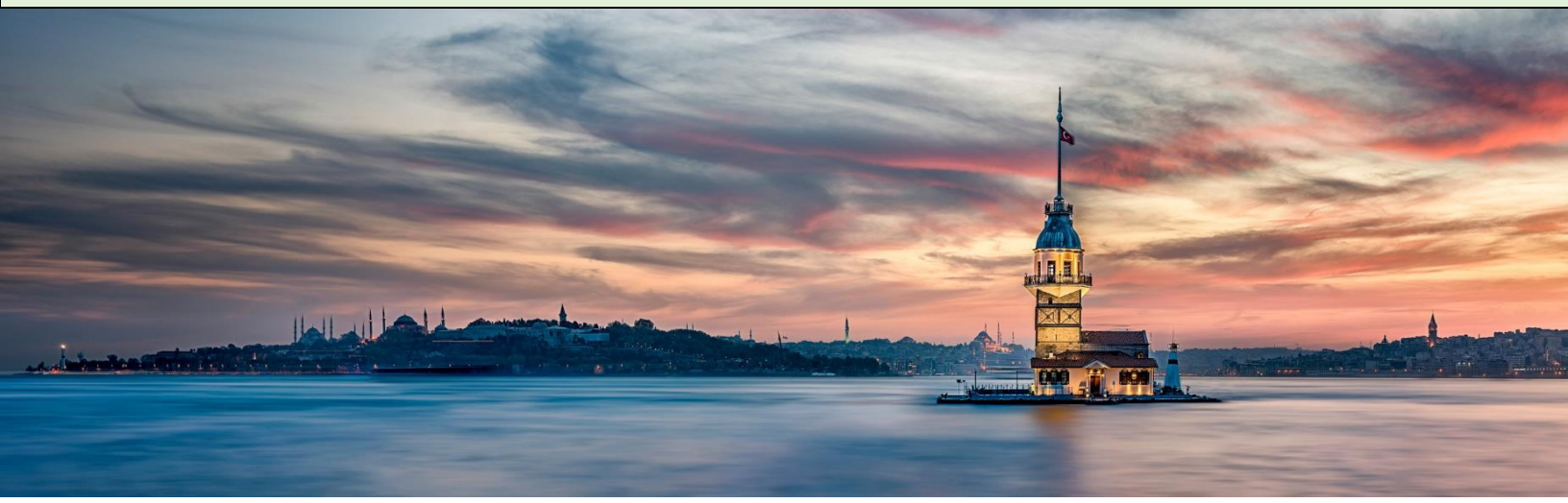
February 19-20, 2021
Istanbul, Turkey

ICMI–2021

1st INTERNATIONAL CONFERENCE ON COMPUTING AND MACHINE INTELLIGENCE

BOOK OF PROCEEDINGS 2021



AIPLUS

İstanbul Sabahattin Zaim Üniversitesi

Organized by

Department of Computer Engineering,
Istanbul, Sabahattin Zaim University,
Istanbul, Turkey.
https://izu.edu.tr

Partners

AIPLUS
308 Daisyfield Centre, Appleby Street
Blackburn Lancashire BB1 3BL,
United Kingdom
https://www.aiplustech.org/

BOOK OF PROCEEDINGS OF

1st International Conference on Computing and Machine Intelligence

(ICMI-2021)

February 19-20, 2021,

Istanbul, Turkey

Editorial Board

Dr. Akhtar JAMIL

Dr. Alaa Ali HAMEED

ISBN: 978-605-06675-7-8

Istanbul Sabahattin Zaim University Yayınları; No. 57.

Istanbul Sabahattin Zaim University
Halkalı Cad. No:2 Küçükçekmece/Istanbul
Tel: 0212 692 96 00
Faks: 0212 693 82 29
www.izu.edu.tr

Copyright © 2021

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright.

The individual contributions in this publication and any liabilities arising from them remain the responsibility of the authors. The publisher is not responsible for possible damages, which could be a result of content derived from this publication

info@aiplustech.org
https://icmi.aiplustech.org

Keynote Speakers

Prof. Dr. Samee Ullah Khan

Mississippi State University,

United States of America

Prof. Dr. Ge Wang

Biomedical Imaging Center

Rensselaer Polytechnic Institute

United States of America,

Associate Prof. Dr. Alfredo Vellido

Polytechnic University of Catalonia (BarcelonaTech)

Spain.

Committees and Scientific Boards

General Chairs

Dr. Ismail Kuçük, Istanbul Sabahattin Zaim University, Turkey.

Dr. Hasan Kömürcügil, Eastern Mediterranean University, Turkey.

Dr. Akhtar Jamil, Istanbul Sabahattin Zaim University, Turkey.

Technical Program Chairs

Dr. Naim Ajlouni Musleh , Istanbul Aydin University, Turkey.

Dr. Alaa Ali Hameed, Istanbul Sabahattin Zaim University, Turkey.

Dr. Fadi Al-Turjman ,Near East University, Turkey.

Dr. Bulent Bayram, Yildiz Technical University, Turkey.

Dr. Dursun Zafer Seker, Istanbul Technical University, Turkey.

Dr. Chawki Djeddi, University of Rouen, France.

Dr. Nadeem Iqbal, Abdul Wali Khan University Mardan, Pakistan.

Dr. Luca Romeo, Istituto Italiano di Tecnologia, Italy.

Dr. Paulo Batista, University of Évora, Portugal.

Publication Chairs

Dr. Imran Ahmed Siddiqi, Department of Computer Science, Bahria University, Pakistan.

Dr. Amani Yahyaoui, Istanbul Sabahattin Zaim University, Turkey.

Dr. Vijayakumar Varadarajan, The University of New South Wales, Sydney, Australia.

Registration Committee

Dr. Aydın Tarık Zengin, Istanbul Sabahattin Zaim University, Turkey.

Dr. Jawad Rasheed , Istanbul Sabahattin Zaim University, Turkey.

Publicity Committee

Erdal Alimovski , Istanbul Sabahattin Zaim University, Turkey.

Hasibe Büşra Doğru, Istanbul Sabahattin Zaim University, Turkey.

Sahra Tilki , Istanbul Sabahattin Zaim University, Turkey

Ayşenur Gençdoğmuş, Istanbul Sabahattin Zaim University, Turkey

Mirsat Yeşiltepe, Yildiz Technical University, Turkey

Enes Albay, Istanbul Technical University, Turkey

Program Committee

Dr. Abdullah Sönmez, Department of Computer Engineering, Istanbul Sabahattin Zaim University, Turkey.

Dr. Abdullahi Abdu Ibrahim, Department of Computer Engineering, Altinbas University, Turkey.

Dr. Adem Ozyavaş, Department of Computer Engineering, Istanbul Aydin University, Turkey.

Dr. Aftab Ahmed Khan, Department of Computer Sciences, Karakoram International University, Pakistan.

Dr. Ahmad Din, Department of Computer Engineering, Comsats University Islamabad, Pakistan.

Dr. Ahmet Gürhanli, Department of Computer Engineering, Istanbul Aydin University, Turkey.

Dr. Alaa Sheta, Deparment of Computer Sciences, Southern Connecticut State University, USA.

Dr. Ameur Bensefia, Computer & Information Sciences Division, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates.

Dr. Atta Ur Rehman, College of Engineering and IT, Ajman University, United Arab Emirates.

Dr. Azeem Hafeez, Department of Electrical and Computer Engineering, University of Michigan, USA.

Dr. Chawki Djeddi, Laboratoire d'informatique de traitement de I'information et des systemes (LITIS), University of Rouen, France.

Dr. Dostdar Hussain, Department of Computer Sciences, Karakoram International University, Pakistan.

Dr. Fadi Al-Turjman, Research center for AI and IoT, Near East University, Turkey.

Dr. Fatma Bozyiğit, Department of Computer Engineering, İzmir Bakırçay University, Turkey.

Dr. Gabriella Casalino, Department of Informatics, Università degli Studi di Bari Aldo Moro Bari, Italy .

Dr. Hakan Gençoğlu , Department of Computer Engineering, Istanbul Sabahattin Zaim University, Turkey.

Dr. Hayder Al-Kashoash, Department of Computer Engineering, Southern Technical University, Iraq.

Dr. Huda Algayyim, Department of Computer Engineering, Southern Technical University, Iraq.

Dr. Imene Yahyaoui, Universidad Rey Juan Carlos, Applied Mathematics Materials Science and Engineering and Electronic Technology, Spain.

Dr. Ismail Duru, Department of Computer Engineering , Istanbul Sabahattin Zaim University, Turkey.

Dr. Jafar Alzubi, Department of Computer Engineering, Al-Balqa Applied University, Jordan.

Dr. Kevser Nur Çoğalmiş , Department of Computer Engineering, Istanbul Sabahattin Zaim University, Turkey.

Dr. Korhan Cengiz, Department of Electrical-Electronics Engineering, Trakya University, Turkey.

Dr. M. Basel Bazbouz, The Nanoscience Centre Department of Engineering, University of Cambridge, UK.

Dr. Mehmet Fatih Amasyali , Yildiz Technical University, Turkey

Dr. Meryem Uzun-Per, Department of Biomedical Informatics, Wake Forest University, USA

Dr. Muhammad Abdul Basit, Montana Technological University, Butte Montana, USA.

Dr. Muhammad Fahim , Institute of Information security and cyberphysical systems, Innopolis University, Russia.

Dr. Muhammad Ilyas, Department of Electrical and Electronics Eng., Altinbas University, Turkey.

Dr. Muhammed Davud , Department of Computer Engineering, Istanbul Sabahattin Zaim University, Turkey

Dr. Nadeem Iqbal, Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan.

Dr. Naim Ajlouni, Department of Software Engineering, Istanbul Aydin University, Turkey.

Dr. Prateek Agrawal, Department of Computer Science, University of Klagenfurt, Austria.

Dr. Sadeq Alhamouz, Deparment of Computer Sciences, WISE University, Jordan.

Dr. Sibel Senan, Deparment of Computer Engineering, Istanbul University, Turkey.

Dr. Sobhan Sarkar, Management Science Division of Business School, University of Edinburgh, UK.

Dr. Syed Attique Shah , Balochistan University of Information Technology Engineering and Management Sciences, Pakistan

Dr. Vira V. Shendryk, Deparment of Computer Sciences, Sumy State University, Ukraine.

Dr. Zahra Elmi, Department of Computer Engineering, Istanbul Sabahattin Zaim University, Turkey.

Dr. Zeynep Orman, Deparment of Computer Engineering, Istanbul University, Turkey.

CONTENTS

S.No.	Title / Authors	Page
1	Design and Implementation of a Genetic Framework for Project Scheduling Problem Anum Kaleem, Saima Jawad and Urooj Rafiq	1
2	Design and Implementation of a Convolutional Artificial Neural Network Based Mask Detection System Sonay Duman and Mehmet Ali Aktaş	6
3	Automatic insulin delivery: Artificial pancreas controlled by machine learning trained algorithm compared to other therapies for diabetes treatment Joan Carles Peiro-Ruiz	10
4	Classification of Fruits by Sample-based Image Processing Hasibe Busra Dogru, Yahya Sirin, Sahra Tilki, Mirsat Yesiltepe, Akhtar Jamil and Alaa Ali Hameed	15
5	Indoor Localization Technique for Small Cell Networks Muhammad Ilyas and Oguz Bayat	20
6	An Integrated Real-Time Water Quality and Usage Monitoring and Control System Argho Das and Abdur Rahman Rubayet	24
7	A Brief Survey on Deep Learning based Recommender Systems and Applications Muhammad Sanwal and Alper Özcan	29
8	Design and Implementation of a Snake-like Robot with Amplitude-Controlled Phase Oscillator-based Motion Control Serkan Karacol, Deniz Korkmaz and Gonca Ozmen Koca	34
9	Performance Analysis of XGBoost Classifier with Missing Data Zeliha Ergul Aydin and Zehra Kamisli Ozturk	39

10	Using Machine Learning Methods for Detecting Phishing Websites (A Comparative Analysis) Hamdullah Karamollaoğlu and Ahmet Albayrak	44
11	Big Data Interoperability Measurements and Decision Making Weiam S. Elsaghair and Dogu Cagdas Atilla	49
12	The Real-Time Detection of Red Californian Worm Eggs Ali Çelik and Sinan Uğuz	54
13	The Influence of Quality and Size of the Training Dataset on the Performance of the Support Vector Machine Algorithm. Evan Kurpiewski and Ilya Samokhvalov	59
14	Evaluating the Performance of Agents for Deceptive Levels Noor Us Sabah and Adeel Zafar	64
15	Clustering Performance Analysis of Traditional and New-Generation Meta-Heuristic Algorithms Şüheda Semih Açmali and Yasin Ortakci	70
16	Computer Vision and Artificial Intelligence For Assessing Plant Diseases In Data-Driven Agriculture: A Case Study Of Downy Mildew In Grapevine Javier Tardaguila, Ines Hernandez, Salvador Gutierrez, Ignacio Barrio, Ruben Iñiguez, Fernando Palacios and María P. Diago	76
17	Computer Vision for Assessing Downy Mildew In Grapevine Leaves Under Laboratory Conditions Ines Hernandez, Salvador Gutierrez, Sara Ceballos, Ruben Iñiguez, Ignacio Barrio, Fernando Palacios, Silvia L Toffolatti, Giuliana Maddalena, María P. Diago, Javier Tardaguila	81
18	Application of Deep Learning And Computer Vision For Grapevine Flower Counting in Digital Viticulture Fernando Palacios, Gloria Bueno, Jesús Salido, María P. Diago, Ruben Iñiguez, Javier Tardaguila	85

19	Determining Covid-19 with Relieff-Based Machine Learning Algorithms Using Biochemistry Parameters Çağla Danacı, Seda Arslan Tuncer, Hakan Ayyıldız and Mehmet Kalaycı	89
20	Data Science Concept, Scope and Technological Development: A Review Ahmet Albayrak and Hamdullah Karamollaoğlu	93
21	Comparative Analysis of Modified-P&O and Modified-PSO Based MPPTs for Partial Shading Conditions Semih Cam, Haluk Gozde and Mustafa Aktas	98
22	Analysis of Cracks in Photovoltaic Module Cells From Electroluminescence Images by Deep Learning Miktat Aktaş, Ferdi Doğan and İbrahim Türkoğlu	103
23	A Study on Automatic Counting of Steel Bars with Image Processing Ali Apalı, Ahmet Yavuz, Murat Demirtaş and Burcu Ceren Sarıoğlu	108
24	Machine Learning Methods for Land Cover Classification from Multi-Spectral Images Fatma Kiraç, Akhtar Jamil, Alaa Ali Hameed, Jawad Rasheed, Mirsat Yesiltepe and Bulent Bayram	112
25	Smart Home Automation System Design Based on IoT Device Cloud Muhammad Ilyas, Osman Nuri Uçan and Yehya El Mohamad	116
26	Recyclable Material Detection in Video Streams using Neural Networks Enes Bayturk, Abdullah Agca, Halil Balamur, Levent Kutlucan and Ulas Vural	124
27	Tuberculosis and Lung Cancer Prediction Using Support Vector Machines and Artificial Neural Network Amani Yahyaoui, Amir Karaj, Merve Hamzaoğlu, Akhtar Jamil and Nejat Yumuşak	128

28	A Recipe for Social Media Analysis Shahid Alam and Juvariya Khan	132
29	Self HARMing Malware - An Adaptive IoT Honeypot for Automated, Repetitive Malware Seamus Dowling	137
30	Predict the Match Outcome in One Day International Cricket Matches Using Machine Learning Zia Ur Rehman, Muhammad Munawar Iqbal and Hamza Safwan	143
31	Rumor Detection Based on Temporal Dimensions Arfa Hussain and Saira Karim	150
32	ARM Based Development of Embedded System for an Energy and Harmonic Analyzer Ugur Polat and Ergün Erçelebi	155
33	New Geometric Based Features for Facial Expression Recognition Nuri Özbey and Mehmet Bilginer Gülmezoğlu	160
34	Spectrogram Images Based Identification of Bird Species Using Convolutional Neural Networks Jutyar Awrahman and Hakan Kutucu	165
35	Analysis and Investigation of Malicious DNS Queries Using CIRA-CIC-DoHBrw-2020 Dataset Muosa Tayseer Jafar, Mohammad Al-Fawa'Reh, Zaid Al-Hrahsheh and Shifa Tayseer Jafar	170
36	Deep-Immune-Network Model for Vulnerable Clone Detection Canan Batur Şahin	175
37	Perspectives of Big Data Analytics and Explainable Machine Learning in Identification of Probable Biomarkers of Alzheimer’s Disease Afreen Khan, Swaleha Zubair and Samreen Khan	180

38	Identification of Medical Forum Posts on Hypertension and Cholesterol Based on Machine Learning Mansur Alp Toçoğlu and Aytuğ Onan	185
39	Towards a Privacy Preserving Machine Learning-based Access Control for the Internet of Things Aissam Outchakoucht, Hamza Es-Samaali, Oussama Mounnan, Anas Abou El Kalam and Siham Benhadou	190
40	Evaluation of Outlier Algorithms for Anomaly Detection Pınar Ersoy, Mustafa Erşahin and Deniz Kılınç	196
41	Link Analysis and Web Search: A review Badaruddin, Ali Raza and Abdul Aziz	202
42	A Real-Life Predictive Maintenance: A Case Study from Industry İlknur Kurban, Mehmet Tekeli, Özgür Selmanoğlu, Onur Tekir, Murat Şahin and Deniz Kılınç	206
43	Assessment of Current Computational Intelligence Methods on Benchmark Functions Soner Kızıloluk, Umit Can and Bilal Alatas	210
44	Comparative Analysis Of Machine Learning Algorithms For Mapping Of Debris Covered Glacier Through Remote Sensing Data: A Case Study Of Hunza Basin Rahila Parveen, Aftab Ahmed Khan and Dostdar Hussain	214
45	Landslide Hazard Assessment for KKH (Karakorum Highway) Using Machine Learning and Deep Learning Approaches Wajid Hussain, Aftab Ahmed Khan and Israr Hussain	221
46	Music Makam Recognition by Using Convolutional Neural Network Yucel Cimtay	226
47	A Comparison of Obstacle Dependent Gaussian and Hybrid Potential Field Methods for Collision Avoidance in Multi-Agent Systems Fethi Candan, Yusheng Peng and Lyudmila Mihaylova	230

48	Nonholonomic Path Planning for A Mobile Robot Based On Voronoi And Q-Learning Algorithm Mustafa Al-Hassow, Oguz Ata and Doğu Çağdaş Atilla	236
49	Classification of Power Quality Events Using Deep Learning Hammad Khalid and Abdulfetah Shobole	240
50	Improving Sentiment Analysis Based on Gated Recurrent Unit Model by Using Feature Selection Mohammed Hussein Abdalla and Fatih Özyurt	245
51	Real Time Emotion Recognition Using Convolutional Neural Network Abdoulaye Bah	250
52	Decov-CNN: A Simple CNN Model for Detection Of COVID-19 Using Chest X-Rays S Suba and Nita Parekh	256
53	Firefly Algorithm Based Maximum Power Point Tracking for Photovoltaic System Under Partial Shading Condition Hasan Basri Karakaya and O. Fatih Kececioglu	261
54	Developing A Protective – Preventive and Machine Learning Based Model on Child Abuse Fatih Mert, Muhammed Ali Aydin and Abdul Halim Zaim	265
55	Automated Biometrical Fingerprint Recognition Scheme Using Synthesized Images Erdal Alimovski and Jawad Rasheed	272
56	Remedial Directed Topic Map on Personalized Scaffolding Adaptive Learning Management System Yulia Wahyuningsih, Arif Djunaidy and Daniel Oranova Siahaan	277
57	Application of Machine Learning methods for Prediction and Diagnosis of Urology Diseases Yasemin Hande Sitki, Erhan Gokcay and Yusuf Şevki Günaydın	282

58	Impact of Local Histogram Equalization on Deep Learning Architectures for Diagnosis Of COVID-19 On Chest X-Rays Suleyman Serhan Narli and Gokhan Altan	287
59	Hybridizing A Conceptual Hydrological Model with Neural Networks to Enhance Runoff Prediction Zeynep Beril Ersoy, Umut Okkan and Okan Fıstıkoğlu	292
60	Development of A Weighted Ensemble Approach for Prediction of Blood Glucose Levels Shashank Bhargav, Shruti Kaushik, Abhinav Choudhury and Varun Dutt	297
61	Underlying Concepts and Understanding of Internet of Things (IoT): Case Study Md Sarwar Morshedul Haque, Md Rafiul Hassan, Mohammad Kamal Hossain, Sk Md Mizanur Rahman and Md Arifuzzaman	302
62	Controlling Driver Behavior in ADAS With Emotions Recognition System Oleg Evstafev Vladimir Bespalov, Sergey Shavetov and Mikhail Kakanov	311
63	Comparative Analysis of Deep Learning and Traditional Machine Learning Models for Turkish Text Classification Hasibe Busra Dogru, Sahra Tilki, Alaa Ali Hameed and Akhtar Jamil	316
64	Comparative Analysis of Different Algorithms for Image Denoising Mohammad Ikhsan Zakaria	323
65	Skin Lesions Segmentation and Classification for Medical Diagnosis Merve Gün, Alaa Ali Hameed, Mirsat Yesiltepe and Akhtar Jamil	327
66	Gender Classification Using Deep Learning Techniques Sahra Tilki, Hasibe Busra Dogru, Alaa Ali Hameed, Akhtar Jamil, Jawad Rasheed and Erdal Alimovski	332
67	A Comparative Study of Multi Label Classification Methods with Unlabeled Data Oumaima Stitini, Soulaimane Kaloun and Omar Bencharef	337

68	Adaptive Neuro Fuzzy Inference System Based Control of a Wind Turbine and Validation of the Real-time Dataset Meer Abdul Mateen Khan, Mohammad Kamal Hossain and Md Sarwar Haque	342
69	Determining of Alzheimer from DNA Sequences with One-Dimensional Capsule Networks Suat Toraman and Bihter Daş	347
70	Artificial Neural Networks Based Survival Prediction of Heart Failure Using Only Serum Creatinine and Ejection Fraction Zehra Karapinar Senturk	351
71	Using Data Science to Detect Software Aging Fatma Bozyigit, Kadir Sert, Murat Şahin, Dursun Dinçer and Deniz Kılınç	356
72	Hypermeters Optimization in Recurrent Neural Networks-LSTM Approach for Human Activity Recognition Ayşenur Topbaş, Alaa Ali Hameed and Akhtar Jamil	360
73	COVID-19 Detection from Chest X-ray Images using CNN Elif Aşıcı, Alaa Ali Hameed, Akhtar Jamil, Jawad Rasheed, Sahra Tilki and Hasibe Büşra Doğru	366
74	Electricity Loss and Fraud Prediction with Deep Learning for Dicle Region Orçun Kitapcı, Alaa Ali Hameed and Akhtar Jamil	371
75	Radiological Medical Reports Classification Alda Kika and Suela Maxhelaku	376
76	A Survey of Artificial Intelligence Driven Blockchain Technology: Blockchain Intelligence Naim Ajlouni, Adem Özyavaş and Mustafa Takaoğlu	380
77	Developing a Clinical Decision Support System by Classification of Melanoma using Machine Learning Techniques Betül Kara, Şeyma Büyücek, Mehmet Gamsızkan and Pakize Erdoğmuş	387

78	Implementation of One Stop Shop model for Government Services Ina Hyseni and Endri Xhina	391
79	Identifying Pneumonia in SARS-CoV-2 Disease from Images using Deep Learning Abdulkadir Şahiner, Alaa Ali Hameed and Akhtar Jamil	397
80	Weather Forecasting Using Back Propagation Feed Forward Neural Network and Multiple Linear Regression Doğancan Ulutaş, Alaa Ali Hameed and Erdal Alimovski	404
81	Augmented Fake News Detection using LSTM and Particle Swarm Optimization Ghulam Mustafa, Shahid Alam, Muhammd Asif Khan and Basit Shahzad	410

Design and Implementation of a Genetic Framework for Project Scheduling Problem

Anum Kaleem, Saima Jawad
Department of Computer Science,
Bahria University, Islamabad, Pakistan.
anumkleem.buic@bahria.edu.pk, saima@bahria.edu.pk

Urooj Rafiq
Department of Information Technology,
S&P Global Market Intelligence, Islamabad, Pakistan.
uroojrafiq@hotmail.com

Abstract —
The quality of every project is determined by time, cost and scope – the triple constraint. Change in one factor incontrovertibly affects the others and attempting to minimize the cost and duration may prove to be an arduous task. Therefore, a successful project necessitates effective utilization of resources. To facilitate the project planning process, an automated intelligent tool is required to generate an optimal project schedule. Research has shown that genetic algorithms (GAs) can be used as an optimization tool in the project management domain. The main objective of this research is to investigate the use of GAs to enhance the efficacy of the project planning process by aiming to resolve time and cost conflicts. The research also includes a study of the effect of genetic environment parameters (population size, crossover rate, mutation probability, number of generations) on the solution efficiency. The paper describes the design and object oriented implementation of a genetic framework for the solution of the project scheduling problem. The proposition has been implemented in the C# language using Microsoft's dot Net platform. The test cases depict that optimal project cost and task duration are achieved as generation size increases. In conclusion, an increase in the permutations of project schedule possibilities increases the chances to achieve minimum project cost.

Keywords— Genetic Algorithm; Project Scheduling Problem; Project Planning; Project Management

I. INTRODUCTION

One of the fundamental project management challenges is effectively scheduling tasks and allocating resources. Attempting to minimize both the cost and duration may prove to be an arduous task as reducing cost increases the project duration and vice versa.

Planning and scheduling requires integration of data pertaining to task details, resource constraints and the relationships among them. Depending on the kind of project, a variety of tasks may be included. For example, project planning involves requirements gathering, task analysis, design, development and testing and integration of modules. The resources employed may be both renewable and non-renewable encompassing manpower, machines, budget and raw materials. Effective assignment and management of the tasks and resources can make the difference between products that hit or miss the market.

Thus, the optimization and automation of task scheduling and resource allocation is essential for good results. To solve this problem a multitude of exhaustive search methods have been used which may require extensive execution time as size of the problem increases. Genetic Algorithms (GAs) can provide an efficient and near optimal schedule for projects.

This paper is divided into seven main sections. Section II presents the existing work in genetic algorithms along with the problems in project and resource scheduling. Section III discusses the proposed genetic framework and section IV elaborates on the application scenario. Section V contains the discussion of the acquired results. Future considerations are discussed in section VI and section VII concludes the research.

II. BACKGROUND STUDY

The core objectives of project scheduling are to minimize the cost and duration of the project and to minimize the number of products that are delivered late [1]. For effective project scheduling, it is necessary to assign the tasks with minimum number of resources to ensure completion in minimum duration. This complex multi-objective management problem belongs to the category of NP-hard problems [2, 3]. In order to find optimal solutions for these problems researchers have formulated numerous approaches including Genetic Algorithms (GAs), a branch of Evolutionary Computing. GAs are population-based search algorithms that reduce the possibility of finding the local optima because of the multiple search points. GAs can also be used to evolve polynomial time solutions for NP-hard problems [4]. A myriad of studies have been conducted on the use of GAs for solving the Project Scheduling Problem (PSP). A hybridized GA which integrates the Priority Rule-base Parallel Schedule Generation Scheme (PRPSGS) with conventional GA has been proposed for the Resource Constrained Project Scheduling Problem (RCPSP) with multi-skilled workforce [5]. Amr Kandil and Khaled El-Rayes implemented a parallel multi-objective genetic algorithm capable of optimizing large scale construction projects [6]. A hybrid greedy GA to reduce the make span of a project and production of good quality schedules has been proposed by Delgoshaei et al [7]. For optimization of RCPSP from different domains Rajeevan et al used GAs [8]. Findings have shown GA to be capable of searching the best known solutions from the problem space. In a study conducted by Sebt et al a Multi-Mode Resource Constraint Problem (MMRCPSP) was used to find the feasible mode and starting time for project scheduling. Here the authors have enhanced GA with an efficient fitness function and mode list based mutation operator [9]. The results show that using GA reduces the time duration for these projects. Sebt et al have proposed three different variants for GAs namely: Standard GA, Stud GA and Jumping Gene. In particular, the results show the effectiveness of Stud GA in solving RCPSP [10]. Xin Shu et al discuss a model taking into consideration priority of activities and a storage adjacency matrix to solve resource constraint multi project scheduling problem (RCMPSP). They find a solution for medical resource scheduling to enhance the feasibility of the problem solution [11]. Similarly, Firoz Mahmud et al propose a GA to solve heterogonous project scheduling problem which indicates the need of optimization of PSP [12]. In [13] Huu Dang Quoc et al perform a comparative analysis between GA, Greedy Search and differential evolution

algorithm. The result suggest that differential evolution algorithm is an enhanced model for solving PSP [13].
In short, many advanced GA techniques have been used to solve PSP, RCPSP, RCMPSP, PRPSGS and MMRCPSP successfully. However, there is a need to implement an automated tool based on these sophisticated techniques. The main objective of this research is to investigate application of GAs for the solution of PSP. The proposed genetic framework is designed using object oriented paradigm and C# language and the dot NET platform are used for the implementation.

III. Genetic Project Scheduling Framework

A GA evolves the solution of a problem. The algorithm begins with an initial set of solutions called population (represented by chromosomes – a collection of genes). Fitness of the chromosomes is calculated on the generation of the population. Subsequently, offspring are formed from parent solutions which have been selected according to their fitness. Every solution has an associated quality value which represents its chances to succeed. Crossover and Mutation are two genetic operators which are applied next. Crossover is implemented to form a new child by exchanging the information of two parent chromosomes. It is expected that the new population will be better than the old one. Mutation refers to arbitrary change in the information of a randomly selected gene from the population. Until certain conditions (such as the number of populations or improvement of the best solution) are satisfied, the entire process is repeated
Different environment parameters influence the efficiency of a GA process. Among these, the size of chromosome population size is the most important factor. A larger population size may increase the probability of computing the global optimum but on the other hand, the processing time increases for larger population. Crossover Probability refers to the number of times a crossover will be performed. The offspring is an exact copy of the parents in the absence of a crossover. In the presence of a crossover, however, the offspring is a mixture of parts of the parents' chromosome. That is to say, if the probability of the crossover is 100%, then all offspring are different from their parents. If the crossover probability is 0%, whole new generation is an exact copy of chromosomes from the parents. Mutation Probability refers to how frequently parts of chromosome are mutated. If there is no mutation, the offspring, taken after crossover (or copy), is unchanged. If mutation does in fact occur, part of chromosome is changed. Thus, in case of a 100% mutation probability, the entire chromosome is changed, whereas, if the probability is 0%, the chromosome remains unchanged. Mutation prevents falling of GA into local extreme, however, frequent occurrence should be avoided because that will in fact change GA to random search.

A. Solution Methodology

Figure 1 illustrates the devised methodology for the implementation of the genetic framework. First of all, the user inputs the genetic environment parameters and the project plan requirements. In order to find the fittest schedule, GA functions like crossover, mutation and fitness are applied on the initial requirements. Until a project schedule is formulated and a terminating condition is met, this process continues.

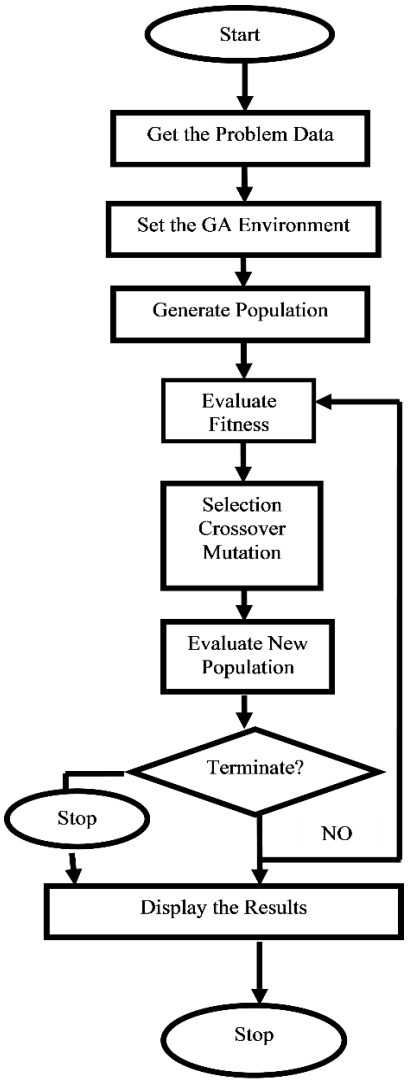


Figure 1: Solution methodology

Different states of the genetic algorithm are explained in Figure 2 where first population is generated according to the population size. Fitness of each chromosome is calculated. Chromosomes are arranged in ascending order of their fitness value. Two-point crossover is performed and mutation probability is added to all the less fit chromosomes. Chromosomes with acceptable fitness values are placed in the new generation. Processing is terminated till best fit chromosome or schedule with optimal/minimum project cost is obtained.

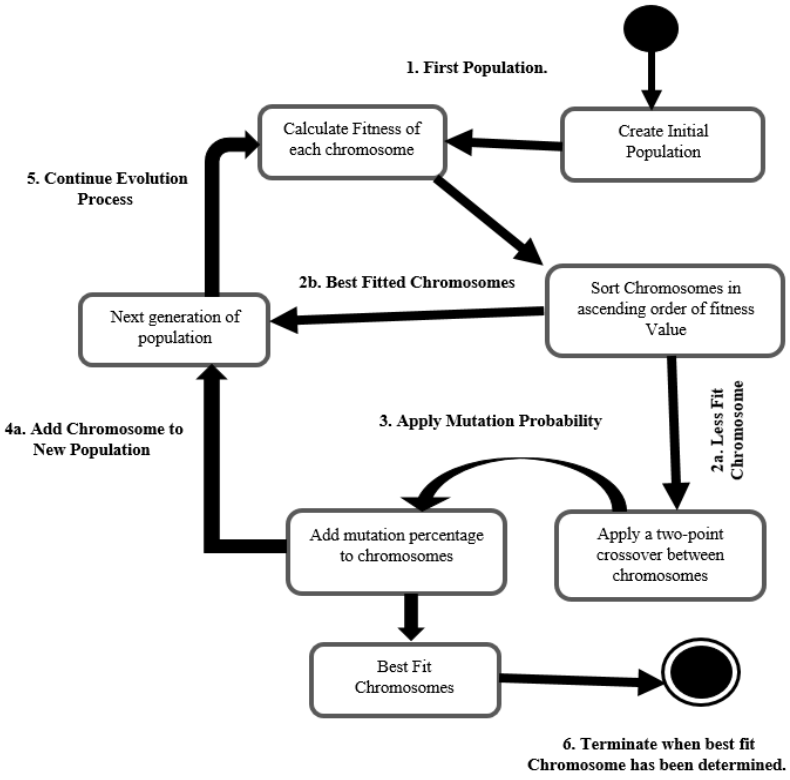


Figure 2: State diagram of solution methodology

B. Object Oriented Structure

Figure 3 illustrates the object oriented structure of the genetic framework. The structure comprises of gene, chromosome, population and generations. The smallest component of the structure is a gene. A gene contains the fundamental information about the problem. A chromosome is a schedule formed considering employee and project details. A specified population size determines the generation of multiple chromosomes. Various population generations are generated according to the size of the generation. Finally, until an optimal schedule is attained, fitness, crossover and mutation functions are applied on the project schedule. Different classes have been constructed in order to implement object oriented GA structure.

C. Genetic Algorithm

1. Set the GA environment parameters: population size ***p***, generation size ***g***, crossover rate ***c*** and termination criteria ***s***.
2. Initialize ***employee class*** consisting of attributes such as ***salary, experience in years, availability*** and ***employee skill set []***.
3. Initialize ***project details class*** consisting of attributes such as ***title of project, project skill set [], number of employees required and minimum duration for task completion***.
4. Initialize ***gene class*** by initializing required ***employee class*** and ***project detail class objects***.
5. Initialize an array of ***chromosome class C[]*** with respect to the population size ***p.*** In each chromosome perform the following:
   a. Compare ***employee skillset*** from ***employee class*** and ***project skill set*** from ***project detail class*** to formulate a ***schedule***.
   b. Calculate ***project cost of each schedule.***
   c. Calculate ***fitness value*** of each schedule
6. Arrange population of chromosomes in ascending order of their fitness value.
7. Initialize an array of ***generation class G[]*** with respect to ***generation size g.***
8. Calculate ***fitness threshold value.***
9. Perform ***crossover*** using ***cross-over rate c*** on chromosomes with project cost greater than ***fitness threshold***.
10. Perform ***mutation*** on chromosomes with project cost greater than fitness threshold.
11. Continue till ***best fit schedule*** with ***minimum project cost*** is obtained.

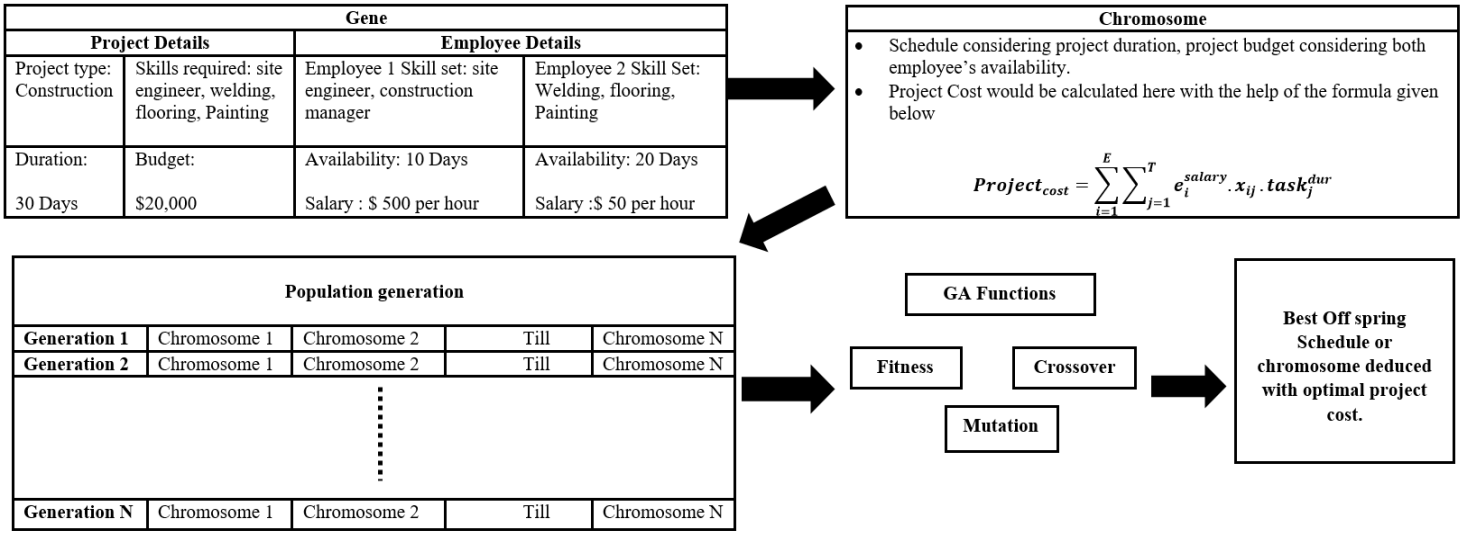


Figure 3: Object oriented GA structure

C.1 Population Generation

The initial population of ***schedule (chromosome)*** and ***generations of population G[]*** are generated according to the generation size ***g***. One generation consists of ***multiple variants of schedules*** with different project costs. A schedule is formed according to ***employee availability***, ***salary***, ***skillset*** and ***project skill requirement***. Each schedules ***project cost varies*** in accordance to the ***assignment of project tasks*** and ***employee skill set***.

C.2 Fitness Evaluation

The fitness of a schedule is computed using minimum project cost and accumulative project time duration. In accordance to calculate the project duration, the time duration of each task being performed needs to be calculated [4].

1. Let $task_j^{dur}$ be the time period of each task. This can be computed by dividing the effort for current task performance t_j^{effort} by the accumulative degree of employee dedication $\sum_{i=1}^{E} x_{ij}$. This value lies between 0 and 1, if an employee is unable to carry out project tasks then x_{ij} is assigned the value of 0. The equation for project task period computation is given below:

$$task_j^{dur} = \frac{t_j^{effort}}{\sum_{i=1}^{E} x_{ij}}$$

2. Let $Project_{cost}$ be the cost of current schedule. It can be computed by multiplication of employee salary, time duration of each task and employee dedication [4]. The mathematical equation of which is stated below:

$$Project_{cost} = \sum_{i=1}^{E}\sum_{j=1}^{T} e_i^{salary} . x_{ij} . task_j^{dur}$$
$$\sum_{i=1}^{E} x_{ij} > 0$$

C.3 Cross-over Function

A two dimensional single point crossover is applied where, rows and columns of parents are swapped on the basis of random selection [4]. Figure 4 below explains the above mentioned process in detail. In a way similar to that of a coin toss, the crossover is performed in order to determine which parent gene would be passed to the offspring. Until the crossover probability ***c*** is reached, alternate schedules from the present generations are selected to be placed in the new offspring.

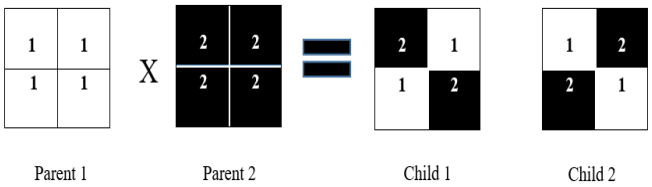


Figure 4: 2D Single point crossover

IV. GA FRAMEWORK APPLICATION DESIGN

To implement the GA framework, an application titled "GAP Scheduler" was designed and developed utilizing C# language dot NET platform. The application requests the user to enter task and resource skill set information. This data is entered in separate interfaces and is subsequently stored in a local repository database for optimal schedule determination. User then specifies GA parameters which are cross-over rate, mutation rate, population size and generation size. If user chooses not to specify GA parameters, then default parameter values are applied. Finally, the optimal schedule is displayed by the application in the form of a Gantt chart.

V. RESULTS AND ANALYSIS

The relationship between framework input variables, GA environment parameters and the output variables is displayed in Table 1. The input parameters include the total employees E_{Total}, total project tasks P_{Tasks}, the skills each task requires PT_{skills}, and furthermore, the skills each employee possesses Emp_{skills}. The generation size G_{size}, population size P_{size}, crossover rate C_{rate} and mutation rate M_{rate} are the GA environment parameters. The last two columns represent the project cost $Project_{cost}$ and the time duration $Task_{dur}$ which are the two output variables calculated for each project schedule.

For the analysis, test cases with different number of employees, tasks and GA parameters are generated. The results depict that the more the permutations of a schedule the more the chances of achieving a schedule with minimum cost and time. Figure 5a and 5b illustrate this relationship proving that project cost decreases by increasing the generation size. The results also show that the population size and crossover rate have no effect on project cost.

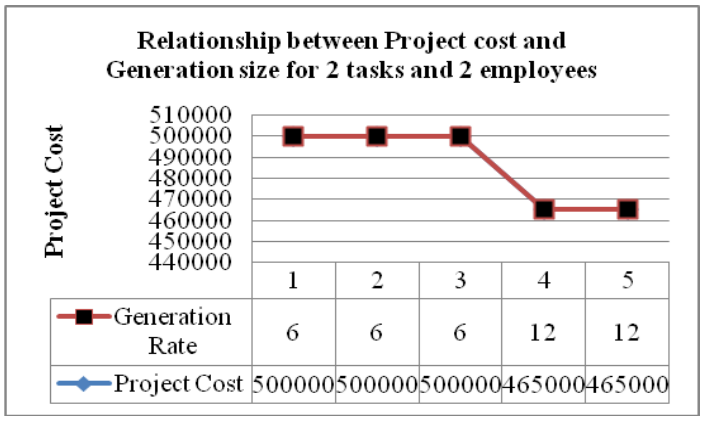


Figure 5a: Relationship between project cost and generation size for two task and two employees

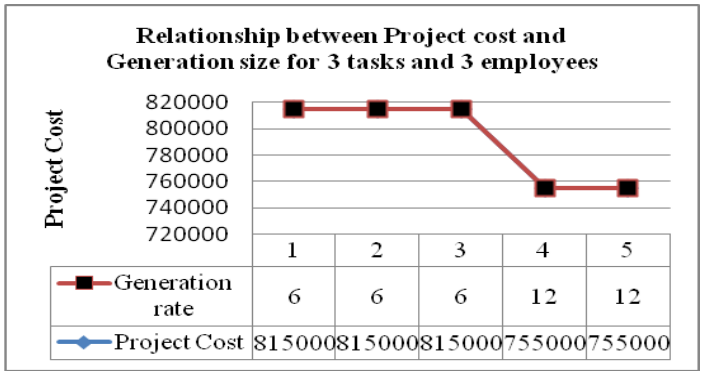


Figure 5b. Relationship between project cost and generation size for three tasks and three employees

Table 2: Result of test cases

	E_{total}	P_{task}	PT_{skills}	Emp_{skills}	G_{size}	P_{size}	C_{rate}	M_{rate}	$Task_{dur}$	$Project_{cost}$
1	2	2	Task 1: 6 , Task 2 : 3	Emp1:6 , Emp2:3	6	10	0.7	0.09	29	500000
2	2	2	Task 1: 6 , Task 2 : 3	Emp1:6 , Emp2:3	6	20	0.7	0.09	29	500000
3	2	2	Task 1: 6 , Task 2 : 3	Emp1:6 , Emp2:3	6	10	1	0.09	22	500000
4	2	2	Task 1: 6 , Task 2 : 3	Emp1:6, Emp2:3	12	20	1	0.09	22	465000
5	2	2	Task 1: 6 , Task 2 : 3	Emp1:6 , Emp2:3	12	10	0.7	0.09	22	465000
6	3	3	Task1:4 , Task2:1 , Task3:2	Emp1:7 , Emp2:2 , Emp3:4	6	10	0.7	0.09	42	815000
7	3	3	Task1:4 , Task2:1 , Task3:2	Emp1:7 , Emp2:2 , Emp3:4	6	20	0.7	0.09	49	815000
8	3	3	Task1:4 , Task2:1 , Task3:2	Emp1:7 , Emp2:2 , Emp3:4	6	10	1	0.09	49	815000
9	3	3	Task1:4 , Task2:1 , Task3:2	Emp1:7 , Emp2:2, Emp3:4	12	10	0.7	0.09	42	755000
10	3	3	Task1:4 , Task2:1 , Task3:2	Emp1:7 , Emp2:2 , Emp3:4	12	20	1	0.09	42	755000

VI. FUTURE CONSIDERATIONS

To manage conflicting requirements in the project scheduling problem the proposed framework can further be improved. The current approach is focused on project scheduling for a single project. This framework would be more effective if it could manage finding optimal schedule cost for multiple projects simultaneously. It would provide project managers efficient solutions for PSP with minimum permutations collectively. Another consideration would be the use of priority based crossover instead of single point crossover [14]. By incorporating priority based crossover project managers would be able to set precedence to which gene is being passed to the offspring instead of random selection. Priority based crossover example is given in the table 2. Here user can select which set of activity should be passed over to the new offspring. In such crossover user would be able to input crossover task priority instead of crossover rate.

Table 2: Priority based crossover

Parent 1								
Priority	1		2				3	
Task/Activity List	1	3	6	5	7	4	8	2
Parent 2								
Priority	1		2				3	
Task/Activity List	1	2	4	6	7	5	3	8
Exchange priority no 2 from both parents lists								
Child 1								
Priority	1		2				3	
Task/Activity List	1	3	4	6	7	5	8	2
Child 2								
Priority	1		2				3	
Task/Activity List	1	2	6	5	7	4	3	8

VII. CONCLUSION

In practical life, projects mostly have strict constraints of time and cost. Automation of finding optimal solution to PSP can save project resources. In this research work we implemented a GA based Project Scheduling application called GAP Scheduler to facilitate the project managers. The application will aid them to build an efficient schedule consisting of minimal cost and duration. Moreover, the utilization of resources is automatically optimized. This GA based tool can be used in the project management domain for the optimal assignment of workforce to the tasks. This scheduler is a step forward in the application of evolutionary algorithms for the solution of real world problems. The efficiency of the existing implementation can be further increased by using multithreading for the genetic computations. In the future we plan to enhance the functionality of application by incorporating provision of multiple projects' scheduling simultaneously.

References

[1] M. B. Wall, "A Genetic Algorithm for Resource-Constrained Scheduling by," 1996.

[2] D. Sundar, B. Umadevi, and D. K. Alagarsamy, "Multi Objective Genetic Algorithm for the optimized Resource usage and the Prioritization of the Constraints in the Software project planning," *Int. J. Comput. Appl.*, vol. 3, no. 3, pp. 1–4, 2010.

[3] B. K. T. Trzaskalik, "Genetic algorithm in multi-criteria multi-project scheduling problem References:," pp. 2–3.

[4] E. Alba and J. F. Chicano, "Software project management with GAs," vol. 177, pp. 2380–2401, 2007

[5] C. Liu and S. Yang, "A Hybrid Genetic Algorithm for Integrated Project Task and Multi-skilled Workforce Scheduling," vol. 6, pp. 2187–2194, 2011.

[6] A. Kandil and K. El-Rayes, "Parallel Genetic Algorithms for Optimizing Resource Utilization in Large-Scale Construction Projects," *J. Constr. Eng. Manag.*, vol. 132, no. May, pp. 491–498, 2006.

[7] A. Delgoshaei, M. Khairol, M. Ariffin, B. T. H. Tuah, B. Baharudin, and Z. Leman, "Minimizing makespan of a resource-constrained scheduling problem: A hybrid greedy and genetic algorithm," International Journal of Industrial Engineering Computations, vol. 6, pp. 503–520, 2015.

[8] M. Rajeevan and R. Nagavinothini, "Time Optimization for Resource-Constrained Project Scheduling Using Meta-Heuristic Approach," vol. 4, no. 3, pp. 606–609, 2015.

[9] M. H. Sebt, M. R. Afshar, and Y. Alipouri, "An efficient genetic algorithm for solving the multi-mode resource-constrained project scheduling problem based on random key representation," vol. 2, no. 3, pp. 905–924, 2015.

[10] M. H. Sebt, M. H. F. Zarandi, and Y. Alipouri, "Genetic algorithms to solve resource-constrained project scheduling problems with variable activity durations," vol. 11, no. 3, 2013.

[11] X. Shu, Q. Su, Q. Wang, and Q. Wang, "Optimization of Resource-Constrained Multi-Project Scheduling Problem Based on the Genetic Algorithm," *2018 15th Int. Conf. Serv. Syst. Serv. Manag. ICSSSM 2018*, pp. 0–5, 2018, doi: 10.1109/ICSSSM.2018.8465086.

[12]F. Mahmud, F. Zaman, R. Sarker, and D. Essam, "Heuristic Embedded Genetic Algorithm for Heterogeneous Project Scheduling Problems," *2020 IEEE Congr. Evol. Comput. CEC 2020 - Conf. Proc.*, 2020, doi: 10.1109/CEC48606.2020.9185712.

[13]H. D. Quoc, L. N. The, C. N. Doan, and T. P. Thanh, "New Effective Differential Evolution Algorithm for the Project Scheduling Problem," *2020 2nd Int. Conf. Comput. Commun. Internet, ICCCI 2020*, pp. 150–157, 2020, doi: 10.1109/ICCCI49374.2020.9145982.

[14]S. U. Kadam and S. U. Mane, "A genetic-local search algorithm approach for resource constrained project scheduling problem," *Proc. - 1st Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2015*, pp. 841–846, 2015, doi: 10.1109/ICCUBEA.2015.168.

Design and Implementation of a Convolutional Artificial Neural Network Based Mask Detection System

Sonay Duman†
Computer Engineering
Toros University
Mersin, Turkey
sonay.duman@toros.edu.tr

Mehmet Ali Aktaş
Computer Engineering
Toros University
Mersin, Turkey
mehmet.aktas@toros.edu.tr

ABSTRACT

By the spread of the corona virus to the world, all countries have started to take some precautions to protect them from the virus in the pandemic process. Individuals have been warned not to engage in dialogue at less than 2 meters, not to go out without a mask, and to pay attention to their hand hygiene. Among these measures, wearing a mask has been shown to significantly reduce virus spread. For this reason, it is important to wear a mask especially in closed places such as schools, shopping malls, and workplaces. In this study, a system based on a convolutional neural network has been designed to identify people who do not wear a mask by a camera in the entrance of places where wearing a mask is mandatory.

KEYWORDS

YOLO, CNN, mask detection, real-time, corona virus

1 Introduction

According to the most recent data published by the WHO as of November; 45.942.902 corona virus cases and 1.192.644 deaths were recorded worldwide [1]. Many new scientific studies, especially in the field of health, have been carried out regarding the new corona virus (COVID-19). These studies generally contain information about the definition of the virus, its spread and protection routes. The concept of corona virus is derived from the Latin word corona, which means crown or wreath. On the surface of the virus, which creates an image reminiscent of the crown or sun figure, there are viral (spreading viruses) pointed peplomers consisting of proteins. Corona virus which causes respiratory infections that can be fatal in humans is found in mammals and birds. More lethal varieties come to the fore as SARS, MERS and COVID-19. Vaccines or antiviral drugs have not yet been developed to prevent or directly treat corona virus infections that can be passed from animals to humans [2].

Since international interaction has played a major role in the spread of the virus worldwide, restrictions have been imposed on transportation and tourism in all countries. Activities such as worship travel, sports activities and scientific congresses increase the interaction with the interaction of large masses of people. For this reason, social activities were suspended in all countries and infection control guides were published. Infection guidelines basically explain the rules of mask, distance and hygiene and provide information to individuals to comply with these three rules. With these measures, unmasked access to confined spaces is restricted, including public transportation vehicles such as shopping malls, markets, buses, trains, and aircraft. The countries that managed to keep the virus density under control by reducing it to a certain number have entered a normalization process in order to restart economic, social and cultural activities. With the start of these activities, it has been made compulsory to wear masks in indoor/outdoor areas with high human density. Mask control is started at the entrances of areas such as shopping malls, markets, public institution buildings, schools, conference rooms where there is no natural air flow. These checks are usually carried out by people assigned at the entrance gates.

In this study, a real-time mask detection system based on artificial convolutional neural networks has been designed that can be used at the entrance of areas where masking is mandatory. The algorithm can work on an embedded hardware to detect a mask with a camera. In the system, YOLO [3] real-time object recognition algorithm has been trained with a database containing images of masked and unmasked individuals, and mask detection has been provided.

2 Related Work

With the spread of the corona virus, many social measures have been taken to prevent the spread of the disease. Studies in the field of health and technology to ensure that precautions are taken and followed have gained speed. During the pandemic, determining whether the COVID-19 virus has settled in the lungs of patients has been an important pre-diagnostic criterion. For this reason, different studies have been carried out using artificial intelligence and deep learning methods [4, 5, and 6]. Also, because of the great danger of entering closed areas without a mask, mask recognition system studies have begun to be carried out [7, 8]. In this study, a low cost and portable system was designed and

implemented on Raspberry pi [9] so that the mask recognition system can be integrated into all places.

3 System Structure and Modules

The proposed system is designed to run the mask detection algorithm on the Raspberry pi. Its purpose is to enable the system to be easily integrated into different structures. In the system prototype, which is operated on the images obtained by using the Raspberry pi camera module, it is determined whether or not people wear masks with the YOLO algorithm and gives audible warning by Google Text to Speech library [10]. Figure 1 illustrates the block diagram of the system.

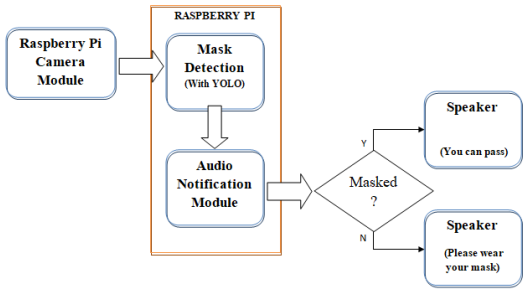


Figure 1: Block Diagram of the Proposed System

3.1. Raspberry Pi and Camera Module

The designed system has been implemented on Raspberry Pi 3 Model B and its prototype was created. Raspberry Pi was preferred to make the prototype low cost and portable. The Raspberry Pi is a low cost credit card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a small device that allows people of all ages to explore. It can perform everything a desktop computer can do, such as surfing the internet, playing high definition video, preparing spreadsheets. V2 camera module has been used for the prototype which has a Sony IMX219 8 mega pixel sensor [9].

Figure 2: Raspberry Pi 3 Model B and Camera Module [9]

3.2. Creating Voice Notifications with Google Text to Speech Module

Google Text-to-Speech (gTTS) is a Speech Synthesis technology that allows computers to speak human words like humans. Computers can interpret text and produce a 'human-like' speech voice with the ability to speak to users (in different languages) with appropriate intonation. The Google Text-to-Speech (gTTS) module is a Python library that has been used for interacting with Google's Text-to-Speech service and creating an mp3 file in Python that can be played on headphones / speakers [10].

3.3. Convolutional Neural Network and You Only Look Once (YOLO) Algorithm

CNN is a specially developed type of multilayer neural networks developed by Yann LeCun in 1988 [11]. Besides having different architectures, they have advanced propagation algorithm like classical artificial neural networks. CNN was developed to recognize visual patterns directly from pixel images, keeping the preprocessing volume to a minimum. It gives good results against sudden changes in patterns and geometric transformations. CNN contains feature extraction methods that are different from other algorithms, including feature replication and summarizing layers. CNN basically consists of 4 layers. These layers are; convolution layer, activation function layer, pooling layer and normalization layers [8]. In addition to these layers, layers with different properties are developed, but for the multi-classification in the last layer, usually the softmax layer is preferred. There are several CNN based algorithms such as YOLO (You Only Look Once) [3], R-CNN (Region Based CNN) [12], Fast R-CNN [13], Faster R-CNN [14] which are widely used in the field of image processing. YOLO is one of the most popular CNN algorithms which can make real-time object detection.

YOLO can detect various objects in different sizes, works pretty fast and provides real-time extraction on several devices. It takes images as inputs, passes them through the neural network, and as outputs the bounding boxes and class predictions take the vector. YOLOv3 uses DarkNet-53 to feature detection and then make curved layers. Darknet-53 [15] is a 53-layer Convolution Neural network, trained on ImageNet, and consists of 3 × 3 and 1 × 1 filters with jump connections, such as the ResNet. YOLOv3 estimates bounding boxes using size sets as anchor boxes. Anchor boxes predefined different shapes and were calculated in the COCO [16] dataset using k-mean clustering. YOLOv3 uses a total of 9 anchor boxes. As an example, if a system wants to detect people, it usually looks for vertical rectangular boxes, if calls the car, it will probably be a horizontal rectangular box.

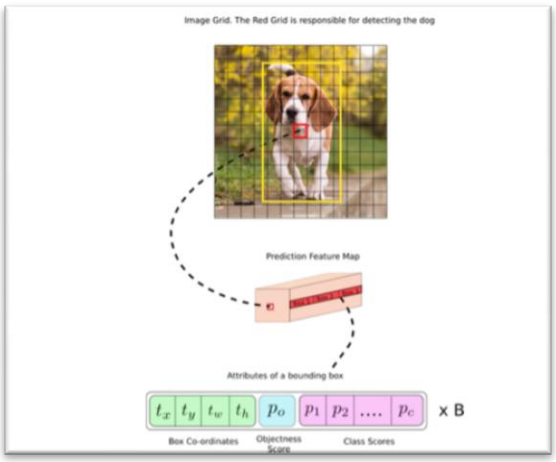


Figure 3: Working Principle of YOLOv3 [7]

In this study, the light version of the real-time object recognition system YOLOv3-tiny [17] which can be used in hardware with less processing capability has been used. The implementation of the system will be explained in the next section.

4 Experimental Results

In the system implementation phase, YOLO model has been trained with a ready dataset that consists of different size of 690 with mask and 686 without mask individuals' images [18]. Before, starting train the model, images have been resized to 224x224 pixels. The dataset has been split into 80% training and 20% testing data.

4.1. Training

After setting the data set, labeling is made on the data, because YOLO labels the bounding boxes of images. The BBOX-Label-Tool [19], a tool developed for labeling, is used in the data labeling part. First, the images containing the object to be detected are collected under a folder. Images and Labels files include two folders one of them for the train set and the other for the validation set. Before starting the training, weights previously trained with big data for Darknet-53 are downloaded into the darknet folder. The Darknet-53 architecture has been shown on Figure 4.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4: Darknet-53 Architecture [15]

After downloading the weights, model becomes ready for training. The training starts with the required command while the terminal is under the darknet directory. Figure 4 shows the training loss and accuracy graph of the system.

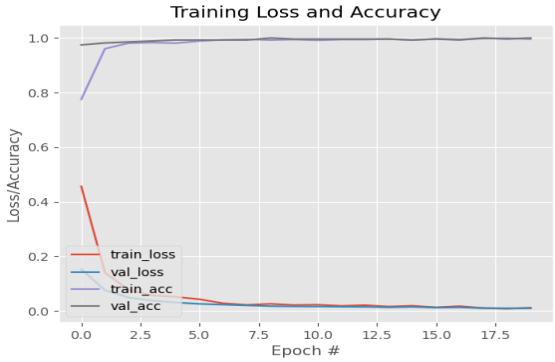


Figure 5: Training Loss and Accuracy Graph

4.2. Testing

After the training is over, the last weighting file has been taken from the backup and moves it into the cfg file. The network is tested by entering the required command with images to be in the data set from the terminal screen. Figure 4 and 5 shows the results of testing the image which have masked individuals. Images with unmasked individuals were also tested. Figure 6 shows the image of an individual without a mask. Although rarely, the system has detected some elements that are similar form of the mask but not can be used as a mask. Figure 7 shows incorrect mask detection. According to the results obtained in the study, the system which was tested on 276 different images was able to detect a mask in about 7 seconds by operating on Raspberry Pi with 99.7% accuracy. The recall and precision values of the system have been calculated as 0.98 and 0.99.

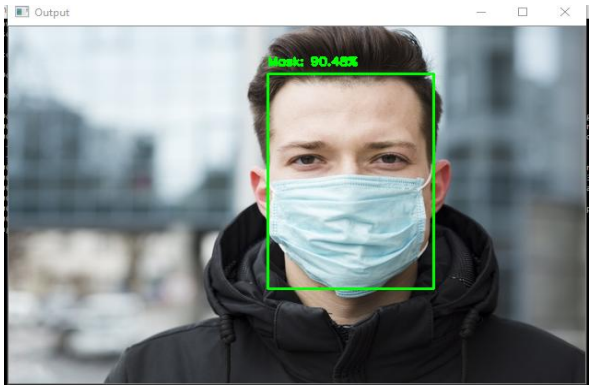


Figure 6: Mask Detection Result

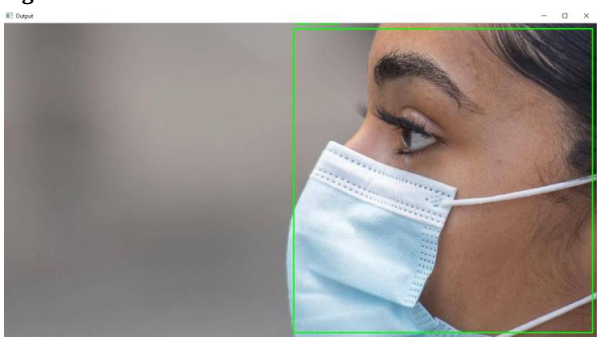


Figure 7: Mask Detection Result

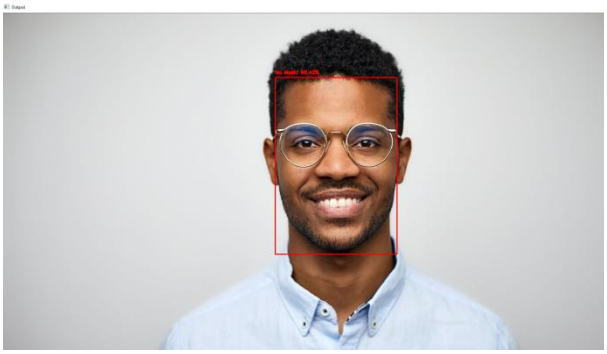


Figure 8: Unmasked Detection Result

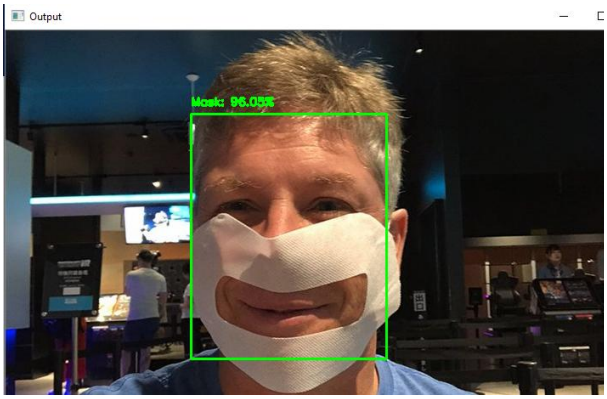


Figure 9: Incorrect Mask Detection Result

5 Conclusion

In this study, a system that automatically detects whether individuals are wearing a mask or not has been designed and implemented. The developed system will make it easier to inspect, especially in closed areas where mask is required. Developing the system by training and testing it with more images and increasing its usability in every field are determined as future studies.

REFERENCES

[1] Anon. WHO Coronavirus Disease (COVID-19) Dashboard. Retrieved November 2, 2020 from https://covid19.who.int/

[2] Lawrence S. Sturman and Kathryn V. Holmes. 1983. The Molecular Biology of Coronaviruses. *Advances in Virus Research* (1983), 35–112. DOI:http://dx.doi.org/10.1016/s0065-3527(08)60721-6

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). DOI:http://dx.doi.org/10.1109/cvpr.2016.91

[4] Ismael, Aras M., and Abdulkadir Şengür. "Deep Learning Approaches for COVID-19 Detection Based on Chest X-ray Images." *Expert Systems with Applications* 164 (2021): 114054. Print.

[5] Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., & Singh, V. (2020). A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos, Solitons & Fractals*, *140*, 110190.

[6] Rasheed, J., Jamil, A., Hameed, A. A., Aftab, U., Aftab, J., Shah, S. A., & Draheim, D. (2020). A Survey on Artificial Intelligence Approaches in Supporting Frontline Workers and Decision Makers for COVID-19 Pandemic. *Chaos, Solitons & Fractals*, 110337.

[7] Alexandra Lorenzo. 2020. Face Mask Detector using Deep Learning (YOLOv3). (May 2020). Retrieved November 2, 2020 from https://medium.com/face-mask-detector-using-deep-learning-yolov3/face-mask-detector-using-deep-learning-yolov3-209b57f77e92

[8] Rushad Mehta. 2020. Real-Time Mask Detection with YOLOv3. (May 2020). Retrieved November 2, 2020 from https://towardsdatascience.com/real-time-mask-detection-with-yolov3-21ae0a1724b4

[9] Anon. 2015. What is a Raspberry Pi? (August 2015). Retrieved November 2, 2020 from https://www.raspberrypi.org/help/what- is-a-raspberry-pi/

[10] Anon. Text-to-Speech Client Libraries | Cloud Text-to-Speech Documentation. Retrieved November 2, 2020 from https://cloud.google.com/text-tospeech/docs/libraries

[11] Y.Le Cun et al. 1989. Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magazine* 27, 11 (1989), 41–46. DOI:http://dx.doi.org/10.1109/35.41400

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2016. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 1 (2016), 142–158. DOI:http://dx.doi.org/10.1109/tpami.2015.2437384

[13] Girshick, R.. Fast R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 71-13 (2015), 1440-1448.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE *Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (2017), 1137–1149. DOI:http://dx.doi.org/10.1109/tpami.2016.2577031

[15] Anon. Papers with Code - Darknet-53 Explained. Retrieved November 2, 2020 from https://paperswithcode.com/method/darknet-53

[16] Anon. Common Objects in Context. Retrieved November 2, 2020 from https://cocodataset.org/

[17] Joseph Redmon.Retrieved November 2, 2020 from https://pjreddie.com/darknet/yolo/

[18] Prajnasb. prajnasb/observations. Retrieved November 2, 2020 from https://github.com/prajnasb/observations/tree/master/mask_classifier/Data_Generator

[19] Puzzledqs. puzzledqs/BBox-Label-Tool. Retrieved November 2, 2020 from https://github.com/puzzledqs/BBox-Label-Tool

Automatic insulin delivery: Artificial pancreas controlled by machine learning trained algorithm compared to other therapies for diabetes treatment

Profesor Joan Carles Peiró, MSc
Bioengineering research institute
Open University La Salle
La Massana, Andorra
joancarles.peiro@salle.url.edu
ORCID 0000-0003-1366-6605

***Abstract*—Diabetes therapy by means of new automatic insulin delivery artificial pancreas system consisting on insulin pump controlled by an algorithm trained with machine learning technology fed by a continuous glucose monitoring sensor provides better glycemia control compared to previous therapies such as multi-daily injection, insulin pump without continuous glucose reading or "sensor assisted" insulin pump.**

Keywords— artificial pancreas, AP, closed-loop, hybrid closed-loop, automatic insulin delivery, AID, machine learning, CGM, MDI, sensor assisted insulin pump

I. INTRODUCTION

The maintenance of glycemia in range is one of the biggest challenges in the treatment of patients with diabetes. This implies that the calculation of precise doses of insulin is critical and must be individually adapted. As a key performance indicator, an average plasma glucose concentration of glycated hemoglobin (HbA1C) with values <7% was recommended by the American Diabetes Association [1] and has been proven to reduce development and progression of micro-vascular problems and cardiovascular complications by 76% [2]

The complexity of delivery of insulin and the objective of maintaining normoglycemia require a complex, personalized, adaptive and flexible algorithm, which can be achieved with the use of automatic learning techniques as shown in some approaches [3] [4]. Adequate machine learning algorithms can analyze training data, recognize complex patterns, and based on these patterns apply the knowledge to other data to predict their behavior [5]. Since the development of the first AP [6], significant improvements have been made, advancing further with the introduction of smartphones as AP control management systems.

Open-loop control algorithms manually calculate the dose of insulin based on blood glucose and external information [7]. On the other hand, closed-loop algorithms automatically calculate the dose of insulin based on parameters such as food intake and glucose levels measured by CGM. closed-loop models can be classified into closed-loop models and hybrid closed-loop models. In a closed-loop model, insulin dosing decisions are based exclusively on the parameters measured in the patient's body, without requiring any external information such as food intake or exercise habits. hybrid closed-loop models use both closed-loop CGM control and external information, such as carbohydrate intake. Weinzimer et al. [8] compared fully closed-loop control with hybrid closed-loop control. The hybrid model reduced post-meal hyperglycemia without inducing hypoglycemia, and the same result was seen in a similar model tested by another group in a hospital. D.Shi et al. [9] reviews why an artificial pancreas (AP) requires an algorithm that adapts and evolves not only on the data but also on the lifestyle of the patient. Their work is based on the adaptation of the basal and carbohydrate-ratio profile. C.Boughton et al. [10] reviews the progress that technology has achieved for diabetes treatment. His paper covers with detail the advance in "close-loop" artificial pancreas (AP) systems. It is explained that four hybrid closed-loop AP systems have already been approved by regulatory bodies. It also explains that systems are not yet fully automated and require patient intervention to adjust meal boluses by informing carbohydrate intakes, therefore should be classified as hybrid closed loop. Benhamou et al. [11] explains that closed-loop insulin delivery is expected to become a standard treatment for patients type 1 diabetes.

JC Peiro [12], explains how machine learning techniques can provide a suitable algorithm to control a hybrid closed-loop AP. Machine Learning hybrid closed-loop algorithm has been personalized, trained and tested using in-vivo data collected by DT1 patient. Comparison results between MDI and closed-loop AP have been validated in-silico through the UVA/Padova T1DMS [13][14] simulator on ten adults, ten adolescents and ten children using the ML trained closed-loop regression algorithms. The Use of T1DMS simulator has been approved by FDA.

II. METHODS

Statistical analysis : Accu-chek Smartpix [15] platform has been used to collect glycemia control values from Aug 2004 to Aug 2019. The software provides control statistics on % of time with glycemia in range (%TIR), % above and below range, % of hypoglycemia, mean and median values for glycemia control values. It also provides high blood glycemic index (HBGI) and low blood glycemic index (LBGI) [16]. These indexes measure the probability of situations of hyperglycemia and of hypoglycemia. HBGI has a positive correlation with glycated hemoglobin (HbA1c). LBGI and HBGI are calculated as follows:

$$LBGI = \frac{1}{N}\sum_{i=1}^{N} rl\,(BG_i) \qquad HBGI = \frac{1}{N}\sum_{i=1}^{N} rh\,(BG_i)$$

Where BG is blood glucose reading, r(BG) is a measure of the risk associated with a certain BG level; rh(BG) represents the risk of hyperglycemia while rl(BG) represents the risk of hypoglycemia. LBGI is a quantity that increases when the number and/or extend of low BG readings increases. Similarly, the HBGI increases when the number and/or extend of high BG readings increases. The level of risk is explained in Table I.

TABLE I. BLOOD GLYCEMIC INDEX

Risk	LBGI	HBGI
Minimal	≤ 1.1	
Low	> 1.1 and ≤ 2.5	≤ 4.5
Moderate	> 2.5 and ≤ 5.0	> 4.5 – ≤ 9.0
High	> 5.0	> 9.0

III. RESULTS

ML trained algorithm is used to control the insulin pump based on continuous BG readings and carbohydrate intakes to provide the optimal insulin infusion in 5 minutes intervals necessary to maintain blood glucose at the optimal value.

Table II provides the comparison of glycemic control parameters for hybrid closed-loop AP therapy, MDI therapy, insulin pump without CGM therapy, and "sensor assisted" insulin pump therapy.

TABLE II. EVOLUTION COMPARING TREATMENTS

	Multi-Injection	Insulin pump no CGM	Insulin pump + CGM	AP closed Loop	AP closed Loop (Optimal)
# days	543	314	259	112	13
# tests	1165	1544	1497	376	39
Tests/Day	2.1	4.9	5.8	3.4	3.0
Sensor coverage			88%	97%	97%
Mediana Gluc	144.0	154.6	129.5	115.0	113.0
Standard Deviation	76.3	73.9	72.1	55.0	49.0
SD / Median CGM	53.0%	47.8%	55.7%	47.8%	43.4%
Avg Gluc	144.0	154.6	137.4	121.0	116.0
SD	76.3	73.9	54.1	42.0	34.0
SD / Avg	53.0%	47.8%	39.4%	34.7%	29.3%
Above > 180	30.7%	32.5%	19.6%	9.0%	5.0%
In range	48.0%	58.9%	73.3%	84.0%	89.0%
Below < 70	21.3%	8.7%	7.1%	7.0%	6.0%
Hypoglucemia < 50	11.3%	2.5%	1.5%	1.0%	0.0%
(% / Below)	53.1%	29.2%	21.7%	14.3%	0.0%
HGBI	7.0	7.7	5.7	2.3	1.3
LGBI	6.1	1.9	1.7	1.6	1.3
HbA1c	8.3%	7.9%	6.9%	6.6%	6.5%

A. *Comparing hybrid closed-loop AP to MDI therapy*

TABLE III. EVOLUTION COMPARING TO MDI THERAPY

Variation vs Multiinjection	Multi-Injection	Insulin pump no CGM	Insulin pump + CGM	AP closed Loop	AP closed Loop (Optimal)
Gluc Avg		7.3%	-4.6%	-16.0%	-19.4%
Gluc Med		7.3%	-10.1%	-20.1%	-21.5%
HbA1c		-4.3%	-15.9%	-20.1%	-21.4%
HGBI		11.1%	-18.3%	-67.0%	-81.3%
LGBI		-69.1%	-72.8%	-73.8%	-78.7%
Above > 180		5.9%	-36.0%	-70.7%	-83.7%
In range		22.6%	52.7%	75.0%	85.4%
Below < 70		-59.4%	-66.8%	-67.2%	-71.9%
Hypoglucemia < 50		-77.7%	-86.4%	-91.2%	-100.0%

Key remarks on the analyzed data collected in table III comparing hybrid closed-loop AP to MDI therapy can be resumed as follows:

- TIR% presents an improvement of +75%, increasing from 48% to 84% in range
- % above range (> 180 mg/dL) decreases -70.7% (30.7% to 9%)
- % below range (< 70 mg/dL) decreases -67.2% (21.3% to 7%)
- The value of hypoglycemia (< 50 mg/dL) decreases 91.2% (11.3% to 1%)
- LBGI index improves 73.8%, reducing from 6.1 (high risk) to 1.0 (minimal risk)
- HBGI index improves 67.0%, reducing from 7.0 (moderate risk to 2.3 (low risk)
- Mean glycemia reduces -16%, from 144 mg/dL with SD 76.3 to 121 mg/dL with SD 42.0
- Median glycemia reduces -20.1%, from 144mg/dL with SD 76.3 to 115mg/dL with SD 55.0
- Glycated hemoglobin HbA1c improves -20.1%, reducing on average from 8.3% to 6.6%

B. *Comparing hybrid closed-loop AP to insulin pump without CGM*

TABLE IV. EVOLUTION COMPARING TO INSULIN PUMP WITHOUT CGM THERAPY

Variation vs Pump no CGM	Insulin pump no CGM	Insulin pump + CGM	AP closed Loop	AP closed Loop (Optimal)
Gluc Avg		-11.1%	-21.7%	-25.0%
Gluc Med		-16.2%	-25.6%	-26.9%
HbA1c		-12.2%	-16.5%	-17.9%
HGBI		-26.4%	-70.3%	-83.2%
LGBI		-11.9%	-15.1%	-31.0%
Above > 180		-39.6%	-72.3%	-84.6%
In range		24.5%	42.7%	51.2%
Below < 70		-18.3%	-19.2%	-30.7%
Hypoglucemia < 50		-39.3%	-60.5%	-100.0%

Key remarks on the analyzed data collected in table IV comparing hybrid closed-loop AP to insulin pump without CGM therapy can be resumed as follows:

- TIR% presents an improvement of 42.7%, increasing from 58.9% to 84% in range.
- % above range decreases -72.3% (32.5% to 9%)
- % below range decreases -19.2% (8.7% to 7.0%)
- % of hypoglycemia reduces -60.5% (2.5% to 1%)
- LBGI reduces -15.1%, down from 1.9 to 1.6
- HBGI reduces -70.3%, down from 7.7 to 2.3
- Glycemia average reduces -21.7%, from 154.6 mg/dL to 121.0 mg/dL
- Median glycemia reduces -25.6%, from 154.6 mg/dL to 115mg/dL

  Glycated hemoglobin (HbA1c) improves -16.5%, reducing on average from 7.9% to 6.6%

C. *Comparing hybrid closed-loop AP to sensor assisted insulin pump therapy*

TABLE V. EVOLUTION COMPARING TO SENSOR ASISTED INSULIN PUMP THERAPY

Comparació vs Pump + CGM	Insulin pump + CGM	AP closed Loop	AP closed Loop (Optimal)
Gluc Avg		-11.9%	-15.6%
Gluc Med		-11.2%	-12.7%
HbA1c		-4.9%	-6.5%
HGBI		-59.6%	-77.2%
LGBI		-3.7%	-21.7%
Above > 180		-54.2%	-74.5%
In range		14.6%	21.4%
Below < 70		-1.0%	-15.2%
Hypoglucemia < 50		-34.9%	-100.0%

Key remarks on the analyzed data collected in table V comparing hybrid closed-loop AP to "sensor assisted" insulin pump therapy can be resumed as follows:

- TIR% % presents an improvement of +14.6% increasing from 73.3% to 84% in range.
- % above range decreases -54.2% (19.6% to 9%)
- % below range decreases -1% (7.1% to 7%)
- % of hypoglycemia reduces -34.9% (1.5% to 1%)
- LBGI index reduces -3.7%, down from 1.7 to 1.6
- HBGI index improves reducing -59.6%, down from 5.7 to 2.3
- Glycemia average is reduced -11.9%, from 137 mg/dL to 121 mg/dL
- Median glycemia is reduced -11.2%, from 129mg/dL to 115mg/dL
- Glycated hemoglobin HbA1c improves -4.9%, reducing on average from 6.9% to 6.6%.

IV. DISCUSSION

We will compare glycemia control results of therapy using hybrid close loop artificial pancreas system to MDI, to insulin pump without CGM and to "sensor assisted" insulin pump.

A. MDI therapy

The analysis of glycemia variation using MDI therapy shows a big dispersion on the measured results in the analyzed periods. The weighted average of the analyzed periods using MDI therapy indicates the following statistical control parameters:

- Number of samples: 1165
- Number of days: 543
- Mean blood glucose: 144 mg/dL
- Standard deviation: 76 mg/dL
- % deviation/average: 53%
- HBGI: 7.0
- LBGI: 6.1
- % above range(> 180 mg/dL): 30.7%
- TIR% (< =180 mg/dL and >= 70 mg/dL): 48.0%
- % below range (< 70 mg/dL and >= 50 mg/dL): 21.3%
- % Hypoglycemia (< 50 mg/dL): 11.3% with respect to the total, representing 53.1% of the values under range
- HbA1c: 8.3%

B. Insulin Pump without CGM therapy

The weighted average of the periods analyzed with 1544 blood glucose readings are:

- Mean glycemia: 154.6 mg/dL
- Standard deviation: 73.9 mg/dL
- % deviation/mean: 47.8%
- HBGI: 7.7
- LBGI: 1.9
- TIR%: 58.9%
- % above range: 32.5%
- % below range: 8.7%
- % Hypoglycemia: 2.5%, representing 29.2% of the values below range
- HbA1c: 7.9%

C. "Sensor assisted" insulin pump therapy

The weighted average of the two periods analyzed with CGM "sensor assisted" insulin pump, with 1497 BG readings and 259 days of continuous glucose measurement is:

- Average blood glucose: 137.4 mg/dL
- Standard deviation: 54.1 mg/dL (39%)
- HBGI: 5.7
- LBGI: 1.7
- TIR%: 73.3%
- % over range: 19.6%
- % below Range: 7.1%
- % hypoglycemia: 1.5%, representing 21.7% of the values below range
- HbA1c: 6.9%

D. Artificial pancreas with hybrid closed-loop ML trained control algorithm:

- 376 BG tests are collected equivalent to 3.4/day
- Glucose Median of 115 mg/dL is obtained with standard deviation of 55 mg/dL
- TIR%: 84%
- Above range%: 9%
- Below range%: 7%
- 1% of hypoglycemia episodes
- HBGI index: 2.3
- LBGI index: 1.6
- HbA1c: 6.6%

Fig. 1. Statistics collected using hybrid closed-loop AP therapy

Fig. 2. Metabolic control chart using hybrid closed-loop AP therapy

Fig. 1 shows the period from August 1st, 2019 to November 21st, 2019. Fig. 2 shows the metabolic control chart, indicating 97% sensor coverage, a mean CGM of 121 mg/dl with a standard deviation of 42 mg/dL (35%). We can observe a high stability of glucose values. TIR% is 84%, with 9% above range, 7% below range and 1% of periods of hypoglycemia. HBGI index is 2.3 (indicating low risk) and LBGI is 1.6% (very low risk). Glycated hemoglobin is 6.6% All values represent a very significant improvement with respect to MDI, insulin pump without CGM and sensor assisted insulin pump.

Period from August 12th, 2019 to August 25th, 2019 is the period achieving highest performance of the hybrid closed-loop AP system. This cannot be considered the average value, but the optimum value that can be obtained with a hybrid closed-loop AP system controlled using algorithm trained with machine learning technology. This period includes 39 BG tests performed for 13 days, which equals to 2.8 capillary/day measurements. During this period of optimal control and performance we can see a 95% TIR% o, 5% above range and there is a total elimination of measures below range and hypoglycemia. The average glucose in this period is 114 mg/dL with a standard deviation of 34mg/dL (29%). Analyzing the trend for seven days a week shows a high stability of measured values, always within the control ranges established. Both HBGI and LBGI index are in 1 index, well below the low risk index. All glucose measured values are within the high stability area within the range. Metabolic control chart with 97% sensor coverage, indicates a median of CGM of 113 mg/dL with a variability of 49mg/dL, an average of 116 mg/dL with a standard deviation of 34 mg/dL.

E. In-silico analysis comparing closed-loop AP to MDI:

Finally, we will perform in-silico validation comparing closed-loop AP to MDI therapy using UVA/Padova T1DMS simulator [13] using 30 individuals: 10 adult, 10 adolescents and 10 pediatric which is the max that the simulator provides.

TABLE VI. GLYCEMIC CONTROL USING MDI THERAPY (T1DMS SIMULATION FOR 30 INDIVIDUALS)

ID	Mean BG	% <70	TIR %	% > 180	LBGI	HBGI
adult	173.28	0.00	64.12	35.88	0.00	7.35
adolescent	139.79	0.00	84.18	15.82	0.06	2.76
child	64.96	59.61	38.93	1.46	28.92	0.10

Table VI presents the glycaemia stability analysis for MDI therapy; it provides the average values for the 30 individuals grouped by adult, adolescent and pediatric populations.

TIR% for adults' average is 64.12%, for adolescents' average is 84.18% and for children's average is 38.93%. Mean BG is 173.28 for adults' average, 139.79 for adolescents' average and 64.96 for children's average.

Fig. 3 presents the upper and lower 95% confidence bond matrix showing 0% in A zone, 36% B zone, 52% C zone, 12% D Zone and 0% E zone. A+B is 36%.

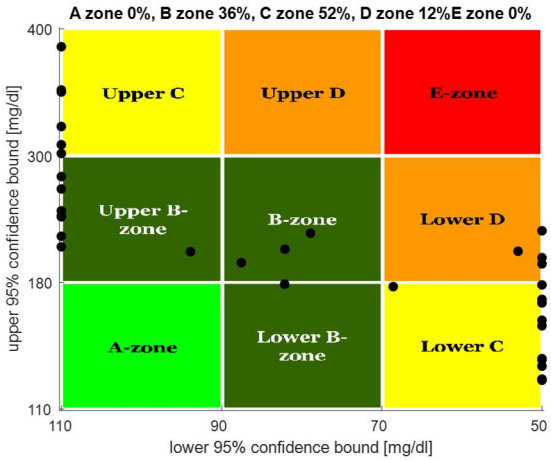


Fig. 3. Confidence bound matrix T1DMS in-silico simulation for MDI therapy

TABLE VII. GLYCEMIC CONTROL USING CLOSED-LOOP AP THERAPY (T1DMS SIMULATION FOR 30 INDIVIDUALS)

ID	Mean BG	% <70	TIR %	% > 180	LBGI	HBGI
adult	130.35	0.00	95.07	4.93	0.07	1.46
adolescent	145.18	0.00	80.85	19.15	0.02	3.43
child	151.44	0.00	80.78	19.22	0.00	4.20

Table VII presents the glycemia stability analysis for closed-loop AP therapy calculating the average values for the 30 individuals using T1DMS simulator.

TIR% improves significatively compared to MDI therapy. TIR% for adult average reaches 95.07%, adolescent average reaches 80.85% and pediatric average reaches 80.78%. Mean BG for adults' average reduces to 130.35, adolescents' average is 145.18 and children's average is 151.44.

Fig. 4 presents the upper and lower 95% confidence bond matrix, showing 12% in A zone, 82% in B zone, 6% in C zone, and 0% D and E zones. A+B is 94%

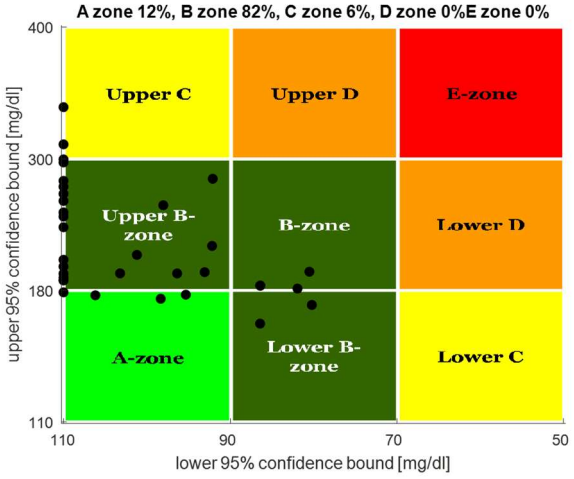


Fig. 4. Confidence bound matrix T1DMS in-silico simulation for closed-loop AP therapy

According to in-silico test T1DMS simulation, closed-loop AP therapy presents significant improvements in glycemia stability showing a A+B zone with 94% in comparison to the MDI therapy with A+B zone with 36%.

V. CONCLUSIONS / INTERPRETATION

We can conclude that "hybrid closed-loop" AP with control algorithm trained with machine learning technology provides very significant improvement in glycemia control compared to the MDI, insulin pump without CGM and sensor assisted insulin pump therapies according to the presented in-vivo analysis and the T1DMS in-silico simulation.

In this analysis, the improvement is obtained consistently with all the glycemic control parameters. Therapy with hybrid closed-loop AP reduces the % of periods out of range (above and below range), TIR% increases significantly, HBGI and LBGI indexes improves reducing high and low blood glycemia risk, reduces the mean and the median of glycemia, and consequently, it reduces the glycated hemoglobin (HbA1c).

REFERENCES

[1] "Standards of medical care in diabetes-2011," *Diabetes Care*, vol. 34, no. SUPPL.1. Jan-2011, doi: 10.2337/dc11-S011.

[2] "The Effect of Intensive Treatment of Diabetes on the Development and Progression of Long-Term Complications in Insulin-Dependent Diabetes Mellitus," *N. Engl. J. Med.*, vol. 329, no. 14, pp. 977–986, Sep. 1993, doi: 10.1056/NEJM199309303291401.

[3] I. Contreras and J. Vehi, "(10) (PDF) Artificial Intelligence for Diabetes Management and Decision Support: Literature Review," *Journal of Medical Internet Research*, 2018. [Online]. Available: https://www.researchgate.net/publication/325162420_Artificial_Intelligence_for_Diabetes_Management_and_Decision_Support_Literature_Review. [Accessed: 24-Dec-2019].

[4] A. K. El-Jabali, "Neural network modeling and control of type 1 diabetes mellitus," *Bioprocess Biosyst. Eng.*, vol. 27, no. 2, pp. 75–79, Apr. 2005, doi: 10.1007/s00449-004-0363-3.

[5] C. M. Bishop, *Pattern Recogniton and Machine Learning*. Springer-Verlag New York, 2006.

[6] A. H. Clemens, P. H. Chang, and R. W. Myers, "The development of Biostator, a Glucose Controlled Insulin Infusion System (GCIIS).," *Horm. Metab. Res.*, vol. Suppl 7, pp. 23–33, 1977.

[7] T. G. Farmer, T. F. Edgar, and N. A. Peppas, "The future of open- and closed-loop insulin delivery systems," *J. Pharm. Pharmacol.*, vol. 60, no. 1, pp. 1–13, Jan. 2008, doi: 10.1211/jpp.60.1.0001.

[8] S. A. Weinzimer, G. M. Steil, K. L. Swan, J. Dziura, N. Kurtz, and W. V. Tamborlane, "Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas," *Diabetes Care*, vol. 31, no. 5, pp. 934–939, May 2008, doi: 10.2337/dc07-1967.

[9] D. Shi, E. Dassau, and F. J. Doyle, "Multivariate learning framework for long-term adaptation in the artificial pancreas," *Bioeng. Transl. Med.*, 2019, doi: 10.1002/btm2.10119.

[10] C. K. Boughton and R. Hovorka, "The artificial pancreas," *Curr. Opin. Organ Transplant.*, 2020, doi: 10.1097/MOT.0000000000000786.

[11] P. Y. Benhamou *et al.*, "Closed-loop insulin delivery in adults with type 1 diabetes in real-life conditions: a 12-week multicentre, open-label randomised controlled crossover trial," *Lancet Digit. Heal.*, 2019, doi: 10.1016/S2589-7500(19)30003-2.

[12] JC. Peiro, "Selection of the optimal machine learning technique for the development of a personalized algorithm to control a 'hybrid' closed-loop artificial pancreas."

[13] "T1DMS - The Epsilon Group." [Online]. Available: https://tegvirginia.com/software/t1dms-2014/. [Accessed: 24-Jan-2020].

[14] K. Zarkogianni, A. Vazeou, S. G. Mougiakakou, A. Prountzou, and K. S. Nikita, "An insulin infusion advisory system based on autotuning nonlinear model-predictive control," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2467–2477, Sep. 2011, doi: 10.1109/TBME.2011.2157823.

[15] "Accu-Chek Smart Pix." [Online]. Available: https://www.accu-chek.co.uk/apps-software/smart-pix. [Accessed: 10-Oct-2020].

[16] B. P. Kovatchev, M. Straume, D. J. Cox, and L. S. Farhy, "Risk analysis of blood glucose data: A quantitative approach to optimizing the control of Insulin Dependent Diabetes," *J. Theor. Med.*, 2000, doi: 10.1080/10273660008833060.

Fruits Sorting with Instance Based Image Processing

Hasibe Busra Dogru
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
hasibe.dogru@izu.edu.tr

Yahya Sirin
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
yahya.sirin@izu.edu.tr

Sahra Tilki
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
sahra.tilki@izu.edu.tr

Mirsat Yesiltepe
Mathamatical Engineering
Yildiz Technical University
Istanbul, Turkey
l 0000-0003-4433-5606

Jawad Rasheed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
jawad.rasheed@izu.edu.tr

Muhammad Usman Shahid Khan
Department of Computer Science
COMSATS University Islamabad
Abbottabad, Pakistan
ushahid@cuiatd.edu.pk

***Abstract*— In the agricultural industry which is among the most prominent sectors on a world scale, image processing for efficiency and quality has also become an important domain. Automation systems are working with image processing to increase the efficiency of the classification of fruits. Image processing is a method developed for digital image formatting and performing some operations for obtaining or extracting useful information from a specific image. There are several methods and programs for processing images. In this study, the results obtained by the classification of apple, pear and peach fruits which are similar to each other, but which differ in terms of roughness and color, by using image processing techniques are presented. After the data sets of these fruits are formed, color and roughness are determined by image processing techniques, and success rates are compared by using Instance-Based (IB) and Bayesian Network classifiers in the analysis phase. Finally, the performances of the fruit classifications were investigated using Receiver Operating Character Curve (ROC).**

Keywords—: image processing, morphological Processes, classification, ROC.

I. INTRODUCTION

Applications related to image processing and computer vision have been increasing day by day in recent years and these applications are widely used in many areas such as in-vehicle automation, security systems, military areas as well as in the agricultural sector. Quality assessment and classification processes on food products are an important problem. Since these processes are usually carried out manually and by eye by quality control workers, they cause misclassifications and create great losses in terms of time and labor. However, with the help of image processing, it is possible to automatically classify and determine fruit characteristics that are faster and in accordance with standards. Methods such as digital image analysis, classification, clustering are used to classify the objects under investigation according to their size, type or quality characteristics.

Many different studies have been done on the classification of fruits. In a study related to this, image processing method was applied on various apples in order to determine the color, size and spots on the apples and to classify the spots. In the first stage, photos of apples of different colors were taken and color and size classification was made by the software. The apple photographs to be classified later were taken in real time on a continuous flowing tape. In the software created in this study, the stain problem was tried to be solved by using various functions in it without the need for additional equipment such as filters [1].

In another sample study, it was aimed to classify the hazelnut fruit, and as a result, the objects found in the environment were detected in real time. A camera was used to obtain the image of hazelnut fruits and image processing techniques were used to process the obtained images. The area and size information covered by the hazelnut fruits on the image plane are calculated. By using average-based classification and K-means clustering algorithms on the resulting information, hazelnut fruit is classified into three real-time classes as small, medium and large [2].

Finally, the article reviewed is a study that classifies with image processing, considering the peach and apple characteristics. Diameter information is used in peach classification process, and size and weight information is used in apple classification. In classification, weight is used as a parameter apart from image processing. Load cell is used for weight detection. The load cell outputs an analog signal by looking at the weight information of apple and peach. These two analog information are converted into digital by analog-digital converter to transfer to the computer. Apple and peach classification is made with the software developed by taking weight information and fruit image [3].

In the study proposed by Ishikawa et al., The classification of strawberry fruits with machine learning techniques is presented. Values (MVs) including length, area, fruit length and width, fruit width / length ratio, Ellipse Similarity Index (ESI), Elliptic Fourier Identifiers (EFDs) and Chain Code Extraction (CCS) from digital strawberry images four different types of descriptors have been created. For the classification stage, the descriptors created by the Random Forest method were used together. As a result of the experiments, it was observed that the CCS descriptor makes a classification with a better performance than other descriptors [4].

Comparison of Artificial Neural Network (ANN), Support Vector Machine (SVM) and Random Forest methods was proposed by Piedad et al. For the classification of banana fruit. In the presented study, four characteristics of banana fruit are used for classification. As a result of experimental studies, the best accuracy was obtained with the Random Forest method with a rate of %94.2 [5].

A study on tomato classification was presented by Li Liu et al. An intelligent tomato classification based on machine vision technique is aimed. Accordingly, the histogram of the color model was used for the color characteristics of tomatoes in different sizes and maturity. Classification has been made in three different categories as small, medium and large. As a

result of experimental studies, the average duration of the system was 0.0687s and its accuracy was measured as 90.7% [6].

Another study on tomato classification, proposed to classify the defect types of tomatoes, used fewer features and image processing techniques compared to other systems. In the classification phase, SVM and MSVM methods were applied together and 98% accuracy was achieved [7].

In this study, three stages are applied to obtain the classification results of apple, pear and peach fruits in terms of roughness and color. In these stages, the data from each step is used in the next step. First, the images of various data sets from different locations are passed through the image pre-processing stage and the data obtained is passed through the object finding and feature extraction process in the second stage, and the features such as roughness and color of the objects are extracted. Objects are classified using the data obtained as a result of these stages.

First, 150 fruit images obtained from different places go through the image pre-processing stage. The data obtained as a result of this stage pass through the object finding and feature extraction process in the second stage, and properties such as roughness and color of the objects are extracted. Engineering calculations for preprocessing, object finding and feature extraction up to the classification stage; Matlab program, which is used for general purpose in numerical computing, data solutions and graphics operations, is used [8]. In the cluster obtained after the image pre-processing stage, the distribution pattern is learned by using IB and Bayes Networks classification algorithms, and the ROC curve is created and the performance is tested. For the classification and testing phases, Weka, the data mining application distributed as open source code by Waikato University, where requirements such as machine learning algorithms and data preprocessing are presented together, was used [9].

II. PROPOSED METHODS

A. Image Pre-Processing

At this stage, processes such as graying the data, morphological processes and determining the color are performed respectively. After these processes, the features related to the image are made more prominent and easy to process.

In the step of greying out the image, the colored images are first converted to gray tone so that we can do certain operations on the image in the data set. Image processing is performed on binary images, so grayscale images are converted into binary images. A threshold value is determined for this. By applying thresholding process on the image obtained, a double picture is obtained by assigning the values above the threshold value to 1, ie white, and the values below it to 0, ie black. The value used in the thresholding process is determined experimentally.

Morphological process is applied to eliminate the noises on the resulting image. With this process, the defects were removed by walking around the picture. In the proposed study, erosion, dilation and opening processes are applied on the double image as a morphological process.

The purpose of etching is to refine or shrink an object in the image. It is possible to remove unimportant and small objects in the binary image with this process. When the etching process is applied, the objects in the image shrink and expand if there is a hole.

The enlargement process makes the object in the image thicker or larger. The structural element determines how the thickening or enlargement process will be done. Each pixel of the image to be processed is placed at the center point of the structural element and the enlargement process is applied. In an image where this process is applied, the holes and gaps in the image are expected to be closed.

The process obtained as a result of applying the abrasion process first and then applying the enlargement process on the image to be used is the opening process. After clearing the gaps in the image, the objects remaining on the image become smaller than the original image.

B. Feature Extraction

In the feature extraction stage, the features belonging to each of the binary images obtained as a result of the image pre-processing stage are extracted. The roughness and colors of the objects in the created data set were determined as feature inferences.

The best method to obtain the roughness value takes the difference between the binary image and the image that we apply morphological operations. Thus, if the overlapping pixel values are the same, it will give the value 0, but if the difference between the pixel values is greater than zero, it means that there is a change for that pixel. The greater the difference between them, the closer the pixel value to the white color. When the images are superimposed and subtracted, the different pixels become whiter and distinct. The difference is proportional to the roughness ratio.

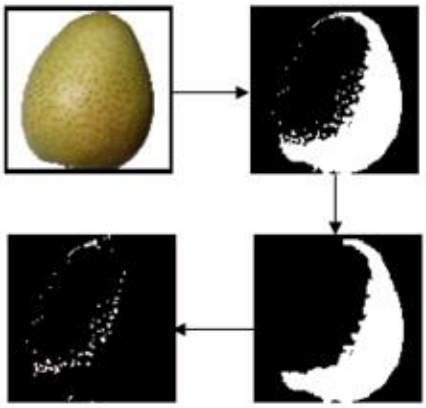


Fig. 1. Pear fruit; original, converted to a binary picture, morphological processes applied and the difference

RGB (Red, Green, Blue) color space was used to determine the colors of the fruits. It consists of RGB color layers. In order to determine the distinguishing feature in RGB color codes; The color code between 0-100 was given a value of 1, the color code between 100-200 was given 2, and the color code between 200-255 was given 3. At the end of this, a 3-digit color value consisting of the numbers 1, 2 and 3 was created for each picture.

In this paper, fruit images are classified using their features in terms of roughness and color after the image pre-processing phase is completed. Two different classification methods, IB and Bayes Networks, are proposed for the classification of objects.

C. Classification

Classification is the distribution of data between these classes by defining classes within a data set. By obtaining the training set, the distribution pattern is learned and the correct classification of the data whose test class is not specified is tried to be made. In this study, fruits are classified using their roughness and color characteristics. Two popular classification methods are suggested for fruits that pass through the image pre-processing stage.

In the IB classification algorithm, there are samples closest to the sample to be classified and all of these samples are dropped against a point in n-dimensional space. For continuous valued functions, the nearest k learning samples are averaged. IB algorithm gives the most seen class value in discrete valued functions. According to the IB algorithm, the value of k is determined first. The suggested value of k in this study is 1. Then the euclidean distances from other objects to the new object are calculated. The distances are listed and the closest neighbors are determined. After the nearest neighbor categories are collected, the most suitable neighbor category is selected. The following formula is used when determining the closest neighbor when the new object arrives [10].

$$Difference_e = \sqrt{(X_i - X_{new})^2 + (Y_i - Y_{new})^2} \quad (1)$$

The characteristic of Bayesian networks is that they are statistical networks and the edges of the nodes that pass between nodes are selected according to statistical decisions. In addition, they are non-directional networks and each node represents a separate variable. Bayes theorem is important for understanding Bayes networks [11]. Bayes' theorem is calculated as follows [12].

$$P(B_i|A) = \frac{P(A \cap B_i)}{P(A)} = \frac{P(A|B_i)P(B_i)}{\sum_{i=1}^{k} P(A|B_i)P(B_i)} \quad (2)$$

D. Criterias Used in Comparison of Classification Algorithms

a) **Confusion Matrix:** In order to evaluate the classification performance, the matrix is obtained by comparing the estimates of the target with the actual values. These classification estimates have one of four assessments.

i. -True to mean right (True Positive - TP) TRUE

ii. -True is false (True Negative - TN) FALSE

iii. -To mean right to the one (False Positive - FP) WRONG

iv. -That means you are wrong (False Negative - FN) TRUE

b) **Error-Accuracy Rate:** This ratio, which is found by the ratio of the number of correctly classified samples (TP + TN) to the total number of samples (TP + TN + FP + FN), is one of the most used methods in performance measurement. The error rate is found by the ratio of the number of incorrectly classified samples (FP + FN) to the total number of samples (TP + TN + FP + FN). Their formulas are shown below.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3)$$

$$Error\ rate = \frac{(FP+FN)}{(TP+FP+FN+TN)} \quad (4)$$

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Precision: Precision is the ratio of True Positive (TP) samples with predicted class 1 to all samples with predicted class 1 (TP + FP) [13].

$$Precision=TP/((TP+FP)) \quad (5)$$

Recall: The ratio of the results obtained by the correct classification of positive samples to the total number of positive samples is called recall.

$$Recall=TP/((TP+FN)) \quad (6)$$

F-messure: Evaluating the precision and recall criteria together, the results are closer to the truth than evaluating each separately.

$$F=2DK/((D+K)) \quad (7)$$

E. Testing

At this stage, ROC curve is drawn with the ratio of TPs to FPs. The Recall and Precision values on the ROC curve consist of different threshold values for each point. Generally, low FPs obtained from threshold values have low TP's. In cases where the results are successful, the TP rate is high, but the FP rate, on the contrary, is low. As the curve obtained to the 'False Positive Rate' axis approaches, the success level decreases. The part under this curve is to show the success of the system with a single value. Expressing the success of the system with a single value is expressed by the area under the ROC curve, and the size of this area is directly proportional to reliability.

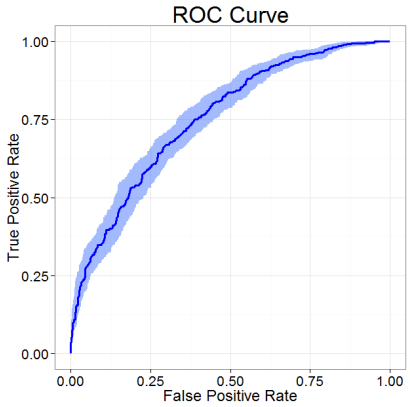


Fig. 2. Sample analysis showing the ROC curve [14]

III. RESULTS AND DISCUSSION

A. Dataset Preparation

A data set consisting of 150 apple, pear and peach images is created to implement the image processing steps. Some of the images are images recorded as a 20-second short film with a white layer on a low-speed traveling tape as a background, and a Logitech C920 camera is used [15]. Others are fruit images that have the same background and size as other images. These images pass through the pre-processing stage and analyze the data obtained according to the feature extractions determined.

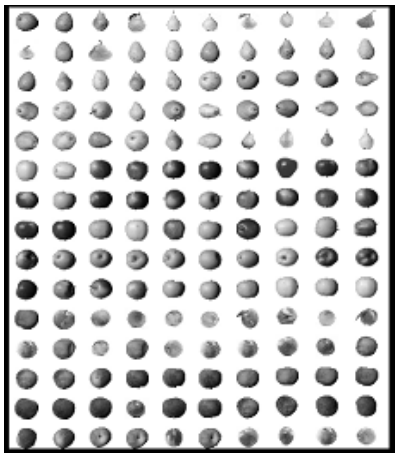


Fig. 3. Fruits-150 data set consisting of 3 classes with 50 shapes per class. Different images of the same objects in columns

B. Experimental Works

After the fruit images in the data set passed the pre-processing phase, the model was created with the IB and Bayes Networks classification algorithms in the Weka library. The performances of the images included in the data set were measured according to the comparison criteria described earlier.

TABLE I. ERROR MATRIX OBTAINED AS A RESULT OF IB CLASSIFICATION ALGORITHM IN WEKA APPLICATION

Classified as	***a***	***b***	***c***
a = pear	50	0	0
b = apple	1	49	0
c = peach	1	0	49

According to the error matrix we obtained in the IB classification algorithm that I determined the K value as 1, it is seen that the whole pear fruit is classified correctly, but one of the apple and peach fruits are mixed with apples. 98.6667 percent was obtained according to the accuracy criterion and 1.3 percent according to the error criterion. According to the precision and recall criteria, the results are 0.987. Since these criteria alone will not be sufficient to evaluate the result, the measured F-criterion also has a result of 0.987.

TABLE II. ERROR MATRIX OBTAINED AS A RESULT OF BAYESIAN NETWORK CLASSIFICATION ALGORITHM IN WEKA APPLICATION

Classified as	***a***	***b***	***c***
a = pear	30	7	13
b = apple	2	42	6
c = peach	4	13	33

According to the data in the error matrix we have obtained in the classification of Bayesian networks, it is seen that 7 of the pear fruits are mixed with apple and 13 are mixed with peach, 42 of the apple fruit are classified correctly, while 2 of them are classified as apples and 6 as peaches. While 33 of peaches were classified correctly, 4 of them were mixed with pear and 13 with apple. Looking at the error matrix, it is seen that IB1 classification is better than Bayesian networks. Bayesian networks have an accuracy rate of 70 percent, and an error rate of 30 percent. According to the precision and recall criteria, the results are 0.715 and 0.700, respectively. The F-criterion required for more accurate results is 0.699. According to these results, it is seen that the IB1 classification algorithm has a better success rate than Bayesian networks.

After the classification process using IB1 and Bayesian networks, the performance of the model is checked and compared by drawing the ROC curve in the test phase with these two classifications.

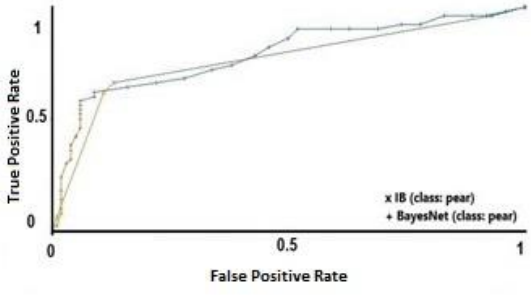


Fig. 4. ROC curve produced by classifying the pear class with IB and Bayes Networks

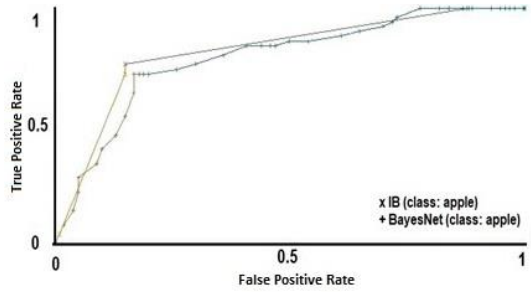


Fig. 5. ROC curve produced as a result of classification of apple grade with IB and Bayes Networks

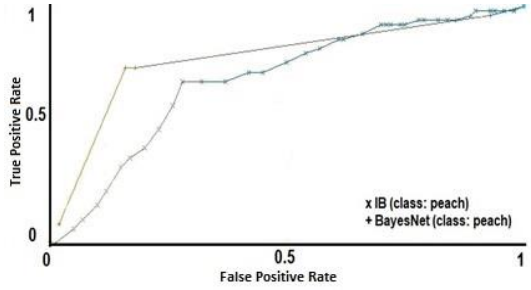


Fig. 6. ROC curve generated as a result of classification of peach grade with IB and Bayesian Networks

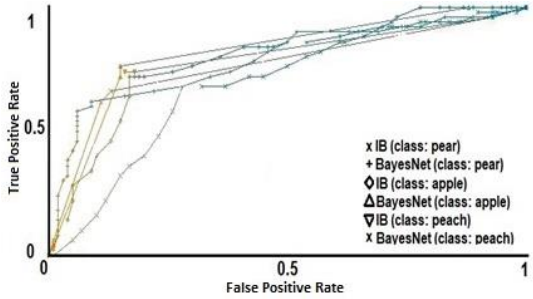


Fig. 7. ROC curve generated by classifying all classes with IB and Bayesian Networks

In the ROC analysis, it is seen that the curve in the Bayesian networks classification is closer to the False Positive axis compared to the curve in the IB classification, and since the size of the area under the curve is directly proportional to the reliability, it is understood that the success level of the IB classification is higher than the Bayesian networks.

IV. Conclusion and Discussion

In this study, the data set consisting of apple, pear and peach fruits was classified using image processing techniques. First, the data passed through the image preprocessing stage was binary for feature extraction, and the roughness ratio was calculated by taking the difference with the picture we applied morphological processes. For the other distinguishing feature, the colors of the data were determined by using the RGB color space. When the success rates were compared by applying the proposed IB1 and Bayesian networks classification to the obtained data, the IB1 method gave a better result with an accuracy of 98.6667 percent compared to the 70 percent success rate obtained in Bayesian networks. The ROC curve is used to compare and evaluate the success level of the classifications made at the last stage. According to the results obtained with the ROC curve, it is understood that the success level of IB classification is higher than Bayesian networks.

In the next study, it is planned to compare the success rates of deep learning and traditional machine learning classification methods using the Generative Adversarial Network (GAN), which can create realistic images to increase the data set.

References

[1] Sofu, Mehmet Mahir, et al. "Elmaların görüntü işleme yöntemi ile sınıflandırılması ve leke tespiti." Gıda Teknolojileri Elektronik Dergisi 8.1 (2013): 12-25.

[2] Solak, Serdar, and Umut ALTINIŞIK. "Görüntü işleme teknikleri ve kümeleme yöntemleri kullanılarak fındık meyvesinin tespit ve sınıflandırılması." Sakarya University Journal of Science 22.1 (2018): 56-65.

[3] Eser, S. E. R. T., Deniz Taşkin, And Nurşen Suçsuz. "Görüntü Işleme Teknikleri Ile Şeftali Ve Elma Siniflandirma." Trakya Üniversitesi Fen Bilimleri Dergisi 11.2 (2010): 82-88.

[4] T. Ishikawa, A. Hayashi, S. Nagamatsu, Y. Kyutoku, I. Dan, T. Wada, ... and S. Isobe, "CLASSIFICATION OF STRAWBERRY FRUIT SHAPE BY MACHINE LEARNING," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, *42*(2), 2018.

[5] J. I. Larada, G. J. Pojas, and L. V. V. Ferrer, "Postharvest classification of banana (Musa acuminata) using tier-based machine learning," *Postharvest Biology and Technology*, *145*, 93-100, 2018.

[6] L. Liu, Z. Li, Y. Lan, Y. Shi and Y. Cui, "Design of a tomato classifier based on machine vision," PloS one, 14(7), e0219803, 2019.

[7] S. D. Kumar, S. Esakkirajan, S. Bama and B. Keerthiveena, "A Microcontroller based Machine Vision Approach for Tomato Grading and Sorting using SVM Classifier," *Microprocessors and Microsystems*, 103090, 2020.

[8] MathWorks: MATLAB. 2016. https://www.mathworks.com/products /matlab.html (Acess Date : 25 January 2020).

[9] Weka, W. E. K. A. "3: data mining software in Java." University of Waikato, Hamilton, New Zealand (www. cs. waikato. ac. nz/ml/weka) 19 (2011): 52.

[10] Uzun E. 2016. Sınıflandırma. https://www.e-adys.com/makine_ ogrenmesi/04-makine-ogrenmesi-siniflandirma/ (Acces Date: 1 February 2020).

[11] Şeker Ş.E. 2008. Bayes Ağları. http://bilgisayarkavramlari. sadievrenseker.com/2008/12/21/bayes-aglari-bayesian-network/ (Acces Date: 1 February 2020).

[12] http://slideplayer.biz.tr/slide/2391226/ (Acces Date: 1 February 2020).

[13] Pourghasemi, Hamid Reza, Masood Beheshtirad, and Biswajeet Pradhan. "A comparative assessment of prediction capabilities of modified analytical hierarchy process (M-AHP) and Mamdani fuzzy logic models using Netcad-GIS for forest fire susceptibility mapping." Geomatics, Natural Hazards and Risk 7.2 (2016): 861-885

[14] Coşkun, Cengiz, and Abdullah Baykal. "Veri madenciliğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması." Akademik Bilişim 2011 (2011): 1-8.

[15] Horea. Fruit Images Dataset. https://github.com/Horea94/Fruit-Images-Dataset (Access Date: 15 February 2020).

Indoor Localization Technique for Small Cell Networks

Muhammad Ilyas
Department of Electrical and Electronics Engineering
Altinbas University
Istanbul, Turkey
Muhammad.ilyas@altinbas.edu.tr

Oguz Bayat
Department of Electrical and Computer Engineering
Altinbas University
Istanbul, Turkey
Oguz.bayat@altinbas.edu.tr

***Abstract*—This article presents a new model for indoor positioning using small cell. The proposed model was tested using both simulations and real-life scenarios. A high accuracy was observed using the new model, with a reported improvement of 8% and 12% from the previous model in simulations and real-life tests, respectively. In addition, the model was further tested using a blind scenario, which resulted in an overall success rate of 92%. The study further highlights issues and challenges faced while characterizing the tracking in such indoor structures, thus providing the way towards novel research.**

Keywords—Femtocells; Indoor localization; Location based services; Message queue telemetry transport protocol; Small cells

I. INTRODUCTION

With the vast deployment of mobile wireless systems and small cell networks, the location based services (LBS's) are getting more popular and can be used with wireless devices, including smart phones. The LBS at outdoor locations typically use Global Positioning System (GPS) services to track a person's location since there is a clear line of sight and it's simple. However, these techniques cannot be used directly in indoor environments due to limited or a lake of a clear line of sight. Another reason is that most of the time GPS signals either cannot be received in indoor locations or the signals are too weak to be used for identification or to estimate position.

There are several well-known techniques used in the area of indoor localization. Among them, Time of Arrival (ToA) [5] and Time Difference of Arrival (TDoA) are the two most popular techniques for position estimation. These techniques require extra setup and configuration to provide valid measurements. Thus, received signal Strength Indication (RSSI) based localization is the most reliable technique used for indoor localization. In this technique, the RSSI readings can be collected from any Access Point (AP) covering the region where position need to be estimated.

Femtocells are indoor low power Access Point (AP's) which provide coverage through an Asymmetric Digital Subscriber Line (ADSL). There are several types of femtocells available in the market which can help the communication operators to provide service inside the structures. Since femtocells can be considered as low power indoor base stations, their coverage faces high interference from other communication devices around especially by Macro cells [1]. Different mitigation techniques are often used to avoid or limit interference from nearby devices. Being the only base stations available for indoor structures they can be used to localize the structures [4]. RSSI can be used [12] for localization of the building. Finger printing [2-3] is one of the most common methods used in the research. There are two phases of finger printing techniques: online and offline; both phases are discussed in detail in [16].

Considering that normal distribution [7-11] is the best statistical technique to get the probabilities of continuous values between any two real numbers. We normalized [13] our data to compare it with the standard Probability Density Function (PDF) of Normal distribution. The simulation is performed in MATLAB to get the Mean Square Error (MSE) [14]. To find the location of a person in a specific zone we used MLE [15]. The simulation was carried out in Matrix Laboratory (MATLAB) and two tests were performed. In the first test, the simulation generates random samples as an input, and the results were automatically compared to our database. The second test was performed for a real scenario where we had a walk with-in the structure, total 100 samples were collected, and the model was applied to the samples to find the location. Which results in 80% of the success rate.

This paper is presenting a system to outperform the efficiency of the system presented in [16]. Here we are presenting a novel method improving the efficiency of our previous model using MATLAB simulation and real-life scenarios. It is shown that the simulation results were improved by 8% and the real-life test results were improved by 12%, compared to [16].

II. FLOOR PLAN AND LOCATOR CLIENT APP

To localize an indoor structure, it is important to first come up with the floor plan with exact measurements. In our case, the floor plan used for this research is shown in Fig. 1. This floor plan is divided into three zones, including two small zones and one larger zone. The measurements of each zone were carefully taken as mentioned in Fig. 1. The Femtocell Access Point (FAP)

was optimized according to the plan so that it can cover all the plan using minimum power. If the FAP is on full-power, there is a considerable chance of interference from the nearby base stations. It is good practice to optimize the FAP power according to the requirements in order to avoid interference mitigation. Finger printing [6] technique was used to localize the plan after putting the grid on the plan [16]. There are two phases of the finger printing techniques, including offline and online phases, which are both explained in [16].

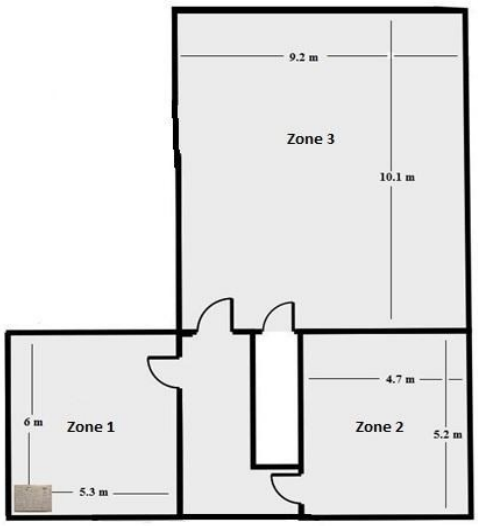


Fig. 1. Floor Plan

A new application was designed to track the RSSI values of the User Equipment (UE). The setup is configured such that whenever there is a change in RSSI values, it will automatically update the server. A high-end server needs to be configured in order to successfully transfer the messages through the network without any delay. The RSSI values are sent to the remote server by using JMS (Java Messaging Service)/MQTT server. First, an MQTT client is configured called publisher; after which it will publish the received RSSI values. Next, the server side is configured in a remote server, called subscriber, to get the incoming RSSI values in real time. The server was on open access so that the server can be easily connected from any remote area through the internet. RSSI values are gathered through an application installed in an android phone. The RSSI values are collected through the Android Telephony Service (ATS). The application can work with any android phone with little configuration. In order to send RSSI values for the phone to the server, a small process of authorization is done. The MQTT allow clients to connect with MQTT and transmit data using a user name and password.

After configuration of the code in JAVA, in its initial configuration, it shows the device Identification (ID) number, software version and the operator as shown in Fig. 2. After finishing the entire configuration, the setup is ready to be tested using a dedicated PC. Since the server is in open access, the PC can be configured at any location, as long as it can access the internet. A test is performed to check the reliability of the setup, application and server side. The Fig. 2 is showing the real time RSSI values collected through the MQTT server. The figure shows that in some cases more than one RSSI value is collected in a second. This indicates that the configuration is not set for any specific duration, but instead, it is set to report any change in RSSI values that occurs.

Fig. 2. Locator Application

III. Proposed Method

A novel method is used to track the foot prints of a person in real time. Some assumptions were made for the Algorithm. Every possible movement of a person at any direction is calculated by developing number of sequences.

The plan was divided into small clusters by applying a grid to the complete floor plan [16]. In order to track the position, we assumed that the initial location must be known. Once the initial location is identified, there are four possibilities of movement in four different directions. To develop the sequence, it is necessary to wait for four readings from the system. It can also be considered as four steps taken in any direction within the plan under the FAP coverage area. The reason for using four readings is to avoid overlapping between the sequences as with three number sequences there is overlapping between zone 2 and zone 3 which will lead to failure.

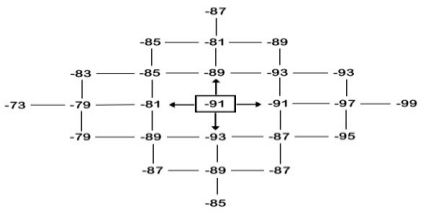


Fig. 3. Possible Directions of Movement

Fig. 3 shows all the possible directions from a specific cluster having an RSSI value of -91 dBm. More

than 1100 sequences were developed and fed into the system for future comparison with real-life tests and randomly generated sequences using MATLAB. A sample four number sequence is as follows

$$-91 \rightarrow -91 - 97 - 99$$

Proposed method used in the simulations first with reasonable improved success rate. This technique improved the success rate by 8% as shown in Fig. 4. The bars in the figure represent the success rate and the line in the figure represent the failure rate of simulation.

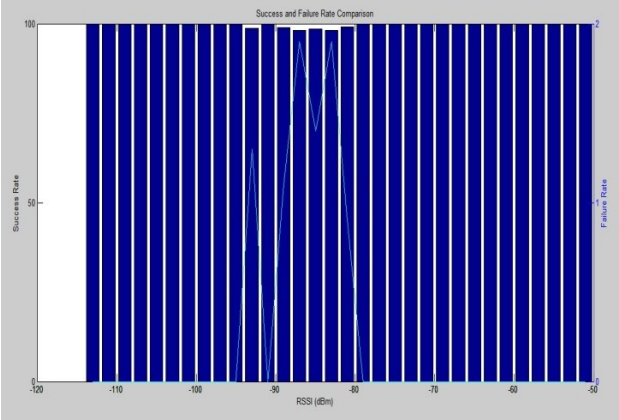


Fig. 4. Success and Failure Rate Using Proposed Method

IV. Blind Testing

To check the accuracy of the proposed model a blind test was performed in a different structure having different conditions by increasing the number of FAP's. This blind testing is regarding the deployment of femtocell network in a convention center, 52m in length and 34m wide. In total six Femtocells are deployed to cover six distinct locations of the plan.

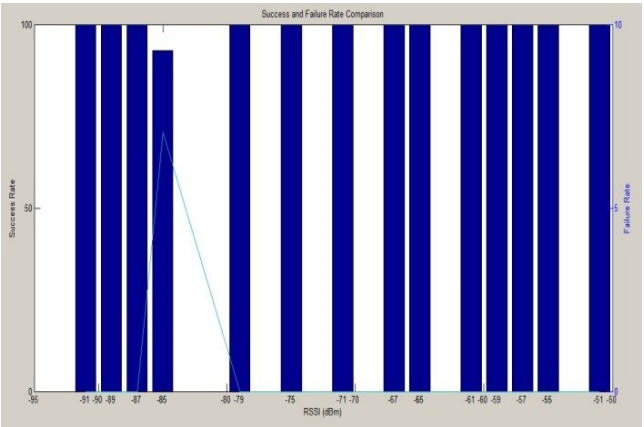


Fig. 5. Success and Failure Rate of Blind Testing

To make successful re-selections from FAP to FAP, re-selection points are optimized. The task is to correctly identify the location even between multiple FAP's. A child access terminal was used as a devise to be tracked by the system. For demonstration purposes, referee having Child access terminal start moving in the network, and the location is detected with a success rate of 92% as shown in Fig. 5. Final results are shown in Table I. The reason we are observing slightly better results for our blind test as compared to our prosed model is because for the blind test we only collect 100 samples plus the number of FAP's were increased to check the model for multiple FAP's.

Table I. Results Comparisons.

Scenarios	Success Rates	Failure Rates
Blind Decision	33%	67%
RF MAP	67%	33%
MLE	84%	16%
Proposed Method	90%	10%
Blind Testing	92%	8%

V. Conclusion

This paper presents a new model for indoor localization by presenting improved success rates as compared to the previous method. The success rates were improved by 8% in the simulation scenario where we run the simulation for a million times. Blind testing was performed to check the accuracy of our proposed method and one hundred samples were collected from the area under coverage. It is shown that the success rates in the real scenarios were improved by 12% under blind testing, further proving the accuracy of the proposed method.

References

[1] M. Morita, Y. Matsunaga, and K .Hamabe, "Adaptive Power Level Setting of Femtocell Base Stations for Mitigating Interference with Macrocells," Vehicular Technology Conference, sept. 2010.

[2] K. Kaemarungsi, and P. Krishnamurthy, "Modeling of Indoor Positioning Systems Based on Location Fingerprinting", INFOCOM, March 2004, vol.2, pp.1012 - 1022.

[3] K. Kaemarungsi, and P. Krishnamurthy, "Properties of Indoor Received Signal Strength for WLAN Location Fingerprinting", MOBIQUITOS, Aug. 2004, pp. 22 - 26.

[4] V. Khaitan, P. Tinnakornsrisuphap, and M. Yavuz, "Indoor Positioning Using Femtocells", Vehicular Technology Conference, Sept. 2011.

[5] I. Guvenc, and C. C. Chong, "A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques", Communication Surveys and Tutorials, 2009, vol. 11, pp. 107 - 124.

[6] J. Calle-Sanchez, M. Molina-Garcia, J. I. Alonso, and A. F. Duran, "Suitability of Indoor Propagation Models for Fingerprinting Based Positioning in Femtocell Networks", Microwave Conference, Nov. 2012, pp. 152 - 155.

[7] M. Zhang, and Y. Zhou, "Approximate Calculation about Standard Normal Distribution with Genetic Programming", Information and Computing, June. 2009, pp. 17 - 20.

[8] H. Okamura, T. Dohi, and S. Osaki, "Software Reliability Growth Model with Normal Distribution and Its Parameter Estimation",

Quality, Reliability, Risk, Maintenance, and Safety Engineering, June. 2011, pp 411 - 416.

[9] Roy D. Yates, and David J. Goodman, " Probability and Stochastic Processes", John Wiley and Sons, 1999, Ch. 9, p. 308.

[10] D. Campbell, M. Whitty, and S. Lim,"Mobile 3D Indoor Mapping Using Continuous Normal Distribution Transform ", Indoor Positioning and Indoor Navigation, Nov. 2012.

[11] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. j. Lilienthal, "Normal Distributions Transform Occupancy Maps: Application to Large-Scale Online 3D Mapping", Robotics and Automation, May. 2013, pp. 2233 - 2238.

[12] E. Goldoni, A. Savioli, M. Risi, and P. Gamba, "Experimental Analysis of RSSI-based Indoor Localization with IEEE 802.15.4", Wireless Conference, April. 2010, pp. 71 - 77.

[13] L. Buera, E. Lleida, A. Miguel, A. Ortega, and O. Saz, "Cepstral Vector Normalization Based on Stereo Data for Robust Speech Recognition", Audio, Speech, and Language Processing, March. 2007, pp. 1098 - 1113.

[14] Y. F. Huang, Y. T. Jheng, and H.C. Chen, "Performance of an MMSE based Indoor Localization with Wireless Sensor Networks", Networked Computing and Advanced Information Management, Aug. 2010, pp. 671 - 675.

[15] N. Pritt, "Indoor Positioning with Maximum Likelihood Classification of Wi-Fi Signals", SENSORS, Nov. 2013.

[16] M. Ilyas, O. Bayat and O. Ileri, "Indoor location estimation by using MLE based algorithm on smallcell networks," *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, Malatya, 2015, pp. 690-693

An Integrated Real-Time Water Quality and Usage Monitoring and Control System

Argho Das
Faculty of Science And Technology
American International University - Bangladesh
Dhaka, Bangladesh
arghodas555@gmail.com

Abdur Rahman Rubayet
Faculty of Science And Technology
American International University - Bangladesh
Dhaka, Bangladesh
arrubayet@gmail.com

***Abstract*—Fresh water is a vital resource for the survival of our population. In countries like Bangladesh where clean water is scarce, overusing or wasting household water limits the availability of it for other communities to use for drinking, cleaning, cooking or growing and thus contributes to disease, illness, or agricultural scarcity/starvation. In relation to this the key concern is to develop an efficient, cost effective and real-time system that monitors leakage in the tank, water overflow and the turbidity of the water. Additionally, the system does not require any user interaction.**

***Index Terms*—Real time, Sensors, Arduino, Water Quality, Ultrasonic transmitter**

I. Introduction

Water is a daily necessary resource for life, health, economic development and the ecosystem all over the world. As water is precious to everyone, its availability and quality are essential. We all need to drink water to keep our bodies hydrated; not just any water, but clean drinking water. It is recommended that adults should consume at least 8 glasses of water per day. Water is involved in every bodily function from digestion and circulation through to the control of body temperature and the excretion of waste products. The water in our bodies is continually being used or lost from the body. This is how we maintain that 70% of body water volume. Food, soups, soft drinks and beverages all use water to make them. Therefore we must try to reduce water wastage and use clean water. Hygiene is next on the priority list. It is said that "cleanliness is next to godliness". Many waterborne diseases like cholera, typhoid, dysentery, dengue fever, and viral hepatitis A are results of people utilizing dirty water for drinking or bathing. Water can get contaminated by chemicals, viruses, or bacteria that might later affect those who use it. Some waterborne diseases, like cholera, are not only contagious if not taken care of in a quickly manner, but they can also be fatal. Malaria may not be a wholly waterborne disease, but is a result of mosquitoes breeding in stagnant water. It is safe to say that stagnant water is polluted water, therefore not safe for consumption. According to kid's health, water helps the body get rid of waste in urine, sweat, and solid waste. It is also found in the lymph nodes, where it helps the body build up and sustain the immune system. Whatever food growing method you are involved in, let it be known that you will not be harvesting if there is no water. This is also true if your crop does not get the required amount of water to fully develop and bear fruit. Whether your crops are rain fed or irrigated, the magic word here is water. Drinking Water Helps Maintain the Balance of Body Fluids. Human body is composed of about 60% water. The functions of these bodily fluids include digestion, absorption, circulation, creation of saliva, transportation of nutrients, and maintenance of body temperature. Water Can Help Control Calories. Water helps keep skin looking good. Our skin contains plenty of water, and functions as a protective barrier to prevent 11 excess fluid loss. Water helps our kidneys. Body fluids transport waste products in and out of cells. Water helps maintain normal bowel function. As we can see that without drinking water human existence is almost impossible. Our proposed system is fully automated. In our proposed system we can always get fresh water with minimum human interaction. Our system does not allow polluted water to be supplied to the user. The system also indicates if it is time to clean the tank or not. Water wastage is a huge problem in our country. Using this system we can reduce the water wastage by a large margin.

II. Related Work

An automated water utilization monitoring system is proposed in [1]. This paper proposes an effective way of controlling the wastage of water at home, colleges, hospitals, industries, shops, and malls by the use of LabVIEW software and Wireless sensor nodes. This particular water usage tracking device includes three separate modules. They are Wi-Fi sensor module, Central module, Server module. The wireless sensor module collects the data and records from each water outlet. At the server all of the computation is performed and the signals are generated and also the central module acts as a user interface between the server and the sensor module. The system is fully automated and Iot based. The detection of the finding leakage is done using fluid mechanics and kinematics physics based on harness water flow rate data gathered using flow liquid meter sensor and Arduino UNO as the microcontroller. In [2] From the given results we can see that the proposed method is able to work stably and efficiently to determine the location of the leakage which has a maximum distance of 2 meters, and it's able to determine the

leak location as close as possible with flow rate about 10 liters per minute. A real time water quality monitoring is done by using data acquisition, method and transmission with increase in the wireless device 15 network technology in the internet of things. [3] The gathered results and values from the sensors are interfaced by microcontroller and the processed results and values remotely to the core controller ARM with a WI-FI protocol. Parameters of water like temperature, pH level, water level and CO2 by multiple different device nodes are selected by WQM. This methodology sends the information directly to the web server. The uploaded data is refreshed within a specific interval time. The data can be accessed from the web server from any part of the world. LDR and LED sensor based water turbidity meter is used in [4]. Nephelometric Method is a method of measuring the turbidity of water by passing a light source on water so that the intensity of light reflected by the substances causing turbidity can be known. With the use of led as a light source and photodiode as a light detector, and combined with processing using Arduino Uno then the voltage obtained from the LDR sensor in the form of analog data is processed into digital data and can be displayed in the LCD. The proposed system in [5] is composed with different sensors such as a water flow tracker, pH detector, water pressure valve, and as a central controller, a raspberry PI. To ensure fair and sufficient flood control for each link, a water control machine is operated via a web portal depending on water flow sensor value. Previous studies on the tracking and identification of water pipelines in water pipes have been performed. In [6], the authors conducted research using web services and also Zigbee as a communication system, as well as some sensors, such as water flow sensors, level sensors and temperature sensors in order to track the level of water flow. In addition to web monitoring, the sensor owner may also provide the owner's personal cell phone number with important details about the flow of water via SMS. The authors conducted an experiment in this [7] to detect water pipeline leakage. The study investigated the effect of different pipe diameters on water flow pressure in the pipes and changes in the temperature around the pipe. An FSR sensor has been used to track changes in the diameter of the pipe, and temperature sensors are used to measure the temperature within the pipe. In this analysis, the authors used a 40 mm diameter Plastic pipe with a steady concentration of 3 bar, as well as the other research in [9] studied movement in the tube wall caused by collisions between water flows to the walls of the pipe. Using an MEMS sensor the vibration is calculated. If there is a leak in the vessel, leakage is evaluated by comparing the vibration of the normal water flow and the vibration. Tests are determined by varying the pressure from 3 to 10 bar with a continuous rate of 300 m3 / h of flow of water. In [10], a study was carried out that evaluated the usage savings of water from the tap using wireless sensor nodes. This study utilizes Rfbee sensors as a transmitter and an information receiver to collect data obtained from the flow rates transducer. Data obtained by Rfbee sensors will be distributed to a device connected directly to the Rfbee sensor via a wireless connection. In [8], research has been carried out to track and regulate flow of water via a web application. With the Arduino Hall Effect Flow Sensor, liquid flow is determined, while Raspberry PI manages the solenoid electro valve used to close or open the liquid motion through to the pipe.

III. Current System

The current system is totally manual. As a result a lot of water is wasted. If there is any leakage in the tank or any pipes then there is no way of detecting in the current system. Water loss through leaking pipes constitutes a major challenge to the operational service of water utilities. A significant amount of water is lost in the water supply system. Water leakages have been a major problem for many regions around the world. Fresh water is a vital resource for the survival of our population. Seeing as less than 1% of the world's water is freshwater and available for us to consume there are limitations that factor into our carrying capacity as a population on Earth including the availability and distribution of freshwater. Reduction of water leakages is an important goal for many countries in the world as it will mean a reduction in the amount of money and energy required on producing and pumping 12 water and also satisfaction of consumer needs through improved reliability of the system. Water is a fundamental human need. Each person on Earth requires at least 20 to 50 liters of clean, safe water a day for drinking, cooking, and simply keeping themselves clean. Polluted water is not just dirty, it's deadly. Some 1.8 million people die every year of diarrhea diseases like cholera. Tens of millions of others are seriously sickened by a host of water-related ailments which many of which are easily preventable. So clean water is essential in our day to day life. In the current system there is no way of detecting dirty water. Water overflow is a major issue in our country. When the water tank is empty people turn on the pump to fill up the tank but people usually forget about the pump, as a result water falling from the overflow pipe is seen. After seeing the overflow pipe people usually turn off the motor. By this time a lot of water has been wasted. Sometimes people ignore or don't even notice the overflow pipe and let the water fall until they finish the work at hand. This is an alarming problem in our society. The proposed system will solve water turbidity, water leakage and overflow problems of the tank. This system will save electricity and reduce wastage of water.

IV. Proposed System

Our proposed system gives the current state of the water turbidity, water level and leakage detection in real time. If there is any problem the system will alert the user. All the features are integrated together in one system. This system is cost effective than the conventional systems we use. The whole system is totally automated. The tank will contain two pipelines. One is the supply line which connects the tank with the pump and another one is the overflow pipe. The overflow pipe is used to release the excess water when the tank becomes full. Leakage may occur from either pipe or from the tank. The

condition of the pump must be checked too. If the pump is damaged then the water level will not rise. It is very hard to detect a leakage in a system. The current water level of the tank can be checked using the sensors placed inside the tank. Three sensors will be used for detecting the leakage of the system. The system monitor will display the condition of the water constantly. If the water becomes muddy or dirty then the sensor will detect the turbid and inform the user that the water is dirty. It will be shown on the monitor. If the water is too dirty and is unusable then the sensor will inform the user 13 that the water is too dirty. If the water is constantly dirty then it will inform the user to clean the tank.

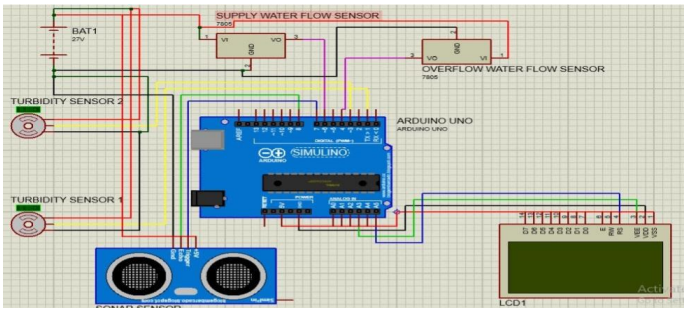


Fig. 1. System Design

The functionality of the proposed system is given below,

- Initially a sensor will be placed in the tank. That sensor will be placed at the top of the tank.
- Then two more sensors are needed at the overflow pipe and the supply pipe to ensure that water is flowing through those pipes.
- First we must calculate how long it takes to fill up the water tank. Using this information we will be able to find out if there is any leakage or not. If there isn't any leakage then the water tank will be filled with in the allocated time. If any significant delay occurs during this process then we can assume that there must be leakage in the pipe or the water tank or the pump is faulty.
- When the water is flowing through the supply pipe, the sensor under the supply pipe will detect water first. This ensures that there is no leakage in the pipe and the pump is working properly because if there was any leakage then the water wouldn't flow to the tank and the sensor would not detect any water.
- Another sensor is placed at the top of the tank and it detects the water level.
- A sensor will scan the tank from top to bottom and give feedback to the user about the condition of the tank's water level. If the water is at the top of the tank then the sensor will inform the user and the system monitor will show that the tank is full. If the water is at the middle of the tank then the monitor will inform the user that the tank is half full. If the water is at the bottom of the tank then the monitor will inform that the tank is empty and the pump will be turned on automatically by the system.
- The sensor which measures water flow in the supply pipe will ensure that water is in the supply pipe but it will not ensure that the water is going to the tank. To ensure that both sensors must detect water in the tank, only then we can be sure that water is flowing through the supply pipe into the tank.
- For the pump to stop pumping water both the top sensors need to be triggered. If the tank is in good condition without any leakage then 14 we can assume that the pump or the supply pipe is damaged or faulty.
- The sensor is responsible for ensuring that the water in the tank is usable and drinkable.
- Two more sensors will be used in the system. One will be placed at the top of the tank and another will be placed at the bottom of the tank.
- Two sensors are used because each sensor can scan 100 liters of water each. The sensors will give feedback to the user about the condition of the water.
- If the water is clean then the monitor will show that the water is clean. If the water is dirty then the monitor will have the message that water is dirty. If the water is very dirty and totally unusable then the system monitor will show that the water is too dirty.
- If the water is constantly dirty for more than 15 days then it will inform the user to clean the tank.

V. Hardware Details

For this system we use some sensors like turbidity sensor, ultrasonic sonar sensor, 5V Relay Module Arduino and water flow sensor. We also use some other hardware like Arduino UNO, LCD display, battery, breadboard, water pump and jumper wire for their connection. The model numbers, specifications and details of these sensors and other hardware are given below:

Ultrasonic Sensor (HC-SR04): HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that is: Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turn on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a micro-controller or microprocessor.

VI. Proposed Algorithm

We developed our full system by separating it into two parts. One is Turbidity checking and the other one is Water flow and leakage checking. In the Turbidity checking part we use the algorithm implemented for water turbidity checking. Where we have to consider ultrasonic sonar sensor's data first because the tank might be empty. So we compare the sonar sensor result with the 5% of the tank's maximum height. If the tank is not empty then we will consider turbidity sensors data for measuring water quality. Total result will be the average of

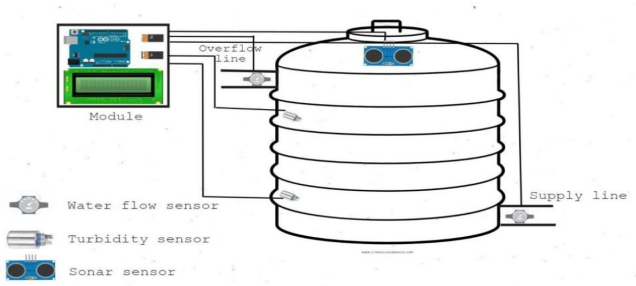


Fig. 2. System Implementation

those turbidity sensors data which will output non zero data. Because if any turbidity sensor outputs zero then it means at that height there is no water. Like, if there is X no of turbidity sensors and the tank is 50% filled up so turbiditySensor1 and turbiditySensor2 gives non zero data others give zero as data. Then the total result will be the average of turbiditySensor1 and turbiditySensor2. So, total result = (turbiditySensor1+ turbiditySensor2)/X Then the clean Water Range, dirty Water Range, very Dirty Water Range one after one. If the total result of sensors is lesser than dirty Water Range then water will be considered clean water. If the total result of sensors is greater than or equal to dirty Water Range but lesser from very Dirty Water Range then water will be considered dirty water. If the total result of sensors is greater than or equal to the very Dirty Water Range then water will be considered as very dirty water. If the water is dirty or very dirty we have to wash the tank manually and an alarm will be sounded. In the Water flow and leakage checking part we use the algorithms implemented for both no flow, underflow and overflow checking and leakage and faulty pump checking. Here we derive a hybrid solution from these two algorithms. Where we have to consider the supply water flow sensor's reading because supply water flow sensor's reading will assure if the water is flowing into the tank or not. If the reading of the supply water flow sensor shows no reading then it will be considered as no flow. But if there is not no flow then we have to consider either normal flow or underflow. If the reading of the supply water flow sensor shows lower reading than our regular water flow criteria then it will be considered as an underflow. But if it is not underflow then we have to consider either normal flow. When water flow is normal then there can be two consequences: Overflow, Leakage or faulty pump. For checking overflow we have to consider supply water flow sensor, overflow water flow sensor and also ultrasonic sensor. If the water is flowing from the supply pipe into the tank (supply water flow sensor reading is within average limit) and at the same time the overflow water flow sensor is also giving reading then it will define that the water may be reached to the tank's maximum storing limit. Though it can be falsified if there is water in the sensor when the water level of the tank is not actually that high. So we need to consider the ultrasonic sensor because its reading will assure if there is actually an overflow or not. If the reading of the ultrasonic sensor shows that water level of the tank has reached 90% of the total height of the tank then it will be considered as an overflow. 39 For checking leakage or faulty pump we have to consider an ultrasonic sensor. The sensor will be placed to the top of the tank. It will measure the current water level distance. Whenever the pump is on continuous time interval calculation will be started. If the interval time is 10minutes and the water level cannot reach the lowest calibration of the water level system will raise an alarm that the pump is not working properly or there may be a leakage. In both cases overflow detection and leakage or faulty pump detection, the system will automatically turn off the pump For further discussion we have used a table for better understanding and we define the variable name in details which we have used in our algorithm. Configuration Table:

VII. Experimental Results

I have developed a lab prototype for simulating the project we are proposing. There is reference picture of our prototype system - Some typical dust was put on the surface of the water

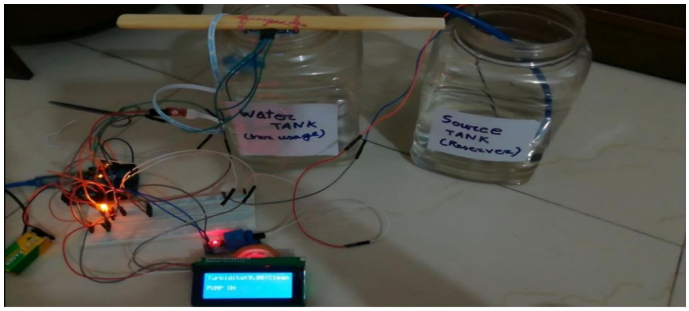


Fig. 3. System Prototype

tank. Turbidity sensor was kept active for 1 hours without any interruption.

VIII. Conclusion

The paper presents a detailed discussion and ways to improve the existing water distribution systems. Also, a low cost, less complex and much more efficient water distribution system is proposed. The implementation enables one sonic sensor to provide the information of water level to consumers through a LED display, two turbidity sensors to measure the turbidity level of the water and give appropriate feedback and suggestions to the user and two water flow sensors to detect any occurrence of leakage and overflowing from the system.

TABLE I
CONFIGURATION TABLE

System Configuration Table	
Data	Description
water level	Get water level height of the tank using ultrasonic sensor.
minimum level	Estimated time to reach minimum level
getquality	Get water quality of the tank using turbidity sensor (tSensor), "getquality" will be the average result of total (X) turbidity sensors data. So, getquality=(tSensor1+ tSensor2+.....+ tSensorX)/X
NTU	Nephelometric Turbidity Units
cleanWaterRange	0-599 NTU
dirtyWaterRange	600-900 NTU
veryDirtyWaterRange	Above 900 NTU
M	Maximum height of tank
MINsf	Minimum limit of water flow through supply water flow sensor
AVGsf	Average limit of water flow through supply water flow sensor
MINof	Minimum limit of water flow through overflow water flow sensor
AVGof	Average limit of water flow through overflow water flow sensor
flow rate	Water flow sensor reading
current time	Current calculated time
Estimated time	Estimated time to reach minimum level
current height	Get current height using ultrasonic sensor
min level	Minimum level to reach with estimated time (t)
supply flow	Current Flow Rate from flow sensor of supply pipe
estimated flow	Estimated flow rate exist in the flow sensor

TABLE II
TABLE TO TEST CAPTIONS AND LABELS

Turbidity Sensor Data Collection Table			
Time	Dust Position	Turbidity level (upper sensor) NTU	Turbidity level (lower sensor) NTU
0 minutes	At the top	489-683	97-102
30 minutes	Almost at the bottom	205-244	689-773
60 minutes	Totally at the bottom	97-254	665-753

In conclusion from our current system we can successfully measure the turbidity level of the water and give feedback to the user, water level in the container, detect leakage and stop the water from overflowing. This allows us to save time and reduce water wastage since the whole system is fully automated and no user interaction is required. The experimental setup can be improved by making it IoT (Internet of things) based. Our current system requires wires to transfer data to the LED display in order to inform the user. If we convert the system into IoT (Internet of things) then the use of wires will be limited and the user can check from anywhere

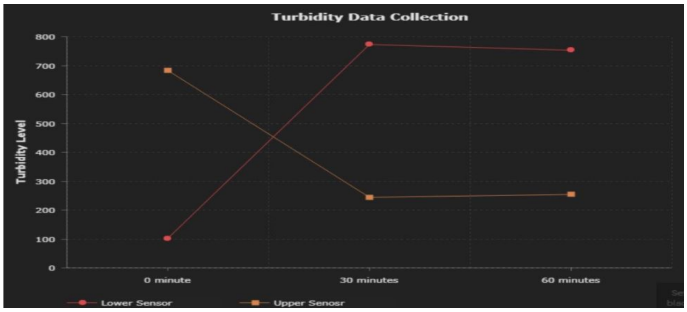


Fig. 4. Turbidity Sensor Data Collection Visualization

and anytime from his electronic devices such as smartphones, laptops and tablets. This will make the interaction between the user and system much more easy and comfortable. Also by implementing this feature we can make the system faster and efficient.

REFERENCES

[1] Arun Mozhi Devan P., Pooventhan K., Mukesh Kumar C., Midhun Kumar R. (2019) IoT Based Water Usage Monitoring System Using LabVIEW. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, Cham.

[2] R F Rahmat, I S Satria, B Siregar, R Budiarto "Water Pipeline Monitoring and Leak Detection using Flow Liquid Meter Sensor" IOP Conference Series: Materials Science and Engineering, Volume 190

[3] K. Spandana, V.R. Seshagiri Rao "Internet of Things (Iot) Based Smart Water Quality Monitoring System" Internet of Things (Iot) Based Smart Water Quality Monitoring System Vol 7, No 3.6 (2018)

[4] A. P. U. Siahaan, Nogar Silitonga , Muhammad Iqbal, Solly Aryza, Wirda Fitriani, Zuhri Ramadhan, Zuraidah Tharo, Rusiadi, Rahmad Hidayat, H. A. Hasibuan, M. D. T. P. Nasution, Ali Ikhwan, Zulfi Azhar, Mhd. Irwan Dwitama Harahap "Arduino Uno-based water turbidity meter using LDR and LED sensors" International Journal of Engineering & Technology, Vol 7, No 4 (2018)

[5] Joy Shah "An Internet of Things Based Model for Smart Water Distribution with Quality Monitoring" International Journal of Innovative Research in Science, Engineering and Technology, Vol. 6, Issue 3, March 2017

[6] Rani M U, Kamalesh S, Preethi S, Shri C K C and Sungaya C 2013 Web based service to monitor water flow level in various applications using sensors Int. J. Biological, Ecological and Environmental Sciences (IJBEES) 2 119-122

[7] Sadeghioon A M, Metje N, Chapman D N and Anthony C J 2014 SmartPipes: Smart wireless sensor networks for leak detection in water pipelines J. Sensor and Actuator Network (JSAN) 3 64-78

[8] Suresh N, Balaji E, Anto K J and Jenith J 2014 Raspberry PI based liquid flow monitoring and control Int. J. Research in Engineering and Technology (IJRET) 3 122-125

[9] Rizwan M and Paul I D 2015 Leak detection in pipeline system based on flow induced vibration methodology in pipeline Int. J. Science and Research (IJSR) 4 3326-3330

[10] Yano I H, Oliveira V C, Araujo E V, Campagnuci A G, Fabiano B and Demanboro A C 2014 Wireless sensor networks for measuring the consumption of save water taps American J. Applied Sciences 11 899-905

A brief Survey on Deep Learning based Recommender Systems and Applications

Muhammad Sanwal
Department of Computer Engineering
Antalya Bilim University
Antalya, Turkey
muhammad.sanwal@antalya.std.edu.tr

Alper Özcan
Department of Computer Engineering
Nişantaşı University
Istanbul, Turkey
alper.ozcan@nisantasi.edu.tr

***Abstract*—In the current era, the information on the internet is increasing at a rapid pace on daily basis. The Internet contains essential and non-essential information and retrieving a piece of useful information is very challenging for users. To counter such issues conventional recommender systems have been proposed in the previous decade. Recently, deep learning methods have shown promising results in the field of image processing and computer vision. By realizing such a boost in performance researchers have proposed deep learning-based models for recommendation systems. In this paper, we have analyzed deep learning-based recommendation models that suggest appropriate items to the target users. Further, we have discussed the advantages and disadvantages of the deep learning methods in this specific area. Moreover, this survey highlights the important factors and popular applied methods in the literature to give an appropriate sight to those researchers who are interested in this area of research.**

Keywords—Deep Learning, Recommender Systems

I. INTRODUCTION

Nowadays, the data on the internet is increasing exponentially; it is challenging to organize and handle such an amount of data. Many online platforms, such as Amazon, Netflix, Ali Express, Facebook, and YouTube, take a large amount of data daily. These platforms offer various products to the customers, and it will be challenging and time consuming for the customers to choose an item from such a vast platform. Recommender systems are useful and powerful tools that provide suggestions to the user by using different techniques, especially in the E-commerce domain [1]. In such platforms, people are usually interested in product ratings and reviews before purchasing any item. In such a situation, personalizing user requirements is a better strategy to facilitate a better user-experience [2].

There are multiple strategies to build a recommender system, such as user-based, item-based. In general, a user profile is generated based on his preferences, interactions, purchasing habits, and previous history [3]. Mostly, the recommender systems are divided into three main categories: content-based systems, collaborative filtering (CF), and hybrid systems [4].

- *Collaborative Recommendations:* Collaborative recommendations: This method uses the experiences and past behaviors of a specific user and measures the similarities between previous and current products of interest to the user.
- *Content-based Recommendations:* This approach recommends the items from the database which are similar to the previously liked items from a particular user, and it computes the similarity between the characteristics of the items.
- *Hybrid Approaches:* It recommends the items by combining the above approaches, which can be more precise and generate better recommendations.

In the previous decade, deep learning models have shown promising improvements and development in image processing, computer vision, information theory, and natural language processing [5]. Because of its sophistication, deep learning models are becoming more popular in every field of study. In industry, multiple organizations are using deep learning-based recommender systems to enhance the user experience on websites and mobile applications [6].

In this survey paper, we are elaborating on different deep learning techniques used in recommender systems and their applications. Though the popularity of recommender systems has increased in the past decade, there are still limited surveys present in the literature. To the best of our knowledge, there are systematic and in-depth reviews that have been published for the recommender system, but deep learning-based reviews with proper systematic study along with its application have not been published frequently in recent years. Two detailed studies [46, 47] have been published in the literature about the deep learning-based recommender systems. Both of these studies are published a couple of years ago and it doesn't include some latest researches present in the literature. In this survey, the new state of the art and latest deep learning-based recommender systems have been discussed briefly to give insights to the users interested in this topic such as [48].

Betru et al. [10], and Liu et al. [11] have categorized the deep learning-based models, but it lacks some essential insights such as its applications in the real world. Recently, new deep learning methods have been implemented that utilize some advanced techniques for classification and recommendation purposes.

In this paper, we have analyzed different deep learning methods that have been implemented in recent years, along with applications. This study will provide deep insights into deep learning-based modern recommender systems that are state of the art. This paper aims to provide an overview and

deep insights into the recommender systems for the researchers interested in this particular field of study. The rest of the article is organized as follows:

Section 2 will discuss an overview of recommender systems and previous deep learning methods present in the literature. Section 3 will discuss the current state of the art deep learning methods and their categories along with comparisons. In Section 4, future research directions and issues will be discussed.

II. OVERVIEW OF RECOMMENDER SYSTEMS

The deep learning methods are highly anticipated in recommender systems because of their precision and sophistication. In the past and current decade, many researchers have applied various deep learning methods for recommender systems. In this section, we will describe some basic terminologies for the recommender system and an overview of previous studies.

Recommender systems are estimators to predict the user's preferences, such as an item of their interest. Researchers are also applying deep learning methods and multiple well-known machine learning techniques such as matrix factorization and collaborative filtering used in [12, 13, 14]. Deep learning is the branch of machine learning consisting of neurons and different layers such as input layer, hidden layers, and output layer. The primary purpose of the deep learning models is to learn multiple representations on multiple levels on the given data and optimize objective functions such as stochastic gradient descent (SGD) [15]. Deep learning methods have shown promising results in both supervised and unsupervised tasks. Below some of the famous architectural paradigms are given for a better understating of this survey.

- *Convolutional Neural Network (CNN):* It is a type of neural network which consists of input, output, and multiple hidden layers. It is a forward propagated method with some convolutional layers, along with pooling operations [16]. Generally, it enhances efficiency and precision by capturing the dataset's global and local features.
- *Recurrent Neural Network (RNN):* It uses sequential data and used for temporal problems such as language translation, Natural Language Processing (NLP), and speech recognition [16].
- *Autoencoders (AE):* It is an unsupervised artificial neural network that encodes the data at the beginning, and decodes the compressed data to such a representation that is similar to the original data. It is used in tasks such as image denoising and feature extraction [17].
- *Restricted Boltzmann Machine (RBM):* It is a shallow neural network consist of two layers; input and hidden layer. It is used for regression, feature learning, and classification problems.

Chang et al. suggested a customized music recommendation framework (PMRS) based on the CNN approach. The CNN approach classifies music according to the audio beats of the music in different genres. Collaborative filtering (CF) was used as a suggestion to merge CNN output and log files that suggest music to the user. The log file contains all PMRS users' history. The PMRS extracts the history of the user from the log file and suggests each kind of music. For evaluation, a million-song dataset (MSD) was used [18].

Y. Chu et al. proposed the use of gated recurrent units in recurrent neural networks to solve the time series problem. The network considers recent ratings or actions of a user as a chain, and each hidden layer models the rating or behavior of a user. Moreover, the gated recurrent unit and backpropagation neural network are combined to improve the precision [19].

To mitigate problems such as cold start and data sparsity, H. Chu et al. [20] proposed an autoencoder algorithm that integrates autoencoders and collaborative filtering algorithms knowns as the AE-CF for recommendation purposes. The proposed AE-CF algorithm learns deep latent factors from the user's data and ratings. Using MovieLens, a public movie dataset, they achieved better results from the previous studies.

F. Yang et al. [21] proposed an algorithm to further extend the RBM process to interact with implicit feedback. In the proposed method, model parameters can be effectively learned with the contrasting divergence algorithm. The proposed RBM approach can explicitly predict preferences for a new consumer concerning a few items compared to other methods. It also preserves confidentiality by keeping user information locally and only sends modified parameters to a central server. The findings on real data with multiple million records show that it works superior to other standard methods.

A. Benefits of Deep Learning

Deep learning models in the domain of recommender systems have several benefits over the traditional machine learning methods such as Matrix Factorization. Neural networks have a sophisticated architecture that helps to predict better in case of recommendations. The most prominent feature that all neural architecture contains is that they are end-to-end differentiable and it contains the biases for the inputs. As such, if the model can control an intrinsic structure, deep neural networks should be useful [22].

Moreover, we sum up the strengths of profound deep learning-based recommendation models that readers should consider when attempting to use for practical purposes.

- *Non-linearity:* Unlike linear models, deep neural networks can model non-linearity in data by non-linear functions such as ReLu, sigmoid, tanh, etc. This property allows complex user interaction patterns to be captured. Conventional approaches, such as matrix manufacturing and sparse linear models, are linear [23].
- *Representation Learning:* In learning underlying explanatory factors, deep neural networks are very effective and efficient. Generally, information about the users and items is present in the real world, such as their location, purchasing habits, and comments about specific items. By making the use of such information about users and items leads us to develop better-personalized recommender systems.
- *Sequence Modeling:* In sequential input data and their applications, the deep neural networks have already demonstrated impressive results. The applications such as natural language processing and chatbots are

the most used applications in the literature and industry. Moreover, RNNs and CNNs are commonly used and versatile in data mining tasks.

- *Flexibility:* Deep learning models are highly flexible because of highly qualitative frameworks present in the industry, such as Keras, Tensorflow, and Pytorch.

B. Limitations

In this section, we will summarize some of the significant challenges and limitations in deep learning-based recommender systems.

- *Data Requirement:* The data is the key to train your deep learning-based model. If the data is too small, it can end up with a poor approximation. Afterward, the problem of underfitting and overfitting are most likely to originate with a too-small dataset. Generally, the deep learning-based models are considered data-hungry models that require a large amount of data.
- *Parameter Tuning:* Generally, parameter tuning is considered a typical machine learning problem, but, in some cases, deep learning could introduce new hyperparameters for specific models, such as in this research [24].

III. Deep Learning-Based Models and Applications

This section will categorize the deep learning-based recommender systems and describe the applications presented in the previous studies within this domain. We have classified the recommender systems into two following categories.

- Recommendations with Neural Networks
- Recommendations with Hybrid Approaches

A. The Recommendation with Neural Networks and Hybrid Approaches

These deep learning-based models utilize CNNs, RNNs, RBMs, and MLP, based on the recommender system. These techniques have been used extensively in the literature for a decade because of their high computation and flexibility. Moreover, some of the modern studies utilize hybrid techniques by combining multiple techniques in the recommendation systems. The benefit of hybrid techniques is to borrow such elements from specific architecture to perform better when integrated with conventional methods. Below, we summarize some of the studies that are using the above-mentioned techniques in the literature.

To address the limitation of previously proposed model, Donghyun et al. proposed a CNN-based model. The first limitation was to ignore the contextual information in words given in the user-item comments sections because of their modeling methods. The second limitation in the proposed literature is that it does not consider gaussian noise differently along with scores, while gaussian noise depends on item statistics. In the proposed model, they integrated CNN with probabilistic matrix factorization (PMF) to extract contextual information and gaussian noise individually [25]. The proposed approach was tested on three different real-world datasets, and this model outperformed the previous state of the art models. Moreover, the use of CNN's in other proposed methods can be studied in [26, 27].

Hanjun Dai et al. [28] proposed a Deep Coevolutionary Network to address the limitations of strong parametric assumption in the evolving relationship between user and item and their latent features, which does not reflect reality. This paper proposed a model that learns user and item features based on an interaction graph between them. Recurrent Neural Networks (RNNs) are used to capture the mutual influence between the user and item. The proposed model showed promising improvements from the previously well-known methods. The other applications of RNNs can be studied in [29, 30, 42].

Kostadin et al. proposed a Restricted Boltzmann Machines (RBM) model based system on Collaborative Filtering Framework that extends the previous RBM method in many important directions[31]. Firstly, whereas earlier RBM research was based on modeling the correlations between item scores, in a single hybrid, non-IID framework, they modeled both user-user and item-item correlations. In comparison to multinomial variables, real values are used in the visible layer that takes advantage of the natural order between user ratings. Finally, they explored the ability to merge the original training data with the RBM model data in a bootstrapping way. Furthermore, similar approaches consisting of RBM are used in [32, 33, 43, 44].

Paul et al. suggested a deep learning model to recommend YouTube videos to the users and the proposed work is divided into two stages, based on the classical distinction of retrieval. Firstly, a detailed deep generation model is generated, and then a separate deep ranking model is defined. They also offer practical lessons and perspectives from the architecture, iteration, and maintenance of a massive user-faceted recommendation framework [34]. Further, similar studies regarding MLP are [35, 36].

Hanbit et al. suggested a hybrid Deep Learning News Recommender method [37].In the proposed model, the instantiation of CHAMELEON architecture is used that is based on composed of deep learning architecture. The architecture consists of two modules: learning the representation of news articles based on their text and metadata, and the other for providing suggestions based on sessions using Recurrent Neural Networks. In this research, they predicted the next possible news which a user might read in a specific session.

Yin Zheng et al. [38] suggested the implicit CF-NADE, an independent neural model for collaborative filtering with implicit feedback. First the implicit input of an entity was converted into a similar vector and then the likelihood of a vector weighted by the confidence vector was modeled. They measured the output of CF-NADE implicitly on a dataset from a common digital TV streaming service. Then the output of the implicit CF-NADE model is compared with common implicit matrix factorization. Experimental findings indicate that implicit CF-NADE exceeds the baseline models substantially.

The following table illustrates the different categories mentioned in this study, along with reviewed publications.

TABLE 1: CATEGORIZATION OF DIFFERENT STUDIES WITH PUBLICATIONS

CATEGORY	PUBLICATIONS	ACCURACY (RMSE OR MSE)
CNNS	[25, 26, 27]	0.78 - 0.80
RNNS	[28, 29, 30, 42, 46]	0.72 – 0.82
RBMS	[31, 32, 33, 43, 44]	0.66 – 0.92
MLP	[34, 35, 36]	0.75 – 0.94
HYBRID METHODS	[37, 38, 39, 40, 41, 45]	0.65 – 0.84
OTHERS	[8, 9]	-

In the table above, the accuracy of models is given, as many of the proposed researches calculates RMSE, MAE, and MSE. The perception about these measurement can not be considered accurate as each of the model uses different datasets for performance evaluation. But the above table provides the performance sight of all mentioned categories.

IV. CONCLUSION

The increasing application of recommendation systems, such as videos, social networks, etc., makes the recommendation system a critical and incredibly challenging task. In recent years, the research trend has been to tackle such tasks by creating deep neural networks and learning about the latent information to structure the information and its relationship with users. Most traditional approaches construct hand-crafted features for recommendations and are time-consuming, inefficient, and contain issues like sparsity and cold start. The deep learning methods are more efficient when enough data is provided and can automatically extract raw data features.

In this article, we evaluated and analyzed some important work on deep learning systems of modern literature. We have categorized the literature publications into different categories and highlighted some important work in this specific field. Further, we also addressed the benefits and drawbacks of deep learning methods. We hope that the survey will provide readers with an understanding of the main aspects of this area, clarify the most important progress and clarify future studies

REFERENCES

[1] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to Recommender Systems Handbook. In: Ricci F., Rokach L., Shapira B., Kantor P. (eds) Recommender Systems Handbook. Springer, Boston, MA, 1-35.

[2] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. Recommender systems: an introduction.

[3] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17, 6 (2005), 734–749.

[4] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2018. Deep Learning based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 1, 1, Article 1 (July 2018), 35 pages. DOI: 0000001.0000001.

[5] Naiyan Wang and Dit-Yan Yeung. 2013. Learning a Deep Compact Image Representation for Visual Tracking. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13), C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates Inc., , USA, 809-817.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In Recsys. 7–10.

[7] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netfix recommender system: Algorithms, business value, and innovation. TMIS 6, 4 (2016), 13.

[8] Gao, Tingwei & Li, Xiu & Chai, Yueting & Tang, Youhua. (2016). Deep learning with consumer preferences for recommender system. 1556-1561. 10.1109/ICInfA.2016.7832066.

[9] B. Ouhbi, B. Frikh, E. Zemmouri and A. Abbad, "Deep Learning Based Recommender Systems," 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, 2018, pp. 161-166, doi: 10.1109/CIST.2018.8596492.

[10] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. 2017. Deep Learning Methods on Recommender System: A Survey of State-of-the-art. International Journal of Computer Applications 162, 10 (Mar 2017).

[11] Juntao Liu and Caihua Wu. 2017. Deep Learning Based Recommendation: A Survey.

[12] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in Computer, vol. 42, no. 8, pp. 30-37, Aug. 2009, doi: 10.1109/MC.2009.263.

[13] Jinbo Zhang, Zhiqing Lin, Bo Xiao and Chuang Zhang, "An optimized item-based collaborative filtering recommendation algorithm," 2009 IEEE International Conference on Network Infrastructure and Digital Content, Beijing, 2009, pp. 414-418, doi: 10.1109/ICNIDC.2009.5360986.

[14] R. Singla, S. Gupta, A. Gupta and D. K. Vishwakarma, "FLEX: A Content Based Movie Recommender," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154163.

[15] Li Deng, Dong Yu, and others. 2014. Deep learning: methods and applications. Foundations and Trends® in Signal Processing 7, 3–4 (2014), 197–387.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

[17] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. arXiv preprint arXiv:1206.4683 (2012).

[18] S. Chang, A. Abdul, J. Chen and H. Liao, "A personalized music recommendation system using convolutional neural networks approach," 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, 2018, pp. 47-49, doi: 10.1109/ICASI.2018.8394293.

[19] Y. Chu, F. Huang, H. Wang, G. Li and X. Song, "Short-term recommendation with recurrent neural networks," 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, 2017, pp. 927-932, doi: 10.1109/ICMA.2017.8015940.

[20] H. Chu, X. Xing, Z. Meng and Z. Jia, "Towards a Deep Learning Autoencoder algorithm for Collaborative Filtering Recommendation," 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Jinzhou, China, 2019, pp. 239-243, doi: 10.1109/YAC.2019.8787614.

[21] F. Yang and Y. Lu, "Restricted Boltzmann Machines for Recommender Systems with Implicit Feedback," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4109-4113, doi: 10.1109/BigData.2018.8622127.

[22] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In WSDM. 565–573.

[23] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In WWW. 173–182.

[24] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking.

[25] Donghyun, Kim & Park, Chanyoung & Oh, Jinoh & Yu, Hwanjo. (2017). Deep Hybrid Recommender Systems via Exploiting Document Context and Statistics of Items. Information Sciences. 417. 10.1016/j.ins.2017.06.026.

[26] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In WWW. 507–517.

[27] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In AAAI. 144–150.

[28] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep coevolutionary network: Embedding user and item features for recommendation. arXiv preprint. arXiv preprint arXiv:1609.03675 (2016).

[29] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, and Alexander J Smola. 2016. Joint Training of Ratings and Reviews with Recurrent Recommender Networks. (2016).

[30] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In WSDM. 495–503.

[31] Kostadin Georgiev and Preslav Nakov. 2013. A non-iid framework for collaborative € ltering with restricted boltzmann machines. In ICML. 1148–1156.

[32] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In MM. 627–636.

[33] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. 2014. Exploration in interactive personalized music recommendation: a reinforcement learning approach. TOMM 11, 1 (2014), 7.

[34] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In Recsys. 191–198.

[35] Zhenghua Xu, Cheng Chen, Tomas Lukasiewicz, Yishu Miao, and Xiangwu Meng. 2016. Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. In CIKM. 1921–1924.

[36] Zhenghua Xu, Tomas Lukasiewicz, Cheng Chen, Yishu Miao, and Xiangwu Meng. 2017. Tag-aware personalized recommendation using a hybrid deep model. (2017).

[37] Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Note Recommendation in Dialogue using Deep Neural Network. In SIGIR. 957–960.

[38] Yin Zheng, Cailiang Liu, Bangsheng Tang, and Hanning Zhou. 2016. Neural Autoregressive Collaborative Filtering for Implicit Feedback. In Recsys.

[39] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In SIGKDD. 353–362.

[40] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In IJCAI.

[41] Yogesh Singh Rawat and Mohan S Kankanhalli. 2016. ConTagNet: exploiting user context for image tag recommendation. In Proceedings of the 2016 ACM on Multimedia Conference. 1102–1106.

[42] Ruobing Xie, Zhiyuan Liu, Rui Yan, and Maosong Sun. 2016. Neural Emoji Recommendation in Dialogue Systems. arXiv preprint arXiv:1612.04609 (2016).

[43] I. A. S. Jabbar, R. S. Alhamdani and M. N. Abdullah, "Analyzing Restricted Boltzmann Machine Neural Network for Building Recommender Systems," 2019 2nd International Conference on Engineering Technology and its Applications (IICETA), Al-Najef, Iraq, 2019, pp. 133-137, doi: 10.1109/IICETA47481.2019.9012981.

[44] N. IDRISSI, O. HOURRANE, A. ZELLOU and E. H. BENLAHMAR, "A Restricted Boltzmann Machine-based Recommender System For Alleviating Sparsity Issues," 2019 1st International Conference on Smart Systems and Data Science (ICSSD), Rabat, Morocco, 2019, pp. 1-5, doi: 10.1109/ICSSD47982.2019.9003149.

[45] I. M. A. Jawarneh et al., "A Pre-Filtering Approach for Incorporating Contextual Information Into Deep Learning Based Recommender Systems," in IEEE Access, vol. 8, pp. 40485-40498, 2020, doi: 10.1109/ACCESS.2020.2975167.

[46] Katarya, R., Arora, Y. Capsmf: a novel product recommender system using deep learning based text analysis model. Multimed Tools Appl 79, 35927–35948 (2020). https://doi.org/10.1007/s11042-020-09199-5.

[47] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv 52, 1, Article 5 (February 2019), 38 pages. DOI:https://doi.org/10.1145/3285029.

[48] Batmaz, Z., Yurekli, A., Bilge, A. et al. A review on deep learning for recommender systems: challenges and remedies. Artif Intell Rev 52, 1–37 (2019). https://doi.org/10.1007/s10462-018-9654-y.

[49] Kiran R, Pradeep k, Baharat B. DNNRec: A novel deep learning based hybrid recommender system, Expert Systems with Applications, volume 144 (April 2020).

Design and Implementation of a Snake-like Robot with Amplitude-Controlled Phase Oscillator-based Motion Control

Serkan Karacol
Department of Mechatronics Engineering
Firat University
Elazig, Turkey
karacol.serkan21@gmail.com

Deniz Korkmaz
Department of Electrical and Electronics Engineering
Malatya Turgut Ozal University
Malatya, Turkey
ORCID: 0000-0002-5159-0659

Gonca Ozmen Koca
Department of Mechatronics Engineering
Firat University
Elazig, Turkey
ORCID: 0000-0003-1750-8479

Abstract*—In this study, the design and implementation of an 8-joint, autonomous snake-like robot with Central Pattern Generator (CPG) based motion control are presented. The implementation of the robot is based on a biomimetic structure. The robot prototype consists of three main parts as a head including infrared sensors, an 8-joint body propulsion mechanism, and a passive tail connected to the last link. All parts of the robot are designed in a SolidWorks environment and produced with three-dimensional (3D) printing technology. In order to perform snake-like rhythmic, stable, oscillatory, and robust locomotion patterns, a biomimetic motion control structure based on Amplitude-Controlled Phase Oscillators (ACPOs) is designed. The designed structure is constructed with a bidirectional chain network topology. In 2D space, S-shape and C-shape motion patterns are performed to analyze the robot prototype. The experimental results show that the constructed network provides effective and smooth motion performances and the designed robot can perform basic snake-like motion patterns.

***Keywords*—snake-like robot, biomimetic robots, amplitude-controlled phase oscillator, motion control**

I. INTRODUCTION

Biomimetic systems are an engineering technology that enables motions to take place in the robotic field by observing the living forms in nature and by comparing the real-life mobility of these forms. Robotic systems, inspired by nature, are used in many areas such as artificial intelligence, search-rescue, exploration-observation, military, and surgery. Robotic designs have gained mobility with hints taken from the muscles, spinal cords, and joints of the living models in nature [1-3]. Snake robots are examples of the appropriate designs of these systems and have become a very popular topic with their high mobility. Snake robots can be used in places that people cannot reach. Their flexible mobility enables these robots to move across flat and sloping terrains, underwater, water surface, and in complex environments. Due to their very large number of Degrees of Freedom (DoF), it is very difficult to control their motion. The Central Pattern Generators (CPGs) can be a suitable solution for the motion control of snake-like robots. This solution method takes an important place in the formation and coordination of locomotion and also rhythmic patterns such as headway, crawling, swimming, and climbing can be generated [4-8].

CPGs are the biological neural networks that generate rhythmic and oscillatory behaviors and exists in the spinal cords. These networks provide multiple link control by synaptic connections. CPG-based locomotion control has three main advantages: (i) output signals are stable and exhibit limit cycles; (ii) they can control the robot motion without the need for a complex dynamic model; (iii) closed loop sensory feedback structure can be easily adapted [9,10].

In the literature, various models of CPGs have been used for the motion control of snake-like robots. Transeth et al. [11] offered a mathematical model for the snake robot obstacle-aided locomotion. The framework of nonlinear dynamics and convex analysis were given to systematically and accurately add isotropic friction forces using both unilateral contact forces (obstacles) and Coulomb's coordinated force laws. A comparison between numerical results and experimental studies were also presented. Nor and Ma [9] introduced a simplified CPG structure that can produce rhythmic snake patterns. the designed CPG network was constructed with a unidirectional Amplitude-Controlled Phase Oscillator (APCO) to control the harmonic motions. The forward and backward motions were controlled with a single parameter. The proposed CPG network provided a simple structure that exhibits less complex, less mathematical computation, fast convergence speed, and boundary loop behavior. In another study, a unidirectional ACPO network was designed and obstacle-avoidance control was performed. The phase transition method was used to obtain a smooth body shape transition and a continuous angle capable of adapting to the environment. The proposed method was tested in a simulation environment [10]. Nakajima et al. [12] a wheeled simultaneous controlled snake-like robot. the trajectory tracking was performed and three-dimensional (3D) motion performance was analyzed. Wang et al. [13] constructed a Hopf oscillator model and analyzed various network topologies. In the experiments, forward and backward motion control were presented. Takemori et al. [14] developed a snake robot by connecting curve segments. The experiments were performed on a pipe and a debris field. Yin et al. [15] proposed an adaptive robust control method for a soft robotic snake with a smooth-zone approach. In the simulations, the proposed method was also proved. Omisore et al. [16] analyzed a deeply-learned damped least-squares method for the inverse kinematics of snake-like robots. For fast control and singularity avoidance, a deep network was designed and the unique damping factor required for each target point of the robot motion space was predicted. These studies can be shown that the snake-like robotic designs and their locomotion control strategies are still in need of further researches.

The aim of this study is to design and implement an 8-joint, autonomous, biomimetic snake-like robot with a CPG-

This work was funded by a research grant from Firat University Scientific Research Projects Unit (FUBAP) under Grant No: TEKF.20.02.

based locomotion control framework. The designed prototype consists of three main parts as a rigid head including infrared sensors, an 8-joint body propulsion mechanism, and a passive tail connected to the last link. All of the parts are designed as a modular structure in a SolidWorks environment and produced with 3D printing technology. The CPG network is constructed with ACPO-based bidirectional chain network topology. In 2D space, S-type and C-type motions are performed to analyze the locomotion performance of the robot. The experimental results show that the constructed CPG network generates smooth motion patterns and the designed robot can perform basic snake-like motions.

The rest of this paper is organized as follows: Section 2 presents the design and implementation of the snake-like robot prototype. Section 3 describes the network structure, mathematical model of the network and topology type. The experimental results of the control method are given in Section 4. Finally, conclusions are given in Section 5.

II. Snake-Like Robot Prototpye

A. *Design Procedure*

In order to imitate a real snake, a biomimetic robot prototype is designed and implemented with an 8-joint body propulsion mechanism actuated by RC servo motors. In the studies, the number of joints of snake robots varies according to the environment to which the robot is adapted. The reason why the robot is designed with 8 joints is that real snakes can be imitated with a better performance with at least 8 joints according to the results obtained with different experiments [4-7]. The snake robot is modeled in 3D and designed in SolidWorks environment. The details of the robot with modular parts are given in Fig. 1 (a). Fig. 1 (b) shows the configuration of the planar structure when all joints are at the central positions. Center of Gravity (CoG) represents the motion according to the Earth-fixed plane. It consists of a rigid head, propulsive body, and a tail. All of the parts are connected to each other and developed with a modular and serial structure. The propulsive parts give the motion pattern shape to the robot and generate the propulsion force. There are ±80˚ motor angle ranges between the joints. The designed body components of the robot are given in Fig. 2. All parts are drawn according to the servo motor dimensions.

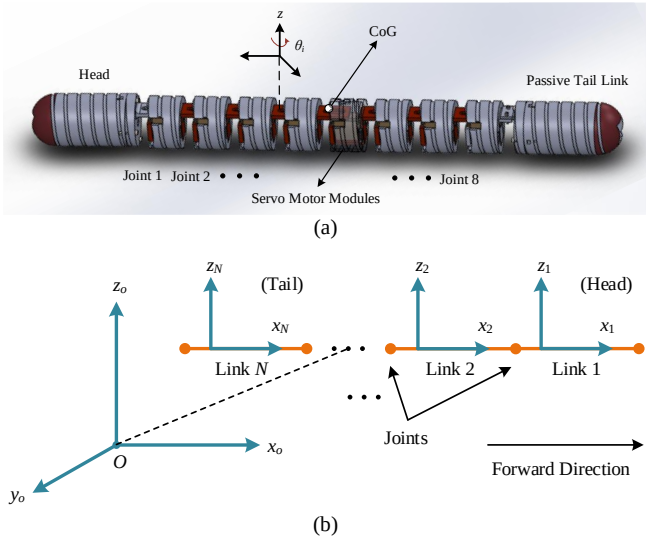


Fig. 1. The design procedure; (a) SolidWorks design of the snake robot, (b) planar configuration of the links

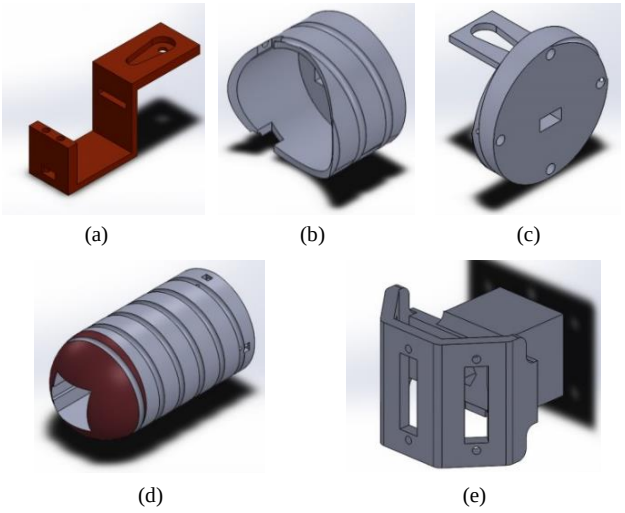


Fig. 2. The designed body components; (a) motor assembly component, (b) outer skin component, (c) head/tail and joint connection, (d) head and tail parts, (e) sensors packet

The motor assembly component is designed for mounting the both servo motor and motor shaft. Considering the outer skins, the joint bodies are formed. Head and tail connections are attached to the hinge joints with their body parts. The connections of the motors pass through the cable channels located in the outer skins. The sensor packets are mounted on the head and tail parts. These parts are designed for the vertically placed three infrared sensors. Each joint is connected to the next link and this structure composes a serial chain link mechanism. The position of the front sensor is placed in the same direction on the x-axis and the left and right sensors are located ±45˚ intervals from the front sensor. It is noted that the sensor packet is mounted to only the head module. The control unit and the other electronic parts are located and centered on the rigid head. The width of the interior head is designed according to the size of these parts. Fig. 3 presents the general view of the head component and motor assembly component with a servo motor. The assembly and distribution of all components on the prototype are carefully realized to provide the balancing of the CoG point. Therefore, the designed snake-like robot can be as compact as possible.

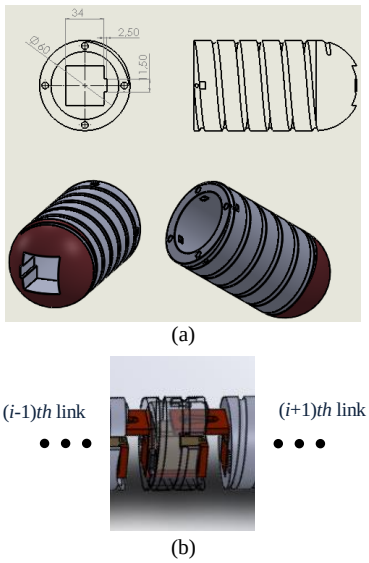
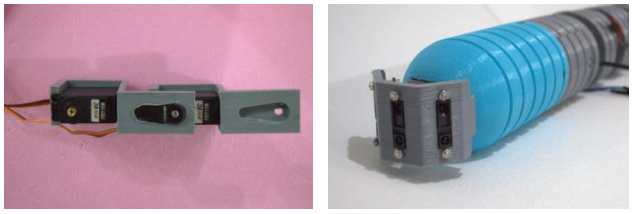


Fig. 3. General view of the components; (a) head component, (b) servo motor connection

B. *Implementation of the Prototype*

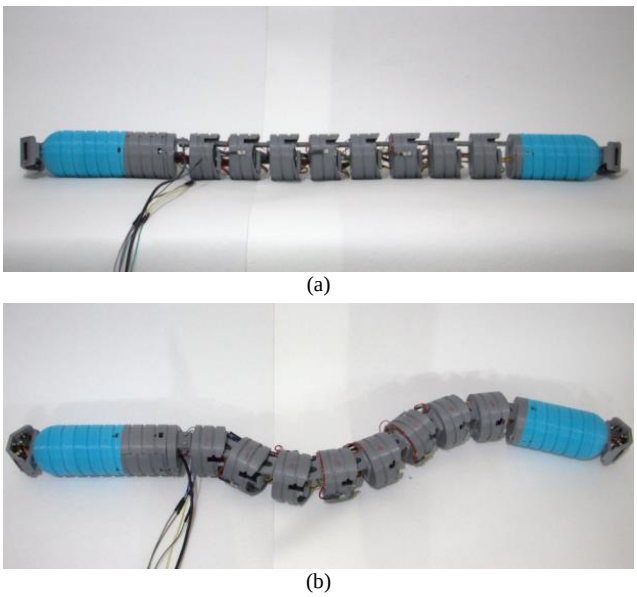
Based on the designed procedure, the snake robot prototype is implemented. In the production process, 3D designs of the components are transferred to the Simplify3D printing program in the STL file formats. The voxel formats are composed and print settings are configured. Then, the prepared parts are converted to G codes and the production phase is performed with a Prusa i3 3D printer. For production, 1.755 mm gray Poly Lactic Acid (PLA) filament is used as the raw material of the printer. The assembly of the parts is carried out with metric-3 bolts. All motor components are connected as a serial link as shown in Fig. 4 (a). The connected sensors to the head are given in Fig. 4 (b).

The developed snake robot prototype is presented in Fig. 5. The electronic system of the robot includes integrated hardware and software. The propulsion mechanism is actuated with 8 RC servo motors. An 11.1 V power supply is used for the power generation. In order to detect the obstacles, three Sharp GP2Y0A21 infrared sensors are connected to the front head module. The measurement range of the sensors is 10-80 cm. There is a 32 bit 180 MHz ARM Cortex M4 microprocessor for the control unit. It has 256 KB RAM and 1 MB flash memory. The servo motors are driven with PWM signals. The control unit performs the rhythmic and simultaneous motions with the control algorithm and evaluates the sensor data received from ADC inputs. The prototype is approximately 980 mm long and 80 mm in diameter. The total mass is also approximately 2.8 kg.



(a) (b)

Fig. 4. Implementation of the robot parts; (a) serial servo motor connection, (b) connected infrared sensors



(a) (b)

Fig. 5. The developed snake robot prototype; (a) all joints are at the central position, (b) an example photo of the S-shape

III. Control Method

There are many methods for controlling robots according to system requirements. The locomotion control of snake-like robots and synchronous gait planning is a difficult task due to their multi-connected body structures. At this point, simultaneous motions are necessary and the gait sequences should not be affected by external disturbances. One of the control methods that can be applied for biomimetic systems is CPG. CPGs are found in many vertebrates and can be modeled as a locomotion controller. In addition, these models do not need any dynamic model of the system unlike conventional controllers. By imitating the spinal cord, the CPG-based locomotion controller provides rhythmic and stable outputs.

A simple and effective CPG model is ACPO. Kuramoto oscillator-based ACPOs can produce rhythmic outputs against external disturbances through their features and limit cycle behaviors [3,17]. This oscillator type is also based on coupled oscillators like neural networks in animals and is given by:

$$\dot{\emptyset}_i = \omega_i + \sum_j^N \left(\mu_{ij} r_j \sin\left(\emptyset_j - \emptyset_i - \varphi_{ij}\right)\right) \quad (1)$$

$$\ddot{\mathrm{r}}_\mathrm{i} = \mathrm{a_r}\left(\frac{\mathrm{a_r}}{4}\left(\mathrm{R_i} - \mathrm{r_i}\right) - \dot{\mathrm{r}}_\mathrm{i}\right) \quad (2)$$

$$\ddot{x}_i = a_x\left(\frac{a_x}{4}\left(X_i - x_i\right) - \dot{x}_i\right) \quad (3)$$

$$\theta_i = x_i + r_i \cos(\emptyset_i) \quad (4)$$

Where $\emptyset_i, r_i,$ and x_i are the state variables that determine the phase, amplitude, and offset of the oscillation, respectively. $R_i, X_i,$ and w_i are the desired amplitude, bias, and frequency, respectively. N is the number of oscillators with ith and jth oscillators. a_r and a_x are the positive convergence speed constants. The oscillator outputs are expressed as θ_i. The parameters μ_{ij} and φ_{ij} are the coupling weights and phase differences between ith and jth oscillators. r_j also defines the correction rate of the jth coupling. The designed model is coupled and constructed as a bidirectional chain type network. Equations (2) and (3) are the damped second order differential expressions and the amplitude and bias will asymptotically converge to R_i and X_i by the following form:

$$\theta_i^\infty(t) = X_i + R_i \cos(w_i t + \varphi_0) \quad (5)$$

Here, φ_0 is the initial phase of the oscillator. For any initial conditions, the outputs converge to the desired control parameters and the outputs are not affected by external disturbances. Therefore, the amplitude and offset of the oscillation can be easily modulated and the coupled oscillator outputs exhibit steady-state robust snake motion patterns.

The control architecture of the robot is given in Fig. 6. Each oscillator corresponds to each link of the robot. After receiving the set-point parameters, the CPG network generates the desired sinusoidal angles, and then PWM signals are applied to drive the servo motors. Practically, the designed network determines the following four system requirements as an amplitude regulator, a frequency and phase regulator, an offset control, and an output magnitude. In the designed network, all coupling weights are set to 1.

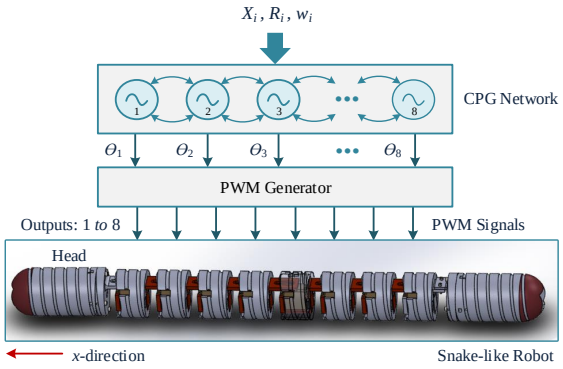


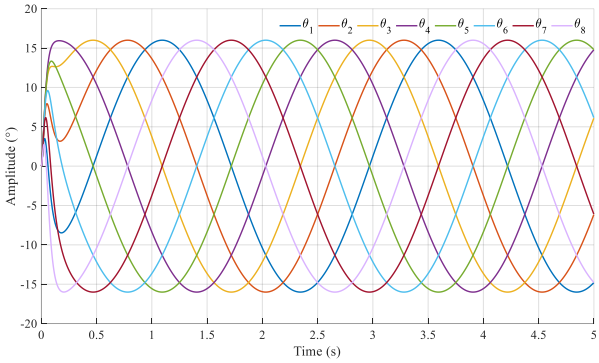
Fig. 6. Control architecture of the snake robot: Input parameters are the desired amplitude, bias, and phase-relation of the links.

IV. Experimental Results

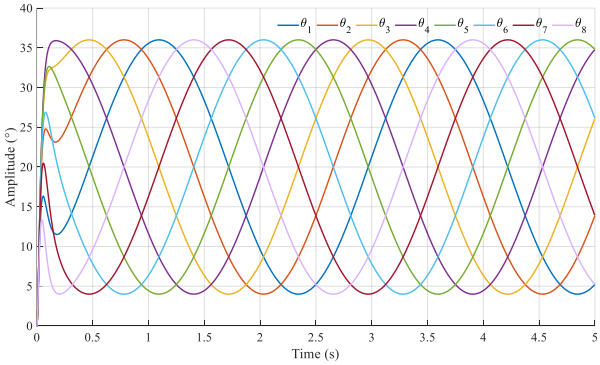
In order to evaluate the developed robot prototype, two main locomotion patterns as S-shape and C-shape are performed in the experiments. CPG network parameters are given in Table 1. According to these parameters, oscillator outputs are presented in Fig. 7. The experiments are utilized during 10 s for two motion patterns. Fig. 8 presents the amplitude change of the oscillator outputs for both two motions. The motion sequences of the robot for S-shape and C-shape are given in Fig. 8 and Fig. 9, respectively. It is noted that the motion is realized by the interaction of the robot with the ground.

TABLE I. CPG Network Parameters

Network Parameters	Values	
	S-shape	C-shape
R_i (°)	16	16
f (Hz)	0.4	0.4
φ (°)	45	45
X_i (°)	0	20



(a)



(b)

Fig. 7. Oscillator outputs; (a) S-shape motion, (b) C-shape motion

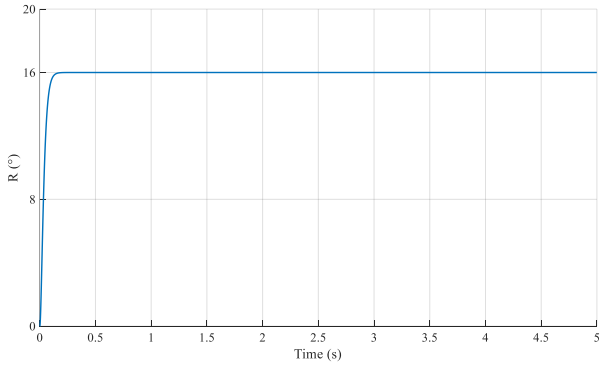


Fig. 8. Amplitude change of the oscillator outputs for two motion patterns

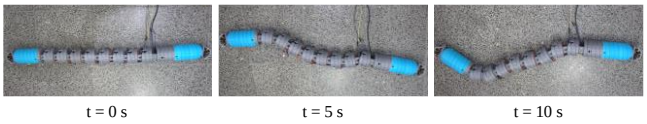


Fig. 9. S-shape sequences of the snake robot

Fig. 10. C-shape sequences of the snake robot

It can be seen from Fig. 8 that the oscillator outputs rapidly reach the desired amplitude and maintain the stable behavior in steady-state. According to the motion sequences, the phase difference is 45° and the bias is 0°. The transient-state time is nearly 0.31 s. During 10 s, the robot can perform the S-shape motion effectively. If the amplitude reduces, the robot narrows the motion patterns. In addition, the applied amplitude value to the links is quite enough. The key parameter is the bias and it is not 0° for C-shape. According to the oscillator model, the phase difference is 45° and the bias is 20°. In this motion, the body of the robot is taken the C-shape starting from its linear position. The transient-state time is nearly 0.33 s. During 10 s, the robot can perform this motion effectively. If the amplitude increases, more torsion occurs.

These results show that the designed network architecture is well suited and generates rhythmic and robust outputs for both two motion patterns.

V. Conclusion

In this study, 8-joint, autonomous, biomimetic snake-like robot with a CPG-based locomotion control framework is presented. The designed robot consists of three main parts as a rigid head including infrared sensors, a multi-joint body propulsion mechanism, and a passive tail connected to the last link. The robot is designed as a serial link mechanism and all of the parts are designed as a modular structure. The robot components are drawn in a SolidWorks environment and produced with 3D printing technology. The ACPO network is constructed with bidirectional chain type network topology. In the experiments, S-type and C-type motions are performed to analyze the locomotion performance of the robot. The experimental results show that the designed network gives rapid and rhythmic oscillation outputs and the developed robot can perform basic snake-like motion patterns effectively.

REFERENCES

[1] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A review on modelling, implementation, and control of snake robots," Robotics and Autonomous systems, 60(1), 2012, pp.29-40.

[2] S. Karacol, D. Korkmaz, G. Ozmen Koca, and N. Karabulut, "A study of hopf and amplitude-controlled phase oscillators for snake robot locomotion," International Journal of Intelligent Systems and Applications in Engineering, 7(3), 2019, pp.153-158.

[3] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," Neural networks, 21(4), 2008, pp.642-653.

[4] T. Matsuo, T. Sonoda, and K. Ishii, "A design method of CPG network using energy efficiency to control a snake-like robot," 2012 IEEE Fifth International Conference on Emerging Trends in Engineering and Technology, 2012, pp.287-292.

[5] Z. Lu, Y. Xie, H. Xu, J. Liu, L. Mao, C. Shan, and B. Li, "Design of a MNSM-based controller for the swimming motion of a snake-like robot," 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015, pp.2050-2055.

[6] S. Hasanzadeh and A. A. Tootoonchi, "Adaptive optimal locomotion of snake robot based on CPG-network using fuzzy logic tuner," 2008 IEEE Conference on Robotics, Automation and Mechatronics, 2008, pp.187-192.

[7] S. Fukunaga, Y. Nakamura, K. Aso, and S. Ishii, "Reinforcement learning for a snake-like robot controlled by a central pattern generator," IEEE Conference on Robotics, Automation and Mechatronics, 2, 2004, pp.909-914.

[8] G. Qiao, Y. Zhang, X. Wen, Z. Wei, and J. Cui, "Triple-layered central pattern generator-based controller for 3D locomotion control of snake-like robots," International Journal of Advanced Robotic Systems, November 2017, pp.1-13.

[9] N. M. Nor and S. Ma, "A simplified CPGs network with phase oscillator model for locomotion control of a snake-like robot," Journal of Intelligent and Robotic Systems, 75(1), 2014, pp.71-86.

[10] N. M. Nor and S. Ma, "CPG-based locomotion control of a snake-like robot for obstacle avoidance," 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp.347-352.

[11] A. A. Transeth, R I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle-aided locomotion: modeling, simulations, and experiments," IEEE Transactions on Robotics, 24(1), 2008, pp.88-104.

[12] M. Nakajima, M, Tanaka, and K. Tanaka, "Simultaneous control of two points for snake robot and its application to transportation," IEEE Robotics and Automation Letters, 5(1), 2019, pp.111-118.

[13] Z. Wang, Q. Gao, and H. Zhao, "CPG-inspired locomotion control for a snake robot basing on nonlinear oscillators," Journal of Intelligent and Robotic Systems, 85(2), 2017, pp.209-227.

[14] T. Takemori, M. Tanaka, and F. Matsuno, "Gait design for a snake robot by connecting curve segments and experimental demonstration," IEEE Transactions on Robotics, (99), 2018, pp.1-8.

[15] H. Yin, Y. H. Chen, D. Yu, H. Lü, and W. Shangguan, "Adaptive robust control for a soft robotic snake: a smooth-zone approach," Applied Mathematical Modelling, 80, 2020, pp.454-471.

[16] O. M. Omisore, S. Han, L. Ren, A. Elazab, L. Hui, T. Abdelhamid, N. A. Azeez, L. Wang, "Deeply-learnt damped least-squares (DL-DLS) method for inverse kinematics of snake-like robots," Neural Networks, 107, 2018, pp.34-47.

[17] Z. Bing, L. Cheng, K. Huang, M. Zhou, and A. Knoll, "CPG-based control of smooth transition for body shape and locomotion speed of a snake-like robot," In 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp.4146-4153.

Performance Analysis of XGBoost Classifier with Missing Data

Zeliha Ergul Aydin
Department of Industrial Engineering
Eskisehir Technical University
Eskisehir, Turkey
0000-0002-7108-8930

Zehra Kamisli Ozturk
Department of Industrial Engineering
Eskisehir Technical University
Eskisehir, Turkey
0000-0003-3156-6464

***Abstract*—XGBoost algorithm has become popular due to its success in data science competitions, especially Kaggle competitions. Missingness in a dataset is a challenging problem and needs extra processing. Missing data imputation, which is filling missing data with plausible values, is one of the solutions to this problem. One of the most important reasons why researchers prefer XGBoost is that it can work with missing data without processing. With this study, we aim to show the impact of missing data imputation methods on the XGBoost classifier. We compare the performance of the XGBoost classifier trained on non-imputed data with the XGBoost classifier trained on imputed data. For comparative analysis, we choose K-nearest neighbor (KNN), Soft-Impute, multivariate imputation by chained equations (MICE), mean, and class-mean as the imputations methods, and ten datasets from KEEL repository as the datasets. We perform the Friedman test to compare the classification models' F-score statistically. Our analysis shows that the missing data imputation methods don't have any effect on XGBoost classifier performance.**

***Index Terms*—missing data, missing data imputation, classification, XGBoost, KEEL**

I. Introduction

XGBoost also called eXtreme Gradient Boosting, is a machine learning algorithm that is becoming widespread as it won many Kaggle data science competitions. It provides satisfactory results in many applications such as disease prediction [1], diesel fuel brands identification [2], estimation of the tunnel boring machine advance rate [3], prediction of concrete electrical resistivity for structural health monitoring [4], hotel reviews sentiment analysis [5], star/galaxy classification [6], prediction of vehicle occupants injury at signalized intersections [7]. Researchers widely prefer XGBoost because of its pros given below:

- handling missing data internally
- does not require data scaling and normalizing
- high computational speed by using parallel processing
- avoiding the overfitting problem with regularization
- high prediction accuracy

Xgboost's handling of missing data internally is one of the essential factors in the widespread use of XGboost because missing data handling is a challenging problem and needs extra processing. There is no comprehensive study analyzing the XGBoost performance in handling missing data to the best of our knowledge. To fill this gap, we perform a comprehensive experiment to show the impact of missing data imputation methods on the XGBoost classifier with this study. We use K-nearest neighbor (KNN), Soft-Impute, multivariate imputation by chained equations (MICE), mean, and class-mean imputations methods, and ten datasets from KEEL repository as the datasets for comparison. The rest of the paper is organized as follows. In Section II, we give a brief synopsis of the related works. Section III describes the XGBoost classifier and missing data imputation methods used in this study. In Section IV, we present our experimental design, results and our discussion about the results. Finally, the conclusion and future works are given in Section V.

II. Related Works

There are many comparative missing data imputation analyzes for traditional classifiers in the literature. Most of these studies proved that missing data imputation methods increase the classifier's performance. Farhangar et al. [8] showed that imputation methods, except for the mean imputation, improve the performance of RIPPER, C4.5, KNN, support vector machine with polynomial and RBF kernels, and Naive-Bayes classifier, with paired t-test. Luengo et al. [9] performed a comprehensive analysis with 23 different classification methods and 14 different imputation methods. They concluded that missing values imputation methods could improve the classification accuracy. Missing data imputation gave similar results on ANN, decision tree, and random forest classifiers [10]. Contrary to these findings, Acuna and Rodriguez [11] showed that the case deletion method, mean imputation, median imputation, and the KNN imputation don't have a significant effect on the accuracy of Linear Discriminant Analysis and the KNN classifier.

There are also some comparison based studies for classifiers C4.5, CN2, and XGBoost that can handle missing data internally. Batista and Monard [12] indicated that missing data KNN imputation method could outperform the internal methods used by C4.5 and CN2 classifier. However, they didn't perform any statistical test to compare the classifiers' performance on imputed and non-imputed datasets. Rusdah and Murfi [13] showed that the XGBoost without any imputation gives a comparable classification accuracy score to one of the XGBoost with KNN and mean imputation method for

risk prediction in life insurance. They performed the analysis on only one dataset from an insurance company without any statistical test. Hence, their findings cannot be generalized. Based on the literature, it is seen that there is a need for a comprehensive analysis to mention general results.

III. Methods

The idea behind ensemble classification is to construct multiple classifiers to obtain better classification performance. Bagging and boosting are the two forms of ensemble classification. There are three types of boosting algorithm; Adaptive Boosting (AdaBoost), Gradient Boosting, and eXtreme Gradient Boosting (XGBoost). We focus on XGBoost classifier in subsection A. Besides, data missingness is a critical issue in the classification task because most classification algorithms work with complete datasets. We briefly explain missing data imputation methods used in this paper, in subsection B.

A. XGBoost

XGBoost presented by Chen and Guestrin [14] is a gradient boosted decision tree model. XGBoost algorithm trains decision trees on training data sequentially. The algorithm adds a new decision tree at each iteration to the previous decision trees to improve the objective function's value. The objective function, which is aimed to minimize, consists of the loss term (l) and the regularization term (Ω). Equation (1) defines the objective function of the t-th iteration (L_t), where y_i is the actual class label of instance i ,$\hat{y}_i$ is the predicted class label of instance i, f_k is the function of tree, n is the number of instances in the training set, and Ω is regularization term.

$$L^t = \sum_{i=1}^{n}[l(y_i, \hat{y}_i^{t-1} + f_t(x_i))] + \Omega(f_t) \tag{1}$$

$\Omega(f_t)$ (2) penalizes the model complexity to avoid overfitting, where γ and λ are the hyperparameters, T is the number of leaves in the tree, and w is the weight of each leaf.

$$\Omega(f_t) = \gamma T_t + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2 \tag{2}$$

The second-order Taylor expansion is applied to approximate the value of the loss function in (3), where g_i represents the first order gradient statistics on the loss in (4), and h_i represents the second order gradient statistics on the loss in (5).

$$L^t \cong \sum_{i=1}^{n}\left[l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{3}$$

$$g_i = \partial_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \tag{4}$$

$$h_i = \partial^2_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \tag{5}$$

For a fixed tree structure, the optimal leaf weight in leaf node j, and the corresponding optimal value of objective function are given by (6) and (7) respectively, where I_j denotes the instance set of leaf j. Equation (7) is used as a scoring function to evaluate the quality of tree structure q.

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} \tag{6}$$

$$L(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T \tag{7}$$

Nevertheless, all possible tree structures must be evaluated to obtain the optimal tree structure. However, it is impossible to consider all possible tree structures because of computational cost. In practice, XGBoost adopts a greedy algorithm to find an optimal tree structure.

Missing data are handled internally in XGBoost without requiring any imputing and deleting process. XGBoost classifies an instance with missing feature in to default direction at each node as shown in Fig. 1. There are two options as right and left nodes for default direction. The direction with the maximum gain in training set is selected as a default direction.

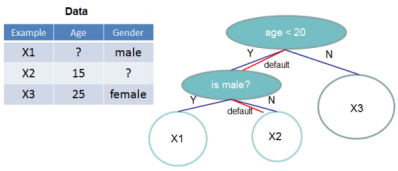


Fig. 1. An example of how XGboost handle missing data [14]

B. Missing Data Imputation Methods

Little and Rubin [15] classified missing data mechanisms that lead to missingness as missing completely at random (MCAR), missing at random (MAR), or not missing at random (NMAR). In a nutshell, missingness on MCAR is independent of both missing and known data. The missingness on MAR depends on known data, and missingness on NMAR depends on missing data. Missing data imputation methods fill the missing data with plausible data under the assumptions of different mechanisms. Missing data imputation, which is filling missing data with plausible values, is one of the solutions to this problem.

In this paper, we consider five commonly used imputation methods that can work with the MCAR mechanism. These are K-nearest neighbor (KNN), Soft-Impute, multivariate imputation by chained equations (MICE), mean, and class-mean. KNN imputation replaces missing data in an instance with the mean, mode, or median of the k nearest neighbors of this instance. The nearest neighbors are determined using Euclidean, Manhattan, Hamming, or Jaccard similarity measures. Mean imputation fills the missing data in a feature with the mean, median, or mode of this feature. In class mean imputation, missing data in a feature are completed with the mean, median, or the mode of this feature instance, which belongs to the same class label as the missing data. Mean and

median are used for continuous features, the mode is used for discrete features in mean, KNN on and class mean imputation. MICE [16] imputes missing data by chained regressions. A regression model is trained on other features to predict missing data in a feature, and these missing data replace with the predictions. This process continues iteratively with a chain until it reaches the number of multiple imputations and iteration parameters. Soft-Impute [17] iteratively replaces the missing data with values obtained from a soft-thresholded singular value decomposition.

IV. Experiments and Results

A. Datasets

Ten datasets which include missing value were taken from KEEL repository [18] for comparative analysis. In the KEEL repository, Datasets are split into training and test sets with 10-folds cross-validation procedure. They are artificially introduced missing data according to MCAR mechanism in training sets. Table I summarizes the characteristic of these datasets.

TABLE I
Description of Datasets

Dataset	Feature	Feature Type	Instance	Classes	Missing Percentage%
Iris	4	Continous	150	3	32.67 %
Pima	8	Continous	768	2	50.65 %
Wine	13	Continous	178	3	70.22 %
Australian	14	Mixed	690	2	70.58 %
Newthyroid	5	Mixed	215	3	35.35 %
Ecoli	7	Continous	336	8	48.21 %
Satimage	3	Discrete	6435	7	87.80 %
German	6	Mixed	1000	2	80.00 %
Magic	10	Continous	1902	2	58.20 %
Shuttle	9	Discrete	2175	7	55.95 %

B. Classification Models

We constructed six different XGBoost classifier models by combining different missing data imputation methods for each dataset. Table II shows the description and name of these models. Fancyimpute [19] and Scikit-learn [20] Python packages are used for the implementation of these methods.

TABLE II
Description of Classification Models

Model Name	Description
XGB	XGBoost classifier trained on non-imputed data
XGB+KNN	XGBoost classifier trained on imputed data with KNN
XGB+MICE	XGBoost classifier trained on imputed data with MICE
XGB+SI	XGBoost classifier trained on imputed data with Soft-Impute
XGB+MI	XGBoost classifier trained on imputed data with mean
XGB+CMI	XGBoost classifier trained on imputed data with class-mean

C. Parameter Setting

We set the *k* parameter as 1 in KNN, the number of multiple imputations parameters and iterations as 5, and optimized the XGBoost classifiers' hyperparameters by using grid search method with nested 10-fold cross-validation procedure.

D. Evaluation Metrics

The macro-averaged F-score computed with 10-fold cross-validation is used to evaluate the models' performance. F-score is calculated based on the confusion matrix, which is given in Table III. The calculation of F-score is given in (8).

TABLE III
Confusion Matrix

	Actual Positive	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

$$F\text{-}score = \frac{TP}{TP + \frac{(FP + FN)}{2}} \tag{8}$$

E. Statistical comparison of classifiers

We performed Friedman test [21] to compare the classification models' F-score statistically. The Friedman test is a useful non-parametric statistical test to compare multiple classification models over multiple data sets [22]. The null hypothesis of this test states that there is no difference between the classification models. It ranks the classification models' performance for each dataset in ascending order, then computes each classification model's average rank over all datasets. Equation (9) shows the Friedman test statistic calculation with K denotes the number of the classification model, N denotes the number of datasets, and AR_i denotes the average rank of each classification model over all datasets. If the test statistic value exceeds the critical value obtained from the chi-squared distribution table with K-1 degrees of freedom, we can reject the null hypothesis. The p-value approach can also be used for hypothesis testing. If the corresponding p-value of the test statistic is less than or equal to the significance level, we can reject the null hypothesis.

$$\tilde{\chi}_F^2 = \frac{12N}{K(K+1)}\left[\sum_{i=1}^{K} AR_i^2 - \frac{K(K+1)^2}{4}\right] \tag{9}$$

F. Results and Discussions

Average macro F-scores of six classification models using 10-fold cross-validation are tabulated in Table IV. The bold values indicate the best F-score for each dataset in this table. It is worth noting that XGB+CMI does not give the best F-score for any data set. We can say that the reason for this is the imbalanced structure of the data sets used. The mean of the minor classes' features may not be informative due to the small instance size in this class. XGB and XGB MICE models give the best F-score for three datasets, XGB+KNN and XGB+MI models provide the best F-score for two datasets, and the XGB+SI model provides the best F-score for just a dataset.

We also visualize the results in Fig. 2. The models' F-score values are very close to each other; hence, it is difficult to say that one model is superior to the others. Here, it is necessary to use a statistical test to mention a statistical difference between models.

TABLE IV
F-SCORES OF THE CLASSIFICATION MODELS

Dataset	XGB	XGB+KNN	XGB+MICE	XGB+SI	XGM+MI	XGM+CMI
Australian	0.8560	**0.8692**	0.8562	0.8545	0.8618	0.8647
Ecoli	**0.7272**	0.6815	0.7121	0.6786	0.7251	0.6880
German	**0.7054**	0.6843	0.6799	0.6836	0.6736	0.6893
Iris	0.9463	0.9463	0.9463	0.9596	**0.9597**	0.9530
Magic	0.7999	0.7982	0.7898	**0.8039**	0.7898	0.7928
Newthyroid	0.9286	**0.9620**	0.9325	0.9448	0.9393	0.9270
Pima	0.7122	0.7172	**0.7227**	0.7003	0.6982	0.7089
Satimage	0.8942	0.8980	**0.9015**	0.8982	0.8928	0.8840
Shuttle	0.9227	0.9404	0.8974	0.8963	**0.9471**	0.9223
Wine	**0.9778**	0.9721	**0.9788**	0.9635	0.9726	0.9666

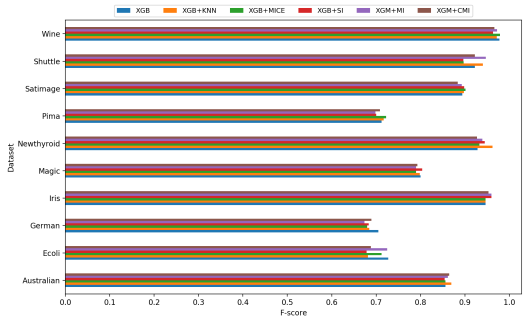


Fig. 2. F-scores of the classification models on all datasets.

We used the Friedman test for statistical comparison of the six classification models. The p-value of the Friedman test is 0.708, so we fail to reject the null hypothesis at 95% confidence level. There is no strong evidence to support that these classification models are significantly different. So we can conclude classification models used in this study have the same performance. Our findings show that missing data imputation methods don't have an impact on XGBoost classifier performance. The findings also support Rusdah and Murfi [13].

V. CONCLUSIONS

XGBoost is a commonly used classifier that can handle missing data internally and doesn't require any missing data treatment for missing data. This study analyzes the impact of missing data imputation methods on the XGBoost classifier. Firstly, we impute missing data in ten datasets from the KEEL repository with five imputation methods, namely KNN, Soft-Impute, MICE, mean, and class mean, to analyze. We train XGBoost classifiers on these imputed datasets and non-imputed datasets. Finally, we compare the F-score of the classifiers statistically. This study shows that the XGBoost classifier trained on non-imputed datasets gives statistically the same results as the XGBoost classifier trained on imputed datasets.

Future works can consider including different missing data imputation methods and perform analysis in more data sets with XGBoost classifier. In this study, we only consider the artificially generated MCAR mechanism, limiting the generalization of our results. We will analyze the XGBoost classifier performance on MAR and NMAR mechanism as future work to overcome this limitation.

ACKNOWLEDGMENT

This study is supported by Eskisehir Technical University Scientific Research Projects Committee (ESTUBAP-20DRP025).

REFERENCES

[1] K. Budholiya, S. K. Shrivastava, V. Sharma, An optimized XGBoost based diagnostic system for effective prediction of heart disease, Journal of King Saud University - Computer and Information Sciences (10 2020). doi:10.1016/j.jksuci.2020.10.013.

[2] S. Wang, S. Liu, J. Zhang, X. Che, Y. Yuan, Z. Wang, D. Kong, A new method of diesel fuel brands identification: SMOTE oversampling combined with XGBoost ensemble learning, Fuel 282 (12 2020). doi:10.1016/j.fuel.2020.118848.

[3] J. Zhou, Y. Qiu, S. Zhu, D. J. Armaghani, M. Khandelwal, E. T. Mohamad, Estimation of the TBM advance rate under hard rock conditions using XGBoost and Bayesian optimization, Underground Space (7 2020). doi:10.1016/j.undsp.2020.05.008.

[4] W. Dong, Y. Huang, B. Lehane, G. Ma, XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring, Automation in Construction 114 (6 2020). doi:10.1016/j.autcon.2020.103155.

[5] X. Zhang, Q. Yu, Hotel reviews sentiment analysis based on word vector clustering, in: 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), IEEE, 2017. doi:10.1109/CIAPP.2017.8167219.

[6] L. Chao, Z. Wen-hui, L. Ji-ming, Study of Star/Galaxy Classification Based on the XGBoost Algorithm, Chinese Astronomy and Astrophysics 43 (4) (10 2019). doi:10.1016/j.chinastron.2019.11.005.

[7] E. Kidando, A. E. Kitali, B. Kutela, M. Ghorbanzadeh, A. Karaer, M. Koloushani, R. Moses, E. E. Ozguven, T. Sando, Prediction of vehicle occupants injury at signalized intersections using real-time traffic and signal data, Accident Analysis & Prevention 149 (1 2021). doi:10.1016/j.aap.2020.105869.

[8] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition 41 (12) (2008) 3692–3705. doi:10.1016/j.patcog.2008.05.019.

[9] J. Luengo, S. García, F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, Knowledge and Information Systems 32 (1) (2012) 77–108. doi:10.1007/s10115-011-0424-2.

[10] J. Poulos, R. Valle, Missing Data Imputation for Supervised Learning, Applied Artificial Intelligence 32 (2) (2018) 186–196. doi:10.1080/08839514.2018.1448143.

[11] E. Acuña, C. Rodriguez, The Treatment of Missing Values and its Effect on Classifier Accuracy, in: Classification, Clustering, and Data Mining Applications, Springer Berlin Heidelberg, 2004, pp. 639–647. doi:10.1007/978-3-642-17103-1_60.

[12] G. E. Batista, M. C. Monard, An analysis of four missing data treatment methods for supervised learning, Applied Artificial Intelligence 17 (5-6) (2003) 519–533. doi:10.1080/713827181.

[13] D. A. Rusdah, H. Murfi, XGBoost in handling missing values for life insurance risk prediction, SN Applied Sciences 2 (8) (8 2020). doi:10.1007/s42452-020-3128-y.

[14] T. Chen, C. Guestrin, XGBoost, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2016. doi:10.1145/2939672.2939785.

[15] R. J. A. Little, D. B. Rubin, Statistical Analysis with Missing Data, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2002. doi:10.1002/9781119013563.

[16] S. van Buuren, K. Groothuis-Oudshoorn, mice: Multivariate imputation by chained equations in R, Journal of Statistical Software 45 (3) (2011) 1–67. doi:10.18637/jss.v045.i03.

[17] R. Mazumder, T. Hastie, R. Tibshirani, Spectral Regularization Algorithms for Learning Large Incomplete Matrices, Journal of Machine Learning Research 11 (80) (2010) 2287–2322.

[18] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2-3) (2011) 255–287.

[19] fancyimpute · PyPI.
URL https://pypi.org/project/fancyimpute/

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (85) (2011) 2825–2830.

[21] M. Friedman, A Comparison of Alternative Tests of Significance for the Problem of m Rankings, The Annals of Mathematical Statistics 11 (1) (3 1940). doi:10.1214/aoms/1177731944.

[22] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, Tech. rep. (2006). doi:10.5555/1248547.1248548.

Using Machine Learning Methods to Detecting Phishing Websites (A Comparative Analysis)

Hamdullah KARAMOLLAOĞLU
Dept. of IMS and Education
Electricity Generation Company Inc.
İstanbul, Turkey
h.karamollaoglu@euas.gov.tr

Ahmet ALBAYRAK
Computer Engineering
Düzce University
Düzce, Turkey
ahmetalbayrak@duzce.edu.tr

***Abstract*— The number of phishing websites created for accessing important accounts by capturing personal data such as passwords, credit cards, identity or address information is increasing day by day and causing material and moral damages to end users. For this reason, it is of great importance to develop methods that can automatically detect and eliminate these phishing web pages. In this study, Support Vector Machine, Multilayer Perceptron, Random Forest, Naïve Bayes, K-Nearest Neighbors, Logistic Regression, Decision Trees and AdaBoost, which are among the most used machine learning methods in the literature, are used to identify phishing websites. A dataset obtained from Machine Learning repository (UCI) is used in the study. In this dataset, there are 1353 pieces of data including 9 attributes of websites. The classification performance of the methods used in the study on the relevant dataset is measured and it is seen that the most successful machine learning method is the Random Forest Classifier with an accuracy of 89.4%.**

Keywords—phishing, phishing website, phishing detection, machine learning

I. INTRODUCTION

The number of phishing websites created for fraudulent purposes by accessing users' personal data such as identity, credit card, password, address is increasing day by day. According to APWG's report for the third quarter of 2020, the number of unique phishing websites detected on the internet is 171040 in July, 201591 in August and 199133 in September [1] and according to the Webroot Threat Report, each month nearly 1500000 new phishing websites are created [2]. Turkey in the first five months of 2020, a total of 14120 phishing sites are blocked with joint efforts made by General Directorate of Security Department of Combating Cyber Crimes and Information Technology Communications Authority [3].

The number of phishing websites detected and the rapid increase in this number every year reveal the seriousness of the situation. For this reason, it is of great importance to detect and eliminate phishing websites with the help of various methods. In this study, machine learning algorithms have been used to detecting phishing websites. For this purpose, Random Forest, K-Nearest Neighbors, Support Vector Machine, Multilayer Perceptron, Naïve Bayes, Logistic Regression, Decision Trees and Adaboost, which are among the most used machine learning methods in the literature, are used to identify phishing websites. In addition, a data set obtained from the Machine Learning Pool (UCI) [4] is used in the study to be applied in machine learning methods.

In the second section of the study, the literature studies on detecting phishing websites are discussed. In the third section, the metarial and method used in the study are emphasized. Experimental results have given in the fourth section and the results are evaluated in the last section.

II. RELATED WORKS

In this section, studies in the literature with the detection of phishing webpages are examined. The studies discussed are summarized below.

Li et al. [5], proposed a system for detecting phishing web pages using a multi-layered stacking model that combines XGBoost and LightGBM methods. This model has been tested on a data set consisting of approximately 50000 web pages and the classification process has been performed with a success rate of 98.60%.

Abeywardena et al. [6], proposed a CNN architecture to detecting phishing websites. In this architecture, FPN used to determine the similarity ratio of the content of web pages and ResNet used to determine the structural similarity between websites. As a result of the analysis, it has been observed that CNN architecture gives 20% more successful results in detecting phishing sites compared to other baseline methods.

Şahingöz et al.[7], proposed to detecting phishing websites using natural language processing and machine learning methods on a dataset of 36400 legit, 37175 phishing URLs. It has been observed that the best method to determine phishing web pages is the Random Forest Method, which performs accurate classification with a rate of 98.7%.

Jain et al. [8], proposed to detect phishing websites using machine learning methods (1428 for phishing and legitimate for phishing) on 2544 URL dataset compiled from the Phishtank dataset and Alexa's top websites etc. As a result of the analysis, it has seen that Logistic Regression is the most successful classification method with 98.4% accuracy.

Roopak et al. [9], suggested using the RIPPER method to detect phishing web pages. For the performance measurement, the "Phishing Web Site Data Set", which is available on the UCI website and contains 30 features of 1055 websites, is used. In the study, classification success is obtained with accuracy of 92%.

Tan et al. [10], proposed a total of 17 web graphics features created from the web graphics of 500 legitimate and 500 phishing webpage's ego network which obtained from PhishTank, OpenPhish, Alexa and Common Crowl Archive and used them with machine learning methods to detecting phishing webpages. Experimental results show that the best machine learning method for determining phishing web pages using the recommended graphic properties is C4.5 with 98.7% correct classification rate.

Opara et al. [11], proposed to using Convolutional Neural Networks (CNNs) for detecting phishing webpages based on learning the textual contents and the semantic relationships of other characters on their pages. A data set of 47000 legitimate and 4700 phishing website URLs has collected within 60 days from phishing and legitimate web pages with a web browser.

As a result of the analysis with using the relevant data set, results show that, Convolutional Neural Networks (CNNs) has achieved an accuracy of 93% in detecting phishing web pages.

Ding et al. [12], proposed the SHLR method in which three methods, namely Search and Heuristic Rule and Logistic Regression, are used together in detecting phishing web pages. In this method, first the URLs of the pages are searched in the Baudi search engine. If relevant URLs are included in the top 10 results, this website is labeled as legitimate. If the web page is not labeled as legitimate in the first stage, it is checked whether it is a phishing page according to the heuristic rules defined by character features in the second stage. Finally, a logistic regression classifier is used to increase the performance accuracy of the method. As a result of the performance analysis, it is seen that the SHLR method has an accuracy rate of 98.9% in the classification of phishing web pages.

Chatterjee and Namin [13], proposed a reinforcement learning model that uses the deep neural network to detect phishing web pages. Ebbu2017 Phishing Data Set was used to measure the performance of this study. The dataset contains 73575 URLs, 36400 of which are legitimate and 37175 are phishing. As a result of the experimental studies, it has seen that the reinforcement learning model using the deep neural network performs classification with an accuracy of 86.7%.

Wu et al. [14], proposed the Support Vector Machine (SVM) method, one of the machine learning methods for detecting phishing websites. The dataset used for the performance analysis of the study includes 5000 phishing web pages from Phishtank and 10000 legitimate web pages from DMOZ. As a result of the experimental studies, it has seen that SVM has achieved an accuracy rate of 92.8% in the classification of phishing websites.

When the studies in the literature are examined, it is seen that the systems used to detecting phishing web pages are generally developed using content-based, URL-based and machine learning-based methods.

III. Study Methodology

In this section, the data set used to detecting phishing web pages in the experimental study and the machine learning methods applied in this data set are presented. Then, the performance comparisons of the machine learning methods used in the study and the results obtained have analyzed.

A. Dataset and Feature Extraction

In the study, a dataset containing information about a total of 1353 URLs consisting of 702 phishing URLs, 103 suspicious URLs and 548 legitimate URLs obtained from various internet sources (Phishtank, Yahoo...etc.) by Abdelhamid et al [4].

The sample data in the first five lines of the dataset used in the study are shown in Fig. 1.

	0	1	2	3	4	5	6	7	8	9
0	1	-1	1	-1	-1	1	1	1	0	0
1	-1	-1	-1	-1	-1	0	1	1	1	1
2	1	-1	0	0	-1	0	-1	1	0	1
3	1	0	1	-1	-1	0	1	1	0	0
4	-1	-1	1	-1	0	0	-1	1	0	1

Fig. 1. The first five lines of the dataset

As seen in Fig.1, in dataset, there are 9 features of each URL. These features and their explanations are seen in Table 1.

TABLE I. Properties of Dataset Attributes

No	*Feature Name*	*Explanation*
0	SFH (-1,0,1)	If SFH contains 'about:blank' or empty string: phishing, if refers to a different domain: suspicious, otherwise:legitimate
1	popUpWindow (-1,0,1)	Whether popUp Window opens when the URL is accessed and if rightClick disabled: phishing, if rightClick giving alert: suspicious, otherwise: legitimate
2	SSLfinalState (-1,0,1)	If URL have a https tag, have a SSL certificate and age of this certificate more than two: legitimate, if URL have a https tag but doesn't have a SSL certificate: suspicious, otherwise: phishing
3	RequestURL (-1,0,1)	The webpage URL and most of objects are not loaded from the same domain: phishing (if percentage of URL request >61%), if percentage of URL request >21% and <61%: suspicious and otherwise: legitimate
4	URLofAnchor (-1,0,1)	In the webpage if the anchor (<a>) points to a different domain more than the webpage's domain (if percentage of URL of anchor >66): phishing, if percentage of URL of anchor >30% and <66%: suspicious and otherwise: legitimate
5	WebTraffic (-1,0,1)	If the webpage's domain has no traffic or is not recognized by Alexa database (www.alexa.com): phishing, if the webpage rank>100.000: suspicious, if the webpage rank<100.000: legitimate
6	URL_Length (-1,0,1)	If URL length<55: legitimate, if URL length>54 and <76: suspicious, otherwise: phishing
7	AgeOfDomain (-1,1)	If age of domain>6 months: legitimate, otherwise: phishing
8	HavingIPAddress (-1,1)	If an IP is used in part of the URL:phishing, otherwise: legitimate
9	Class (-1,0,1)	Phishing, suspicious and legitimate

As in Table 1, each feature of URLs and the result class value takes one of the values "-1" (phishing), "0" (suspicious), and "1" (legitimate), taking into account the conditions specified in the descriptions.

The heatmap of Spearman's correlation matrix which showing the colleration degrees of the features with each other is shown in Fig. 1.

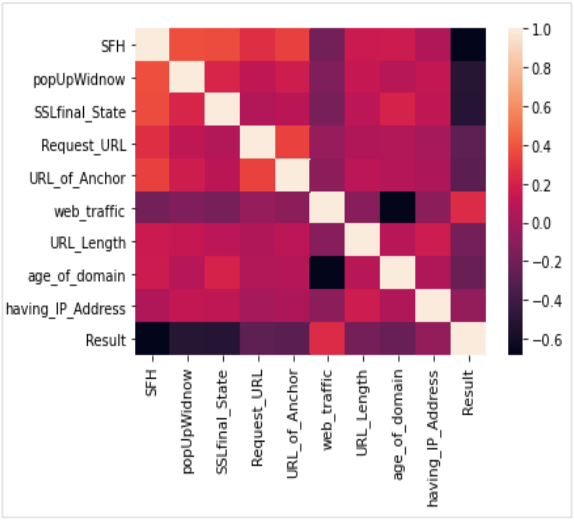


Fig. 2. Correlation analysis using correlation matrix with heatmap

When the heatmap in Fig. 2 is examined, it is seen that the result class value has a weak positive relationship with the web_traffic feature (0.243896) and a negative relationship with the other features. It is also seen that that the highest negatively correlated feature with the result class is SFH (-0.678277) and the lowest negatively correlated feature is URL_Length (-0.183061).

B. Model Selection

In this section, machine learning methods used to detect phishing web pages are discussed in detail. In the study, the methods applied on the relevant dataset have implemented by using Scikit-Learn (sklearn) which is an open source machine learning library in Python. Machine learning methods used in the study and information about these methods are presented below.

1) Logistic Regression (LR): Logistic Regression is a statistical model that is used in the classification of categorical and numerical data and can be used when the dependent variable has only two different values[15].

2) Support Vector Machine (SVM): SVM is a supervised learning method based on the statistical learning theory. Support Vector Machines are mainly used to distinguish between two classes of data in an optimal way. An optimum hyperplane is determined to separate the two classes. Thus, samples with similar characteristics can be categorized by assigning them to predetermined classes [16].

3) Multilayer Perceptron (MLP): A multilayer perceptron is a class of artificial neural network (ANN). MLP consists of the input layer where the information is entered, the hidden layer (s) where the data is processed and the models are performed, and an output layer. There are transitions between layers of the MLP, called forward and backward spans. In forward propagation, the output and error value of the network is calculated. In backpropagation, the connection weight values between the layers are constantly updated in order to minimize the calculated error value [17].

4) K-Nearest Neighbors (KNN): KNN is a supervised learning method which proposed by T. M. Cover and P. E. Hart in 1967. In this method, when a new data is added to a sample space consisting of (labeled) data, the new data is classified by looking at the distance to its nearest k neighbors [18].

5) Naïve Bayes Classifier (NB): The Naïve Bayes classifier is a method that makes statistical inference based on Bayes' theorem [19]. This theorem describes the relationship between conditional probabilities and a priori (marginal) probabilities for a random variable. The Naïve Bayes Classifier aims to identify the class of new data submitted to the system, using a set of calculations defined according to probability principles [20].

6) Decision Trees (DT): Decision Trees method is a supervised learning technique that can be used for both classification and regression problems. It is a tree structured classifier in which internal nodes represent the properties of a data set, branches represent the decision rules, and each leaf node represents a result. A Decision tree has two nodes, a decision node and a leaf cluster. Decision nodes are used to make any decision, and leaf nodes are the output of these decisions [21].

7) Random Forest (RF): Random Forest is a supervised learning method used for both classification and regression. The random forest model is also an ensemble method that trains several trees in parallel and uses the majority decision of trees as the final decision of the model [22].

8) AdaBoost (AB): AdaBoost, short for "Adaptive Boosting" is an ensemble learning method which first created in 1996 by Freund and Schapire to increase the efficiency of binary classifiers. AdaBoost focuses on classification problems and uses an iterative approach to learning from weak classifiers' errors, and aims to transform them into strong ones [23].

IV. Empirical Study

Of the dataset used in the study, 70% is decomposed for training and 30% for test data. Thus, 947 training data and 406 test data have obtained.

In the study, the confusion matrix has created to measure the classification success of machine learning methods on the relevant data set, and then the precision, recall, F-score and accuracy values have calculated based on the confusion matrix.

The confusion matrix is a method used to determine the performance values of algorithms with a matrix based on the numbers of real values and predicted values, based on the data obtained as a result of the operations performed on test data by machine learning methods. Table 2 shows the contents of the confusion matrix.

TABLE II. Confusion Matrix

Confusion Matrix		***Actual***	
		Class 1	Class 2
Predicted	Class 1	True Positives (TP)	False Positives (FP)
	Class 2	False Negatives (FN)	True Negatives (TN)

According to Table 2, the confusion matrix obtained based on the prediction results of each machine learning method for the relevant test data set is shown in Fig.3, Fig.4, Fig.5 and Fig.6.

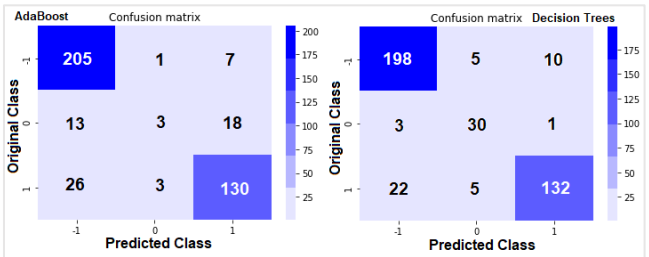


Fig. 3. Confusion matrices for AdaBoost and Decision Trees

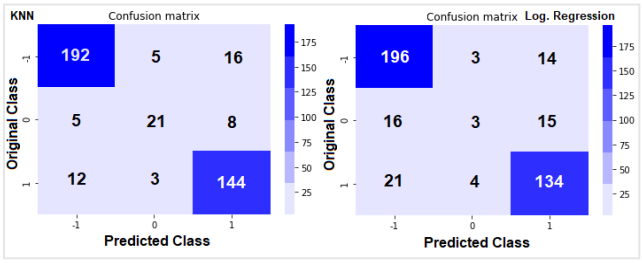


Fig. 4. Confusion matrices for KNN and Logistic Regression

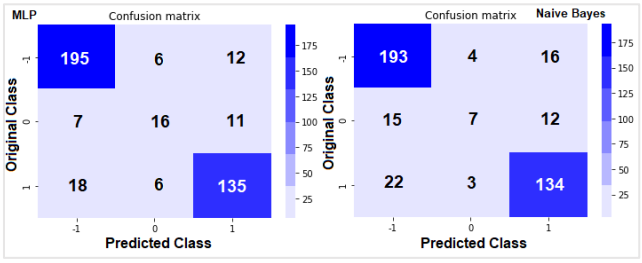


Fig. 5. Confusion matrices for MLP and Naïve Bayes

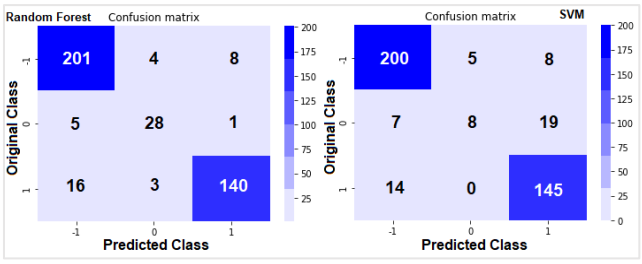


Fig. 6. Confusion matrices for Random Forest and SVM

Precision and recall are two values that are used to evaluate the classification performance of algorithms used in solving a problem. Precision is defined as the proportion of relevant instances among all retrieved instances. Precision value is calculated as in (1). Recall, is the proportion of retrieved instances among all relevant instances. Recall value is calculated as in (2). F-Score is used for the purpose of obtaining more reliable results by evaluating the precision and recall values. F-Score value is calculated as in (3). Accuracy is the ratio of correctly classified samples to the total number of samples. Accuracy value is calculated as in (4).

$$Precision\ (P)\ = \frac{TP}{(TP+FP)} \tag{1}$$

$$Recall(R) = \frac{TP}{(TP+FN)} \tag{2}$$

$$F-Score = 2 * \frac{(P*R)}{(P+R)} \tag{3}$$

$$Accuracy(A) = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{4}$$

Table 3 shows the performance values of the classification algorithms calculated on the basis of (1), (2), (3) and (4).

TABLE III. COMPARISON OF CLASSIFICATION PERFORMANCES

Method	*Class*	*Precision*	*Recall*	*F-Score*	*Accuracy Score*
KNN	-1	0.92	0.90	0.91	0.87931
	0	0.72	0.62	0.67	
	1	0.86	0.91	0.88	
LR	-1	0.92	0.84	0.88	0.82019
	0	0.09	0.30	0.14	
	1	0.84	0.82	0.83	
DT	-1	0.93	0.88	0.90	0.88177
	0	0.85	0.74	0.79	
	1	0.82	0.92	0.87	
SVM	-1	0.94	0.90	0.92	0.86945
	0	0.24	0.62	0.34	
	1	0.91	0.84	0.88	
MLP	-1	0.92	0.89	0.90	0.85221
	0	0.47	0.57	0.52	
	1	0.85	0.85	0.85	
NB	-1	0.91	0.84	0.87	0.82266
	0	0.21	0.50	0.29	
	1	0.84	0.83	0.83	
RF	-1	0.91	0.92	0.92	0.89408
	0	0.76	0.76	0.76	
	1	0.90	0.88	0.89	
AB	-1	0.96	0.84	0.90	0.83251
	0	0.09	0.43	0.15	
	1	0.82	0.84	0.83	

When Table 3 is examined, it is seen that the Random Forest Method is the machine learning method that performs the most successful classification on the relevant test dataset.

V. CONCLUSION

The widespread use of the internet day by day brings some problems with it. One of the most important of these problems is phishing websites. Phishing websites created for fraud purposes by accessing users' personal data such as identity, credit card, password, address can cause serious material and moral damage to users.

In this study, SVM, MLP, Random Forest, Naive Bayes, KNN, Logistic Regression, Decision Trees and Adaboost algorithms, which are among the most used machine learning methods in the literature, have used to detect phishing websites. A data set obtained from the Machine Learning Pool (UCI) [4] is used to be applied in machine learning methods. This dataset contains a total of 1353 data, including 702 phishing, 103 suspicious and 548 legitimate URLs, including 9 attributes.

As a result of the experimental studies conducted to detect phishing web pages, it is seen that the machine learning method that performs the most successful classification on the dataset used is Random Forest with an accuracy rate of 89.4%.

In future studies, it is planned to identify phishing websites based on the hybrid use of content-based methods.

REFERENCES

[1] APGW Trends Report. (2020, September 16) [Online] Avaiable: https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf

[2] Webroot Threat Report. (2020, September 16) [Online] Avaiable: https://www.webroot.com/us/en/about/press-room/releases/nearly-15-million-new-phishing-sites

[3] Anadolu Agency. (2020, September 16) [Online] Avaiable: https://www.aa.com.tr/tr/turkiye/14-bin-120-oltalama-sitesi-erisime-kapatildi/1850849

[4] N. Abdelhamid, A. Ayesh and F. Thabtah, “Phishing detection based associative classification data mining,” Expert Systems with Applications, vol.41, no:13, pp.5948-5959, 2014.

[5] Y. Li, Z.Yang, X. Chen, H. Yuan, and W. Liu, “A stacking model using URL and HTML features for phishing webpage detection,” Future Generation Computer Systems, vol. 94, pp. 27-39, 2019.

[6] K. Abeywardena, J. Zhao, L. Brent, S. Seneviratne, and R. Holz, “Triplet Mining-based Phishing Webpage Detection,” In Local Computer Networks, 2020.

[7] O. K. Sahingoz, E. Buber, O. Demir and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol.117, pp.345-357, 2019.

[8] A. K. Jain and B. B. Gupta, “A machine learning based approach for phishing detection using hyperlinks information,” Journal of Ambient Intelligence and Humanized Computing, vol.10, no.5, pp.2015-2028, 2019.

[9] S. Roopak, A. P. Vijayaraghavan and T. Thomas, “On Effectiveness of Source Code and SSL Based Features for Phishing Website Detection,” 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), pp. 172-175, 2019.

[10] C. L. Tan, K. L. Chiew, K. S. Yong, J. Abdullah and Y. Sebastian, “A Graph-Theoretic Approach for the Detection of Phishing Webpages,” Computers & Security, vol. 101793, 2020.

[11] C. Opara, B. Wei and Y. Chen, “HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis,” International Joint Conference on Neural Networks (IJCNN), 2020.

[12] Y. Ding, N. Luktarhan, K. Li and W. Slamu, “A keyword-based combination approach for detecting phishing webpages,” Computers & Security, vol.84, pp.256-275, 2019.

[13] M. Chatterjee and A. S. Namin, “Detecting phishing websites through deep reinforcement learning,” 43rd Annual Computer Software and Applications Conference (COMPSAC), Vol.2, pp. 227-232, 2019.

[14] C. Y. Wu, C. C. Kuo and C. S. Yang, “A Phishing Detection System based on Machine Learning,” International Conference on Intelligent Computing and its Emerging Applications (ICEA), pp.28-32, 2019.

[15] J. Tolles and W. J. Meurer, “Logistic regression: relating patient characteristics to outcomes” Jama, vol.316, no.5, pp.533-534, 2016.

[16] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no.3, pp.273-297, 1995.

[17] A. Arı and M. E. Berberler, “Yapay sinir ağları ile tahmin ve sınıflandırma problemlerinin çözümü için arayüz tasarımı,” Acta Infologica, vol.1, no.2, pp.55-73, 2017.

[18] G. D. Cavalcanti and R. J. Soares, “Ranking-based instance selection for pattern classification. Expert Systems with Applications,” vol. 150, no.113269, 2020.

[19] H. Karamollaoğlu, İ. A. Doğru and M. Dörterler, “Detection of Spam E-mails with Machine Learning Methods,” Innovations in Intelligent Systems and Applications Conference (ASYU), pp.1-5, 2018.

[20] M. Granik and V. Mesyura, “Fake News Detection Using Naive Bayes Classifier,” First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pp.900-903, 2017.

[21] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology” Transactions on Systems, Man, and Cybernetics,vol. 21, no.3, pp. 660-674, 1991.

[22] S. Misra, Y. Wu, “Machine learning assisted segmentation of scanning electron microscopy images of organic-rich shales with feature extraction and feature ranking,” Machine Learning for Subsurface Characterization, vol. 289, 2019.

[23] T. Hastie, S. Rosset, J. Zhu and H. Zou, “Multi-class adaboost. Statistics and its Interface,” vol.2, no.3, pp.349-360, 2009.

Big data interoperability measurements and decision making

Weiam S. Elsaghair
Institute of Graduate Studies
Altınbaş university
Istanbul, Turkey
Weiam2005@gmail.com

Dogu Cagdas Atilla
School of Engineering and Natural Sciences
Altinbas University, Turkey
cagdasatilla@gmail.com

Abstract— **Big data arise in many varieties of forms than classical data, and it is gathered with a great rate of volume. Imagine all the data that was generated by different social media applications, this data is valuable because it is amount, variety and growth acceleration, but it is hard to manage. Big data interoperability concerns with the possibility to proficiently exchange data between organizations, knowledge and services between organizations. That knowledge and information were extracted from the raw data and big data. This complex mutual process growth organization's performance. This study aims to measure big data interoperability and investigate its effect on decision making. The issue is that the interoperability can be enhanced means the measuring metrics can be illustrated. To measure the interoperability between systems, interoperability assessment model is required. This paper deals with the big data interoperability in concern of decision making. The analysis provides a set of recommendations to improve an interoperability assessment.**

Keywords—Big Data interoperability, Interoperability assessments, Decision making.

I. INTRODUCTION

There were many definitions of big data from different aspects of information, technology, methods and impact. For instance, from a technical aspect, big data is a complex large data silo which should be deal with beneficial application analytical techniques, and advanced particular storage with efficient management, analysis, and data visualization [1]. Moreover, another brief method definition is the Data that cannot be handled and processed straightforwardly[1], also in concern of information big data conducted as vast datasets, with characteristics as a huge amount of data, speed of growth and variety, that need adaptable structure to achieve an efficient performance. In my opinion, big data is a huge amount of data and it has different features (velocity, variety, value, volume, veracity). Therefore, and to achieve better using of the big data, it has to be exchanged between different heterogeneous systems, and the exchangeability comes with another concept which is big data interoperability and that impose modern challenges to management and operational challenges.

Big data interoperability influences on the whole organization performance, because big data interoperability and big data architecture is a complex process with different heterogeneous systems and data sources because the data is stored and integrated as a single entity. So, many challenges have appeared. Those challenges are data inconsistency, Data quality and poor resources, Personal data protection, Different Schema and conflicts, Data analytics and processing time, Data duplication, Query optimization.

So, the organizations need to improve their data interoperability because it will affect the firm's resources as a whole system. Also, that supports the major types of information systems inside business organizations (TPS, MIS, DSS, and ESS). On the other hand, decision making as strategy level and their decision support systems, and executive support systems in the organizations and it is related to big data interoperability becomes a crucial issue.

Dss is a complex platform storing and managing data, Dss should strongly and smoothly be integrated and interacting with other components like management information system and transaction processing systems. those components bring data from external resources and use it to be the raw material for Dss. So, exchanging the data efficiently and effectively sturdily related to the interoperability concept.

II. BIG DATA INTEROPERABILITY

Big data interoperability concerns with big data features and how the ability of these features to be exchangeable between two systems or components. Data interoperability interfered with other terminologies like data integration and data exchange. Because implementing data interoperability needs to adopt data integration and data exchange. Therefore, many definitions of data interoperability have emerged, as the diverse heterogeneous systems to be able to interconnect with each other and use the exchanged information that has been generated [2]. Another definition of data interoperability as the possibility to accurately manipulate data that override the enterprise's boundaries [4]. Both Data and interoperability concepts are hard to be perfectly recognized. The term "data" takes different forms, collected and generated by observations, statistics, records, artefacts…etc.

And is aggregated for many reasons, with different approaches.

So often it is hard to collect and interpret this kind of data which is dropped into the category of big data with their specific features (Velocity, Variety, Value, Volume, Veracity). Therefore, the term "Interoperability" does not yet have a well-defined can be collaborative with the community. IEEE with incorporate the US Department of Defense they defined the interoperability as providing the services by system groups and accept them from other ones of system groups, or forces to provide services to and accept services from other systems, units, or forces, and to use the services so exchanged to enable them to operate effectively together [5].

Big data interoperability has three levels. Technical Interoperability, conceptual interoperability, organizational interoperability. Technical interoperability discusses different aspects:

1) *Software aspect*: system components or applications to establish communication, interconnection services, data integration services and share resources.
2) *Hardware aspect:* deals with communication protocols and infrastructure. Moreover, the conceptual interoperability categorized into two features syntactic and semantic.
   a) *Syntactic:* it is mainly related to data formats. It can be defined as the communication standardization between the clients and the servers [6].
   b) *Semantic:* it is about the content definition, concerns with the information interpretation of two communicated systems. That interpretation has to be compatible, in considering different database and large corporation systems [7].

On the other hand, the organizational interoperability concern with the ability of the organizations to successfully communicate and transfer the relevant data. Achieving the business goals by improving the processes and enhance the organizational collaboration to exchange the data between different internal architectures.

III. INTEROPERABILITY BARRIERS

Interoperability barriers present the inconsistency of information systems (platform, architecture, and framework…etc.), those barriers show the fundamental issues that prevent the overall society from participially work. Many studies conducted the interoperability hurdles in three types of problems (Conceptual, technological, and organizational), those categories cover different aspects related to interoperability levels and how these categories concern with incompatibilities of information exchangeability. Also, some studies proposed a correlation between data integration and data interoperability in challenges perspective as following: [9]

1) *Accommodate scope of data*: New domains can be created to accommodate the huge range of data and improve response time with query optimization, also introduce better reliability and availability.
2) *Data Inconsistency:* Structured and unstructured data in heterogeneous sets lead to data inconsistency.
3) *Query Optimization*: queries could be optimized by decreasing the queries amount and using all attributes of the relational database and their sequence queries language. Also, improve the latency and response time by using multiprocessing technique. So, the previous optimization and processing techniques can be achieved to conquered these hurdles.
4) *Scalability:* This issue appears when the data from different resources are combined with the traditional ones.
5) *Inadequate Resources*: lack of different assets like skilled professionals, the appropriate budget and implementation costs like getting a license and modern tools, can effect on organization performance and the ability to handle projects.
6) *ETL Process*: For data integration, a systematic Extract Load Transform process is recommended.
7) *support system Implementation*: to achieve the integrity of the data, by performing the updates and fixing errors, the training module is recommended. and this will be beneficial for future updates or modifications. Therefore, other studies summarized the challenges of data interoperability into three issues:
   a) insufficient definition of the problem.
   b) dealing with diversity.
   c) Agreeing on the possibility of reusing the data.

Defining all the facets of the interoperability in a structured and unified manner by a common framework. That framework used to illustrate interoperability approaches to ensure it will be used regards to the domain in which it has been developed. Also, managing heterogeneous different independent data resources with respect of type, accuracy, size, semantic reveals a very broad aspect of data variety and its effect on interoperability. Likewise, data have to be re-used in considering the value, to make that possible contextual information about such data has to be available.

Enterprises concerns could be data, service, process, and business. Those concerns are another issue that can be relevant to the interoperability barriers (Syntactic and Semantic, Organisational, legal and technological). Table1 conducts the barriers of interoperability with interoperability concerns interest [10].

A. Conceptual barriers (syntactic/ Semantic): refers to the modelling of data format and content definition and interpretation.

B. Technological barriers: regards to heterogeneous technical compatibility standards systems, and sharing data between different systems.

C. Organisational barriers: concerns with organization architecture, business functions and management approaches.

D. Business concerns: organizations work in a concerted way despite different legislations, cultures and commercial

strategies.

E. Process concerns: the ability of various processes to work together.

F. Data concerns: sharing data of different database systems and different framework.

G. Services concerns: refers to stimulate various applications to work together.

Therefore, according to the previous, interoperability barriers related to different aspects. Furthermore, poor data quality, data protection considerations and legal barriers, and decoupling data. Are other interoperability barriers issues have to be considered.

TABLE I. DESCRIBES THE INTEROPERABILITY BARRIERS IN CONSIDER THE ENTERPRISE'S CONCERNS.

		The barriers of Interoperability		
		Conceptual	***Technological***	***Organizational / Legal***
Interoperability Concerns	Business	Visions, strategies, Organization cultures, understanding.	- IT infrastructure.	- organization structure. - Legislations. - Business rules.
	Process	Syntax and semantics processes	Process interfaces and supporting tools.	Procedures of work, processes organization.
	Service	Semantics to name and describe services.	Interface, architecture.	Responsibility/ authority to manage services.
	Data	- Data representation and semantics. - Data restriction rule.	Data exchange formats.	Responsibility authority to add/delete, change/ update data.

IV. INTEROPERABILITY MEASUREMENTS

Data interoperability measurements use to disclose strengths and weakness of partner's interoperation exchanging information and services, also to avoid possible problems and achieving suitable enterprise collaboration, and better services and support. So, to improve organizations data interoperability and their ability to interoperate with the other's, so they have to be Conscious of their current interoperability situation [11]. Interoperability measurement has an essential rule to support enterprise collaboration, an organizations ability to continuously improve their information exchange, operational process, sustainable procedures, and strategic plans. So, data interoperability regularly needs to be optimized. Here in this paper, three types of data interoperability measurements will be considered:

A. The potentiality measurement.

B. The compatibility measurement.

C. The performance measurement.

Those three assessment types covered different aspects begins with overcoming the possible obstacles when interacting with a potential partner. The objective of potentiality measurement is to assess the potentiality of organizations applications and process, and how their systems can be flexible and accommodate with respect of the prospective partner. The potential interoperability refers to the assessment of enterprises indicating to the four types of obstacles (technological, conceptual, organizational and legal), and the concerns of interoperability (process, business, data and services), for each of them there are five scales of potentiality:

1) Isolated: determine the entire inability to interoperate.

2) Initial: enormous effort is required to enforce the interoperability.

3) Executable: concern with the possibility of interoperability with a high rate of encountering problems.

4) Connectable: refers to the far partnership, and how to enable easy interoperability.

5) Interoperable: controls the levels of interoperability and their evolution.

Another interesting aspect stated that potentiality conduct that each enterprise components have some relative interoperability attributes, to be adequately convenient to interoperate with other enterprises. Also, potentiality means an inside enterprise assessment of interoperability without exact need to know the partner [15]. tabel2 describes the properties with high potentiality.

TABLE II. SHOWS THE IMPORTANT PROPERTIES GIVING HIGH POTENTIALITY.

Open (1)	Decoupled (1)	Decentralized(1)	Configurable(1)
Close(0)	Coupled(0)	Centralized(0)	Not-configurable(0)

On the other hand, the following assessment has to be achieved through the engineering phase, the compatibility measurement, determine the conflicts of two interacted systems that conflicts cause or may cause problems, the evaluation could be before or after interoperation. The objective is to check the different incompatibility aspects (conceptual, organisational and technological compatibilities) with respect to interoperability barriers. That can be conducted using interviews or questionnaire.

For instance, in table 3, the coefficient matrix has been described with 1 sign for incompatibility or 0 sign for known incompatibilities. Then, we have the performance measurement, which conducted to be evaluated during run time, also there are three criteria are investigated to measure operational performance.[13]

TABLE III. SHOWS THE COMPATIBILITY MEASUREMENTS MATRIX

Barriers / Concerns	Conceptual		Legal		Organizational			Technology		
	Syntactic	*Semantic*	*Data Privacy*	*Legislations*	*Authorities*	*Organization rules*	*Business Processes*	*platform*	*Communication*	*Data Integrity*
Business	1	1	1	0	0	0	1	0	0	1
Processes	1	1	1	1	0	1	1	0	1	1
Services	0	1	1	1	0	0	1	0	1	1
Data	0	0	0	1	1	1	0	0	0	0

1) The interoperate cost: refers to the cost of removing the barriers, and system modification. This cost could be the cost of interoperation or the expected cost of using the exchanged information.

$C_{ex}= (E_{Cex} - R_{Cex})/ E_{Cex}$. (1)

C_{ex}: Cost of exchange.

E_{Cex}: Expected cost of the exchange.

R_{Cex}: Real cost of the exchange.

2) The interoperate time: it is the difference between the time of the information that has been requested and the time that the information has been used.

$T_{ex}=(E_{tex}-R_{tex})/ E_{tex}$. (2)

T_{ex}: Time of the exchange.

R_{tex}: The real-time of the exchange.

E_{tex}: The expected time of the exchange

3) The interoperate quality: information quality is another measurement in consideration of performance measurement, according to that three types of quality are expressed.

The quality of exchange: indicates to the data that are successfully posted.

The quality of utilization: indicates the amount of acquired data in comparison to the demanded.

$Q_{ex}= A_{Succex}/A_{tex}$ (3)

Q_{ex}: Quality of the exchange.

A_{Succex}: Amount of succeeded exchange.

$A_{tex:}$ Total amount of exchange.

4) Conformity: concerns with the received information, and the usability rate of that information.

Cex=Aconex/Arecex (4)

C_{ex}: Conformity of the exchange.

A_{conex}: Amount of information that is conforming.

A_{recex}: Amount of information received.

V. WEIGHTED DECISION MATRIX

a weighted decision matrix is a quantitative technique, we can use it to assign weightings to factors, those weights assigned from high to low weights according to their interesting rate. This weighted decision matrix can be useful when you have choices with different features and you need to select one of them. Those previous features could have cost, market value, risk, importance, reliability…. etc. [21]

TABLE IV. DESCRIBES A WEIGHTED DECISION MATRIX.

Weighted decision matrix							
		Options					
Criteria	***w/t***	***1***		***2***		***3***	
		Sre	***Ttl***	***Sre***	***Ttl***	***Sre***	***Ttl***
Potential measurement							
Isolated potential measurement	5	10	50	7	35	5	25
Initial potential measurement	3	6	18	8	24	6	18
Executable potential measurement	4	3	12	7	28	8	40
Connectable potential measurement	5	8	40	8	40	6	30
Interoperable potential measurement	3	9	27	9	27	4	12
Incompatibility measurement							
Conceptual barries	5	10	50	9	45	7	35
Legal barries	4	8	32	10	40	6	24
Organizational barries	5	9	40	9	45	7	35
Technolgy barries	5	9	40	9	45	7	35
Performance measurement							
The cost of interoperate	3	9	27	7	21	4	12
The time of interoperate	5	8	40	8	40	6	30
The quaility of interoperate	5	9	45	8	40	6	30
			421		430		326

w/t: Criteria weight.

Sre: score.

Ttl: Total.

Criteria weight from 5 to 1.

The score of assessments from 1 to 10.

VI. CONCLUSION

The process of measuring big data interoperability and its effect on the decision making clearly appeared. Many interoperability barriers have been revealed (conceptual barriers, Technological barriers, Organizational barriers, Process concerns, Services concerns). Those barriers have to be eliminated, because they have a negative effect on interoperability performance, and the decisions that are not based on high interoperability indicators Cannot be reliable. that will be conducted in the study, where three important indicators of interoperability measurement will be analyzed and these indicators definitely impact on decision making. Also, these indicators negotiate the following:

1) *how the organisation's applications can be perfectly dealled with the expected partner.*
2) *check the different incompatibility aspects with respect of interoperability barriers.*
3) *criterions of information exchange like the exchange cost, exchange time, and data quality*

Through, the previous points which directly affect the organization's interaction and information transfer, and by measuring potentiality the deficiency can be located, it is clear that making appropriate decisions due to the rates of previous indicators must be discussed.

REFERENCES

[1] Andrea De Mauro,"A formal definition of Big Data based on its essential features," Emerald Group Publishing Limited, Vol. 65 No.3,2016, pp122-136.

[2] "Definition of Interoperability," HIMSS Board of Dirctors, April 5, 2013.

[3] Marijan janssen, Elsa Estevez and Tomasz janowski, "Interoperability in big, open, and linked data organizational maturity, Capabilities, and data portfolios," IEEE Computer Society, pp. 44, October 2014.

[4] Olaronke Iroju, Abimbola soriyan, Ishaya Gambo, and Janet Olaleke, "Interoperability in healthcare: Benefits, Challenges and Resolutions", International Journal Of Innovation and Applied Studies, ISSN 2028-9324 Vol. 3 No. 1 May 2013, pp.262-270.

[5] Reza Rezaei, Thiam-Kian Chiew, Sai-peck Lee,"A review of interoperability assessment models," Journal of zhejiang university-science (computer&Electronics) , Aug 21, 2013.

[6] Martin Sudmanns, Dirk Tiede, Stefan Lang and Andrea baraldi, "Semantic and syntactic interoperability in online processing of big Earth observation data," International Journal of Digital Earth, 06 june 2017.

[7] F.B Vernadat,"Interoperability enterprise systems: Principles, concepts, and methods," DG-DIGIT JMO C2/103 Luxemburg, 6 April 2007.

[8] F.B Vernadat,"Technical, sematic and organizational issues of enterprise interoperability and networking," LGIPM: Laboratoire de Ge´nie Industriel et Production Me´canique, ENIM/Universite´ de Metz, Ile du Saulcy, F-57045 Metz Cedex, France, 22 Feb 2010.

[9] Anirudh Kadadi, Rajeev Agrawal, Christopher Nyamful, Rahman Atiq*,"Challenges of Data Integration and Interoperability in Big Data,"IEEE International Conference on Big Data,2014.

[10] Gabriel Leal, Wided Guedria, Herve Panetto,"Interoperability Assessment: A Systematic Literature Review," Elsevier, 2019, 106, pp.111-132.

[11] Gabriel da Silva Serapiao Leal, Wided Guedria, Herve Panetto, ,"Interoperability assessment: A systematic Literature review," Luxembourg Institute of Science and Technologie (LIST), 11 January 2019.

[12] Nicolas Daclin, Bruno Vallespir, David chen,"Enterprise interoperability measurement- Basic concepts," University Bordeaux · January 2006.

[13] Yves Ducq, David Chen,"How to measure interperability: Concept and Approach," IMS-LAPS/GRAI, University of Bordeaux, UMR 5218 CNRS, 2008.

[14] David chen, Bruno vallespir, Nicolas Daclin,"An approach for enterprise interoperability measurement," Proceedings of MoDISE-EUS 2008.

[15] N. Daclin, D. Chen, B. Vallespir,"Methodology for enterprise interoperability," The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

[16] Reza rezaei, Thiam-kian chiew, sai-peck Lee,"A review of interoperability assessment models," Journal of Zhejiang University-SCIENCE C (Computers & Electronics), 2013.

[17] Par wided guedria,"A contribution to enterprise interoperability maturity assessment," University of Bordeaux, 2012.

[18] Pasquale Pagano, Leonardo Candela, and Donatella Castelli,"Data Interoperability," Data Science Journal, Volume 12, 23 July 2013.

[19] François B.Vernadat,"Technical, semantic and organizational issues of enterprise interoperability and networking," LGIPM: Laboratoire de Ge´nie Industriel et Production Me´canique,2010.

[20] David Ball,"A weighted decision matrix for outsourcing library services," volume16. Number1.2003.

The Real-Time Detection of Red Californian Worm Eggs

Ali ÇELİK
Dept. of Mechatronic Engineering
Isparta University of Applied Sciences
Isparta, Turkey
https://orcid.org/0000-0003-3517-7255

Sinan UĞUZ
Dept. of Computer Engineering
Isparta University of Applied Sciences
Isparta, Turkey
https://orcid.org/ 0000-0003-4397-6196

Abstract*— Deep learning techniques, which are being used effectively in many areas today, will become more and more important every day and will increase the competitiveness of producers by contributing significantly to the digital transformation in the agricultural sector. In this study, worm eggs (cocoons) detected by convolutional neural network (CNN) model created using single shot multibox detector (SSD) architecture were separated from compost. The data set consists of 1000 compost images containing 3809 worm eggs and a single class named "cocoon" representing the worm eggs. The success of different iteration numbers in object detection has been observed in the SSD architecture used by transferring learning over VGG16.

Keywords—Object detection, CNN, SSD, Eisenia Fetida, Red Californian Worm, Vermicompost, Worm egg, Cocoon.

I. INTRODUCTION

In recent years, it has been observed that applications related to deep learning based on artificial neural networks, which is a sub-field of machine learning, have come to the fore in artificial intelligence research. With the architectures developed in deep learning, significant successes have been achieved in problems such as classification, object detection, object segmentation. One of the application areas where deep learning studies are common is the agriculture sector. It is seen that especially object detection studies have an important place in agricultural practices. It can be said that the use of organic fertilizers has made relatively less progress in our country compared to other agricultural countries in the world. For this reason, more qualified scientific studies are needed for the widespread use of organic fertilizer, which is an important element of organic agriculture. The production of innovative technologies by investigating the process from the production of worm fertilizer, which is widely used in the world, to its use, can undoubtedly be seen as an important gain for our country. In this study, it is aimed to detect worm eggs using the Single Shot Multiple Detection (SSD) architecture below.

A. RED WORM AND VERMICOMPOSTING

It is commonly called worm compost as "vermicompost", and worm compost production as "vermicomposting". Although the Latin name is Eisenia Fetida, the production of worm compost is generally made by the worm species known as the Red Californian Worm [1].

Worm castings are obtained as a result of separating plant and animal organic wastes and passing through the digestive systems of worms. It is an organic fertilizer type that increases the yield, endurance and quality of vegetables, fruits and plants, and strengthens the resistance against diseases around the root. Its appearance is similar to black soil and it does not bother with its odorless feature compared to other fertilizer types. The most important feature is that it is the number one alternative to chemical fertilizers due to its organic structure. (Şimşek Erşahin, 2013).

B. RELATED STUDIES

Studies that are close to the subject of this study can be examined in sub-categories such as the identification, classification of agricultural products and disease or defect detection. Below is a summary of recent literature studies related to these topics.

Reference [2] proposed a PLC-controlled conveyor system based on image processing that sorts apples according to their class, weight and size and can detect apples affected by rotting. The image of an apple moving on the conveyor from 4 different positions can be captured and processed in 0.52 seconds. Accuracy values between 73% and 96% have been obtained in the experimental studies. It is seen that there is more limited work in the classification and separation of smaller sized products based on image processing.

Reference [3] in their studies, achieved a maximum accuracy of 93% in the experiments they carried out in an image processing-based study they set up with a camera and conveyor belt system to classify quality wheat grains.

Reference [4] conducted a study based on both image processing and CNN architectures for the detection of apple defects. They have developed a four-line fruit sorting machine that has the potential to process 5 fruits per second. It was stated that the results obtained in the experiments they performed using CNN gave 10% better results than the results obtained with the image processing technique.

Reference [5] with the shrimp classification machine based on the LeNet-5 architecture, one of the CNN architectures, was able to classify 9 different shrimp classes with 96% success in 0.47 hour period.

Reference [6] their work is on the use of CNN architectures in the livestock industry. Accordingly, using the data set consisting of 14,728 images, the aim of the training with the Fusion Single Shot MultiBox Detector (IFSSD) (the architecture they use for the development of the SSD architecture) is based on the instant detection of sick chickens for poultry breeding. As a result of the experiments performed with IFSSD, they obtained a 99.7% mAP score for the IoU> 0.5 threshold value. The results show that sick chickens in a flock can be successfully detected automatically and the method has the potential to facilitate efficient flock management.

Reference [7] developed an image processing-based system for classification of white mushrooms with the help of OpenCv library. With the camera and conveyor belt

system, they were able to classify 102 mushrooms per minute with 97% accuracy according to their size. When compared with the manual classification results, they achieved an improvement of 38% in classification speed and 6.8% in accuracy.

Reference [8] with the model based on the VGG-16 architecture, they developed a system that includes a conveyor mechanism that allows unwashed eggs to be divided into three different classes as solid, cracked and bloody. The number of samples was increased with the data augmentation and as a result, a classification success of 94% was achieved.

Reference [9] conducted a study on the classification of relatively smaller objects. They developed a CNN-based model that allows the hickory type walnuts, which cannot be separated easily from the shell and the core, to separate in real time on the conveyor system after they are broken. With the image of 15,000 broken walnuts, the CNN model was trained and separated by the developed model by classifying three class labels as "kernel, shell and unsorted" with an accuracy between 94% and 98%.

Reference [10] aimed to create a united convolutional neural networks (CNNs) architecture based on an integrated method that would enable to distinguish common grape diseases i.e., black rot, esca and isariopsis leaf spot from healthy leaves. The combination of multiple CNNs enables the proposed UnitedModel to extract complementary discriminative features. Thus the representative ability of UnitedModel has been enhanced. The UnitedModel has been evaluated on the hold-out PlantVillage dataset and has been compared with several state-of-the-art CNN models. The experimental results have shown that UnitedModel achieves the best performance on various evaluation metrics. The UnitedModel achieves an average validation accuracy of 99.17% and a test accuracy of 98.57%, which can serve as a decision support tool to help farmers identify grape diseases.

Reference [11] made classification with a CNN model was created with the VGG-16 structure in order to overcome the difficulties encountered in traditional methods in distinguishing healty dates from defective ones and estimating their maturity levels. The model makes a classification of four outputs. Three of these classes represent different maturity levels of healty dates and the fourth represent defective dates. A total of 1357 images were taken by mobile phone in a palm garden in Jahrom (Iran) in October 2018. 899 of these images are healty and 458 are defective dates. Then, the training dataset was augmented using rotation, height shift, width shift, zoom, horizontal-flip, and shear intensity. The data augmentation technique created 37,056 images for the training process. 30,688 images were used for training of the model, 6,368 images for validation, 199 images for testing. The average performance of the model per class varied between 96% and 99%.

II. Convolutional Neural Networks (CNNs)

Object detection is one of the important application areas with CNN. A more complex process takes place in object detection compared to classification problems. In object recognition problems, more than one class can take place on an image. The main purpose is to correctly predict the location and determine the class of the object. In this respect, a more complex process takes place compared to classification problems.

In neural networks, Convolutional neural network is one of the main categories to do images recognition, images classifications. CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat). Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The Fig. 1. is a complete flow of CNN to process an input image and classifies the objects based on values.

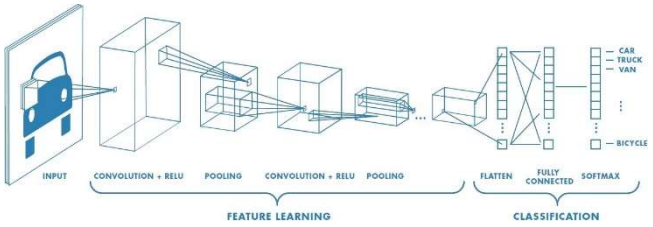


Fig. 1. General structure of CNN architecture

A. Convolution Layer

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

B. Non Linearity (ReLU)

ReLU's purpose is to introduce non-linearity in CNN. The output is $f(x) = \max(0,x)$. There are other non linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.

C. Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

D. Fully Connected Layer (FC)

The output of convolution/pooling is flattened into a single vector of values, each representing a probability that a certain feature belongs to a label. The objective of a fully connected layer is to take the results of the convolution/pooling process and use them to classify the image into a label. An activation function such as softmax or sigmoid is used to classify the outputs as cat, dog, car, truck etc [12].

III. SINGLE SHOT MULTIPLE DETECTION

SSD architecture does not consist of two different stages that first determine the places of objects in the images and then the types of these objects, like classical structures, and does them all at once. Thus, by decreasing the calculation load, the hardware requirements are reduced and faster results are obtained, especially in real-time object detection [13]. For this reason, the data set consisting of worm eggs was trained using SSD architecture in this study.

When examining SSD architecture, it is necessary to talk about VGG architecture first. Because SSD architecture is based on VGG architecture that provides a lower feature map. The backbone of the SSD architecture was created with VGG16 in order to benefit from the superior performance of VGG16 in image classification. In this way, transfer learning is performed by using the features learned by VGG16 in SSD. In the backbone of the SSD architecture, VGG16 layers are used almost excluding the last fully connected layer. Differently, the first two fully connected layers are transformed into the conv6 and conv7 layers as shown in Fig. 1 and 2.

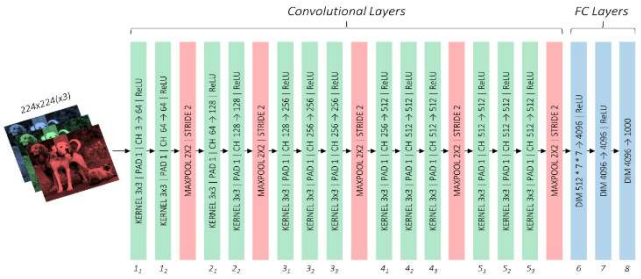


Fig. 2. Original structure of VGG16

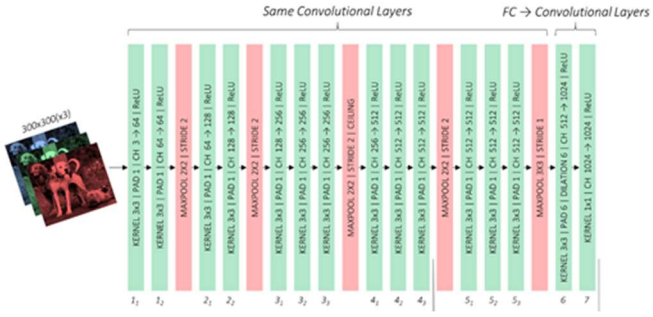


Fig. 3. Converted structure of VGG16 to SSD

Convolution layers other than VGG16 in SSD architecture are expressed as extra feature layers [13] and shown in Fig. 4.

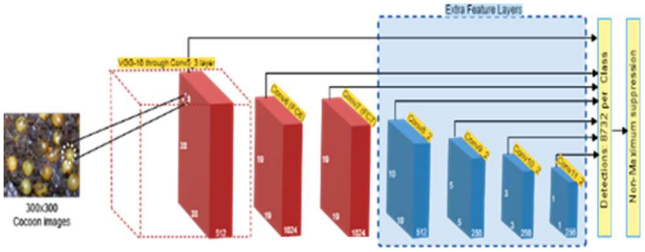


Fig. 4. General structure of SSD architecture

IV. MATERIAL AND METHOD

In this study SSD was used to detect worm eggs in worm castings. This network is preferred because it is fast enough for real-time object detection.

A. Data set

The compost and eggs whose images were used in the data set were obtained from a local vermicompost production facility [14]. The dataset created in this study consists of three folders named *JPEGImages, Annotations and ImageSets.* In the JPEGImages folder there are 1000 compost images with an average of four and 3809 eggs in total. In the literature search, the worm egg data set was not reached within our knowledge. Images obtained at high resolution (4000 x 3000 pixels) using the Panasonic DMC-FZ100 digital camera were resized in RGB format and 800 x 600 pixels in order to reduce the computational load and increase image processing efficiency. Sample images from data set are given in Fig. 5.

Fig. 5. Sample images from data set

B. Data Labeling

The coordinates of the eggs in the images were labeled one by one with the *LabelImage* program and saved in the folder *Annotations* in XML format. A sample image with five eggs labeled in *LabelImage* program is given in Fig. 6.

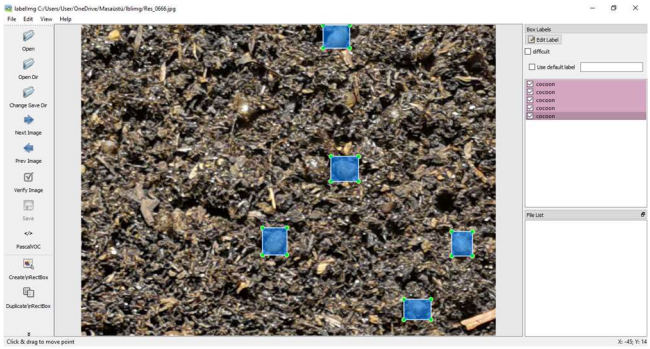


Fig. 6. Cocoon labelling

In the *ImageSets* folder, there are files containing image information used during training, validation and test. These files are text files that contain information on which image is an object and which image is not an object. In Table I, after the data set is divided into two groups for training and test, the number of images in each group and the number of eggs in the images are shown.

TABLE I IMAGE AND EGG NUMBERS IN DATA SET

Data Set	Number of Images	Number of Eggs in Images
TRAIN	800	2854
TEST	200	955
TOTAL	1000	3809

C. *Data Augmentation*

The process of increasing data is the process of obtaining new training examples by applying different techniques (such as rotating, cropping, zooming in and out) to the images in the data set [15]. In this study data augmentation was done only on training data with following principles:

- Brightness, contrast, saturation and hue adjustments were adjusted by 50% randomly.
- Zoom out is done on the image with a probability of 50% and up to 4 times. This process helps small objects to be detected and learned.
- Crop was done in a ratio of 0.5 to 2 and by random cropping. This process helps to detect and learn large objects.
- Rotation was done at a rate of 50% and randomly.
- As required by SD300 architecture, all images are resized to 300x300 pixels.
- All boxes have been resized proportionally according to the transformations made.
- In the last stage, all images were normalized according to the mean and standard deviation values of ImageNet data.

The data is a set of 1000 images and one class named "cocoon". Since there are 3804 cocoon labels in total, cross validation has not been used.

V. RESULTS AND DISCUSSION

In this study, it is aimed to separate Red Californian Worm eggs from compost by using deep learning and image processing techniques in vermicompost production and recycling them into production. For this purpose, the data set was created by labeling 3809 worm eggs in 1000 images obtained from compost prepared under controlled conditions as a single class “cocoon” and made ready for training. With this data set, a model was created that detects worm eggs in images using SSD architecture. Detected eggs on an image is given in Fig. 7.

Fig. 7. Image of detected eggs

The reason why SSD architecture is preferred is because this structure does not consist of two different stages that first determine the places of objects in the images, then the types of these objects, and does them all at once, thus reducing the computational load and reducing the hardware requirements and obtaining faster results, especially in real-time object detection.

The model was trained in 1000 epochs with a data set consisting of 1000 different images. The trained model was tested on 200 different images and average precision rates were determined. Classification performances were compared by making trials on different levels of threshold values. Average precision graph of 0.5 threshold value is given in Fig. 8.

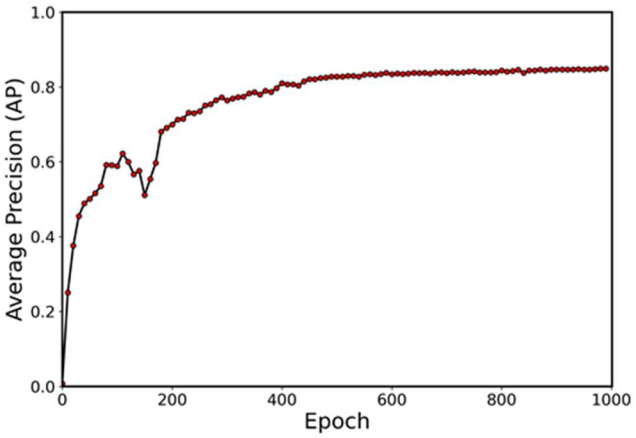


Fig. 8. Average precision (AP) graph

AP value increased up to 0.8 at the 500th epoch, after that it became convergent and reached 84.9% in the 1000th epoch. In the calculation of the total loss, for localization loss Categorical Cross-Entropy and for condidence loss Smooth L1 Loss functions are used. Learning rate (α) is taken as 1e-5. The graph of the total loss obtained at the end of the study is given in Fig. 9.

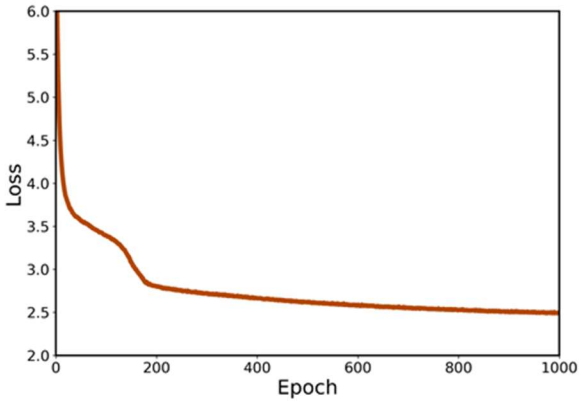


Fig. 9. Total Loss graph

When the total loss is examined, it is seen that the loss value of the model decreases to 2.5 at the 800th apoch, and it becomes convergent from the 1000th epoch. The lower this value means, the more successful the training is. Since there was no significant decrease in the total loss after the 1000th epoch, the training was completed at this level. Total training time lasted 20 hours.

VI. CONCLUSION

In this study, it is aimed to separate Red California Worm eggs from compost by using deep learning and image processing techniques to increase vermicompost production. For this purpose, SSD was used because it obtains faster results especially in real time object detection. Network was trained in 1000 epochs by augmenting the data with 1000 images. As a result of the training, it was determined that the loss decreased to 2.527 in the 800th epoch and to 2.494 in the 1000th epoch, and after that it made a horizontal course with converging values. The AP of the model increased up to 84.9% in the 1000th epoch. In the light of these results, the model consisting of a single class with relatively small dimensions can be considered quite successful.

The current data set consists of images of worm eggs in compost obtained from only one source. Although worm eggs are similar to one species as the Red Californian worm, composts of different sources come in different colors and textures. For this reason, it would be more appropriate to use composts obtained from more than one source and data sets created with larger sizes.

REFERENCES

[1] C. Saday, "Vermikültür Üretimi, Yaşanılan Yasal Zorluklar ve Çözüm Yolları ile Üretim Süreçleri ve Gelişimi Konusundaki Deneyimlerin Aktarılması," in *Tema Vakfı Ulusal Vermikültür Çalıştayı*, Ankara, 2013, pp. 20-36: TEMA.

[2] M. M. Sofu, O. Er, M. C. Kayacan, and B. Cetişli, "Design of an automatic apple sorting system using machine vision," *Computers and Electronics in Agriculture,* vol. 127, pp. 395-405, 2016/09/01/ 2016.

[3] E. Kaya and İ. Saritas, "Towards a real-time sorting system: Identification of vitreous durum wheat kernels using ANN based on their morphological, colour, wavelet and gaborlet features," *Computers and Electronics in Agriculture,* vol. 166, p. 105016, 2019/11/01/ 2019.

[4] S. Fan *et al.*, "On line detection of defective apples using computer vision system combined with deep learning methods," *Journal of Food Engineering,* vol. 286, p. 110102, 2020/12/01/ 2020.

[5] Z. Liu, X. Jia, and X. Xu, "Study of shrimp recognition methods using smart networks," *Computers and Electronics in Agriculture,* vol. 165, p. 104926, 2019/10/01/ 2019.

[6] X. Zhuang and T. Zhang, "Detection of sick broilers by digital image processing and deep learning," *Biosystems Engineering,* vol. 179, pp. 106-116, 2019/03/01/ 2019.

[7] F. Wang, J. Zheng, X. Tian, J. Wang, L. Niu, and W. Feng, "An automatic sorting system for fresh white button mushrooms based on image processing," *Computers and Electronics in Agriculture,* vol. 151, pp. 416-425, 2018/08/01/ 2018.

[8] A. Nasiri, M. Omid, and A. Taheri-Garavand, "An automatic sorting system for unwashed eggs using deep learning," *Journal of Food Engineering,* vol. 283, p. 110036, 2020/10/01/ 2020.

[9] Z. Wu, K. Luo, C. Cao, G. Liu, E. Wang, and W. Li, "Fast location and classification of small targets using region segmentation and a convolutional neural network," *Computers and Electronics in Agriculture,* vol. 169, p. 105207, 2020/02/01/ 2020.

[10] M. Ji, L. Zhang, and Q. Wu, "Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks," *Information Processing in Agriculture,* 2019/10/23/ 2019.

[11] A. Nasiri, A. Taheri-Garavand, and Y.-D. Zhang, "Image-based deep learning automated sorting of date fruit," *Postharvest Biology and Technology,* vol. 153, pp. 133-141, 2019/07/01/ 2019.

[12] Prabhu. (04.03.2018). *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[13] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 21-37: Springer International Publishing.

[14] A. ÇELİK. (2014). Available: http://www.verminat.com/

[15] A. Rosebrock. (08.07.2019, 06.06.2020). *Keras ImageDataGenerator and Data Augmentation*. Available: https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/

The Influence of Quality and Size of the Training Dataset on the Performance of the Support Vector Machine Algorithm.

1st Evan Kurpiewski
dept. of Computer Science (of Aff.)
University of North Carolina – Wilmington (of Aff.)
Wilmington, United States
eak3772@uncw.edu

2nd Ilya Samokhvalov
dept. of Computer Science (of Aff.)
University of North Carolina - Wilmington (of Aff.)
Wilmington, United States
ins2632@uncw.edu

***Abstract*—Twitter plays a significant role in our lives as a means of expressing our thoughts, fears, joy, happiness, and other emotions, which collectively can be described as a positive, negative, and neutral sentiment. Such sentiment analysis is essential for any company, organization, or individual that values public opinion towards its products, services, or overall reputation. With the growing number of tweets published every minute, it becomes more and more tedious to track changes in sentiment using human annotators every minute. The accessibility of modern computing power and advancements in Machine Learning makes it feasible to employ a machine to classify a human's sentiments. We researched the influence of the quality and size of the dataset on the Support Vector Machine algorithm's performance and found evidence that the quality of labeled tweets aimed to train the algorithm is paramount, but the size is secondary.**

***Index Terms*—Twitter sentiment analysis, sentiment classification, public relations management, public sentiment, machine learning.**

I. Introduction

Twitter is a microblogging and social networking service on which users post and interact with messages known as "tweets" [1]. The unique way to cross-intertwine messages, topics, interests, and accounts by special characters such as @ and #, made Twitter an irreplaceable instrument in the hands of consumers to attract the desired attention of brands and companies of interest. Adoption of this feature by major social media players, such as Facebook, Instagram, LinkedIn, and others, indirectly confirmed the postulate. The commercial companies, political organizations, and others who wanted to track the public's sentiment began to monitor its customer's satisfaction and audience sentiments by monitoring tweets that contain the keywords of their interest. With the exponential growth of the number of tweets published every minute, every hour, every day, came a necessity of automating the sentiment classification of tweets to make use of changes in moods and consumer preferences. This is where Artificial Intelligence and Machine Learning techniques become indispensable helpers of companies, organizations, and individuals.

Charlotte Teresa Weber and Shaheen Syed [2] studied the public sentiment towards interdisciplinary studies in the academic domain. In their paper "Interdisciplinary optimism? Sentiment analysis of Twitter data," the authors discussed the importance of Twitter as a useful instrument to measure public sentiment. The authors justified their choice of Support Vector Machine (SVM) algorithm by the superior performance of one. They trained seven different algorithms such as SVM, Logistic Regression, Multinomial Naive Bayes, Bernoulli Naive Bayes, Decision Trees, Adaptive Boosting (aka AdaBoost), Multi-Layer Perception algorithms, and compared the F1 metric. The reported F1 values are as following: SVM's F1 score is 0.67, Logistic Regression's F1 score is 0.66, Multinomial NB's F1 score is 0.65, Bernoulli NB's F1 score is 0.64, Decision Trees' F1 score is 0.56, Adaptive Boosting's F1 score is 0.63, Multi-Layer Perception's F1 score is 0.60. Hence, the researchers simply chose the highest score that was achieved by the SVM algorithm.

The statistical test concerning a single mean, Student's t-test, showed us that only Decision Trees' F1 score (0.56) is statistically different from the mean value of the rest of the sample—t-value is -8.056, which is greater than the left rejection point of 4.032 for the level of significance of 1% (two-tailed test, with the degree of freedom equals 5). Hence, we don't see enough evidence to support the theory that the SVM algorithm outperformed the other algorithms, meaning that the choice is unjustified if, and only if, the decision was made based on relative performance, judged by F1 scores. Zainuddin, Selma, and Ibrahim [10] ran the same statistical test to determine whether the accuracy results differ from each other.

Though we don't argue that the SVM algorithm was the right choice to use given the type of classification problems, we argue about the justification and approach chosen to select the SVM algorithm. Our research aims to demonstrate that the size and the quality of the dataset a researcher uses to train an algorithm are essential and substantially influence its performance. We collected three distinct and popular hand-labeled datasets, which were also used by Ms. Weber and Mr. Syed, and trained a single algorithm— SVM — on these datasets. In addition to the comparative evaluation of the algorithm's performance, we intend to use the SVM algorithm, trained separately on three distinct datasets, to label our target

dataset, which we collected directly from Twitter Application Programming Interface (API) through Python library Tweepy. The target dataset contains tweets with the following keywords: "UNCW," "#UNCW," "UNCW." We supposed that we will get inconsistent and different results, which in turn should demonstrate the influence of training sets on classification performance.

II. Related Work

The overall difficulty of the sentiment classification problems is aggravated by specifics of the Twitter user's manner of speech, the prevalence of sarcasm, and other specifics of sentiment expression. On the other hand, the messages' length limitation forces the public to invent and use shortcuts such as emojis, emoticons, and other means of graphical and textual sentiment expression. This in turn, allows the development of a lexicon, a collection of symbols that define certain sentiments by public perception. Thereby, practitioners and scientists prefer to use two major approaches to classify tweets' sentiment—lexicon-based, supervised-learning, and a third hybrid approach, which combines the variation of both approaches. During the research of related work, we found that most studies focus on a comparative evaluation of different algorithms but used the same datasets available online. We followed the steps of Ms. Weber and Mr. Syed, but investigated the datasets' influence on the single algorithm—SVM.

Rout et al. [3] investigated different machine learning algorithms' relative performance concerning the classification of tweets acquired from the Twitter public domain. The Multinomial Naive Bayes (MNB), Maximum Entropy and Support Vector Machine algorithms were trained based on the unsupervised approach and lexicon-based approach. The authors reported an accuracy of 0.807 using the unsupervised approach and 0.752 using the lexicon-based approach, while the MNB algorithm achieved an accuracy of 0.67 using the unigram feature. Saif et al. [4] introduced a novel lexicon-based approach called SentiCircles. The method is built on a dynamic representation of words that are supposed to capture contextual semantics. The three datasets that authors used to measure the accuracy of the approach are all relatively small. The Obama-McCain Debate (OMD) dataset was collected during the first presidential TV debates between presidential candidates Obama and McCain in 2008 and eventually provided 393 positive and 688 negative tweets. The Health-Care Reform dataset represents a collection of tweets containing hashtag #hcr. It was collected in 2010 and included 397 positive and 957 negative tweets. The third dataset was built by researchers from the Stanford Twitter Sentiment Corpus, which was initially automatically labeled based on emoticons regardless of the tweet's remaining content. It contained 632 positive and 1402 negative tweets. Ren et al. [5] discussed the topic-enhanced word embedding for Twitter sentiment classification problems. In the paper, they mention Pang et al. [6] pioneering the bag-of-word approach for text classification analysis. The authors looked at the classification problem from an overall sentiment perspective rather than from the perspective of topic-based classification. The authors tested three algorithms overall – Naive Bayes, Maximum Entropy, and SVM. The accuracy of all three of them ranged from 0.73 to 0.83, with a mean of 0.77 and a standard deviation of 0.0253. Ren et al. [5], in turn, claimed that the experimental results showed the topic-enhanced word embedding model outperformed other word-representation models. The authors used the SemEval 2014 Dataset during their experiments.

Ryan Ong [7] described his challenge identifying and categorizing offensive language within Twitter. The author experimented with variations of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) layers. The researcher concluded that the ordering of layers is extremely important for optimal model architecture. Appel et al. [8] studied the comparative performance of the hybrid approach to sentiment analysis against Naive Bayes and Maximum Entropy algorithms trained on the same datasets—movie reviews, and sentiments Twitter datasets. The hybrid approach combines a Sentiment Lexicon, Semantic Rules, Negation Handling, Ambiguity Management, and Linguistic Variables. The authors reported the accuracy of 0.76 for the hybrid approach against 0.62 for the alternative method. Ghosh and Sanyal [9] investigated the comparative performance of the Machine Learning classification algorithms concerning the use of feature selection methods such as Information Gain (IG) Ratio, Chi- square, and Gini-index against unigram and bigram feature sets. The investigators used Recall, Precision, and F1 score measures to compare the performance of the algorithm. They reported the accuracy in the limits of 0.87

Zainuddin et al. [10] proposed their version of the hybrid approach to the sentiment classification problem. The researchers compared the performance of the Principal Component Analysis (PCA), Latent Semantic Analysis (LSA), and Random Projection (RP) feature selection methods. They used three different datasets for the analysis, such as the Hate Crime Twitter Sentiment dataset, which consists of 1078 tweets in total, Stanford Twitter Sentiment dataset, which consists of 353 tweets, and Sanders Twitter Corpus dataset, that consists of 1091 tweets. Additionally, the researchers conducted t-tests to figure out whether the difference in results is statistically different from each other. They concluded that the hybrid sentiment classification, which combines the Association Rule Mining with a heuristics combination in part-of-speech patterns, provides statistically meaningful results at a p-value of 5%. However, the highest reported accuracy was reached by the SVM algorithm and was 0.76. The reported performance of the Extreme Learning Machine of Twitter aspect-based sentiment classification showed quite inconsistent results between average training accuracy of 0.84 and average testing accuracy of 0.58, which indicates that the model was overfitted. Stojanovski et al. [11] went with an overcomplicated approach and used a CNN for extracting features for further classification experiments on Twitter data. They used three labeled datasets provided by the SemEval challenge. The accuracy for all the sets for 2013, 2014, and 2015 years did not exceed 0.71, with significantly lower

results for the emotion identification classification, with which its accuracy didn't exceed 0.52. Kim et al. [12] provided a straightforward, self-explanatory application of AI algorithms in sentiment analysis of messages on the Internet. As an experiment, they crawled comments that were posted in online communities concerning cryptocurrencies, such as Bitcoin, Ripple, and Ethereum. The researchers claimed that they achieved an average accuracy of around 0.74 in determining the sentiment.

Abassi et al. [13] addressed the important topic of sentiment analysis techniques that facilitate the detection of hate-speech postings and other extremist publications on the Internet. The researchers used three hand-labeled datasets to train the model. One dataset collected the movie reviews, which Pang et al. [6], and later in 2017, Appel et al. [8] also used for their hybrid classification approach. The other two datasets were directly connected to the domain of interest, a US supremacist forum and a Middle Eastern extremist group forum, thus combining English and Arabic content. The authors developed the Entropy Weighted Genetic Algorithm for feature selection and used the SVM algorithm for the classification problem. They claimed the overall accuracy of over 0.91 on the benchmark datasets, but, unfortunately, didn't provide out of sample performance metrics.

III. Dataset

We used three out of the seven datasets that we downloaded along with Mr. Syed's original code from his GitHub repository [14] that is freely accessible online. The first dataset we used is called the Sanders Dataset and was initially available from http://www.sananalytics.com/lab/, but as of December 01, 2020 the link is inaccessible. Zainuddin et al. [10] used the Sanders Dataset to research the performance of the PCA, LSA, and RP feature selection methods for sentiment classification problems. The dataset was collected in October 2011 and initially contained 5,513 tweet IDs related to the topics Apple (Apple), Google (#Google), Microsoft (#Microsoft), and Twitter (#Twitter). We excluded tweets that had "irrelevant" labels, as it refers to non-English tweets and tweet IDs that referred to deleted tweets and accounts. Hence, the actual size of the training dataset was significantly smaller and contained only 529 hand-labeled tweets.

The Semantic Analysis in Twitter Task 2016 dataset [15], also known as SemEval-2016 Task 4, was created for various sentiment classification tasks in 2015. SemEval is an ongoing series of Natural Language Processing (NLP) research workshops whose mission is to advance the current state of the art in semantic analysis and help create high-quality annotated datasets in a range of increasingly challenging problems in natural language semantics [16]. Stojanovski et al. [11] used SemEval Dataset to research CNN extracting features for classification problems. The dataset contained 10,000 tweet IDs, but some corresponding tweets and accounts had been deleted over time. Eventually, SemEval Dataset included 6,771 tweets.

The third training dataset we used was the CLARIN 13 Dataset [17]. It contained a total of 1.6 million labeled tweets from 15 European languages. Tweets were collected in 2013 and represent a small fraction of all publicly available tweets at the time. We used an English subset that originally contained around 90,000 tweet IDs, but eventually, we collected only 32,449 real tweets from Twitter's API.

We used Twitter's API and the Python library Tweepy [18] to collect the tweets that contain the keywords "UNCW," "#UNCW," and "UNCW." We collected tweets from October 17, 2020, till December 01, 2020, and ended up with a total of 4,236 tweets and 1,724 unique tweets. The UNCW dataset is our target dataset, which we use in the comparative evaluation of the SVM algorithm, trained on three above mentioned, publicly available datasets.

IV. Methodology and Experiments

In our research, we followed the steps of Charlotte Teresa Weber and Shaheen Syed, which they set in their paper "Interdisciplinary optimism? Sentiment analysis of Twitter data." We used Mr. Syed's original code from his GitHub repository [14] but with substantial changes and upgrades to meet our stated goal.

To preserve unbiasedness, we didn't do any differential feature engineering techniques besides those described by Ms. Weber and Mr. Syed. We followed the same approach and steps in training the SVM algorithm for each of the training sets. The original code includes eight distinct modules (steps) that can be run consecutively or separately. The code heavily relies on using the mongoDB document database [19] and scikit-learn library [20], a collection of efficient tools for predictive data analysis which is built on NumPY, SciPy, and matplotlib libraries. We significantly upgraded the sixth step by adding calls from the scikit-learn library related to the confusion matrix builder to measure the SVM algorithm's accuracy on different training datasets. We use two-tailed hypothesis testing and Student's t-Distribution statistics to look for evidence to support or reject the hypothesis of significant differences in the accuracy metrics of the SVM algorithm that we collect.

V. Results

For the sake of simplicity, we denote SVM_{Clarin}, $SVM_{Sanders}$, and $SVM_{SemEval}$ as Support Vector Machine algorithm, trained on three distinct datasets, mentioned above—Clarin, Sanders, and SemEval; $F1_{Clarin}$ as an F1 score of the SVM_{Clarin}, $F1_{Sanders}$ as an F1 score of the $SVM_{Sanders}$, $F1_{SemEval}$ as an F1 score of the $SVM_{SemEval}$.

The results of our experiments (Table I) show depressing and inconsistent performance of the SVM algorithm. We conducted the Student's t-test concerning a single mean to test whether the difference between each paired observation is significant. For example, to test whether $F1_{SemEval}$ (0.87) is different from $F1_{Sanders}$ (0.74) and $F1_{Clarin}$ (0.78), we define $F1_{Sanders}$ and $F1_{Clarin}$ as a sample with the mean value of 0.76, the standard deviation of 0.028, and the standard error of 0.02.

The rejection points for a two-tailed test with even a 10% level of significance and degree of freedom equals one lie below -6.314 and above 6.314. Given the t-value of -5.5, we fail to reject the hypothesis that $F1_{SemEval}$ (0.87) is statistically different from $F1_{Sanders}$ (0.74) and $F1_{Clarin}$ (0.78). We conducted the analogous tests for each paired observation and conclude that we don't have enough evidence to reject the hypothesis of equal performance of the SVM algorithm trained on three distinct datasets.

TABLE I. PERFORMANCE OF THE SVM ALGORITHM.

	Training Dataset	**Target Dataset Labels**		
	F1 Score	***Positive Labels***	***Neutral Labels***	***Negative Labels***
SVM_{Clarin}	0.78	338	1282	104
$SVM_{Sanders}$	0.74	115	1393	216
$SVM_{SemEval}$	0.87	1203	423	98
Consistency		62	311	26

We accept the hypothesis of equal performance based on the mentioned metrics. However, $SVM_{SemEval}$ was able to identify 1203 positive, 423 neutral, and 98 negative tweets of our target UNCW dataset, while SVM_{Clarin} and $SVM_{Sanders}$ demonstrated quite different results. SVM_{Clarin} identified 338 positive, 1282 neutral, and 104 negative tweets, and $SVM_{Sanders}$ identified 115 positive, 1393 neutral, and 216 negative tweets. Overall, the consistency [the number of tweets identified alike by all three algorithms] of such identification was 62 positive, 311 neutral, and 26 negative tweets. Given only this discrepancy in classification by the same algorithm, it becomes obvious, that there is substantial deficiency in reporting only F1 and accuracy metrics, hence we visually analyzed the substance of the target dataset and classification results and found some disturbing misclassifications.

The target dataset contained the group of twelve tweets that were almost identical to each other, were made with intervals of 1-2 minutes and went consequentially in the database. This subset contained two types of messages such as "*Dear NAME, today you helped save a life! On behalf of the UNCW Red Cross Club, thank you for your generous blood donation. #giveblood #uncwredcross LINK,*" and "*Dear NAME, thank you for giving blood from all of us at the UNCW Red Cross Club. #giveblood #uncwredcross LINK.*" The messages, apparently bear positive sentiment and one would expect that the computer algorithm won't hesitate to classify the all twelve of them as positive. However, only $SVM_{SemEval}$ classified all of them as positive, while $SVM_{Sanders}$ classified only 6 as positive (True Positive), and SVM_{Clarin} classified 7 as positive (True Positive), giving the neutral label to the rest of the subsample (False Negative).

We admit that the subsample is subject to significant sample bias, but we intend to demonstrate the interdependence of the sample substance and derived metrics. We calculated (Table II) Recall ($SVM_{SemEval}$ = 1, SVM_{Clarin} = 0.583, $SVM_{Sanders}$ = 0.5), Precision (all three metrics equals 1), Accuracy ($SVM_{SemEval}$ = 1, SVM_{Clarin} = 0.583, $SVM_{Sanders}$ = 0.5), and F1 score ($SVM_{SemEval}$ = 1, SVM_{Clarin} = 0.737, $SVM_{Sanders}$ = 0.667) from the confusion matrix for error analysis for the subsample.

TABLE II. ERROR ANALYSIS OF THE TARGET DATASET.

	Assigned Labels		
	SVM_{Clarin}	***$SVM_{Sanders}$***	***$SVM_{SemEval}$***
Subsample Precision	1.000	1.000	1.000
Subsample Recall	0.583	0.500	1.000
Subsample F1	0.737	0.667	1.000
Subsample Accuracy	0.583	0.500	1.000

F1 score of 0.667, for example, demonstrates the performance comparable to the result of a flip of a coin. Hence, we may say that SVM_{Clarin} and $SVM_{Sanders}$ were "most likely" guessing than doing an educated classification task, but showing high F1 scores, comparable to the results the majority of researchers report.

VI. CONCLUSION AND FUTURE WORK

Our research focuses on one important topic - training the Support Vector Machine algorithm so that it would be able to classify a Twitter feed as having a negative, positive, or neutral sentiment. We decided to follow most researchers' steps and use freely available online, hand-labeled datasets to train the algorithm. The central question is whether the quality and size of the training dataset influence the result. If so, how substantial the influence is on the human eye.

We conducted statistical tests concerning a single mean. We discovered that the SVM algorithm's performance that Charlotte Teresa Weber and Shaheen Syed [2] chose in their research wasn't superior (F1 equals 0.67) compared to other algorithms. We followed the authors' steps and trained the SVM algorithm on three distinct datasets available online and used by researchers, including Ms. Weber and Mr. Syed. The classification task of the target dataset by SVM, trained of different sets, revealed substantial inconsistency. The in-depth analysis of classification results showed that the performance metrics' simple reporting sometimes doesn't make any sense and often even misleading.

We want to make the preliminary conclusion that the quality of labeled tweets aimed to train the algorithm is paramount and the size of the dataset is a secondary issue. 32,449 (Clarin dataset) tweets didn't make any difference in the training of the algorithm as 529 can do (Sanders dataset). The in-depth analysis of the classification work made by SVM_{Clarin} and $SVM_{Sanders}$ on the out of sample dataset is no better than simple guessing, while 6,771 tweets could deliver consistent result that you can and should anticipate from the computer algorithm.

The Oxford Dictionary of Phrase and Fable [21] says that the idea (in computing and other spheres) of incorrect or poor-quality input will always produce faulty output is called garbage in, garbage out. The phrase concisely describes the conclusion, which can be made from the results we report even now. However, the science approach recommends finishing the research by properly labeling all 1724 tweets from the target dataset and running the confusion matrix of error analysis to define the overall recall, precision, accuracy, and F1 score. Our primary future task is to increase the classification algorithm's overall efficiency by increasing True Positive and True Negative classifications and decreasing the False Positive and False Negative errors and not by just reaching average accuracy metrics.

REFERENCES

[1] Wikipedia:Twitter
https://en.wikipedia.org/wiki/Twitter

[2] C. Weber, S. Syed. Interdisciplinary optimism? Sentiment analysis of Twitter data. *http://doi.org/10.1098/rsos.190473*. The Royal Society. open sci.6:190473. 2019.

[3] J. Rout, K.Choo,A. Dash, S. Bakshi, S. Jena, K. Williams A model for sentiment and emotion analysis of unstructured social media text. *http://dx.doi.org.liblink.uncw.edu/10.1007/s10660-017-9257-8*. Electronic Commerce Research, 18(1), 181-199. 2018.

[4] H. Saif, Y. He, M. Fernandez, H. Alani Contextual semantics for sentiment analysis of Twitter. Information Processing & Management. *https://doi.org/10.1016/j.ipm.2015.01.005*. Information Processing & Management, 52(1), 5-19. 2016.

[5] Y. Ren, R. Wang, D. Ji A topic-enhanced word embedding for Twitter sentiment classification. *https://doi.org/10.1016/j.ins.2016.06.040*. Infor- mation Sciences, 369, 188-198. 2016.

[6] B. Pang, L. Lee, S. Vaithyanathan Thumbs up? sentiment classification using machine learning techniques. *https://doi.org/10.3115/1118693.1118704*. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 79–86. 2002.

[7] R. Ong Offensive Language Analysis using Deep Learning Architecture. *https://arxiv.org/pdf/1903.05280.pdf*. ArXiv. 2019.

[8] O. Appel, F. Chiclana, J. Carter, H. Fujita Successes and challenges in developing a hybrid approach to sentiment analysis. *http://dx.doi.org.liblink.uncw.edu/10.1007/s10489-017-0966-4*. Applied Intelligence, 48(5), 1176-1188. 2018.

[9] M. Ghosh, G. Sanyal Performance assessment of multiple classifiers based on ensemble feature selection scheme for sentiment analysis. *http://dx.doi.org.liblink.uncw.edu/10.1155/2018/8909357*. Applied Computational Intelligence and Soft Computing, 12. 2018.

[10] N. Zainuddin, A. Selamat, R. Ibrahim Hybrid sentiment classification on twitter aspect-based sentiment analysis. *http://dx.doi.org.liblink.uncw.edu/10.1007/s10489-017-1098-6*. Applied Intelligence, 48(5), 1218-1232. 2018.

[11] D. Stojanovski, G. Strezoski, G. Madjarov, I. Dimitrovski, I. Chorbev Deep neural network architecture for sentiment analysis and emotion identification of Twitter messages. *https://doi.org/10.1007/s11042-018- 6168-1*. Multimed Tools Appl77, 32213–32242. 2018.

[12] Y. Kim, J. Kim, W. Kim, J. Im, T. Kim, S. Kang, C. Kim Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. *https://doi.org/10.1371/journal.pone.0161197*. PLoS ONE 11(8): e0161197. 2016.

[13] A. Abbasi, H. Chen, A. Salem Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *http://doi.acm.org/10.1145/1361684.1361685* . Inform. Syst. 26, 3. 2008.

[14] Github Repository: Shaheen Syed
https://github.com/shaheen-syed/Twitter-Sentiment-Analysis

[15] SemEval Competition
https://alt.qcri.org/semeval2016/task4/

[16] SemEval Github
https://semeval.github.io

[17] M.Igor, G. Miha, S. Jasmina Twitter sentiment for 15 European languages, Slovenian language resource repository. *http://hdl.handle.net/11356/1054.*. CLARIN.SI. 2016.

[18] Tweepy
http://www.tweepy.org/

[19] MongoDB
https://www.mongodb.com

[20] SciKit Learn
https://scikit-learn.org/stable/

[21] Oxford Reference
https://www.oxfordreference.com/view/10.1093/acref/9780198609810.001.0001/acref-9780198609810-e-2835

Evaluating the Performance of Agents for Deceptive Levels

Noor Us Sabah
Riphah International University
Islamabad, Pakistan
noorsaba560@yahoo.com

Adeel Zafar
Riphah International University
Islamabad, Pakistan
adeel.zafar@riphah.edu.pk

***Abstract*—Deceptive levels are levels where agents are trapped in the game and the policy leads them away from the global optimum policy. In this paper, we examine the problem of evaluating deceptive levels. In our previous work, we generated deceptive levels and evaluated them using different agents. The interesting insight from that study was that there was no single agent being able to solve all types of deceptions. In order to further extend our work, we investigate the effect of different metrics on the evaluation of deceptive levels. In our current work, we use different metrics including reward penalization, discounted reward, noisy world, broken world, and broken forward model. Initially, we execute these agents by utilizing deceptive as well as non-deceptive levels to perform a comparison on the basis of total points and win-rate. After comparison, ranks are assigned to the agents. Contrary to our previous evaluation, these metrics give different results and rankings. The experiment implies that each agent has a different set of characteristics and weaknesses.**

***Index Terms*—Procedural Content Generation, General Video Game Artificial Intelligence, General Video Game Level Generation, Video Game Description Language**

I. Introduction

Procedural Content Generation (PCG) is an algorithmic approach to generate content for video games [1]. Different methods are used for content generation in games such as search-based [1], constructive [2] and machine learning-based methods [3].

The evaluation of general video game levels is considered a challenging task due to the diverse nature of games. Various evaluation approaches are already employed by academicians including shallow decision-making analysis [4], ranking approach [5], relative algorithm performance profiles (RAPP) [6]. In our previous work [7], we generated different deceptive levels and evaluated them using different agents. An interesting insight from that study was the limitation of each agent as no agent was able to solve all deceptions. In this paper, we extend our work by evaluating deceptive levels on different set of metrics. We use a rank-based approach, where we assign a rank to the game-playing agents. In addition, five metrics are also used for the ranking of agents including reward penalization, discounted reward noisy world, broken world, and broken forward model.

$$\textit{Formula 1 Scoring System} = \{25, 18, 15, 12, 10, 8, 6, 4, 2, 1\} \tag{1}$$

Equation (1) is used to assign rank to the game-playing agents. The formula assigns 25 points to the best controller and 1 point to the last. Our experiment results show that OLETS is the best-performing agent contrary to our previous experiment where Number27 was the best performing agent. In addition, this approach will help us to create better or optimal agent for deceptive levels.

The remainder of the paper is organized as follows. In section 2, we present related work and background knowledge. Section 3 presents the details of our proposed approach. Details of the experiment are discussed in section 4. Section 5 presents the discussion on the evaluation of agents. Finally, the paper is concluded in section 6.

II. Related Work and Background Knowledge

A. Level Evaluation

Level Evaluation can be performed using multiple approaches including computational metrics, user-based evaluation, and agent-based evaluation. In the next sub-sections, we cover the background knowledge on each one of them.

1) Computational Metrics: Computational metrics measure emergent properties of levels, rather than simply using the same parameters. In [8], authors demonstrated evaluation as a problem in PCG. A comparative evaluation technique was applied for the game of Super Mario Bros.

2) User Based Evaluation: User-based evaluation allows the user to classify the content [9]. Authors proposed a way to solve the problem of automatically generated game levels. Different techniques of modeling and evolutionary algorithms were also introduced for the construction of tracks. Similar user-based studies were also performed by authors in [18], [19] and [20] .

3) Rank-Based Evaluation: Rank-based evaluation allows the user to rank the generated content. In [5], authors presented a research study to test the performance of various game-playing agents. Results indicate that the performances of some controllers were better in almost all conditions while the performances of other controllers were worst.

4) Agent-Based Evaluation: Agent-based evaluation is an evaluation-based approach of game-playing agents, where we test different controllers on game levels without human intervention. In [4], a method to measure the performance of agents was proposed. Results indicate that these methods give them a better understanding of the decision-making process of the agents. In [6], authors provided a technique to rank the content according to win percentage and mean score obtained from various agents.

B. VGDL, GVG-AI, and GVG-LG Framework

Video Game Description Language (VGDL) [10] is a simple, expressive, and human-readable language. Authors highlighted that this language is needed to understand the game logic and it also requires computational agents. The general video artificial intelligence (GVG-AI) framework [11] is proposed to test different artificial intelligent agents. In [12], authors proposed a GVG-LG framework as a benchmark for general video game level generation. It allows competitors to create generators that can generate levels for different games.

C. Deceptive Level Generator

The concept for deceptive games has been introduced by [13]. In [7], authors demonstrate that these games were designed in a way where controllers get rewards that can lead them away from the global optimum policy. Zafar et al. [7] proposed a deceptive level generator. The level generator generates three types of traps including greedy traps, smoothness traps, and generality traps. The results of this study showed that some agents perform poorly on deceptive levels and no single algorithm was able to solve all the deceptions. Fig. 1 shows the deceptive levels for the game of Zelda. Fig. 2 shows the deceptive levels for the game of Solar fox.

Fig. 1. Deceptive levels for game Zelda (The authors have taken copyright permission from (Zafar et al. 2018) to use the figure)

Fig. 2. Deceptive levels for game Solarfox (The authors have taken copyright permission from (Zafar et al. 2018) to use the figure)

III. Method

This section presents the proposed methodology used to evaluate the deceptive levels using a variety of agents. Our proposed approach is a rank-based approach, where we assign ranks to the game playing agents on the basis of total points and win-rate. The flow chart of our proposed evaluation methodology is illustrated in Fig. 3. In total, there are 1250 deceptive and non-deceptive levels executed by all the agents. The deceptive levels used are generated by [7] in the GVG-LG framework and non-deceptive levels are selected from the GVG-AI framework. Five metrics are also used including reward penalization, discounted reward, noisy world, broken world, and broken forward model. By looking at the nature of the algorithm used by each agent, we have embedded these metrics in the GVG-AI framework. After the execution of the agents, we computed results such as total points, win rate and mean score. Then finally, equation (1), is applied that depicts the performance evaluation criteria. The controllers are sorted based on their rankings. To decide the best controller all points of five games are added. After adding up the points, the agent with the highest sum among all other agents is considered as the winner. When the points in the game are considered as a draw, the first position ties the ranking and further proceeds to the second and third positions. Afterwords on the basis of total points and win rate, we decide the best-game playing and worst-game playing agents.

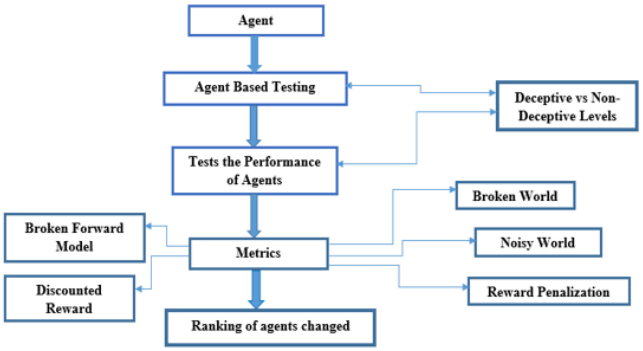


Fig. 3. Flow chart of our proposed evaluation methodology

A. Selection of Games

A brief description of the games [7] used in our research are as follows:

- **Zelda:** The goal of the player in this game is to find the key and escape. The player is also equipped with the sword to kill enemies.
- **Catapults:** The goal of this game is to reach the exit door. The player can not jump on the water and can use catapults to reach it.
- **Cakybaky:** The primary goal for such a game is to bake the cake. In order to bake the cake, the player has to collect some ingredients.
- **Digdug:** The purpose of this game is to gather all gold coins and gems in the cave, exploring the way through it.
- **Solar fox:** The player's objective in this game is to collect diamonds and avoid the attacks of enemies.

B. Selection of Controllers

A brief description of the controllers used in our research are as follows:

- **Number27:** It is an agent with such a population size which involves individuals with a set sequence of actions [14].
- **Thorbjrn:** This agent attempts 49 actions and optimizes various game-playing algorithms and output differences. This agent uses evolutionary and MCTS algorithms [15].
- **OLETS:** OLETS is the version of MCTS, where agent's states are never correlated with the node of the network [5].
- **SampleRS:** The sampleRS controller generates the individuals randomly instead of evolving them. It has length of 10 and it can produce as many sequences in the given time duration [16].
- **SampleOLMCTS:** It is a very common tree search technique which iteratively creates an asymmetric tree in memory to estimate the value of the various actions available from a given state [5].
- **NovTea:** NovTea implements tree-search algorithms and returns the actions [15].
- **CatLinux:** It is an agent that implements Rolling Horizon Evolutionary Algorithms (RHEA). The RHEA algorithms evolves a population of sequences of actions, and the length of each sequence is fixed which is equal to the fixed depth of simulation [6].
- **SampleMCTS:** The sampleMCTS is an MCTS implementation with frequently random roll-outs up to a maximum depth of 10. Once the agent is initialized, the tree policy should be defined, while the heuristic used to determine game states is a combination of the score plus an eventual bonus [5].
- **SampleOneStep:** It rolls the forward model from available actions and selects the best value action. It also determines the score maximization function that minimizes the NPCS distance [4].
- **DoNothing:** It is the basic controller and it always returns *ACTION_ NIL*, so it does not performs any action during the game [17].

IV. Results

This section presents the details of the experiment carried out to evaluate the performance of different game-playing agents. As discussed earlier, five metrics are used to rank the agents. Later, we will discuss the results of the experimentation of each metric.

A. Evaluation of Deceptive vs Non-Deceptive levels

We executed top ten agents including Number27, OLETS, thorbjrn, OLMCTS, sampleRS, NovTea, CatLinux, sampleMCTS, SampleOneStep and doNothing. These agents are selected from the GVG-AI competitions. The performance of the agent is dependent on win-rate and total points. In Table 1, G-0 represents the zelda game, G-1 represents the catapults game, G-2 represents the cakybaky game, G-3 represents the dig dug game and G-4 represents the solarfox game (the same abbreviation is used for all other tables). In the case of deceptive levels, Number27 is the best-performing agent with respect to total points and win-rate, while doNothing is the least-performing agent with respect to total points and win-rate. All agents successfully played the levels of Zelda game, whereas there is no agent which is capable of playing the levels of Solarfox game. Table 1 illustrates the ranking of agents by using deceptive levels. In the case of non-deceptive levels, Number27 is the agent with better performance with respect to total points and win-rate, while sampleOneStep is the agent with worst performance with respect to total points and win-rate. In this scenario, almost all agents are capable of playing the Zelda game levels, while there is no agent that is capable of playing the Solarfox game levels. Table 2 illustrates the ranking of agents by using non-deceptive levels.

B. Reward Penalization

In this metric, we change the settings of the GVG-AI framework in such a way to reduce the score **s** in the game by one point when the action is performed by the agent [5]. So, that better-performing controllers would get reward in the rank. The formula for the calculation is: **s= s-actions**. We compared the rankings of agents with deceptive as well as non-deceptive levels. Table 3 depicts the ranking for this metric. In this setting, the ranks are assigned to the agents according to total points but the ranking is also dependent on the win-rate. All agents performed well in the reward penalization settings except Number27 and sampleRS. Some agents achieved an increase in the win-rate such as CatLinux. The Table 3 show that in this setting, the OLMCTS is the first-ranked agent, OLETS is the second-ranked agent and Number27 is the third-ranked agent.

C. Discounted Reward

In this metric, we changed the settings of the GVG-AI framework in such a way, that it depends on the Discounted Factor **D** when the forward model returns the score **s** [5]. The value of the discounted factor **D** is set to 0.9. The formula for the calculation is: **s = s*D**. We compared the rankings of agents with deceptive as well as non-deceptive levels. Table 4 shows that in this setting, the ranks are assigned to the agents according to total points but the ranking is also dependent on the win-rate. In this setting, the sampleRS agent suffers a drop in win-rate. Thus, the performance of the controllers has the highest impact on the game Zelda. In the previous setting, the first-ranked controller is the OLMCTS, whereas in this setting the first-ranked controller is OLETS, the second-ranked controller is thorbjrn and the third-ranked controller is OLMCTS.

D. Noisy World

In this metric, when action is performed by an agent, noise is added to the actions [5]. The noise is added in the actions when p = 0.25. Table 5 illustrates the rankings of agents obtained with noisy world settings. In this setting, ranks are

TABLE I
RANKING OF AGENTS BY USING DECEPTIVE LEVELS

Agents Ranking									
Rank no.	*Agents*	*Total Points*	*Mean Score*	*Win Rate*	*G-0*	*G-1*	*G-2*	*G-3*	*G-4*
1	**Number27**	125	0.006	52%	**25**	**25**	**25**	**25**	**25**
2	**OLETS**	118	0.63	40%	**25**	**25**	**25**	**25**	18
3	**sampleRS**	118	0.63	28%	**25**	**25**	18	18	**25**
4	**Thorbjrn**	111	0.71	40%	**25**	**25**	**25**	18	18
5	**OLMCTS**	108	0.01	24%	**25**	**25**	**25**	18	15
6	**CatLinux**	108	0.01	20%	**25**	**25**	18	**25**	**15**
7	**sampleMCTS**	103	0.45	16%	**25**	**25**	18	10	**25**
8	**NovTea**	103	0.45	20%	**25**	**25**	**25**	10	15
9	**sampleOneStep**	93	0.13	8%	**25**	**25**	18	10	15
10	**doNothing**	86	0.02	4%	**25**	**25**	**25**	10	1

TABLE II
RANKING OF AGENTS BY USING NON DECEPTIVE LEVELS

Agents Ranking									
Rank no.	*Agents*	*Total Points*	*Mean Score*	*Win Rate*	*G-0*	*G-1*	*G-2*	*G-3*	*G-4*
1	**Number27**	125	0.006	36%	**25**	**25**	**25**	**25**	**25**
2	**OLETS**	118	0.63	30%	**25**	**25**	18	**25**	**25**
3	**sampleRS**	108	0.01	29%	**25**	**25**	18	15	**25**
4	**Thorbjrn**	108	0.01	33%	**25**	**25**	**25**	15	18
5	**OLMCTS**	103	0.45	16%	**25**	**25**	18	10	**25**
6	**CatLinux**	100	0.01	30%	**25**	**25**	**25**	15	10
7	**sampleMCTS**	93	0.13	20%	**25**	18	**25**	15	10
8	**NovTea**	93	0.13	23%	**25**	**25**	18	15	10
9	**doNothing**	90	0.12	8%	**25**	**25**	18	10	12
10	**sampleOneStep**	86	0.02	6%	**25**	**25**	10	1	**25**

TABLE III
RANKING OF AGENTS IN THE REWARD PENALIZATION SETTING BY USING DECEPTIVE LEVELS

Reward Penalization Settings									
Rank no.	*Agents*	*Total Points*	*Mean Score*	*Win Rate*	*G-0*	*G-1*	*G-2*	*G-3*	*G-4*
1	**OLMCTS**	125	0.04	25%	**25**	**25**	**25**	**25**	**25**
2	**OLETS**	118	0.06	35%	**25**	**25**	**25**	18	**25**
3	**Number27**	108	0.03	23%	**25**	**25**	**25**	18	15
4	**thorbjrn**	95	0.05	30%	**25**	**25**	18	15	12
5	**CatLinux**	93	0.96	36%	**25**	18	**25**	15	10
6	**NovTea**	93	0.02	36%	**25**	18	15	**25**	10
7	**sampleMCTS**	90	0.04	17%	**25**	**25**	18	12	10
8	**sampleRS**	88	0.05	10%	**25**	**25**	18	10	10
9	**sampleOneStep**	88	0.64	4%	**25**	18	**25**	10	10
10	**doNothing**	79	0.02	2%	**25**	18	**25**	1	10

assigned to the agents according to total points but the ranking is also dependent on the win-rate. In this setting, the OLETS agent is still in the first position as in the discounted reward setting, second-ranked agent is Number27. OLETS is the best controller tested with the highest points. And doNothing is the least-performing controller. The loss of scores in Zelda, Catapults, Cakybaky, and Digdug is smaller as compared to the Solarfox. The higher scores are achieve-able only in the Zelda game where the key is awarded to the agent in the game.

E. Broken World

In this metric, the real game introduces the noise in the actions, while the froward model is always accurate [5]. But the noise is added in the actions when p = 0.25. In this setting, the agent has no idea how to evaluate states so some states or actions can be missed by the agent. Table 6 shows that in this setting, the ranks are assigned to the agents according to total points but the ranking is also dependent on the win-rate. The results obtained in this setting are completely different as compared to the noisy world settings. Number27 is the highest-ranked agent that achieves the best results in the five games and doNothing is the least-rank agent that achieves poor results in the five games. The agent with the highest increase in the win-rate is OLETS. It is important to note here that the errors are now introduced in the rankings of the agents.

F. Broken Forward Model

In this metric, the forward model introduces the noise with p = 0.25, while the actions provided to the game are not changed [5]. Table 7 shows that in this setting, the ranks are assigned to the agents according to total points but the ranking is also dependent on the win-rate. In the settings of

TABLE IV
RANKING OF AGENTS IN THE DISCOUNTED REWARD SETTING BY USING DECEPTIVE LEVELS

Discounted Reward Settings									
Rank no.	***Agents***	***Total Points***	***Mean Score***	***Win Rate***	***G-0***	***G-1***	***G-2***	***G-3***	***G-4***
1	**OLETS**	118	0.07	35%	18	**25**	**25**	**25**	**25**
2	**thorbjrn**	108	0.04	30%	**25**	**25**	18	**25**	15
3	**OLMCTS**	108	0.04	15%	**25**	18	**25**	15	**25**
4	**Number27**	100	0.01	30%	**25**	15	**25**	10	**25**
5	**CatLinux**	100	0.01	38%	**25**	**25**	15	10	**25**
6	**sampleMCTS**	95	0.02	21%	**25**	18	15	**25**	12
7	**sampleRS**	91	0.05	10%	**25**	18	15	**25**	8
8	**NovTea**	86	0.01	38%	**25**	18	10	**25**	8
9	**sampleOneStep**	69	0.04	13%	**25**	10	8	**25**	1
10	**doNothing**	63	0.01	2%	**25**	12	15	10	1

TABLE V
RANKING OF AGENTS IN THE NOISY WORLD SETTING BY USING DECEPTIVE LEVELS

Noisy World Settings									
Rank no.	***Agents***	***Total Points***	***Mean Score***	***Win Rate***	***G-0***	***G-1***	***G-2***	***G-3***	***G-4***
1	**OLETS**	100	0.07	20%	**25**	**25**	15	10	**25**
2	**Number27**	92	0.01	30%	**25**	15	12	**25**	15
3	**thorbjrn**	87	0.05	20%	**25**	10	12	15	**25**
4	**sampleRS**	87	0.07	15%	**25**	**25**	15	12	10
5	**OLMCTS**	87	0.04	30%	**25**	15	12	**25**	10
6	**CatLinux**	85	0.01	20%	**25**	15	12	8	**25**
7	**sampleMCTS**	83	0.02	16%	**25**	15	8	**25**	10
8	**NovTea**	85	0.01	20%	**25**	15	10	**25**	10
9	**sampleOneStep**	80	0.4	19%	**25**	18	15	12	10
10	**doNothing**	79	0.02	4%	**25**	18	10	**25**	1

TABLE VI
RANKING OF AGENTS IN THE BROKEN WORLD SETTING BY USING DECEPTIVE LEVELS

Broken World Settings									
Rank no.	***Agents***	***Total Points***	***Mean Score***	***Win Rate***	***G-0***	***G-1***	***G-2***	***G-3***	***G-4***
1	**Number27**	108	0.03	23%	**25**	18	**25**	**25**	15
2	**sampleRS**	98	0.06	17%	**25**	**25**	18	15	15
3	**thorbjrn**	93	0.03	18%	**25**	15	18	**25**	10
4	**OLETS**	91	0.04	18%	**25**	18	15	8	**25**
5	**OLMCTS**	88	0.02	31%	**25**	18	10	10	**25**
6	**NovTea**	86	0.04	38%	**25**	18	10	**25**	8
7	**sampleOneStep**	84	36%	0.05	**25**	18	8	8	**25**
8	**sampleMCTS**	83	0.04	15%	**25**	10	15	**25**	8
9	**CatLinux**	80	0.01	20%	**25**	18	15	12	10
10	**doNothing**	69	0.02	4%	**25**	18	10	8	8

the broken forward model, OLETS is the only controller that achieves the highest position in the rankings. The Number27 agent has the same points as the OLETS agent but it is a second-ranked agent. As a result, when the noisy actions are executed, it is only present in the game but not in the forward model. The results show that the behavior of the agents in these metrics is changed due to increase or decrease in win-rate and total points. It is important to note here that OLETS is the best-performing agent. It works better in the metrics including discounted reward, noisy world and broken forward model because in these metrics it is the top-ranked agent as it attains the highest points, whereas in these metrics the points of other agents are comparatively low. OLETS uses Hierarchical Open-Loop Optimistic Planning (HOLOP) for playing different games and it turns out that its performance on a broader scale is better. OLETS's performance is worst in reward penalization metric because in this metric the OLETS agent achieves second position as it attains lowest points. But in this metric, OLMCTS agent performs better because it attains highest points, as it is a variant of Monte-Carlo Tree Search (MCTS) and is designed to work better in stochastic environments. It uses the forward model to reevaluate the actions instead of keeping the states of the game in the nodes of the tree. The performance of OLETS agent is also worst in broken world metric because in this metric it achieves fourth position as it attains lowest points. But in this metric, Number27 is the best-performing agent in the broken world metric as it attains highest points. The Number27 agent consists of a genetic algorithm. It uses multiple action sequences instead of using a single one. As each agent uses different algorithms, so it is also important to look at the the nature of the algorithm to decide best-performing agent. We can conclude here that the

TABLE VII
RANKING OF AGENTS IN THE BROKEN FORWARD MODEL SETTING BY USING DECEPTIVE LEVELS

Broken Forward Model Settings									
Rank no.	***Agents***	***Total Points***	***Mean Score***	***Win Rate***	***G-0***	***G-1***	***G-2***	***G-3***	***G-4***
1	**OLETS**	108	0.03	23%	**25**	**25**	**25**	18	15
2	**Number27**	108	0.03	18%	**25**	**25**	15	**25**	18
3	**Thorbjrn**	100	0.06	17%	**25**	**25**	**25**	15	10
4	**OLMCTS**	85	0.02	31%	**25**	15	**25**	10	10
5	**sampleRS**	100	0.06	17%	**25**	**25**	**25**	15	10
6	**CatLinux**	86	0.01	44%	**25**	**25**	18	10	8
7	**sampleMCTS**	85	0.04	15%	**25**	**25**	15	10	10
8	**NovTea**	80	0.04	38%	**25**	**25**	12	10	8
9	**sampleOneStep**	69	0.05	20%	**25**	18	10	8	8
10	**doNothing**	63	0.02	4%	**25**	15	10	12	1

use of Hierarchical Open-Loop Optimistic Planning (HOLOP) is better for deception problems because when the agent plays any level of a game, the value of next actions are also assessed. It not only includes the average value of the node but also the maximum value among its children.

V. CONCLUSION

The focus of this research is to assign ranks to the game-playing agents. Initially, we execute the agents on deceptive as well as non-deceptive levels to perform a comparison on the basis of total points and win-rate. There are 1250 deceptive levels played by each controller. Five metrics are also used including reward penalization, discounted reward, noisy world, broken world and broken forward model. These metrics analyze how the rankings of the game-playing agents are changed. The proposed approach successfully assigns a rank to the agents based on the characteristics of the agents including total points and win-rate. The experiments indicate better performance of OLETS agent in the majority of metrics. Few agents are not able to perform well in noisy conditions including sampleOneStep and doNothing. By analyzing the performance of the agents in detail, we have a better understanding of the ranking of the agents.

REFERENCES

[1] Togelius, Julian, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. "Search-based procedural content generation." In *European Conference on the Applications of Evolutionary Computation*, pp. 141-150. Springer, Berlin, Heidelberg, 2010.

[2] Shaker, Noor, Julian Togelius, and Mark J. Nelson. *Procedural content generation in games*. Switzerland: Springer International Publishing, 2016.

[3] Summerville, Adam, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. "Procedural content generation via machine learning (PCGML)." *IEEE Transactions on Games* 10, no. 3 (2018): 257-270.

[4] Bravi, Ivan, Diego Perez-Liebana, Simon M. Lucas, and Jialin Liu. "Shallow decision-making analysis in general video game playing." In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1-8. IEEE, 2018.

[5] Pérez-Liébana, Diego, Spyridon Samothrakis, Julian Togelius, Tom Schaul, and Simon M. Lucas. "Analyzing the robustness of general video game playing agents." In *2016 IEEE Conference on Computational Intelligence and Games* (CIG), pp. 1-8. IEEE, 2016.

[6] Nielsen, Thorbjørn S., Gabriella AB Barros, Julian Togelius, and Mark J. Nelson. "General video game evaluation using relative algorithm performance profiles." In *European Conference on the Applications of Evolutionary Computation*, pp. 369-380. Springer, Cham, 2015.

[7] Zafar, Adeel, Hasan Mujtaba, Mirza Omer Beg, and Sajid Ali. "Deceptive Level Generator." In *AIIDE Workshops*. 2018.

[8] Smith, Gillian, and Jim Whitehead. "Analyzing the expressive range of a level generator." In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, pp. 1-7. 2010.

[9] Reis, Willian MP, and Levi HS Lelis. "Human computation for procedural content generation in platform games." In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 99-106. IEEE, 2015.

[10] Ebner, Marc, John Levine, Simon M. Lucas, Tom Schaul, Tommy Thompson, and Julian Togelius. "Towards a video game description language." Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[11] Perez-Liebana, Diego, Spyridon Samothrakis, Julian Togelius, Tom Schaul, Simon M. Lucas, Adrien Couëtoux, Jerry Lee, Chong-U. Lim, and Tommy Thompson. "The 2014 general video game playing competition." *IEEE Transactions on Computational Intelligence and AI in Games* 8, no. 3 (2015): 229-243.

[12] Khalifa, Ahmed, Diego Perez-Liebana, Simon M. Lucas, and Julian Togelius. "General video game level generation." In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pp. 253-259. 2016.

[13] Anderson, Damien, Matthew Stephenson, Julian Togelius, Christoph Salge, John Levine, and Jochen Renz. "Deceptive games." In *International Conference on the Applications of Evolutionary Computation*, pp. 376-391. Springer, Cham, 2018.

[14] Perez-Liebana, Diego, Jialin Liu, Ahmed Khalifa, Raluca D. Gaina, Julian Togelius, and Simon M. Lucas. "General video game ai: A multitrack framework for evaluating agents, games, and content generation algorithms." *IEEE Transactions on Games* 11, no. 3 (2019): 195-214.

[15] Browne, Cameron B., Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. "A survey of monte carlo tree search methods." *IEEE Transactions on Computational Intelligence and AI in games* 4, no. 1 (2012): 1-43.

[16] Anderson, Damien, Philip Rodgers, John Levine, Cristina Guerrero-Romero, and Diego Perez-Liebana. "Ensemble Decision Systems for general video game playing." In *2019 IEEE Conference on Games (CoG)*, pp. 1-8. IEEE, 2019.

[17] Gaina, Raluca D., Diego Pérez-Liébana, and Simon M. Lucas. "General video game for 2 players: Framework and competition." In *2016 8th Computer Science and Electronic Engineering (CEEC)*, pp. 186-191. IEEE, 2016.

[18] Pedersen, Chris, Julian Togelius, and Georgios N. Yannakakis. "Modeling player experience in super mario bros." In 2009 IEEE *Symposium on Computational Intelligence and Games*, pp. 132-139. IEEE, 2009.

[19] Desurvire, Heather, Martin Caplan, and Jozsef A. Toth. "Using heuristics to evaluate the playability of games." In *CHI'04 extended abstracts on Human factors in computing systems*, pp. 1509-1512. 2004.

[20] Sweetser, Penelope, and Peta Wyeth. "GameFlow: a model for evaluating player enjoyment in games." *Computers in Entertainment (CIE)* 3, no. 3 (2005): 3-3.

Clustering Performance Analysis of Traditional and New-Generation Meta-Heuristic Algorithms

Suheda Semih ACMALI
Department of Computer Engineering
Karabuk University
Karabuk, Turkey
ORCID Id: 0000-0001-6834-0835

Yasin ORTAKCI
Department of Computer Engineering
Karabuk University
Karabuk, Turkey
ORCID Id: 0000-0002-0683-2049

***Abstract*—Data clustering, which divides a dataset into a certain number of dissimilar groups, can be considered as NP-hard optimization problem. Many meta-heuristic optimization algorithms have been used in the solution of clustering problem. In this study, we have exploited two meta-heuristic optimization algorithms to solve the clustering problem: Harmony Search as traditional algorithm and Grey Wolf Optimizer as new generation algorithm. We have measured their clustering performance on five well-known, benchmark clustering datasets and compare the obtained results with each other and K-means which is a widely used classical clustering algorithm. The experimental results indicate that, while Grey Wolf Optimizer is generally superior to the other two algorithms in terms clustering success, the computation time of Harmony Search is less than Grey Wolf Optimizer for the same iteration number. In addition, the standard deviation of Grey Wolf Optimizer is smaller than the other two algorithms.**

Keywords—clustering, meta-heuristic optimization, gray wolf optimizer, harmony search

I. INTRODUCTION

Data clustering is one of the most popular unsupervised learning methods for data analysis. It is utilized as an exploratory data analysis model in the fields of machine learning, data mining, image segmentation, marketing, pattern recognition, and many fields of engineering [1]. The main objective of clustering is partitioning dataset into different groups by measuring the similarity of the dataset objects. In addition, clustering aims to while the intra-cluster similarity is maximized, the inter-cluster similarity is minimized. Similarity measures among the data are calculated by taking a set of features in the data into account. Classical clustering algorithms are broadly divided into two groups as hierarchical and partitional clustering. Hierarchical clustering techniques form clusters that have effective order from top-down or bottom-up approach [2]. In top-down approach, the whole dataset is split to data clusters iteratively. In contrast, in bottom-up approach, each data object starts as a separate cluster and these clusters are merged in iterative way until it forms a single cluster [3]. Partitional clustering splits the dataset into a number of cluster by taking some similarity measurements criteria and fitness function in to account [4]. While hierarchical clustering algorithms do not require any preliminary information about the number of clusters, partitional clustering algorithms needs a certain cluster numbers in advance. However, partitional clustering techniques can move dynamically data objects among the cluster during the execution whereas hierarchical clustering techniques is static and not able to move a data object from one cluster to another [5].

Partitional clustering techniques, which try to find the most appropriate cluster centers, can be considered as an NP-hard nonlinear problems problem if cluster number is greater than three [6]. Modeling natural phenomena to solve nonlinear problems has attracted the attention of researchers for many years. Meta-heuristic optimization algorithms, which are nature-inspired approaches, are effective options to solve NP-hard clustering problems in terms of time complexity and clustering accuracy. Meta-heuristic optimization algorithms include a population, each member of which searches the solution space and share its knowledge with the others to find the optimum solution. In the solution of clustering problems with meta-heuristic optimization algorithm, each member of the population tries to find the optimum positions of cluster centers [1]. With their exploratory and exploiter model, meta-heuristic optimization algorithms are superior to the classical clustering algorithms such as K-means which may stick to the local optimums easily [7]. It is clearly observed that new generation meta-heuristic optimization algorithms show better performance in optimization problem. In this study, we will measure the clustering performance of traditional and new generation meta-heuristic optimization algorithms. While Harmony Search Optimization Algorithm (HS) [8] is selected as traditional optimization algorithm, Gray Wolf Optimizer Algorithm (GWO) [9] is selected as new generation optimization algorithm. Their clustering success is compared to each other and also to K-Means [10] which is known as classical clustering algorithm.

II. RELATED WORKS

Meta-heuristic optimization algorithms have been extensively used in the solution of clustering problems. In the literature, meta-heuristic optimization algorithms are basically divided into two categories: traditional algorithms those are found before 2010; new generation algorithms those are found after 2010.

The well-known traditional optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization Algorithm (PSO), Ant Colony Optimization (ACO) have been used to cluster different dataset in many previous researches. Merwe et al. have applied PSO to the clustering problems first [11]. In a similar study, Sherar et al. combined PSO and K-Means for clustering big datasets that they get better results in terms of speed and error rate [12]. Sibil et al. have exploited GA in the clustering of the acoustic emission data and have got better results than K-means [13]. Nihkam et al. combined PSO, ACO, and K-means to increase clustering performance. Generally, a hybrid clustering algorithm, which combines the powerful ways of optimization algorithm are superior to the original ones [14].

The previous studies indicated that new generation optimization algorithms have been showing better clustering

performance than traditional ones. Artificial Bee Colony Algorithm (ABC) [15], which was proposed by Karaboğa in 2010, is of the new generation optimization algorithms. Karaboğa et al. have also adapted ABC to the clustering problems and have got much more successful clustering results according to the traditional optimization algorithms especially in multivariate data sets [16]. In addition, Ji et al. have reported that combining ABC and K-modes algorithm will increase the performance of the clustering in categorical data [17]. Hatamlou have tried Black Hole Algorithm (BH), which was inspired by black holes phenomenon, in the clustering [18]. The obtained results indicated that BH was superior to K-means, PSO and Gravity Search Algorithm. Ozbakır et al. have used Ion Motion Optimization (IMO) and Weighted Superposition Attraction (WSA) optimization algorithms in clustering and they have remarked the performance of algorithms can be increased by applying Deb's rule [19].

III. METHODOLOGY

A. Partitional Clustering

Clustering is a typical unsupervised machine learning technique and partitioning process of a datasets, that do not have any labels, into dissimilar groups without an identifier [20]. Partitional clustering finds out the similarity among the dataset with some measures and scatters the objects in the datasets to similar groups. While the objects in the same group should have similar characteristics as much as possible, the objects from different groups should have dissimilar characteristics.

Let $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_j}, \dots, \overrightarrow{x_N}\}$ represents a dataset which includes N different objects to be clustered. Each object $\overrightarrow{x_j} = (x_{j1}, \dots, x_{jd})$ is d-dimensional feature vector. The goal of partitional clustering is to find K clusters of X, where $C = \{C_1, \dots, C_K\}$ $(K \leq N)$, such that:

- $C_i \neq \emptyset, i = 1, \dots, K$: Each cluster must contain at least one object.
- $\cup_{i=1} C_i = X$: All objects in the dataset must be scattered to a cluster.
- $C_i \cap C_j = \emptyset,\ i, j = 1, \dots, K \text{ and } i \neq j$: An object can be assigned to only one cluster.

Different similarity measures, those evaluate the similarity of two objects, are used in clustering problems [21]. In this study, Euclidean distance [22], which calculates the distance between two objects as shown in Eq.1, is used as a similarity measure.

$$\left\|\overrightarrow{X_\iota}, \overrightarrow{X_J}\right\| = \sqrt{\sum_{\dim=1}^{d} (X_{i,dim} - X_{j,\dim})^2} \quad (1)$$

Partitional clustering is an iterative process that aims to find optimum clustering scheme by minimizing intra-cluster distance and maximizing the inter-cluster distance. Throughout the clustering iterations, a data object can be assigned to different clusters. In each iteration, the quality of clustering scheme is evaluated with a cluster validity index named Within Set Sum of Squared Errors (WSSSE) [23]:

$$WSSSE = \sum_{j=1}^{K} \sum_{i=1}^{N_j} \left\| x_i^{(j)} - c_j \right\|^2 \quad (2)$$

where $x_i^{(j)}$ is the i^{th} data object assigned to the j^{th} cluster and c_j is the center of the j^{th} cluster.

B. Grey Wolf Optimizer (GWO)

Gray Wolves are known as extreme predators living-in groups. This algorithm is based on the mathematical modeling of the social behavior of gray wolves and their hunting mechanism [9]. The wolf group, in which there is a strict hierarchy, contains different types of wolves called alpha, beta, delta and omega. The alpha wolf is the leader of group making the decisions. The beta and delta wolves help the alpha wolf for directing the group. The omega wolves follow the alpha, beta, and delta wolves. The hunting models of gray wolves are considered in three main steps: social hierarchy, encircling prey and attacking prey.

1) Social Hierarchy

In the design of GWO algorithm, the social hierarchy of wolves was taken as a model. In this model, the best solution, the second best solution, the third best solution and the other results are called as alpha (α), beta (β), delta (δ), omega (ω), respectively.

2) Encircling Prey

At this stage, the mathematical models of gray wolves circling their prey are explained. These formulas are:

$$\vec{D} = \left|\vec{C} * \overrightarrow{X_{p(t)}} - \overrightarrow{X_{(t)}}\right| \quad (3)$$

$$\overrightarrow{X_{(t+1)}} = \overrightarrow{X_{p(t)}} - \vec{A} * \vec{D} \quad (4)$$

where t is current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{X_p}$ is the position vector of the prey and $\vec{X}$ shows the position vector of a grey wolf.

The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a} * \overrightarrow{r_1} - \vec{a} \quad (5)$$

$$\vec{C} = 2 * \overrightarrow{r_2} \quad (6)$$

where $\vec{a}$ is a vector linearly reduced from 2 to 0 throughout the iterations and $\overrightarrow{r_1}, \overrightarrow{r_2}$ are random vectors in [0,1] interval.

3) Hunting

Gray wolves hunt by encircling prey. While alpha wolf directs the hunting, beta and delta wolves supports alpha wolf. To simulate hunting behaviors, GWO finds out the best three position vectors in search space and call the best, the second best, and third best solution as alpha, beta, and delta, respectively. Omega wolves update their positions according to these three position vectors. The formulas for this situation are given below:

$$\overrightarrow{D_\alpha} = \left|\left(\overrightarrow{C_1} * \overrightarrow{X_\alpha}\right) - \vec{X}\right| \quad (7)$$

$$\overrightarrow{D_\beta} = \left|\left(\overrightarrow{C_2} * \overrightarrow{X_\beta}\right) - \vec{X}\right| \quad (8)$$

$$\overrightarrow{D_\delta} = \left|\left(\overrightarrow{C_1} * \overrightarrow{X_\delta}\right) - \vec{X}\right| \quad (9)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \left(\vec{A}_1 * \overrightarrow{D_\alpha}\right) \quad (10)$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \left(\vec{A}_2 * \overrightarrow{D_\beta}\right) \quad (11)$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \left(\vec{A}_3 * \overrightarrow{D_\delta}\right) \quad (12)$$

$$\vec{X}_{(t+1)} = \frac{(\vec{X}_1 + \vec{X}_2 + \vec{X}_3)}{3} \quad (13)$$

The pseudocode of the GWO is presented in Algorithm 1.

Algorithm 1. Pseudocode of GWO

INPUT:
n: Population size, **maxIter :** Maximum iteration
OUTPUT:
$\boldsymbol{X_\alpha}$ **:** Optimal result

1: Initialize the grey wolf population $\boldsymbol{X_i}$ $(\boldsymbol{i = 1, 2, \dots, n})$
2: Initialize a, A and C
3: Calculate the fitness of each wolf
4: Find best wolf, $\boldsymbol{X_\alpha}$
5: Find second best wolf, $\boldsymbol{X_\beta}$
6: Find third best wolf, $\boldsymbol{X_\delta}$
7: while (t < maxIter)
8: **for each** wolf
9: Update the position of the current wolf by Eq. 9
10: **end for**
11: Update a, A and C
12: Calculate the fitness of all wolves
13: Update $\boldsymbol{X_\alpha, X_\beta}$ $\boldsymbol{and}$ $\boldsymbol{X_\delta}$
14: t = t +1
15: end while
16: return $\boldsymbol{X_\alpha}$

C. Harmony Search Algorithm (HS)

HS is inspired by a composer searching for the most tuneful harmony [8]. The composer uses past experiences, randomly struck a note, and pitch adjustment to find this harmony. These three principles are adapted to the solution of optimization problems in HS. HS basically starts with random generated harmonies called harmonic memory (HM), of which usage is controlled by Harmony Memory Considering Rate (HMCR) throughout the iterations. Each harmony infers a solution to the optimization problem and a new harmony is created just once in each iteration of HS. In the new harmony composition, bandwidth (BW) and Pitch Adjusting Rate (PAR) are used for pitch adjustment. If the HMCR value is less than random number generated, a new random harmony is created. The pseudocode of HS is given in Algorithm 2 [24].

D. Adaptations of GWO and HS to Clustering Problem

In order to adapt GWO to clustering solution, each wolf is designed to find optimum cluster centroid, namely, a wolf holds the position of cluster centroids $(C_1, C_2, \dots, C_k)$ for k clusters. Similarly, each harmony represents the centroids of clusters in HS.

For instance, Let the number of clusters is three and the dimension of dataset is four. The representation of this agent is as in Fig. 1:

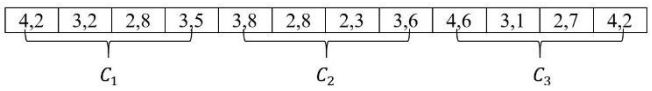


Fig 1. The representation of cluster centers

Algorithm 2. Pseudocode of HS

INPUT:
$\boldsymbol{n}$: Harmony Memory size, **maxIter**: Maximum iteration
HMCR: Harmony Memory Considering Rate
PAR: Pitch Adjusting Rate, BW : Bandwidth
OUTPUT:
$\boldsymbol{H_{best}}$ **:** Best solution

1: Generate Harmony Memory with random harmonies
2: while (t < **maxIter**)
3: **while** (i <= number of dimensions)
4: **if** (rand < **HMCR**)
5: Choose a value from HM for the variable i
6: **if** (rand < **PAR**)
7: Adjust the value by adding BW
8: **end if**
9: **else**
10: Choose a random value
11: **end if**
12: i = i + 1
13: end while
14: Accept $\boldsymbol{H_{new}}$ if better than $\boldsymbol{H_{worst}}$
15: Find $\boldsymbol{H_{best}}$
15: t = t + 1
16: end while
17: Find the current best solution, $\boldsymbol{H_{best}}$

GWO and HS iterate to find out the best clustering scheme until reaching stopping criteria. The clustering schemes produced by GWO and HS, are evaluated with a fitness function which calculates the intra-cluster distance. Both algorithms try to find out the intra-cluster distance using $WSSSE$ function shown in Eq. 2. The pseudocodes of GWO-based clustering and HS-based clustering are very similar to the original pseudocodes of GWO and HS, respectively. The only difference is the fitness function which is customized for clustering as shown in Algorithm 3:

Algorithm 3. Pseudocode of Fitness Funciton

INPUT:
data : $\boldsymbol{d}$-dimensional dataset $\boldsymbol{X} = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_j}, \dots, \overrightarrow{x_N}\}$
centers: Cluster centers
OUTPUT:
fitness : Fitness value

1: function fitnessCalculate(data, centers)
2: fitness = 0.0
3: bestDistance = INF
4: **for each** data
5: **for** center **in** centers **do**
6: localDistance = calcEuclideanDist(data, center)
7: **if**(bestDistance > localDistance)
8: bestDistance = localDistance
9: **end if**
10: fitness = fitness + bestDistance
11: **end for**
12: end for
13: return fitness

IV. EXPERIMENTAL RESULTS

In this study, we measured to clustering performance of GWO, HS, and K-Means on five different classification datasets from UCI database [25]. These are Iris, Wine, Glass, Cancer, and CMC datasets and their total number of data objects, number of clusters, number of features, and cluster sizes are given in Table 1.

The clustering performance tests of each algorithm were conducted on same condition and iteration number were set to 200 for all algorithm. The parameter settings for each algorithm are given in Table 2.

TABLE I. GENERAL STRUCTURE OF DATASET

Dataset	*Dataset Size*	*Cluster number*	*Feature number*	*Cluster sizes*
Iris	150	3	4	50,50, 50
Wine	178	3	13	59,71, 48
Glass	214	6	9	70,76,17, 13, 9, 29
Cancer	683	2	9	444, 239
CMC	1473	3	9	629, 334, 510

TABLE II. HS AND GWO PARAMETER VALUES

Algorithm	*Parameter*	*Value*
HS	*MaxIter(Maximum Iteration)* *n (Harmony Memory size)* *HMCR (Harmony Memory Considering Rate)* *PAR (Pitch Adjusting Rate)* *BW (Bandwidth)*	*200* *20* *0.9* *0.4* *0.2*
GWO	*MaxIter(Maximum Iteration)* *n (Population size)*	*200* *20*

In the experiment test, we run each clustering algorithm 10 individual times on a laptop, the hardware configuration of which is Intel Core i5-8300H and 16 GB DDR4 RAM. The coding of each algorithm is done in Java programming language. The results of experimental tests are presented in Table 3. Table 3 mutually lists the best, average, worst fitness values and standard deviations (Std) of K-Means as a classical clustering algorithm, HS as a traditional meta-heuristic algorithm, and GWO as a new generation meta-heuristic algorithm. The chosen fitness function, which sums up intra-cluster distances as given in Eq. 2, makes the clustering a typical minimization problem.

TABLE III. THE FITNESS VALUES OF CLUSTERING ALGORITHM

Dataset	*Criteria*	*K-Means*	*HS*	*GWO*
Iris	*Best*	*97.32854*	*121.9045*	***96.7311***
	Average	*105.7315*	*140.2566*	***103.9848***
	Worst	*128.4123*	*158.2566*	***121.4135***
	Std	*12.3915*	*12.0500*	***11.1110***
Wine	*Best*	*16554.5325*	*16714.0203*	***16327.1650***
	Average	*16964.1456*	*17408.4856*	***16360.1532***
	Worst	*23856.5638*	*18418.9546*	***16435.5513***
	Std	*1175.4653*	*695.8068*	***33.4340***
Glass	*Best*	*336.06051*	*386.98261*	***332.35125***
	Average	***354.14692***	*429.05975*	*370.64721*
	Worst	***378.89953***	*474.29752*	*419.21072*
	Std	***14.53379***	*29.12444*	*26.93642*
Cancer	*Best*	*2985.86478*	*3448.33241*	***2964.57861***
	Average	*3051.26845*	*4110.71123*	***2964.71563***
	Worst	*5156.23685*	*4459.65454*	***2964.99254***
	Std	*320.54362*	*292.91613*	***0.12475***
CMC	*Best*	***5553.68543***	*5970.54420*	*5641.54187*
	Average	***5556.42185***	*6438.71354*	*5737.45562*
	Worst	***5574.86235***	*6806.64473*	*5809.88417*
	Std	***2.15356***	*257.61613*	*69.43211*

GWO algorithm gave the minimum results at iris, wine, and cancer datasets for all criteria. For the iris dataset, GWO yielded the best, average, and worst solutions are 96.73118, 103.98486, and 121.41351, respectively. For wine and cancer datasets, GWO gave much better results than other algorithm. Even, the worst results of GWO are better than the best results of other algorithms. For glass dataset, while K-means got the minimum values for average, worst, and standard deviation criteria, GWO found the best minimum fitness value. For the CMC dataset, K-means the obtained the minimum fitness values for all criteria.

The decreases of fitness functions (WSSSE) of GWO and HS throughout the iterations for all datasets are presented in Figs. 2-6. As shown in these figures, for all datasets, GWO converges to the minimum fitness value before HS.

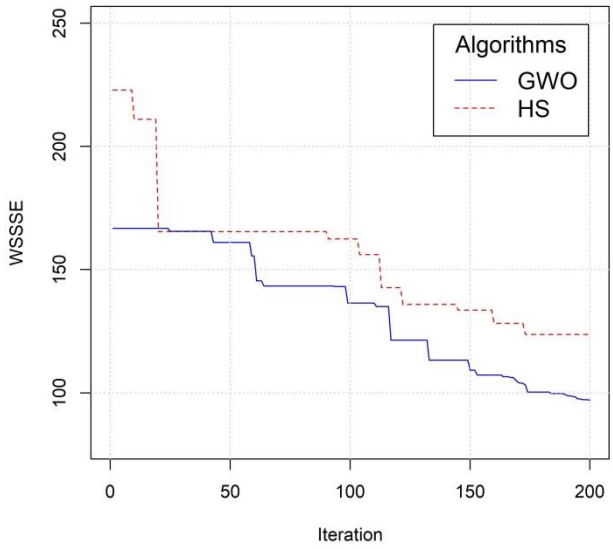


Fig 2. The fitness value-iteration chart of Iris dataset

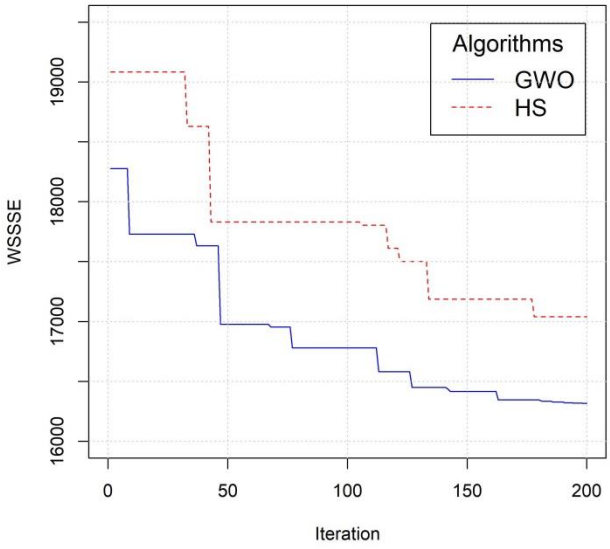


Fig 3. The fitness value-iteration chart of Wine dataset

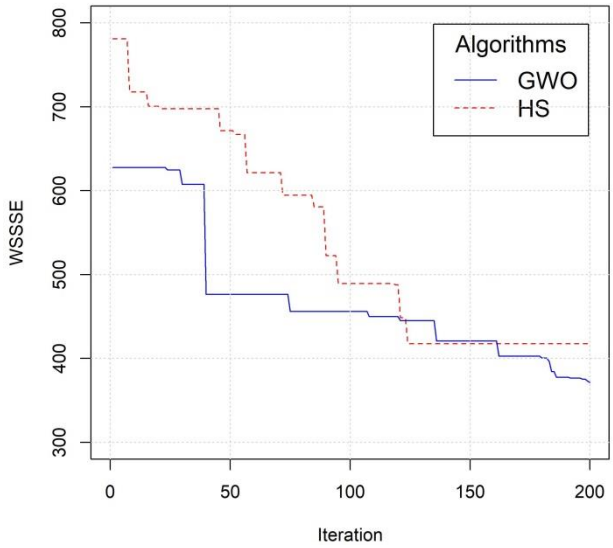


Fig 4. The fitness value-iteration chart of Glass dataset

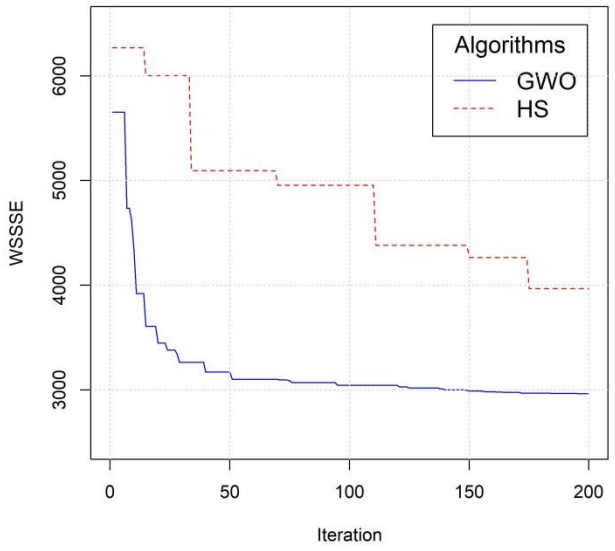


Fig 5. The fitness value-iteration chart of Cancer dataset

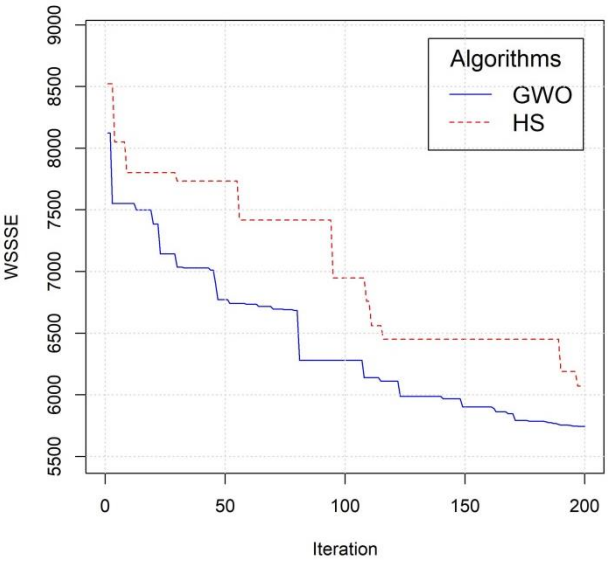


Fig 6. The fitness value-iteration chart of CMC dataset

TABLE IV. AVERAGE RUN-TIME (SEC)

Dataset	***GWO***	***HS***
Iris	*0.106*	***0.011***
Wine	*0.497*	***0.02***
Glass	*0.96*	***0.026***
Cancer	*0.233*	***0.024***
CMC	*2.208*	***0.097***

On the other hand, the average run-times of both algorithms for all datasets are measured in the experimental test and presented in Table 4. Although GWO produces better fitness values than HS, HS seems to reach the optimum results in a much shorter time than GWO. Since HS operates only one harmony in each iteration, the total run-time of HS is much shorter than GWO in all experiment test. In contrast, GWO is a population-based algorithm and operates 20 different agent wolves in each iteration. They also share their information throughout the iterations. If we consider the total evaluation numbers of each algorithm, GWO operates its agent wolves 200(iteration number)x20 (agent number)= 4000 times in an run, while HS runs its harmony just 200 (iteration number) times. This indicates that the time complexity of GWO is much more than HS and if the iteration number of HS is increased, HS will perform better clustering results.

V. CONCLUSION

Clustering problems are considered as an optimization problem and many meta-heuristic optimization algorithms have been implemented to solve clustering problems. In this study, we measured the clustering performance of two meta-heuristic algorithms: GWO as a new generation optimization algorithm and HS as a traditional optimization algorithm. The performance tests are done on the five common benchmark clustering datasets: iris, wine, glass, cancer, and CMC. As a result, while GWO finds out the optimum clustering results, the time complexity of HS is lower than GWO. In addition, the new-generation metaheuristic optimization algorithms can be applied to real clustering problem. On the other hand, as a future work, new algorithms can be used for the solution of clustering problems or some modification can be made on the current optimization algorithms to enhance their clustering performances.

ACKNOWLEDGMENT

We would like to thank Karabuk University BAP Unit for their financial support for this study.

REFERENCES

[1] Ortakci, Y., *Parallel Particle Swarm Optimization in Data Clustering.* International Journal of Soft Computing and Artificial Intelligence. **5**(1): p. 10-14.

[2] Johnson, S.C.J.P., *Hierarchical clustering schemes.* 1967. **32**(3): p. 241-254.

[3] Das, P., D.K. Das, and S. Dey, *A modified Bee Colony Optimization (MBCO) and its hybridization with k-means for an application to data clustering.* Applied Soft Computing, 2018. **70**: p. 590-603.

[4] Nanda, S.J. and G. Panda, *A survey on nature inspired metaheuristic algorithms for partitional clustering.* Swarm and Evolutionary computation, 2014. **16**: p. 1-18.

[5] Jain, A.K., M.N. Murty, and P.J.J.A.c.s. Flynn, *Data clustering: a review.* 1999. **31**(3): p. 264-323.

[6] Pacheco, T.M., et al. *An ant colony optimization for automatic data clustering problem*. in *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018. IEEE.

[7] Tang, R., et al. *Integrating nature-inspired optimization algorithms to K-means clustering*. in *Seventh International Conference on Digital Information Management (ICDIM 2012)*. 2012. IEEE.

[8] Geem, Z.W., J.H. Kim, and G.V.J.s. Loganathan, *A new heuristic optimization algorithm: harmony search.* 2001. **76**(2): p. 60-68.

[9] Mirjalili, S., S.M. Mirjalili, and A.J.A.i.e.s. Lewis, *Grey wolf optimizer*. 2014. **69**: p. 46-61.

[10] Hartigan, J.A. and M.A.J.J.o.t.r.s.s.s.c. Wong, *Algorithm AS 136: A k-means clustering algorithm.* 1979. **28**(1): p. 100-108.

[11] Van der Merwe, D. and A.P. Engelbrecht. *Data clustering using particle swarm optimization*. in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.* 2003. IEEE.

[12] Sherar, M. and F. Zulkernine. *Particle swarm optimization for large-scale clustering on apache spark*. in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017. IEEE.

[13] Sibil, A., et al., *Optimization of acoustic emission data clustering by a genetic algorithm method*. 2012. **31**(2): p. 169-180.

[14] Niknam, T. and B.J.A.s.c. Amiri, *An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis.* 2010. **10**(1): p. 183-197.

[15] Karaboga, D., *Artificial bee colony algorithm.* scholarpedia, 2010. **5**(3): p. 6915.

[16] Karaboga, D. and C.J.A.s.c. Ozturk, *A novel clustering approach: Artificial Bee Colony (ABC) algorithm.* 2011. **11**(1): p. 652-657.

[17] Ji, J., et al., *A novel artificial bee colony based clustering algorithm for categorical data.* 2015. **10**(5): p. e0127125.

[18] Hatamlou, A.J.I.s., *Black hole: A new heuristic optimization approach for data clustering.* 2013. **222**: p. 175-184.

[19] Özbakır, L. and F.J.K.-B.S. Turna, *Clustering performance comparison of new generation meta-heuristic algorithms.* 2017. **130**: p. 1-16.

[20] Xu, R. and D.J.I.T.o.n.n. Wunsch, *Survey of clustering algorithms.* 2005. **16**(3): p. 645-678.

[21] Saxena, A., et al., *A review of clustering techniques and developments.* Neurocomputing, 2017. **267**: p. 664-681.

[22] Das, S., et al., *Automatic clustering using an improved differential evolution algorithm.* 2007. **38**(1): p. 218-237.

[23] Yang, Y., et al. *Temporal clustering of motion capture data with optimal partitioning*. in *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*. 2016.

[24] Khadwilard, A., P. Luangpaiboon, and P.J.T.J.o.I.T. Pongcharoen, *Full factorial experimental design for parameters selection of harmony search Algorithm.* 2012. **8**(2): p. 56-65.

[25] Newman, D.J.h.w.i.u.e.m.M.h., *UCI repository of machine learning databases, University of California, Irvine.* 1998.

Computer vision and artificial intelligence for assessing plant diseases in data-driven agriculture: a case study of downy mildew in grapevine

Javier Tardaguila
Televitis Research Group
University of La Rioja
Logroño, Spain
javier.tardaguila@unirioja.es

Ines Hernandez
Televitis Research Group
University of La Rioja
Logroño, Spain
ines.hernandez@unirioja.es

Salvador Gutierrez
Dep. of Computer Science
University of Cádiz
Puerto Real, Cádiz
salvador.gutierrez@uca.es

Ignacio Barrio
Televitis Research Group
University of La Rioja
Logroño, Spain
ignacio.barrio@unirioja.es

Ruben Iñiguez
Televitis Research Group
University of La Rioja
Logroño, Spain
ruben.iniquez@unirioja.es

Fernando Palacios
Televitis Research Group
University of La Rioja
Logroño, Spain
fernando.palacios@unirioja.es

María P. Diago
Televitis Research Group
University of La Rioja
Logroño, Spain
maria-paz.diago@unirioja.es

***Abstract*—Downy mildew is an important disease in grapevine, which causes large losses of grape commercial production. The aim of this work was to investigate non-invasive sensing technologies and artificial intelligence applications for assessing downy mildew in grapevine under laboratory and field conditions. Machine vision was applied to assess visual symptoms while hyperspectral imaging was used to explore its potential capability towards an early detection of this disease. Image analysis applied to RGB leaf disk images taken under laboratory conditions was used to estimate downy mildew severity in grapevine (*Vitis vinifera* L.). Under laboratory conditions, a determination coefficient (R2) of 0.76 and a root mean square error (RMSE) of 20.53% were observed in the correlation between downy mildew severity by computer vision and expert's visual assessment. Furthermore, an accuracy of 81% was achieved for an early detection of downy mildew using hyperspectral images. The classification of images of healthy and infected leaves, assessed on RGB images taken under field conditions yielded an accuracy of 89% using machine learning and deep learning techniques. These results indicated that non-invasive sensing technologies and computer vision can be applied for assessing and quantify visual symptoms of downy mildew in grapevine leaves. Severity of this key grapevine disease can be evaluated under laboratory conditions, but also under field conditions in commercial vineyards. In conclusion, computer vision, machine learning, and deep learning could be applied for key disease detection in grapevine in data-driven viticulture.**

Keywords—hyperspectral imaging, non-invasive phenotyping tools, machine learning, deep learning, precision viticulture

I. INTRODUCTION

Plant diseases, pests and weeds cause large loses of food production in agriculture. Traditionally, plant diseases are identified by visual observations by the growers in the field or biological techniques in the laboratory [1]. Nevertheless, these techniques are time-consuming, susceptible to human error and/or require qualified personnel [1][2]. Non-invasive sensing technologies as RGB (Red, Green and Blue) images, thermography, multispectral and hyperspectral imaging have been postulated as potential non-invasive technologies for detecting plant diseases in agriculture with several advantages versus conventional methods [1][2][3], even detecting diseases when the symptoms are not visible [4]. Computer vision, machine learning, deep learning and artificial intelligence technologies can be applied to identify, classify and quantify crop diseases in data-driven agriculture [5][6][7][8].

In the European project NoPest (Novel Pesticides for a Sustainable Agriculture), non-destructive proximal sensing technologies and artificial intelligence are being developed for assessing fungal diseases in key crops as grapevine and potato. Downy mildew is a key grapevine disease in world viticulture (Fig. 1). Nowadays, the evaluation of this disease has been based mostly on visual assessment of leaves in the vineyards or histological analyses at the laboratory [9]. Computer vision and artificial intelligence could be very useful to recognize and quantify some diseases in grapevine [10][11][12].

The aim of this work was to examine non-invasive imaging technologies and artificial intelligence applications for assessing downy mildew in grapevine under laboratory

FIGURE 1: DOWNY MILDEW INFECTION IN GRAPEVINE REFLECTING THE SYMPTOMS IN LEAVES (LEFT) AND BUNCH (RIGHT).

This work has been developed as part of the project NoPest (Novel Pesticides for a Sustainable Agriculture), which received funding from the European Union Horizon 2020 FET Open program under Grant agreement ID 828940.

and field conditions in the context of the NoPest project. Computer vision was applied to evaluate visual symptoms while hyperspectral imaging was used for the early detection of downy mildew.

II. Material and Methods

This approach could be separated in three main sections (Fig. 2): i) image acquisition under laboratory and field conditions, ii) image processing using computer vision techniques and hyperspectral pre-processing to improve image features and highlight downy mildew symptoms and iii) machine learning modeling, used for the classification of RGB images of healthy and diseased leaves taken under field conditions and for the location of leaf spectra and early detection of disease using hyperspectral imaging.

A. *Image acquisition under laboratory and field conditions*

In laboratory conditions, leaf disks from grapevine (*Vitis vinifera* L., cv. Tempranillo) plants were placed in Petri dishes with the abaxial side up.

Two groups of study were defined: one group was infected with the downy mildew agent (*Plasmopara viticola*) and the other group was used as control. Images were taken in the laboratory every day until nine days after inoculation using a digital RGB camera (Canon EOS 5D, Japan). Additionally, a push broom hyperspectral visible (VIS) camera (spectral range from 400 to 1000 nm) and the hyperspectral near infrared- short wave near infrared (NIR-SWIR) camera (spectral range from 1000 to 1700 nm) were used for early detection of downy mildew under lab conditions (Fig. 3). For validation, the percentage of leaf area showing downy mildew sporulation in the leaf disks was visually evaluated by a panel of eight experts.

Under field conditions, images were acquired in a commercial vineyard (*Vitis vinifera* L.) located in the northern Spain. Downy mildew symptoms (Fig. 1) were observed in leaves in several grapevine plants. RGB images of the vineyard canopy were taken manually in the field using a digital camera (Canon EOS 5D, Japan). Furthermore, a push broom hyperspectral visible camera (Resonon, USA) was used for imaging grapevine leaf disks taken under field conditions in the vineyard.

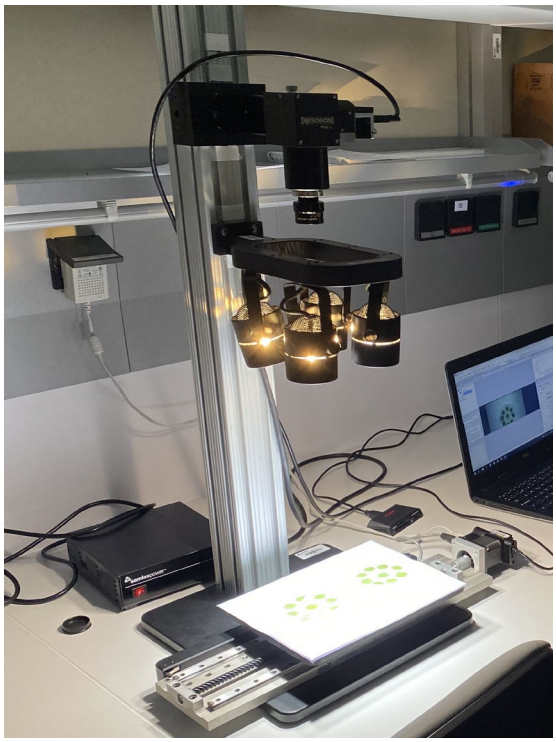


Figure 3: A push broom hyperspectral visible camera (400-1000 nm) used for early detection of downy mildew in grapevine under lab conditions.

B. *Processing of RGB and hyperspectral images*

The RGB images were processed to detect and quantify downy mildew symptoms in grapevine leaves. Different segmentation algorithms were applied to highlight relevant regions of the images.

Watershed [13] was used to separate the leaf disks in the images taken in the laboratory. GrabCut [14] was used to discard the pixels not belonging to the focused leaf in the

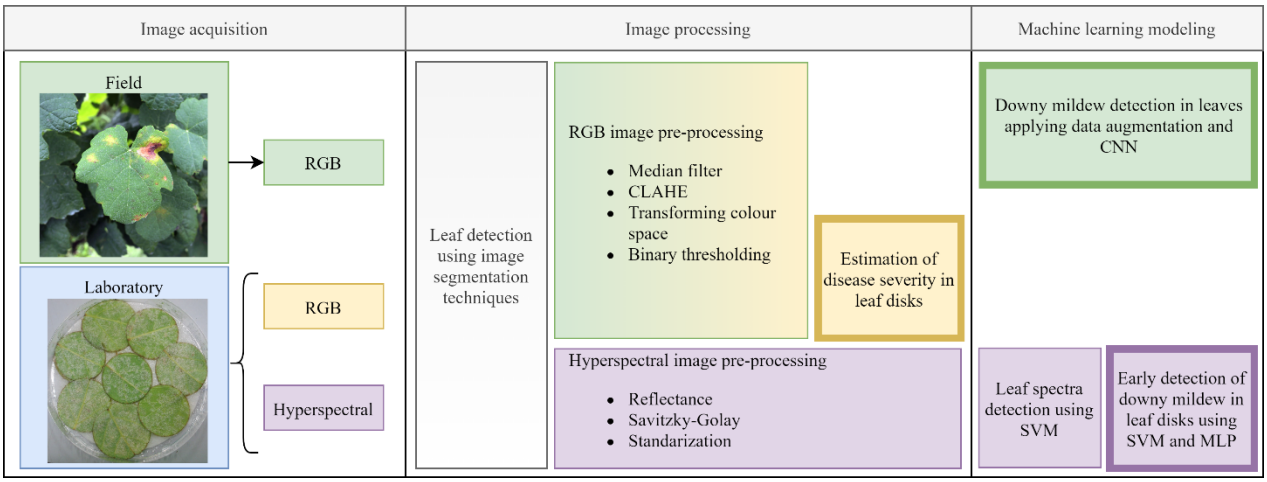


Figure 2: Flowchart of the use of non-invasive sensing technologies for assessing downy mildew in grapevine: RGB images taken under field conditions (green boxes) were used for disease detection, RGB images taken under laboratory conditions (orange boxes) were used for disease severity estimation and hyperspectral images (purple boxes) were applied towards the early detection of the disease.

images taken under field conditions. Classical pre-processing techniques were applied, such as median filter, used to smooth the image, and Contrast Limited Adaptive Histogram Equalization (CLAHE) [15], used to improve contrast. The colour space was transformed to HSV (Hue, Saturation, Value) or HLS (Hue, Lightness, Saturation) to bring the perception of the colour reflected in the digital image closer to the human eye by obtaining separated colour, saturation and brightness values from the red, green and blue values of the images. Depending on the type of symptoms in the leaves captured in the images, different values of the colour space were considered, with yellowish and reddish colours standing out for symptoms in the adaxial side of the leaf and lower saturation values for symptoms in the abaxial side of the leaf. These symptoms were highlighted using a binary threshold with a specific value and using the Otsu method [16], respectively, thus dividing the pixels representing disease symptoms in the leaves from those of the rest of the leaf (Fig. 4). The segmentation of leaf disk images was used to estimate downy mildew severity as the percentage of downy mildew symptoms that appear in the grapevine leaves.

On the other hand, hyperspectral images were processed (Fig. 5). The values (I) were transformed to reflectance (R) with a dark current (DC) and white reference (WR) values using the following equation:

$$R = \frac{I-DC}{WR-DC} \qquad (1)$$

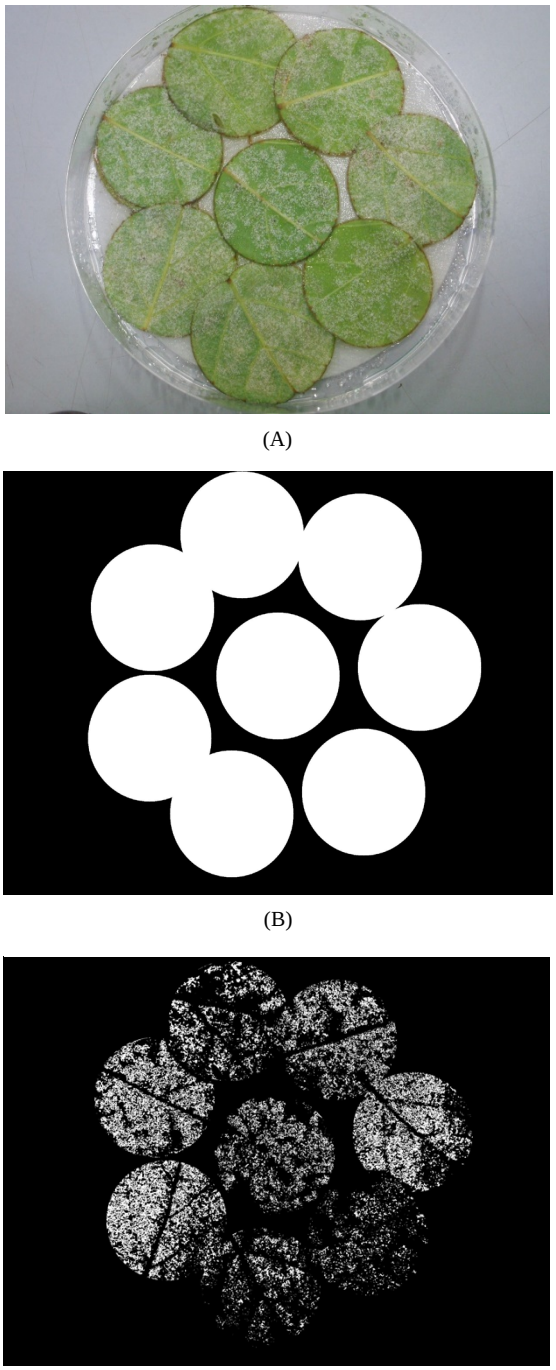
Savitzky-Golay filter [17] was also applied to smooth the spectra of each image, with a grade 2 polynomial and a size 15 window. Finally, a standardization was applied.

C. Machine and deep learning modelling

For hyperspectral images, a two-stage machine learning analysis was designed for (i) the detection of spectra belonging to leaves (segmentation) and (ii) modelling and prediction using the leaf spectra as input. The segmentation of the leaves was carried out by manually selecting spectra belonging to the positive class (leaf spectra) and negative class (spectra from the background elements). From these data, a binary classifier was trained using Support Vector Machines (SVMs) and applied for the automated segmentation of all the images.

The modelling for the pathogen detection was done after leaf spectra extraction in the previous step (thus generating the samples) and the training of the machine learning models using SVMs and Multi-layer Perceptrons (MLPs). Both algorithms were validated using a 5-fold cross validation.

For RGB images taken under field conditions, modelling was performed using deep learning techniques. Data augmentation was applied to improve the dataset, adding new artificial images from original ones. Convolutional Neural Networks (CNNs) were used to binary classify leaves with downy mildew symptoms and healthy leaves. The CNN architecture consisted of a block with convolutional, batch normalization and max pooling layers that automatically extract features of images and a block with fully connected, batch normalization and dropout layers that uses these features for image classification. The model was validated with a hold-out approach.



(A)

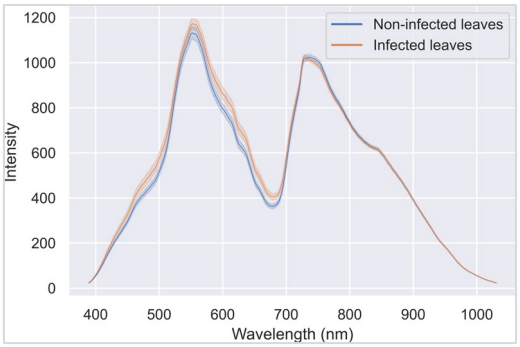
(B)

(C)

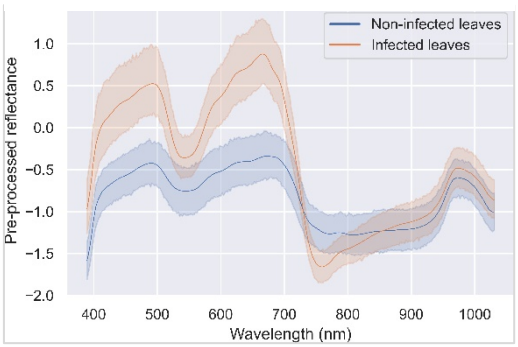
FIGURE 4: COMPUTER VISION WAS USED FOR ASSESSING DOWNY MILDEW SEVERITY IN GRAPEVINE LEAVES. A) ORIGINAL RGB IMAGE B) BACKGROUND REMOVAL AND C) PROCESSED IMAGE WITH DOWNY MILDEW SYMPTOMS (IN GREY) IN LEAF DISKS.

D. Implementation

All the experiments were developed with the Python 3.7.4 programming language. The RGB and hyperspectral images acquired under laboratory conditions were processed using the OpenCV 4.2.0.32 and scikit-learn 0.22.2 libraries on an Intel Core i7 4770 CPU (16 GB RAM). On the other hand, RGB images taken under field conditions were processed with a NVIDIA GeForce RTX 2080 Ti GPU (11 GB memory), optimizing the execution of the CNN developed with the Keras 2.3.1 framework and the Tensorflow 2.1.0 backend.



(A)



(B)

FIGURE 5: SPECTRA OF NON-INFECTED AND INFECTED GRAPEVINE LEAVES WITH DOWNY MILDEW: A) ORIGINAL SPECTRA B) PRE-PROCESSING SPECTRA BY TRANSFORMING THEIR VALUES TO REFLECTANCE, APPLYING A SAVITZKY-GOLAY FILTERING AND STANDARDIZATION.

III. RESULTS AND DISCUSSION

The method employed using computer vision techniques required no training, so its computational cost was low. As well as this method, the one used to classify the hyperspectral images, despite requiring training of the machine learning models, could be executed with a CPU, in less than one hour. On the other hand, for the classification of RGB images taken in the field, a GPU was used to train the CNN, which took several days. This type of neural network has a high computational cost in its training, but its prediction is fast, classifying more than 100 images in less than a minute.

Under laboratory conditions, a strong and significant relationship (determination coefficient of 0.76** and a root mean square error of 20.53%) was observed between downy mildew severity measured using computer vision and visual assessment by the experts in grapevine leaf disks. Additionally, an accuracy of 81% was achieved in the classification of hyperspectral images of grapevine leaves with and without downy mildew symptoms after a few days of the inoculation. Disease severity estimation of downy mildew using simple computer vision techniques obtained similar results to expert evaluation. This severity estimation could consider the expert subjectivity, achieving a greater relationship between automatic and manual assessment, using fuzzy logic [11].

In RGB images taken under field conditions, an accuracy of 89% was obtained in the classification of healthy and infected grapevine plants.

Downy mildew detection using machine learning techniques achieved high accuracies in RGB images and hyperspectral images. These methods could be combined with that used to estimate the severity of the disease to obtain a localisation and quantification of the symptoms or by applying deep learning techniques for image segmentation [7][18].

These results indicate that computer vision can be applied for assessing and quantify visual symptoms of downy mildew in grapevine leaves. Moreover, hyperspectral imaging could be applied for early detection of this disease in grapevine.

Machine learning algorithms can be very useful for downy mildew detection in grapevine, while deep learning techniques can be applied for modelling crop disease incidence. CNNs were applied to classify healthy and downy mildew leaves with high accuracy in grapevine.

Our results indicate that downy mildew can be evaluated under laboratory conditions but also under field conditions using new sensing technologies and artificial intelligence techniques in data-driven agriculture.

IV. CONCLUSIONS

New sensing technologies and data analysis have shown promising results for assessing downy mildew disease in grapevine. Computer vision, hyperspectral imaging and artificial intelligence can be applied for monitoring downy mildew in grapevine.

This key disease can be evaluated under laboratory and field conditions in commercial vineyards. These technologies could be also applied in grapevine and in other key commercial crops in data-driven agriculture for reducing yield losses and environmental impact of pesticides.

REFERENCES

[1] M. M. Ali, N. A. Bachik, N. A. Muhadi, T. N. T. Yusof, and C. Gomes. “Non-destructive techniques of detecting plant diseases: A review,” Physiological and Molecular Plant Pathology, 108, 101426, 2019. https://doi.org/10.1016/j.pmpp.2019.101426

[2] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis. “A review of advanced techniques for detecting plant diseases,” Computers and Electronics in Agriculture, 72 (1), 1–13, 2010. https://doi.org/10.1016/j.compag.2010.02.007

[3] E. C. Oerke, K. Herzog, and R. Toepfer. “Hyperspectral phenotyping of the reaction of grapevine genotypes to *Plasmopara viticola*,” Journal of Experimental Botany, 67 (18), 5529-5543, 2016. https://doi.org/10.1093/jxb/erw318

[4] Z. Gao, L. R. Khot, R. A. Naidu and Q. Zhang. "Early detection of grapevine leafroll disease in a red-berried wine grape cultivar using hyperspectral imaging," Computers and Electronics in Agriculture, 179, 105807, 2020. https://doi.org/10.1016/j.compag.2020.105807

[5] J. G. A. Barbedo. “Plant disease identification from individual lesions and spots using deep learning,” Biosystems Engineering, 180, 96-107, 2019. https://doi.org/10.1016/j.biosystemseng.2019.02.002

[6] A. K. Mahlein, M. T. Kuska, S. Thomas, M. Wahabzada, J. Behmann, U. Rascher, and K. Kersting. “Quantitative and qualitative phenotyping of disease resistance of crops by hyperspectral sensors: seamless interlocking of phytopathology, sensors, and machine learning is needed!,” Current Opinion in Plant Biology, 50, 156-162, 2019. https://doi.org/10.1016/j.pbi.2019.06.007

[7] J. de Paula Gonçalves, F. de Assis de Carvalho Pinto, D. de Queiroz, F. de Melo Villar, J. Barbedo and E. Del Ponte. “Deep learning models for semantic segmentation and automatic estimation of severity of foliar symptoms caused by diseases or pests,” 2020. https://doi.org/10.31219/osf.io/wdb79

[8] C. H. Bock, J. G. A. Barbedo, E. M. Del Ponte, D. Bohnenkamp and A. Mahlein. “From visual estimates to fully automated sensor-based measurements of plant disease severity: status and challenges for improving accuracy,” Phytopathology Research, 2 (1), 9, 2020. https://doi.org/10.1186/s42483-020-00049-8

[9] S. L. Toffolatti, G. Maddalena, D. Salomoni, D. Maghradze, P. A. Bianco, and O. Failla. “Evidence of resistance to the downy mildew agent *Plasmopara viticola* in the Georgian *Vitis vinifera* germplasm,” Vitis - Journal of Grapevine Research, 55 (3), 121-128, 2016. https://doi.org/10.5073/vitis.2016.55.121-128

[10] E. Peressotti, E. Duchêne, D. Merdinoglu and P. Mestre. “A semi-automatic non-destructive method to quantify grapevine downy mildew sporulation,” Journal of Microbiological Methods, 84 (2), 265-271, 2011. https://doi.org/10.1016/j.mimet.2010.12.009

[11] R. Nagi and S. S. Tripathy. “Infected Area Segmentation and Severity Estimation of Grapevine using fuzzy logic,” Advances in Intelligent Systems and Computing, 988, 57-67, 2020. https://doi.org/10.1007/978-981-13-8222-2_5

[12] C. H. Bock, P. E. Parker, A. Z. Cook and T. R. Gottwald. “Visual rating and the use of analysis for assessing different symptoms of citrus canker on grapefruit leaves,” Plant Disease, 92 (4), 530-541, 2008. https://doi.org/10.1094/PDIS-92-4-0530

[13] L. Vincent and P. Soille. “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (6), 583-598, 1991. https://doi.org/10.1109/34.87344

[14] C. Rother, V. Kolmogorov, and A. Blake. “’GrabCut’: interactive foreground extraction using iterated graph cuts,” ACM Transactions on Graphics, 23 (3), 309-314, 2004. https://doi.org/10.1145/1015706.1015720

[15] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. H. Romeny, J. B. Zimmerman, and K. Zuiderveld. “Adaptive histogram equalization and its variations,” Computer vision, graphics, and image processing, 39 (3), 355–368, 1987. https://doi.org/10.1016/S0734-189X(87)80186-X

[16] P. S. Liao, T. S. Chen, and P. C. Chung. “A fast algorithm for multilevel thresholding,” Journal of Information Science and Engineering, 17 (5), 713-727, 2001. https://doi.org/10.6688/JISE.2001.17.5.1

[17] A. Savitzky and M. J.E. Golay. “Smoothing and differentiation of data by simplified least squares procedures,” Analytical Chemistry, 36 (8), 1627-1639, 1964. https://doi.org/10.1021/ac60214a047

[18] G. Wang, Y. Sun and J. Wang. “Automatic image-based plant disease severity estimation using deep learning,” Computational Intelligece and Neuroscience, 2017, 1-8, 2017. https://doi.org/10.1155/2017/2917536

Computer vision for assessing downy mildew in grapevine leaves under laboratory conditions

Ines Hernandez
Televitis Research Group
University of La Rioja
Logroño, Spain
ines.hernandez@unirioja.es

Salvador Gutierrez
Dep. of Computer Science
University of Cádiz
Puerto Real, Cádiz
salvador.gutierrez@uca.es

Sara Ceballos
Televitis Research Group
University of La Rioja
Logroño, Spain
sara.ceballos@uca.es

Ruben Iñiguez
Televitis Research Group
University of La Rioja
Logroño, Spain
ruben.iniquez@unirioja.es

Ignacio Barrio
Televitis Research Group
University of La Rioja
Logroño, Spain
ignacio.barrio@unirioja.es

Fernando Palacios
Televitis Research Group
University of La Rioja
Logroño, Spain
fernando.palacios@unirioja.es

Silvia L Toffolatti
Department of Agriculture
University of Milan
Milan, Italy
silvia.toffolatti@unimi.it

Giuliana Maddalena
Department of Agriculture
University of Milan
Milan, Italy
giuliana.maddalena@unimi.it

María P. Diago
Televitis Research Group
University of La Rioja
Logroño, Spain
maria-paz.diago@unirioja.es

Javier Tardaguila
Televitis Research Group
University of La Rioja
Logroño, Spain
javier.tardaguila@unirioja.es

***Abstract*—Downy mildew is a critical disease in grapevine, which seriously affects production and grape quality. Usually, the detection of this disease is performed by expert's visual assessment of leaves and fruits. The aim of this work was to use computer vision techniques to develop a new algorithm for automatic detection and quantification of downy mildew symptoms in grapevine (*Vitis vinifera* L.) leaves under laboratory conditions. Computer vision techniques were applied for leaf disk location in Petri dishes, image pre-processing and the segmentation of pre-processed images to separate the pixels representing downy mildew infection from the rest. To validate the new computer vision algorithm, the downy mildew severity was visually evaluated by seven experts and average score was used as the reference method. A strong and significant correlation (R^2=0.83** and RMSE=13.1%) was observed between downy mildew severity obtained by the new algorithm and the visual assessment values. These results indicate that downy mildew severity in grapevine can be automatically evaluated by the new algorithm developed using computer vision techniques.**

***Keywords*—digital agriculture, precision viticulture, non-invasive sensing technologies. plant disease detection and crop protection.**

I. INTRODUCTION

Downy mildew is a key disease in world viticulture. Nowadays, the evaluation of the grapevine disease is based mostly on expert's visual assessment of the leaves or histological analysis at the laboratory [1]. New, non-invasive sensing technologies have been used for rapid and accurate disease detection in crops [2]. Computer vision can help to develop accurate, objective and fast methods for disease detection in agriculture using images of the crops [3][4][5]. Disease severity assessment in different crops such as coffee plants, passion fruit and cucumber can be performed applying computer vision techniques to leaf plant images [6]. Plant damages due to diseases or nutritional deficiencies were also detectable with RGB image analysis and the use of machine learning techniques [7]. Moreover, other more complex sensing technologies as hyperspectral imaging have been also employed for disease detection in plants [8]. Machine learning techniques have been also applied for plant disease detection and classification [9][10][11]. Fuzzy logic is another technique widely applied to reflect the subjectivity of results, applied to estimate the severity of diseases in agriculture, accompanied by other techniques such as computer vision and machine learning [12][13].

The aim of this study was to use computer vision techniques for assessing downy mildew disease severity (percentage of infection) in grapevine leaves.

II. MATERIAL AND METHODS

To evaluate the severity of downy mildew disease in grapevine leaves under laboratory conditions a new algorithm has been developed using computer vision techniques. The process carried out is summarised in Figure 1.

A. Plant material and image acquisition

Grapevine (*Vitis vinifera L*) leaf disks were placed in Petri dishes with the abaxial side up. Sporangial inoculum of the downy mildew (*Plamopara viticola*) was sprayed to leaf disks under laboratory conditions.

Images of Petri dishes containing eight inoculated leaf disks were taken with a digital camera (Olympus Stylus V (Olympus Imaging Corp. Japan). 16 RGB images of 2272x1704 size (2272 pixels wide and 1704 pixels high) were taken under laboratory conditions. In total, 128 grapevine leaf disks were imaged.

B. Computer vision algorithm for downy mildew assessment

The new computer vision algorithm for assessing downy mildew in grapevine consisted in four main steps: i) Leaf disk location on Petri dishes, ii) image pre-processing, iii) segmentation of pre-processed images to separate the pixels representing downy mildew infection from the rest, and iv)

This work has been developed as part of the project NoPest (Novel Pesticides for a Sustainable Agriculture), which received funding from the European Union Horizon 2020 FET Open program under Grant agreement ID 828940.

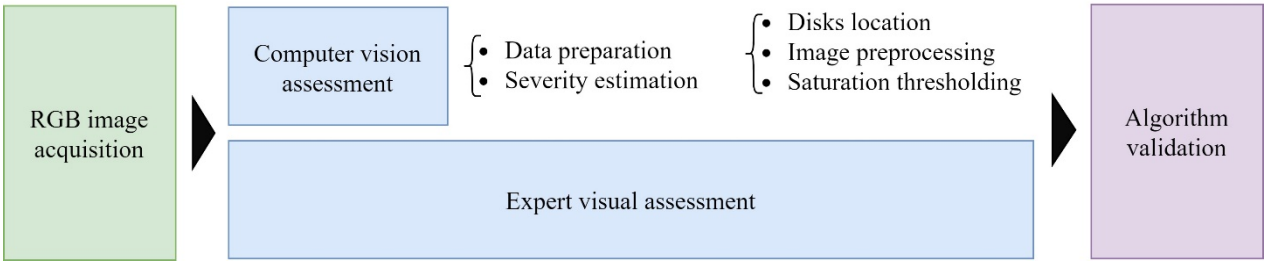


FIGURE 1: FLOWCHART OF THE EVALUATION OF DOWNY MILDEW IN GRAPEVINE LEAVES.

estimation of downy mildew severity computed by the algorithm.

Each leaf disk of the Petri dishes was located using Hough's transform [14]. In this way, despite the disks being of different sizes, they were retrieved in the image. In addition, to help with the location of the disks, the images were smoothed with the median filter used in the initial stage of the mean shift segmentation [15] to reduce image noise, and finally the RGB images were transformed to grayscale.

Image pre-processing were made following the steps summarised in Figure 2. First, the colour space was changed from RGB to HLS (Hue, Lightness, Saturation) to separate colour, lightness and saturation of each image. As shown in Figure 2, the saturation component seemed to highlight the pixels that include downy mildew symptoms. Therefore, only this component of the HLS colour space was used for the downy mildew estimation. Then, a median filter was used in the image that represented the saturation component to blur the image, accompanied by the opening morphological operation with a structuring element of a size of 3x3 pixels, which allowed to remove small holes within homogeneous regions of pixels. In this way, regions of the leaves with a dotted sporulation were unified, making these zones similar to what the human eye could see, taking into account that zone as a whole rather than each point individually. Finally, Contrast Limited Adaptive Histogram Equalization (CLAHE) [16] was applied to minimize the histogram contrast of all images, preventing some images from having significantly high or low lighting.

Segmentation of image pre-processing was made using a threshold that separated the pixels between downy mildew symptoms and the rest of the leaf. To obtain this threshold, one leaf disk, showing distinguishable downy mildew symptoms, was used as a reference. The threshold was obtained automatically based on the histogram of the pre-processed image of this leaf disk, using the Otsu method [17]. The disk chosen has different colours on the leaf, so to obtain a precise segmentation the multi-threshold variant of the Otsu method was used, which allows several thresholds to be found to divide an image. After analysing the thresholds obtained by the Otsu method, all images were segmented considering the threshold that allowed the downy mildew symptoms to be separated from the rest of the leaf. Likewise, a mask was created for each image representing downy mildew infection with ones and the rest of the image with zeros. In addition, to give greater importance to the pixels detected as downy mildew infection, the morphological operation of dilation was applied, thus joining, and enlarging small areas of pixels labelled as symptoms, which in the human eye would be detected as single areas, instead of separate areas that would result in much less severity.

Once the disks were located and the mask that divides the pixels into infection and the rest of the leaf was obtained, the infection severity ($PI_{i,j}$) was calculated considering in each disk the percentage of pixels that represents the infection, according to the following equation:

$$PI_{i,j} = \frac{x_{i,j}\,100}{N_{i,j}} \qquad (1)$$

Where $x_{i,j}$ represents the number of pixels labelled as downy mildew symptoms within disk i denoting the sporulation within the area assigned by the circle assigned to it within plate j, and $N_{i,j}$ represents the total number of pixels assigned to the area of the disk which, depending on the radius ($r_{i,j}$) obtained automatically, follows the equation $N_{i,j}=\pi r_{i,j}^2$.

C. Visual assessment and algorithm validation

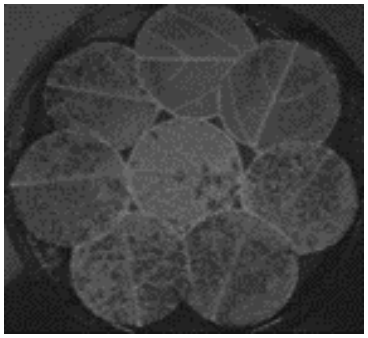
Downy mildew severity in all leaf disks was visually evaluated by seven experts to validate the computer vision algorithm. Each person evaluated each leaf disk using 0-100 scale reflecting the percentage of downy mildew infection. Prior to actual evaluation of leaf disks, experts were trained with additional leaf disks of variable rate of infection.

The downy mildew severity values for each leaf disk obtained by computer vision algorithm were then compared with the expert visual evaluation values (reference method). Determination coefficient (R^2) and root mean square error (RMSE) were computed.

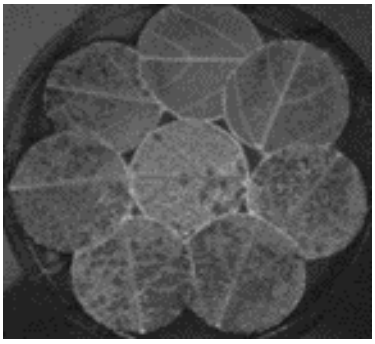
III. RESULTS AND DISCUSSION

As shown in Figure 3, applying the multi-threshold variant of the Otsu method to the reference disk obtained the grouping that can be seen in the segmented image. Therefore, the threshold that divided both groups (pixels representing the

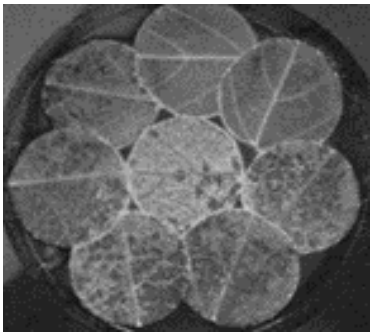
(A)



(B)



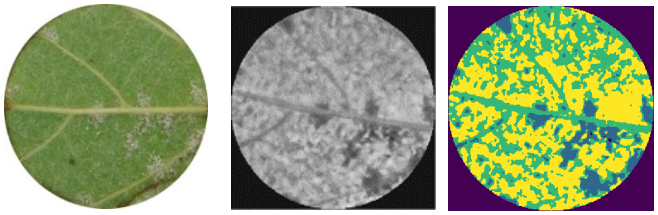
(C)



(D)

FIGURE 2: IMAGE PREPROCESSING STEPS. CHANGE THE COLOUR SPACE FROM ORIGINAL RGB IMAGE (A) TO HLS (B), APPLY THE MEDIAN FILTER TO SATURATION COMPONENT (C) OF THE HLS COLOUR SPACE AND APPLY THE CLAHE EQUALIZATION (D).

symptoms of downy mildew and the rest) was chosen, with the value 98, to be used in the rest of the disks. With the



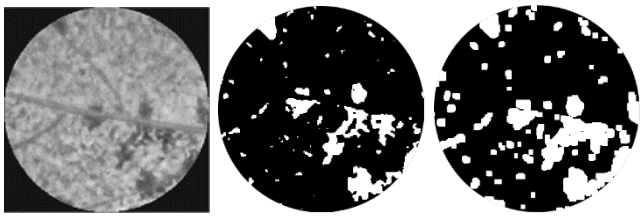
(A) (B) (C)

FIGURE 3: SEGMENTATION OF A GRAPEVINE LEAF DISK. ORIGINAL LEAF DISK (A), PRE-PROCESSED IMAGE (GRAYSCALE) USING THE SATURATION COMPONENT OF HLS COLOUR SPACE (B), THE SEGMENTATION OBTAINED AFTER APPLYING THE OTSU METHOD (C)

threshold chosen considering the reference disk, a mask as represented in Figure 4 was obtained for each disk of Petri dishes. After obtaining a mask for each leaf, the symptoms of downy mildew were detected (Fig. 5) and the severity of downy mildew was computed.

A strong and significant correlation (R^2=0.83**) was observed between downy mildew severity obtained by the new algorithm and the expert visual assessment values. The RMSE obtained by comparing the results of the algorithm with the expert visual assessment was of 13.61%. These results indicate a good estimation of the severity of downy mildew in grapevine leaves. The algorithm seemed to be sensitive to leaf nerves, reflections in water drops or light reflections from the leaf itself, being these the main faults found in the algorithm when detecting them as symptoms of downy mildew. Despite this, most of the disks have obtained an automatic evaluation very similar to that provided by the expert panel. This allows an automatic and rapid evaluation of downy mildew severity in grapevine.

Other works employ more complex techniques such as the use of deep learning [10][11][18], for the detection of diseases in agriculture, or the use of fuzzy logic [12][13] to evaluate the severity of a disease, approaches that require of computational complex pre-processing, training and modelling (in terms of time and memory). In this work, simpler computer vision techniques were used, removing steps of training and modelling, which allow a precise and quick evaluation of



(A) (B) (C)

FIGURE 4: EXAMPLE OF MASK OBTAINED FROM THE COLOUR THRESHOLD OF THE GRAYSCALE IMAGE (A) AND THE DILATION (B) APPLIED TO THE MASK (C).

(A)

(B)

FIGURE 5: ORIGINAL RGB IMAGE OF A PETRI DISH WITH EIGHT GRAPEVINE LEAF DISKS SHOWING DOWNY SYMPTOMS (A) AND PROCESSED IMAGE DISPLAYING DOWNY MILDEW SYMPTOMS (IN RED) LOCATED IN THE LEAF DISKS (B).

downy mildew disease in grapevine leaves, providing a location and quantification of the disease in 128 leaf disks in minutes very similar to the one that could be given by an expert after hours.

In conclusion, these results indicate that downy mildew severity in grapevine can be automatically evaluated by the new algorithm. Following the example of downy mildew in grapevine, it can be suggested that computer vision could be used for assessing major diseases in other key crops in precision agriculture.

REFERENCES

[1] S. L. Toffolatti, G. Maddalena, D. Salomoni, D. Maghradze, P. A. Bianco, and O. Failla. "Evidence of resistance to the downy mildew agent *Plasmopara viticola* in the Georgian *Vitis vinifera* germplasm," Vitis - Journal of Grapevine Research, 55 (3), 121–128, 2016. https://doi.org/10.5073/vitis.2016.55.121-128

[2] E. C. Oerke, K. Herzog, and R. Toepfer. "Hyperspectral phenotyping of the reaction of grapevine genotypes to *Plasmopara viticola*," Journal of Experimental Botany, 67 (18), 5529–5543, 2016. https://doi.org/10.1093/jxb/erw318

[3] C. H. Bock, P. E. Parker, A. Z. Cook and T. R. Gottwald. "Visual rating and the use of analysis for assessing different symptoms of citrus canker on grapefruit leaves," Plant Disease, 92 (4), 530-541, 2008. https://doi.org/10.1094/PDIS-92-4-0530

[4] C. H. Bock, J. G. A. Barbedo, E. M. Del Ponte, D. Bohnenkamp and A. Mahlein. "From visual estimates to fully automated sensor-based measurements of plant disease severity: status and challenges for improving accuracy," Phytopathology Research, 2 (1), 9, 2020. https://doi.org/10.1186/s42483-020-00049-8

[5] E. Peressotti, E. Duchêne, D. Merdinoglu and P. Mestre. "A semi-automatic non-destructive method to quantify grapevine downy mildew sporulation," Journal of Microbiological Methods, 84 (2), 265-271, 2011. https://doi.org/10.1016/j.mimet.2010.12.009

[6] J. G. A. Barbedo. "An automatic method to detect and measure leaf disease symptoms using digital image processing," Plant Disease, 98 (12), 1709–1716, 2014. https://doi.org/10.1094/PDIS-03-14-0290-RE

[7] O. M. O. Kruse, J. M. Prats-Montalban, U. G. Indahl, K. Kvaal, A. Ferrer, and C. M. Futsaether. "Pixel classification methods for identifying and quantifying leaf surface injury from digital images," Computers and Electronics in Agriculture, 108, 155–165, 2014. https://doi.org/10.1016/j.compag.2014.07.010

[8] A. Lowe, N. Harrison, and A. P. French. "Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress," Plant Methods, 13 (1), 80, 2017. https://doi.org/10.1186/s13007-017-0233-z

[9] J. G. A. Barbedo. "Plant disease identification from in- dividual lesions and spots using deep learning," Biosystems Engineering, 180, 96–107, 2019. https://doi.org/10.1016/j.biosystemseng.2019.02.002

[10] G. Wang, Y. Sun and J. Wang. "Automatic image-based plant disease severity estimation using deep learning," Computational Intelligence and Neuroscience, 2017, 1-8, 2017. https://doi.org/10.1155/2017/2917536

[11] J. de Paula Gonçalves, F. de Assis de Carvalho Pinto, D. de Queiroz, F. de Melo Villar, J. Barbedo and E. Del Ponte. "Deep leaning models for semantic segmentation and automatic estimation of severity of foliar symptoms caused by diseases or pests," 2020. https://doi.org/10.31219/osf.io/wdb79

[12] A. Mukherjee. "Analysis of diseased leaf images using digital image processing techniques and SVM classifier and disease severity measurements using fuzzy logic," International Journal of Scientific & Engineering Research, 11 (9), 1905-1912, 2020. https://doi.org/10.14299/ijser.2020.08.12

[13] R. Nagi and S. S. Tripathy. "Infected Area Segmentation and Severity Estimation of Grapevine using fuzzy logic," Advances in Intelligent Systems and Computing, 988, 57-67, 2020. https://doi.org/10.1007/978-981-13-8222-2_5

[14] HK. Yuen, J. Princen, J. Illingworth, and J. Kittler. "Comparative study of Hough Transform methods for circle finding," Image and Vision Computing, 8 (1), 71-77, 1990. https://doi.org/10.1016/0262-8856(90)90059-E

[15] D. Comaniciu and P. Meer. "Mean shift: A robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (5), 603–619, 2002. https://doi.org/10.1109/34.1000236

[16] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. H. Romeny, J. B. Zimmerman, and K. Zuiderveld. "Adaptive histogram equalization and its variations," Computer vision, graphics, and image processing, 39 (3), 355–368, 1987. https://doi.org/10.1016/S0734-189X(87)80186-X

[17] P. S. Liao, T. S. Chen, and P. C. Chung. "A fast algorithm for multilevel thresholding," Journal of Information Science and Engineering, 17 (5), 713–727, 2001. https://doi.org/10.6688/JISE.2001.17.5.1

[18] A. Ramcharan, P. McCloskey, K, Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalahwa, J. Legg and D. P. Hughes. "A mobile-based deep learning model for cassava disease diagnosis," Frontiers in Plant Science, 10, 272, 2019. https://doi.org/10.3389/fpls.2019.00272

Application of deep learning and computer vision for grapevine flower counting in digital viticulture

Fernando Palacios
Televitis Research Group
University of La Rioja
26006 Logroño, Spain
fernando.palacios@unirioja.es

Gloria Bueno
VISILAB group
University of Castilla-La Mancha
13071 Ciudad Real, Spain
gloria.bueno@uclm.es

Jesús Salido
VISILAB group
University of Castilla-La Mancha
13071 Ciudad Real, Spain
jesus.salido@uclm.es

Maria P. Diago
Televitis Research Group
University of La Rioja
26006 Logroño, Spain
maria-paz.diago@unirioja.es

Rubén Iñiguez
Televitis Research Group
University of La Rioja
26006 Logroño, Spain
ruben.iniguez@unirioja.es

Javier Tardaguila
Televitis Research Group
University of La Rioja
26006 Logroño, Spain
javier.tardaguila@unirioja.es

Abstract*—The amount of grapevine flowers at bloom of a vineyard is an early indicator of its final yield at harvest, therefore a precise quantification of the number of flowers is relevant for assessing vineyard yield months before harvest. Red Green Blue (RGB) images were acquired on-the-go under field conditions from a set of 48 vines selected at flowering stage. Their number of flowers was counted and used as the ground truth. A computer vision algorithm was developed to quantify the visible flowers on the images and its performance was tested with the actual number of flowers. For the algorithm, a deep learning semantic segmentation approach was followed. The algorithm comprised two main steps: first inflorescences were segmented, and then flowers of the segmented inflorescences were also individually segmented. A SegNet architecture with a VGG19 network as the encoder was used in this work to perform both steps. A precision of 0.754, a recall of 0.741, and a F1-score of 0.746 were achieved for the individual flower detection step, showing the capability of the network to isolate and identify each flower in the image. Also, a determination coefficient (R^2) of 0.87 was obtained between the number of segmented flowers and the actual number of flowers per vine. These results show that grapevine flowers can be accurately detected and counted in RGB images acquired on-the-go under field conditions using deep learning and computer vision.**

Keywords—Precision viticulture, remote sensing technologies, semantic segmentation.

I. Introduction

Vineyard yield prediction is essential to achieve the desired fruit quantity and quality in wine industry, and to meet specific yield regulations of some countries. Conventional yield forecasting methods tend to be inefficient and inaccurate due to the need of manually collection of data, usually clusters from selected vines, which are then weighted towards assessing the final yield [1] [2].

Innovative, non-invasive technologies enable an efficient acquisition of large collection of data and precise predictions. One of these technologies, widely used in viticulture, is computer vision. Several key parameters in viticulture have been estimated using computer vision, as vine pruning weight [3] or canopy parameters [4]. Yield forecasting is a task that has also been previously addressed using computer vision at different phenological stages, such as pea-size [5], near harvest time [6][7][8] and combining different stages [9]. Previous works have addressed yield forecasting considering several phenological stages and conditions, but none of them considered flower counting of grapevines, which is highly correlated to grapevine final yield [10]. Information about the number of flowers can lead to an adequate estimation of the yield between three to four months before harvest. Some works have presented computer vision algorithms to quantify or estimate the number of flowers on RGB inflorescence images acquired in the field using a dark background to ease the image segmentation and processing [11] [12] [13] [14] [15] [16]. Other authors have overcome this problem presenting algorithms capable of quantifying the number of visible flowers on RGB inflorescence images acquired under natural conditions, using traditional image analysis methods [17] or newer deep learning techniques [18] [19].

Previous works in this topic developped image processing algorithms to address the problem of flower counting on manually acquired images that focused on the inflorescences. However, none of them have presented a solution for flower counting in images focused on the vines and acquired on-the-go using a vehicle or mobile platform. It is a highly relevant task to perform a rapid grapevine flower counting on a high number of vines and to obtain more precise early yield forecasts in commercial vineyards.

The purpose of this work was to develop a computer vision method to detect and quantify the number of visible flowers on RGB vine images acquired using a mobile sensing platform in commercial vineyards in order to provide assistance to grapegrowers with relevant information related to the final vineyard yield.

II. Materials and Methods

A. Experimental layout

A commercial vineyard located in Vergalijo (lat. 42º27'46.0" N; long. 1º48'13.1" W; Navarra, Spain) was used for the experiment. This vineyard was partially defoliated prior to image acquisition at flowering stage, removing two leaves per shoot. A set of 48 vines from three grapevine (*Vitis vinifera* L.) varieties were selected from the vineyard during 2018 season. The vines had 2 m of row spacing and 1 m of vine spacing within the row, and they were trained onto a vertical shoot positioned (VSP) trellis system.

B. *Image acquisition*

RGB images were acquired on-the-go during May 2018, nine days before full flowering, and 109 days before harvest, at night-time, using a mobile sensing platform customized at the University of La Rioja and used in previous works [4] [20]. The night-time acquisition eased the image processing by the algorithm and the differentiation of the vine under evaluation from the vines of the opposite row. An all-terrain-vehicle ATV (Trail Boss 330, Polaris Industries, Minnesota, USA) was used as the base for the platform. The ATV moved at 5 Km/h and incorporated a structure with several elements to allow the image acquisition during night time. These elements included an artificial lightning system and a camera Canon EOS 5D Mark IV RGB camera (Canon Inc. Tokyo, Japan) mounting a full-frame CMOS sensor (35 mm and 30.4 MP) equipped with a Canon EF 35 mm F/2 IS USM lens. The images were acquired at a height of 1 m from the ground and at a distance 0.80 m between the camera and the canopy.

C. *Ground truth validation data*

After on-the-go image acquisition, the actual number of flowers per vine was computed using an estimator developed by Millan et al. [13]. For this purpose, all inflorescences of each vine were manually photographed, each inflorescence individually, using the camera employed for acquiring on-the-go images and a black cardboard as background (Fig. 1). The algorithm presented by Millan et al. [13] processed the inflorescence images, automatically detecting the visible flowers in the images, and obtained an estimation of the actual number of flowers per inflorescence. The actual number of flowers of the inflorescence was estimated using a multilayer feed-forward backpropagation neural network that accepted as input the number of automatically detected flowers and several descriptors extracted from the image. The value of the actual number of flowers for a given vine was determined as the sum of the estimations obtained for all inflorescences belonging to that vine.

Although the estimator presented by Millan et al. [13] is an improvement over a manually flower counting, being more efficient and similarly accurate, it was designed to process images of single inflorescences with a dark background. Therefore, it was not able to work with images acquired without background, containing multiple inflorescences, or images from whole grapevines.

D. *Algorithm for flower counting on images acquired on-the-go*

Prior to counting the flowers in the images, each flower had to be individually identified in a previous step. To isolate each flower, a semantic segmentation approach was followed. Starting from a vine image (Fig. 2A), first the inflorescences were segmented (Fig. 2B), in order to reduce false positive incidence, then each flower of the segmented inflorescences was also individually segmented (Fig. 2C). A SegNet architecture [20] was employed for the semantic segmentation of the inflorescences and the individual flowers, using the VGG19 layers as the encoder layers. Due to the small size of the flowers in the images, the SegNet was not trained using full-resolution images (6720 x 4480) but using image patches. Full image processing was achieved employing a sliding window approach, where on each iteration an image patch is extracted and analysed for segmenting inflorescences. Then, the full image is reconstructed with all inflorescences segmented and only the patches containing inflorescences are reanalysed to segment individual flowers.

For the inflorescences' segmentation training, a set of 61 patches that contained inflorescences, and a set of 126 patches containing non-inflorescence objects (a total of 187 image patches) were selected, and each pixel of those image patches was manually labelled as "background" or "inflorescence". Several data augmentation techniques were applied to the image patches (rotations, flipping, blur and histogram equalization) until a set of 2440 image patches were obtained, of which 1220 contained inflorescences, and 1220 contained other elements. This last set was used as the training set.

For the individual flowers' segmentation training, a set of 35 image patches containing flowers was extracted and the pixels were manually labelled as "center", "contour" (representing the center and the contour of the flowers, respectively) and "background" (representing the rest of the image). The final training set, formed by 700 image patches, was obtained after applying the same data augmentation techniques of the inflorescences' segmentation training step.

As aforementioned, full resolution image processing was achieved following a sliding window approach. Image patches of 480 x 480 pixels were extracted in each iteration and

FIG. 1. GROUND TRUTH IMAGE EXAMPLE. INFLORESCENCE IMAGE ACQUIRED MANUALLY USING A BLACK CARDBOARD AS BACKGROUND TO BE PROCESSED BY THE ALGORITHM OF MILLAN ET AL. [13]

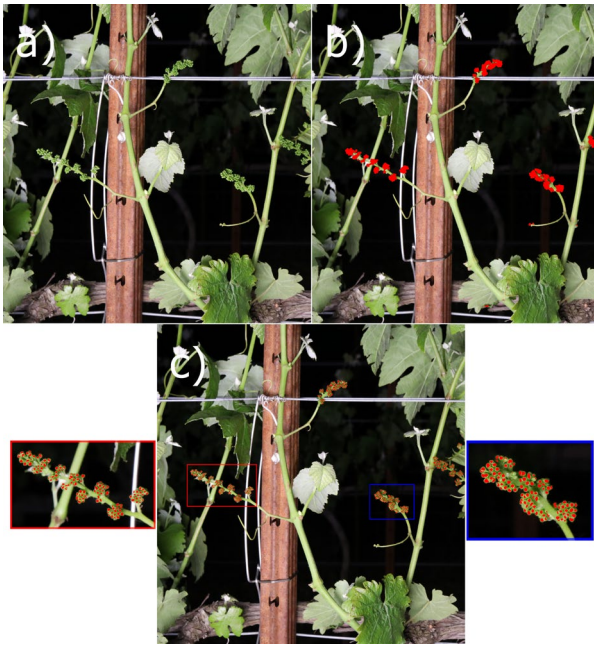


FIG. 2. IMAGE FLOWERS' SEGMENTATION STEPS: A) INITIAL GRAPEVINE IMAGE, B) SEGMENTATION OF INFLORESCENCES, AND C) SEGMENTATION OF FLOWERS.

analysed to segment inflorescences. In order to add robustness at segmenting inflorescences, especially with those close to the edges of the extracted patches, an overlap ratio of 50% between each patch and the previous one (vertically and horizontally) was considered. After inflorescences' segmentation, the full image was reconstructed using the confidence score obtained by SegNet and overlapped regions were averaged. The final inflorescences' segmentation mask was obtained by retaining those pixels with a confidence score above a threshold. Then, individual flowers were segmented and an additional step was performed to remove false positives: segmented groups of pixels from class "center" whose perimeter was not surrounded by "contour" class pixels (at least 50% of the perimeter) were removed. Some wrong artefacts produced by the segmentation were observed after this process.

III. Results and Discussion

In order to test the accuracy of the algorithm, 12 new, full-resolution images were processed and the results were checked. Two masks for each image were manually labelled, one mask contained a pixel-wise labelling of inflorescences and the other one included the checked segmented flowers' centers. The results presented in Table I show similar performance in terms of precision and recall for the inflorescences' segmentation step, being superior for the precision metric. The balance between both metrics is highlighted by the F1-Score, which proves that the algorithm was able to segment inflorescences accurately. Less differences were found between precision and recall for the flowers' detection step, being slightly superior for precision. While the results were not as satisfactory in this step, they proved to be good enough to consider that the algorithm was able to successfully detect and isolate individual flowers visible in images.

TABLE I. Performance metrics average of the inflorescences' segmentation and flower detection steps for the image validation set

Algorithm step	Performance metric		
	Precision	***Recall***	***F1-Score***
Inflorescences' segmentation	0.976	0.914	0.943
Flowers' detection	0.754	0.741	0.746

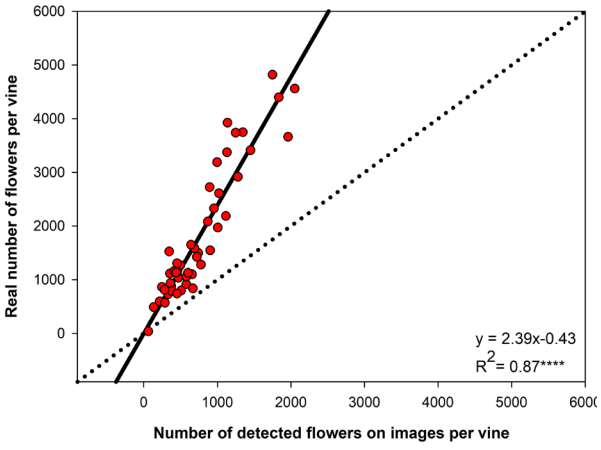


Fig. 3. Correlation between the actual number of flowers per vine and the number of flowers segmented by the algorithm

Fig. 3 shows that a strong and significant linear correlation (R^2 of 0.87***) existed between the number of segmented flowers and the actual number of flowers per vine. This indicates that actual number of flowers in grapevines can be assessed using a linear model. The number of flowers per vine can be used as indicator of yield forecast in commercial vineyards.

IV. Conclusions

The results of this work prove that the new algorithm presented was able to segment grapevine flowers individually on RGB vine images acquired on-the-go. In addition, the number of flowers segmented by the algorithm for a vine was significantly and linearly correlated to its actual number of flowers. This paves the way to an estimation model for the actual number of flowers based on the visible flowers for an early yield indicator.

The algorithm obtained promising results with images acquired during night-time, using artificial illumination. Future work could address the problem of flower counting in images acquired during day-time by adding to the training set additional images acquired under daylight illumination. Also, several defoliation conditions would be tested to study the relationship between the number of detected and the number of actual flowers, and to obtain a precise estimation of the actual number of flowers under high occlusion conditions of the flowers.

Acknowledgment

Fernando Palacios would like to acknowledge the research founding FPI grant 286/2017 by Universidad de La Rioja and Gobierno de La Rioja. Authors would like to thank Eugenio Moreda, Juan Fernández, Saúl Río and Ignacio Barrio for their help during field data acquisition.

References

[1] G.S. Howell. Sustainable grape productivity and the growth-yield relationship: A review. American Journal on Enology and Viticulture 52: 165–174, 2001

[2] S. Martin, R. Dunstone, G. Dunn. How to forecast wine grape deliveries. Technique report, Department of Primary Industries. 2003

[3] A. Kicherer, M. Klodt, S. Sharifzadeh, D. Cremers, R. Töpfer, K. Herzog. Automatic image-based determination of pruning mass as a determinant for yield potential in grapevine management and breeding. Australian Journal of Grape and Wine Research 23, 120–124. 2017

[4] M.P. Diago, A. Aquino, B. Millan, F. Palacios, J. Tardaguila. On-the-go assessment of vineyard canopy porosity, bunch and leaf exposure by image analysis. Australian Journal of Grape and Wine Research 25, 363–374. 2019. https://doi.org/10.1111/ajgw.12404.

[5] A. Aquino, B. Millan, M.P. Diago, J. Tardaguila. Automated early yield prediction in vineyards from on-the-go image acquisition. Computers and Electronics in Agriculture 144, 26–36. 2018. https://doi.org/10.1016/j.compag.2017.11.026

[6] D. Font, M. Tresanchez, D. Martínez, J. Moreno, E. Clotet, J. Palacín. Vineyard Yield estimation based on the analysis of high resolution images obtained with artificial illumination at night. Sensors 15, 8284–8301. 2015. https://doi.org/10.3390/s150408284

[7] S. Liu, M. Whitty. Automatic grape bunch detection in vineyards with an SVM classifier. Journal of Applied Logic 13, 643–653. 2015. https://doi.org/10.1016/j.jal.2015.06.001

[8] B. Millan, S. Velasco-Forero, A. Aquino, J. Tardaguila. On-the-go grapevine yield estimation using image analysis and boolean model. Journal of Sensors 2018, 9634752. https://doi.org/10.1155/2018/9634752

[9] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, S. Narasimhan, S. Singh. Automated visual yield estimation in vineyards. Journal of Field Robotics 31, 837–860. 2014. https://doi.org/10.1002/rob.21541

[10] P. May. Flowering and fruitset in grapevines. Lythrum Press. Adelaide (Australia). 2005

[11] M.P. Diago, A. Sanz-Garcia, B. Millan, J. Blasco, J. Tardaguila. Assessment of flower number per inflorescence in grapevine by image analysis under field conditions. Journal of the Science of Food and Agriculture 94, 1981–1987. 2014. https://doi.org/10.1002/jsfa.6512

[12] A. Aquino, B. Millan, D. Gaston, M.P. Diago, J. Tardaguila. vitisFlower®: development and testing of a novel android-smartphone application for assessing the number of grapevine flowers per inflorescence using artificial vision techniques. Sensors 15, 21204–21218. 2015. https://doi.org/10.3390/s150921204

[13] B. Millan, A. Aquino, M.P. Diago, J. Tardaguila. Image analysis-based modelling for flower number estimation in grapevine. Journal of the Science of Food and Agriculture 97, 784–792. 2017. https://doi.org/10.1002/jsfa.7797

[14] B. Radhouane, K. Derdour, E. Mohamed. Estimation of the flower buttons per inflorescences of grapevine (vitis vinifera L.) by image auto-assessment processing. African Journal of Agricultural Research 11, 3203–3209. 2016. https://doi.org/10.5897/AJAR2016.11331

[15] S. Liu, X. Li, H. Wu, B. Xin, J. Tang, P.R. Petrie, M. Whitty. A robust automated flower estimation system for grape vines. Biosystems Engineering 172, 110–123. 2018. https://doi.org/10.1016/j.biosystemseng.2018.05.009

[16] J. Tello, K. Herzog, F. Rist, P. This, A. Doligez. Automatic flower number evaluation in grapevine inflorescences using RGB images. American Journal of Enology and Viticulture. 2019. https://doi.org/10.5344/ajev.2019.19036

[17] A. Aquino, B. Millan, S. Gutiérrez, J. Tardáguila. Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis. Computers and Electronics in Agriculture 119, 92–104. 2015. https://doi.org/10.1016/j.compag.2015.10.009

[18] R.I. Rudolph, K. Herzog, R. Töpfer, V. Steinhage. Efficient identification, localization and quantification of grapevine inflorescences and flowers in unprepared field images using Fully Convolutional Networks. Vitis: Journal of Grapevine Research 58, 95–104. 2019. https://doi.org/10.5073/vitis.2019.58.95-104

[19] J. Grimm, K. Herzog, F. Rist, A. Kicherer, R. Töpfer, V. Steinhage. An adaptable approach to automated visual detection of plant organs with applications in grapevine breeding. Biosystems Engineering 183, 170–183. 2019. https://doi.org/10.1016/j.biosystemseng.2019.04.018

[20] F. Palacios, M.P. Diago, J. Tardaguila. A non-invasive method based on computer vision for grapevine cluster compactness assessment using a mobile sensing platform under field conditions. Sensors 19. 2019. https://doi.org/10.3390/s19173799

[21] V. Badrinarayanan, A. Kendall, R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 2481–2495. 2017. https://doi.org/10.1109/TPAMI.2016.2644615

Determining Covid-19 with Relieff-Based Machine Learning Algorithms Using Biochemistry Parameters

1st Çağla Danacı
Department of Software Engineering
Fırat University, Faculty of Engineering,
Elazığ, Turkey
191137102@firat.edu.tr

2nd Seda Arslan Tuncer
Department of Software Engineering
Fırat University, Faculty of Engineering,
Elazığ, Turkey
satuncer@firat.edu.tr

3rd Hakan Ayyıldız
Fethi Sekin City Hospital
Biochemistry Department
Elazığ, Turkey
hakan.ayyildiz1@saglik.gov.tr

4th Mehmet Kalaycı
Fethi Sekin City Hospital
Biochemistry Department
Elazığ, Turkey
dr_mehmetkalayci@msn.com

Abstract*— Since its emergence in 2019, the coronavirus epidemic, which has affected the whole world, especially Wuhan, China, continues to spread without being prevented. Early diagnosis plays a major role in preventing and reducing the coronavirus epidemic day by day. Covid-19 disease diagnosis is based on many diagnostic methods such as computed tomography, ultrasound imaging, laboratory tests. Artificial intelligence appears as a helpful tool for these diagnostic methods. Artificial intelligence saves time, cost and labor in diagnosis. Today, PCR (Polymerase Chain Reaction) test is actively used in the diagnosis of Covid-19In this study, biochemistry parameters were used in the diagnosis of Covid-19 disease and an application was developed to determine the priorities of biochemistry parameters in diagnosis. In the study, feature selection process was performed with the relieff method among all the biochemistry parameters and 6 priority parameters were determined. The classification process was performed with the priority parameters and 89.3% accuracy, 93.4% specificity, 85% sensitivity, 92.7% sensitivity and 88.7% F_1 score values were obtained with support vector machines. As a result of the study, it has been observed that the classification made with the features selected with the Relieff algorithm is more successful than the classification made using all biochemistry parameter parameters. It is thought that the work carried out will help early diagnosis in the Covid-19 outbreak, as well as reduce the workload of healthcare workers and save costs with the help of artificial intelligence.

Keywords*— Covid-19, Artificial Intelligence, Feature Selection, Medicine

I. INTRODUCTION

The coronavirus epidemic, which emerged in Wuhan, China in 2019, affected the whole world. The causative virus has been named 2019-nCoV (2019-novel coronavirus) by WHO, and SARS-CoV-2 (severe acute respiratory syndrome coronavirus-2) by the International Virus Taxonomy Committee, and the disease caused by the virus is COVID-19 (coronavirus disease- 2019) was defined as [1]. Basically, the Covid-19 epidemic, which caused the pandemic by affecting the world known to be transmitted by droplets, continues to be effective by bringing severe diseases and deaths.

Multiple diagnostic methods are used for Covid-19, which commonly manifests itself with fever, cough and shortness of breath. The diagnosis of Covid-19 is based on some auxiliary tests such as the patient's clinical symptoms, contact history and determination of viral nucleic acids, revealing lung findings with computed tomography, and showing seroconversion [2].

It is known that radiological methods alone are not sufficient for the diagnosis of Covid-19, and diagnostic laboratory tests also play an important role for the diagnosis of Covid-19.

Artificial intelligence appears as an aid to the analysis of laboratory tests for use in diagnosis. Laboratories constitute the basis of artificial intelligence applications in medicine as the centers where data are created and recorded. Artificial intelligence develops methods in the field of medicine in order to provide early and accurate diagnosis and correct treatment recommendations. Covid-19 diagnosis, treatment, estimation of mortality rates, medicine use, etc. in many areas, artificial intelligence is associated with Covid-19. When the literature is examined, we come across many studies that associate Covid-19 and artificial intelligence. Some of these studies are as follows:

Fernanda Sumika Hojo Souza et al. Conducted a study to predict Covid-19 positive patient outcome through machine learning. They included information from 13,690 patients on cases closed for treatment or death in the study. For the best prediction model, they obtained a ROC AUC of 0.92, sensitivity of 0.88 and specificity of 0.82 [3].

Lin Li et al. performed a machine learning-based study to provide automatic and accurate detection of COVID-19 using chest CT. They included data from 4356 chest CT exams taken from a total of 3.322 patients. They developed a deep learning model, COVID-19 detection neural network (COVNet), to extract visual features from chest CT examinations. At the end of the study, they achieved a sensitivity of 87%, specificity of 92% and AUC of 95% [4].

Felipe Soares developed a machine learning algorithm that takes simple blood exams as input and predicts whether this suspect case will be positive or negative. The study also used public blood data of 81 positive patients obtained from the Albert Einstein Hospital in Brazil, which was reduced to 599 data and obtained from 5,644 patients. At the end of the study, an average specificity of 92.16% and an average sensitivity of 63.98% were obtained [5].

In addition to machine learning-based applications, feature selection process appears within the scope of the study. Feature selection is basically the process of selecting the most meaningful and the most effective features among

the parameters. By using the feature selection process, higher performance can be achieved with fewer parameters. Feature selection process cost, time, etc. it allows us to save on many things. The feature selection process was included in the study, considering the advantages it brings in terms of Covid-19 diagnosis (helping early diagnosis, saving cost with less number of tests, etc.).

This study aims to diagnose Covid-19 with machine learning algorithms using biochemistry parameters. In addition, researches on which of the biochemistry parameters affect the classification process more have been done by using the Relieff feature selection method. For this purpose, the study was organized as follows. While the method and data used in the study are mentioned in the second section, the application is explained in detail in the third section. In section 4, discussion and conclusion are included.

II. Materials And Methods

A. Material

The data used within the scope of the study were obtained from Elazig Fethi Sekin City Hospital. The data set consists of a total of 120 patient data, 60 healthy and 60 diagnosed with Covid-19, with a total of 16 biochemistry parameters (CRP, Dimer, Urea, Krea., Prot, Alb, AST, ALT, LDH, Tbil, WBC, PLT, Neu, Lymp., Mon, Eos). Each of the data has different value ranges, and the value range table for the data set is given in Table 1.

TABLE I. Value Range Table

Parameter	Table Column Head	
	Maximum Value	***Minimum Value***
CRP	289	1
Dimer	3200	115
Urea.	95	11
Krea.	1.66	0.4
Prot	81	60
Alb	51	26
AST	107	13
ALT	197	9
LDH	420	143
Tbil	1.75	20.1
WBC	20.1	2.7
PLT	458	42
Neu	17.68	1.75
Lymp.	3.92	0.48
Mon	1.39	0.17
Eos	1.18	0.01

B. Method

The study is based on two main approaches. The first of these is to classify with machine learning methods using biochemistry parameters. Another subject of the study is to find out which of the biochemistry parameters is more effective for classification using the Relieff algorithm. Classification process was carried out on the data consisting of 16 parameters belonging to 120 patients, using machine learning algorithms (k-nearest neighbor algorithm, support vector machines, decision trees). After the classification process, the Relieff feature selection algorithm was applied on 16 parameters and the most important 6 parameters were selected among 16 parameters. After the parameter selection process, the selected parameters were used in machine learning algorithms and classified with 6 parameters. Here, by reducing the number of features, both cost calculations were made and it was investigated which features are more effective in diagnosis. The performances of the classification algorithms were evaluated and the two applications were compared. The process design of the study is given in Fig.1.

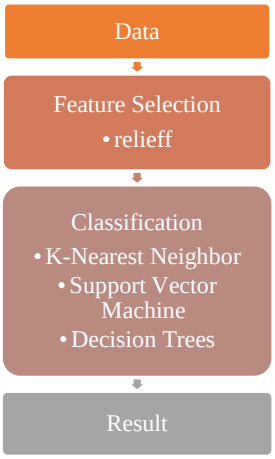


Fig. 1. Process Design

C. Relieff's Algorithm

Feature selection (also known as attribute selection or variable selection) is the process of selecting the best k feature among n attributes in the data set by evaluating the features according to the algorithm used [6].

The Relieff method proposed by Kira and Rendell finds the value of properties by trying to reveal the dependencies between them [6]. This method, which is used for binary classification problems, reaches the result by weighting the closest neighbor samples with a logic similar to the k-nearest neighbor algorithm. In the first step of the Relieff method, the weights of all features are initially set to 0. Then, in each step, he selects a random data from the data set and finds the closest k (k value is one less than the number of classes) data belonging to the same class, then the closest data belonging to each different class are found. In the next step, the weights for each feature are updated using this data. In the last stage, features that do not meet the specified condition are removed from the data set and a new data set is created and feature selection process is performed [7]. The pseudo code of the algorithm is given in Algorithm 1.

Algorithm.1 Relief Pseudo code

Inputs, D: Feature data matrix, n:the number of repeat, K: the number of the neighbors

Output, Vector w: Feature attributes ranking

for j:1 **to** n

Randomly select an instance R_j

Find K nearest hits H and nearest misses M

for i:1 **to** all features

updating estimation w_i by eq.1

end

end

D. Classification

K-Nearest Neighbor Algorithm (KNN)

K-nearest neighbor algorithm is a classification method in which the class in which the sample data point is located and the nearest neighbor are determined according to the value of k [8]. KNN is used by making use of data in a sample set whose classes are known. The distance of the new data that will be included in the sample data set is calculated according to the existing data and k number of close neighbors are checked [9]. The KNN algorithm is as follows [9]:

- First, the K parameter. K is the number of neighbors closest to the given point.
- Existing distances of new data are calculated individually.
- The distances are listed and the minimum distances are the closest neighbors to the hand.
- Knowledge of which class the closest neighbors can take to teach the most frequently recurring classroom.
- The selected class is considered as the class of new data.

Support Vector Machine (SVM)

A support vector machine creates an n-dimensional hyperplane that optimally divides the data into two categories [10]. The support vector machine is based on statistical learning theory and structural risk minimization. In support vector machine, the aim is to maximize the distance between support vectors belonging to different classes. Two situations that can be encountered in SVM are the situations where the data are linear and not linear. For nonlinear data, SVM cannot create a linear hyperplane. SVM uses various core functions such as Linear Function, Polynomial Function, Sigmoid Function for nonlinear cases. An example representation is given in Fig.2 for data that can be separated linearly and in Fig.3 for data that cannot be separated linearly.

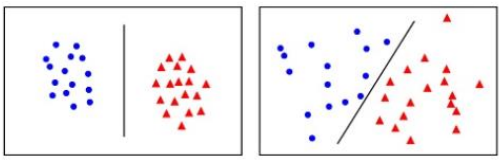


Fig. 2. Linear Separability [11]

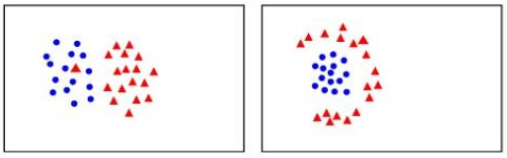


Fig. 3. Linear Non-Separation Condition [11]

Decision Trees

Decision trees are a classification and pattern definition algorithm that has been widely used in the literature in recent years. The most important reason for the widespread use of this method is that the rules used in the creation of tree structures are understandable and simple [12]. The decision tree consists of knots, branches, and leaves. The last part of the tree is called the leaf and the uppermost part is called the root.

The parts remaining between the root and leaves are expressed as branches [13]. The steps of the decision tree algorithm are as follows;

- Which Decision the Decision tree will make is determined.
- The entropy value of the system is calculated.
- The feature to be located in the root determined. For the feature determination process, information gain is calculated for each feature and the root node with the highest information gain is selected.
- All of the examples belong to the same class
- There is no feature left to divide examples ,
- No samples bearing the value of the remaining properties,

Transactions continue until one of their states occurs.

III. Experimental Results

Within the scope of the application, a total of 120 data consisting of 16 parameters, including 60 diagnosed with Covid-19 and 60 healthy patients, were included in the study. First, data was given to the classification algorithms and the classification process was performed using 16 parameters of 120 patients and 5 times cross validation. Optimal parameter selection was made for each classification method in accordance with the data distribution. The k value for the KNN algorithm is set as 1, the kernel scale parameter for the SVM algorithm is 4, and the maximum number of divisions for the decision trees is 10. The accuracy, sensitivity, specificity, sensitivity and F1 score values of the classification process performed with 16 parameters are given in Table 2.

TABLE II. RESULTS OF THE CLASSIFICATION OPERATION PERFORMED USING 16 PARAMETERS

Parameters	Classification		
	% KNN	*% SVM*	*% Decision Trees*
Accuracy	85.1	88.4	81
Specificity	86.8	91.8	83.6
Sensitivity	83.3	85	78.3
Precision	86.2	91.07	82.4
F1 Score	84.7	87.9	80.3

When Table 2 was examined, it was observed that the highest accuracy rate was reached by using the SVM algorithm with a value of 88.4% as a result of the classification process performed with 16 parameters. While the SVM algorithm was followed by the KNN algorithm with 85.1% accuracy, the decision trees took the last placein performance evaluation with 81% accuracy.

After the classification process was carried out, the feature selection process was carried out with the relieff method. As a result of the process, a total of 6 features among 16 features that have the most impact on the result were selected. These properties were determined as MON, WBC, NEU, LYMPHOCITE, ALB and PLT parameters, respectively. Data on parameter weights, which will enable us to better understand the importance levels of the selected features, are given in Fig. 4.

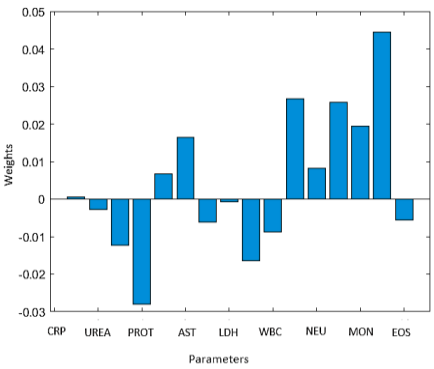


Fig. 4. Parameter Weights

After the feature selection process, the selected features were given to the classification algorithms and the classification process was performed again and a more successful result was obtained from the classification process performed for 16 parameters. Accuracy, sensitivity, specificity, sensitivity and F1 score values of the classification process performed with 6 parameters are given in Table 3.

TABLE III. RESULTS OF THE CLASSIFICATION OPERATION PERFORMED USING 6 PARAMETERS

Parameters	Classification		
	% KNN	*% SVM*	*% Decision Trees*
Accuracy	84.3	89.3	81.8
Specificity	86.9	93.4	81.9
Sensitivity	81.6	85	81.6
Precision	85.9	92.7	81.6
F1 Score	83.7	88.7	81.6

In the application carried out, as a result of the classification process using all parameters, the highest accuracy rate was obtained with support vector machines as 88.4%. As a result of the classification process performed with 6 parameters, it was observed that the target was achieved by achieving an accuracy rate of 89.3% with support vector machines.

IV. CONCLUSION

In this study, a cost-effective and fast approach using biochemistry parameters is proposed to help the diagnosis of Covid-19 disease that causes pandemics worldwide. However, the dominance of each biochemistry parameter on diagnosis was also investigated in the study.

As a result of the study, it was concluded that the application performed with the selected parameters can be a diagnostic system with lower cost than the methods used in the literature. The study is expected to support the physician, to be used for educational purposes in clinical studies and to shed light on future studies.

V. REFERENCES

[1] Alp, Ş., Ünal, S. (2020). Pandemic Caused by New Coronavirus (SARS-CoV-2): Developments and Current Status, *Review*, doi: 10.5578

[2] Togay, A., Yılmaz, N., (2020). Laboratory Diagnosis of SARS-CoV-2, Compilation, *Tepecik Training. And Research. Hospital. Magazine*,2020;30(Additional issue): 70-5, doi:10.5222/terh.2020.13007

[3] Souza, H.,S.,F., Hojo-Souza, S.N., Santos, B.E., Silva, M.C., Guidiono, L.D. (2020). Predicting the disease outcome in COVİD-19 positive patients through Machine Learning: a retrospective cohort study with Brazilian data, doi**:** https://doi.org/10.1101/2020.06.26.20140764

[4] Li ,L., Qin ,L., Xu ,Z., Yin , Y., Wang , X.,Kong , B.,Bai , J.,Lu ,Y., Fang , Z.,Song ,O., Cao ,K., Liu , D.,Wang , G., Xu ,Q., Fang , X., Zhang , S., Xia, J. (2020). Artificial Intelligence Distinguishes COVİD-19 from Community Acquired Pneumonia on Chest CT, doi: 10.1148/radiol.2020200905

[5] Soares, F. (2020).A novel specific artificial intelligence-based method to identify COVİD-19 cases using simple blood exams, Review, doi: https://doi.org/10.1101/2020.04.10.20061036.

[6] Budak. H. (2018). Feature Selection Methods and A New Approach, *Süleyman Demirel University Journal of the Institute of Science*, doi: 10.19113/sdufbed.01653

[7] Kaynar, O., Arslan, H., Görmez, Y., Işık, E.,Y. (2018). Attack Detection with Machine Learning and Feature Selection Methods, *Information Technologies Journal*, Volume: 11, Number:2,doi:10.17671/gazibtd.368583

[8] Babalık, A., Güler, İ., (2007). Use of Expert System in Diagnosis and Treatment of Throat Infections, Selçuk University, Vocational School of Technical Sciences, *Technical-Online Journal*

[9] Bayot, L.M., Naidoo, P., (2019). Clinical Laboratory, National Center for Biotechnology Information Search database

[10] Yakut, E., (2014). Stock Market Index Prediction Using Artificial Neural Networks and Support Vector Machines, *Journal of Faculty of Economics and Administrative Sciences*, Volume: 19, Number: 1, pp.139-157.

[11] Taş, O., (2016). *Support Vector Machines, Oğuzhan TAŞ Academy*. Citation: https://www.slideshare.net/oguzhantas/destek-vektr-makineleri-support-vector-machine

[12] Kavzoğlu, T., Colkesen, İ.,(2010). Classification of Decision Trees and Satellite Images: Kocaeli Case, Electronic Journal of Map Technologies, Volume: 2, Number: 1, pp.36-45.

[13] Bavaş, E., Citation: http://erdoganb.com/2017/07/karar-agaclari-decision-trees-ile-veri-siniflandirma/

Data Science Concept, Scope and Technological Development: A Review

Ahmet ALBAYRAK
Computer Engineering
Düzce University
Düzce, Turkey
ahmetalbayrak@duzce.edu.tr

Hamdullah KARAMOLLAOĞLU
Dept. of IMS and Education
Electricity Generation Company Inc.
İstanbul, Turkey
h.karamollaoglu@euas.gov.tr

***Abstract*— While data science was used in the fields of machine learning, computer science and data processing, which were previously related to the data field of data science, with the developing computer technologies, data science has been used in many different fields, especially in science, education, health, commerce, logistics, media and finance. In this study, the changes in the relationship between the concepts of "Data Science" and "Machine Learning" over the years were discussed by scanning the keywords of the articles on the web of science platform between 1994-2020. While the rate of publications in which data science and machine learning concepts are handled together was 30% in 1997, this rate decreased to 5% in 2020. Although the publications related to data science and machine learning have increased over the years, the reason for the decrease in this rate is the rapid increase in the use of data science with other disciplines.**

Keywords—data science, machine learning, big data, NoSQL, pyhton

I. INTRODUCTION

Data science is an interdisciplinary concept, and it is considered together with different disciplines especially in parallel with the development of computer technologies. One of the basic dynamics affecting the progress of computer technologies is the development of communication technologies. IPv6, which was proposed in 1996 and started to be used gradually in the following years, has an important place in computer communication. Before the transition to IPv6, around 4 billion computers could connect to the internet at the same time [1]. With IPv6, an almost unlimited number of machines have been enabled to connect to the internet and many new concepts have emerged. Some of these new concepts are big data, smartphone technologies, internet of things (IoT) and social networks. These concepts/ technologies contain sub-technologies within themselves. While the transition to IPv6 has enabled the emergence of these sub technologies, it has also empowered the web technologies which these technologies are associated with [2].

Web technologies support interactive and two-way communication, especially with the use of databases, and web 2.0 is the name given to this technology. Web 3.0 is a technology based on the principle that artificial intelligence supported technologies work in browsers and is called the semantic web. Another concept that emerges with the developing information technologies is big data [3].

II. BIG DATA AND DATABASES

Big data is a concept that defines heterogeneous digital data in different volumes and variety, which is difficult to exploit using traditional methods and technologies due to its characteristic features [4]. These data are obtained from digital media such as audio and video recordings, e-mail and other communication platforms, search queries, social networks, scientific data, sensors, mobile phone applications [5]. Big data analytics is the process of revealing useful information by processing data in different formats from many sources with various methods. There are many data analytics methods in the literature [6]. The main ones are shown in Table 1.

TABLE I. DATA ANALYTICS METHODS

Problem	***Method***
Clustering	BIRCH, RKM, TKM, DBSCAN, Incremental DBSCAN
Classification	SLIQ, FastNN, SFFS, TLAESA, GPU-based SVM
Association Rules	CLOSET, FP-tree, CHARM, MAFIA, FAST
Sequential Patterns	SPADE, CloSpan, PrefixSpan, SPAM, ISE

The basic algorithmic code structure of an example data analytics method is shown in Fig. 1.

```
Input data D
Initialize candidate solutions r
While the termination criterion is not met
    d=Scan(D)
    v=Construct(d,r,o)
    r=Update(v)
End
Output rules r
```

Fig.1 Sample data analytics algorithm

As seen in Fig.1., most data analytics methods include initialization, data input and output, scanning, rule construction, and rule update operators.

With the rapid development in computer technologies, the number of data sources is also increasing. While some of these data sources emerge as a new data source, some of them emerge with the transfer of existing data to the digital environment thanks to developing information technologies [7]. The fields and sample applications where data science and big data are combined are shown in Table 2.

TABLE II. RESEARCH AREA AND SAMPLE APPLICATIONS

Research Area	*Sample Application*
Medicine	Disease prediction from healthcare communities [8]
Biology	Prediction of plant biology process [9]
Astronomy	Data analysis in astronomy [10]
Telecom.	Churn prediction in telecommunication [11]
Production	Estimating production data [12]
Investment	Applying ANN to investment analysis [13]
Marketing	A model for marketing research [14]
Banking	Use of data mining in banking [15]
Insurance	Insurance big data analysis [16]
Media	Systems and methods for social media data mining [17]
Energy	Energy and environmental analysis [18]
Security	Information security analysis system [19]
Law	Crime data mining [20]
Tourism	Data mining in tourism data analysis [21]
Goverment	Government data analysis and public benefit [22]

As seen in Table 2, areas that enlarge big data by creating new data or digitizing existing data are generally institutions and organizations that provide services in the fields of transportation, logistics, retail, public services, telecommunications, healthcare, media, life sciences, video surveillance, banking, communication, media, entertainment, health services, education, manufacturing, government services, insurance, retailing, transportation and energy sectors [23][24].

While 10% increase in the rate of data use in organizations increases company productivity between 17% and 49%, it has been observed that the successful inclusion and use of big data in data science processes in companies increases the investment return by 241% [25].

Databases created with new designs are needed for the storage, processing and management of big data that emerges with the complexity of computer networks and the rapid development in communication technologies. New database designs have emerged instead of relational databases in this data environment where the need for performance and flexibility is increasing. The name of this new architecture is called Not-Only SQL (NoSQL). NoSQL, non-relational databases, have been developed to meet the needs of big data [26]. Table 3 includes some of the most used NoSQL databases.

TABLE III. NOSQL DATABASES

Key-value pair based	*Column-oriented graph*	*Graphs based*	*Document-oriented*
Aerospike, Apache Cassandra, Amazon Dynamo DB, Berkeley DB, Couchbase, Memcached, Riak, Redis	MariaDB, CreateDB, ClickHouse, GreenPlum, Apache Hbase, Apache Kudu, Apache Parquet, Hypertable, MonetDB	Neo4j, ArangoDB, OrientDB, Amazon Neptune, FlockDB, DataStax, Cassandra, Titan, Cayley, Azure Cosmos	MarkLogic, InterSystems Cache, MongoDB, OrientDB, Apache CouchDB, IBM Cloudant, CrateDB, Azure

As seen in Table 3, NoSQL databases categorised into four types: key-value pair based, column-oriented graph, graphs based and document-oriented databases [27].

III. DATA SCIENCE AND RELATED FIELDS

Data science has emerged with the extensive use of computer technologies in various disciplines. The use of IT in different disciplines has also led to the development of new data types [28]. Different data types trigger the development of programming languages to be able to interpret these data types. Data science can be defined as a data-oriented approach with the emergence of these new data types. Approaches based on data and extracting meaningful information from data are considered within the scope of data science [29].

The data-oriented approach should provide full control over the data received as input to computer systems. In this context, it is not possible to get data by means of a variable declaration, as in compiler programming languages, while reading data. Today, due to the heterogeneous form of big data and its extremely fast flow, programming languages that enable fast analysis of data and provide full control over the data are preferred. In this context, Python is one of the prominent programming languages [30].

Since its emergence, Python has been a language that has been accepted worldwide and developed applications in many areas. The fact that Python is used by leading organizations in the field of trade, education and health, as well as academic and scientific studies, shows Python's popularity as a programming language [31]. Thanks to its modular structure, Python can quickly adapt to changing computer technologies and can be supported with new packages. The most used Pyhton libraries (packages) are NumPy, Pandas, Matplotlib, Ipython, SciPy, Scikit-Learn, Keras, TensorFlow, Theano, Scrapy, Statsmodels, NLTK, Gensim, Plotly, Cufflinks, Searborn and Bokeh [32].

There are many tools and applications used in data science besides Python libraries, some of the most popular are Xplenty, RapidMiner, Orange, Weka, KNIME, Sisense, SSDT (SQL Server Data Tools), Apache Mahout, Oracle Data Mining, Rattle, DataMelt, IBM Cognos, IBM SPSS Modeler, SAS Data Mining, Teradata, Board, Dundas BI, Intetsoft, KEEL, R Data mining, H2O, Qlik Sense, Birst, ELKI, SPMF, GraphLab, Mallet, Alteryx and Mlpy [33].

When the data science approach is considered separately as data and science, the data part is more related to computer technologies. Fig.2 shows the fields related to data science. Computer science is a discipline that covers the methods and procedures required to solve a problem with the help of computer hardware and software. Mathematics is a branch of science that studies the structures, properties and connections between numbers and quantities with the help of arithmetic, algebra, geometry and calculations. Domain expertise is a discipline that includes in-depth knowledge of a specific field, process, project or program. Statistics is the science that analyzes probability, correlation, or trends in data and often works on big data. Data processing is a discipline that covers the processes of designing and applying computer programs to make data meaningful [34].

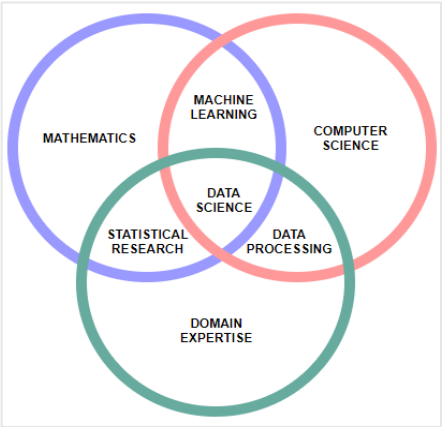


Fig.2. Basic disciplines within data science

As seen in Fig.2, basic disciplines within data science consists of mathematics, statistical analysis and domain expertise. Domain expertise is related to the disciplines in which the data science approach is applied. Mathematical expression of the problem and development of problem-specific mathematical models, which are indispensable for the scientific approach, fall within the scope of data science. Statistical studies and analysis include summarizing data and analyzing them with statistical methods. Within the scope of data science, the data field consists of sub-fields that develop depending on the development of computer technologies. These are computer science, data processing, and machine learning. Computer science and data processing are developing in parallel with the development of hardware, communication and software technologies in computer technologies. This is a historical development that started with the history of computing and continues until the development of today's computer systems.

IV. MACHINE LEARNING

As seen in Fig.3, machine learning is a sub-field of artificial intelligence and refers to a concept that includes both data and science fields in data science. Machine learning techniques consist of supervised and unsupervised learning methods. Supervised learning is a machine learning method that aims to create a function that includes this data and its results using previously observed and labeled data. Unsupervised learning is the process of finding the hidden structure in untagged data. That is, it is the process of revealing the existing but invisible relation between the data [35]. Classical machine learning methods are basically divided into two as supervised and unsupervised methods. Table 4 shows some of the machine learning methods.

TABLE IV. MACHINE LEARNING METHODS

Machine Learning		
Supervised		***Unsupervised***
Classification	***Regression***	***Clustering***
Support Vector Machines	Linear Regression	K-Means, K-Medoids, Fuzzy C-Means
Discriminant Analysis	SVR, GPR	Hierarchical
Naive Bayes	Ensemble Methods	Gaussian Mixture
K-Nearest Neighbor	Decision Trees	Neural Networks
Neural Networks	Neural Networks	Hidden Markov Model

One of the popular sub-field of machine learning is deep learning. Deep learning realizes learning from large data sets with layered architecture. Deep learning provides high power and flexibility thanks to the ability to process many properties on data [36]. In deep learning, data is passed through several layers. Each layer gradually processes the properties and transfers them to the next layer. Some of the most useful deep learning methods are Back Propagation, Stochastic Gradient Descent, Learning Rate Decay, Dropout, Max-Pooling, Batch Normalization, Skip-gram and Transfer Learning [37].

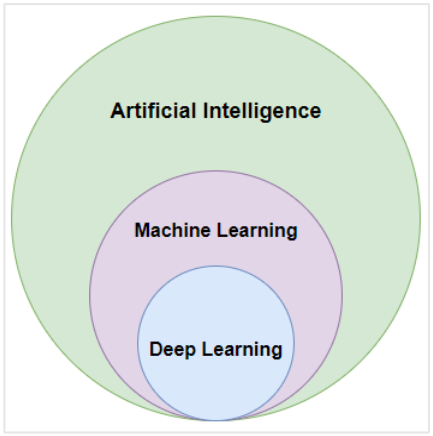


Fig.3. Artificial intelligence, machine learning and deep learning

Deep learning frameworks enable modeling of deep neural networks thanks to their high level programming interfaces. Commonly used deep learning methods are TensorFlow, Keras, PyTorch, Caffe, Neon, Torch, Theano, Deeplearning4j, PaddlePaddle, Chainer and MXNet [38].

Since machine learning is related to mathematics, it can be defined in the field of data in data science, as it is used in various disciplines thanks to computer technologies. In this study, taking into account that data science approach has been adopted intensively by today's researchers, its relationship with machine learning, which includes both data and science sub-fields, is analyzed. Since machine learning is related to mathematics, it can be defined in the field of science in data science, and thanks to computer technologies, it can be defined in the data field as it is used in various disciplines In this study, taking into account that data science approach has been adopted intensively by today's researchers, its relationship with machine learning, which includes both data and science sub-fields, is analyzed.

V. METHODOLOGICAL APPROACH

In this study, a systematic literature review has been made using the PRISMA approach for article selection [39]. The keyword string has been searched on the Web of Science platform. The search terms are structured as "Data Science" And "Machine Learning". The search operation was carried out only in the article title parts and articles between 1994-2020 have taken into consideration. From the results obtained as a result of the screening, only research articles and reviews are separated for evaluation. Conference papers were not evaluated in this study. In Fig.4, the distribution of publications by years is given with the keyword Data Science.

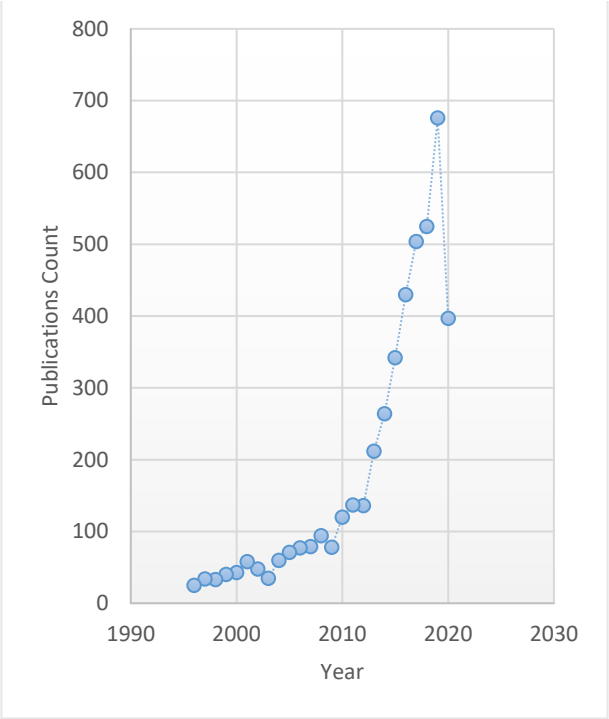


Fig.4. Change of publications of the "Data Science" approach by years

In Fig.4, it is seen that publications related to the concept of data science have increased especially during the period from 2015 to the present day. Fig.5 shows the distribution of the publications related to the machine learning concept by years.

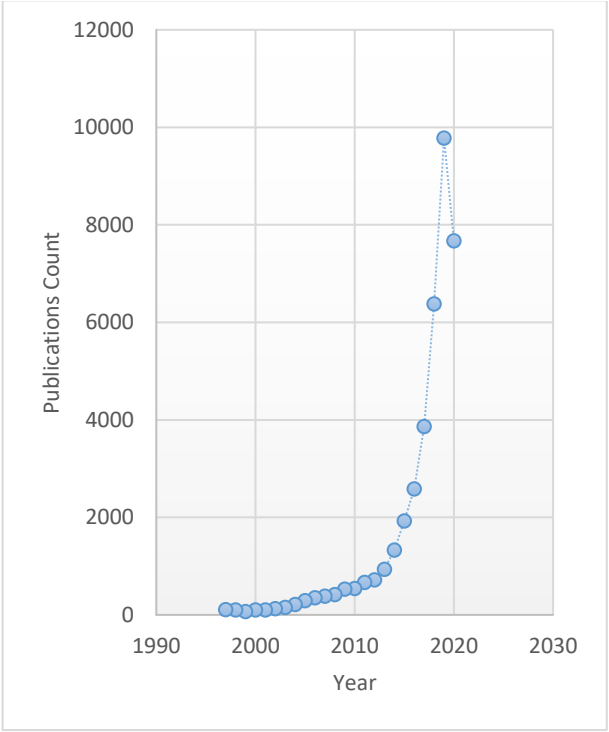


Fig. 5. Change of publications of the "Machine Learning" approach by years

As seen in Fig.5, the publications related to the concept of machine learning have been continuously increasing since 2015, similar to the graph in Fig.4.

Fig.6 shows the rate of data science concept among the publications related to machine learning.

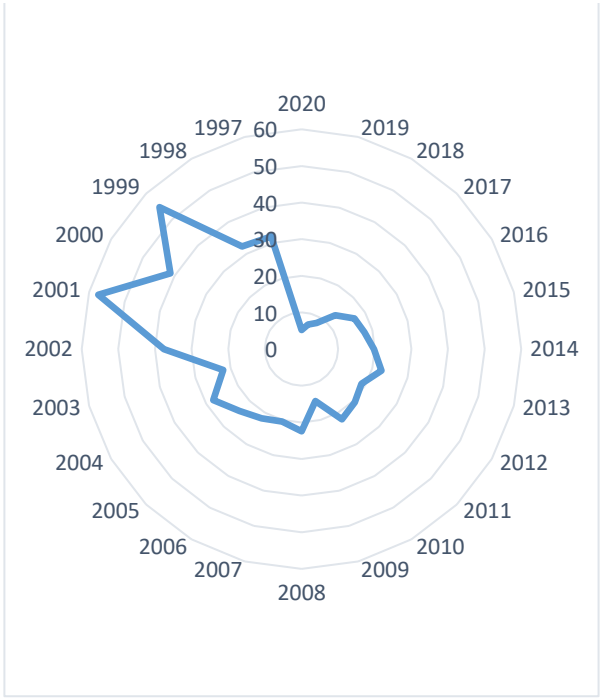


Fig.6. Place of data science articles in machine learning publications by years

As shown in Fig.6, the rate of data science concept among the publications related to machine learning was 30% in 1997, but this rate decreased to 5% in 2020. This expression can also be interpreted the other way around. In other words, the rate of Machine Learning concept among the publications related to the concept of data science was 30% in 1997, while this rate decreased to 5% in 2020. The assumptions about the cause of this change are also the subject of research of this review. This change can be explained by the emergence of data science, which is an interdisciplinary concept, in parallel with the development of computer technologies of the disciplines in which it is adopted as an approach. In addition to physics, chemistry and biology, which are basic sciences also education, ecology, industry, finance, materials science, meteorology and health are among the disciplines that data science has adopted as an approach. The intensive use of computer technologies in these different disciplines causes these disciplines to be divided into sub-disciplines and more research is done in these areas.

VI. CONCLUSION

Data science was previously used and associated with machine learning, computer science and data processing fields, which were mostly related to data science. With the rapid development of computer technologies, the increase in the data obtained from data sources and the emergence of new data sources has increased the use of data science with different disciplines. Areas that reveal big data are generally institutions and organizations serving in the fields of transportation, logistics, retail, public services, telecommunication, health, education, finance and media.

In the study, it is seen that from 1997 to 2020, studies on both data science and machine learning have increased rapidly. Despite this, it is seen that the publications that

include the concepts of data science and machine learning together have decreased from 30% in 1997 to 5% in 2020.

It is revealed that the decline in publications where data science and machine learning are handled together is due to the rapid increase in the use of large data obtained from many different disciplines and fields in data science studies in recent years, thanks to the developments in computer technologies.

With the developments in communication technologies, it is seen that bigger increases will occur in big data and companies that use big data with the help of data science will gain competitive advantage in their business and future planning thanks to data-based decision-making mechanisms.

REFERENCES

[1] H. Soliman, Mobile "IPv6: mobility in a wireless internet," Reading: Addison-Wesley, pp.4-6, 2004.

[2] N. Kushalnagar, G.Montenegro and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," pp.1-11, 2007.

[3] K. Nath, S. Dhar and S. Basishtha, "Web 1.0 to Web 3.0-Evolution of the Web and its various challenges," International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 86-89, 2014.

[4] Y.Gahi, M. Guennoun and H. T. Mouftah, "Big data analytics: Security and privacy challenges," In 2016 IEEE Symposium on Computers and Communication (ISCC), pp. 952-957, 2016.

[5] S. Sagiroglu and D. Sinanc, "Big data: A review," International conference on collaboration technologies and systems (CTS), pp. 42-47, 2013.

[6] R. Altunışık, "Büyük Veri: Fırsatlar Kaynağı mı Yoksa Yeni Sorunlar Yumağı mı?," Yildiz Social Science Review, vol.1, no.1, pp.45-76, 2015.

[7] F. J. Ohlhorst, "Big data analytics: turning big data into big money," John Wiley & Sons., vol.65, 2012.

[8] E. Aktan, "Büyük veri: Uygulama alanları, analitiği ve güvenlik boyutu," Bilgi Yönetimi vol.1, no.1, 2018.

[9] M. O. Gökalp, K. Kayabay, S. Çoban, Y. B. Yandık and P. E. Eren, "Büyük Veri Çağında İşletmelerde Veri Bilimi," 5th International Management Information Systems Conference, 2018.

[10] Y. Daşdemir and B. C. Kara, "Farklı İş Yükleri Altında NoSQL Sistemlerinin Performans Analizi," Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, vol.8, no.4, pp. 1466-1477, 2019.

[11] V. Dhar, "Data science and prediction," Communications of the ACM, vol.56, no.12, pp.64-73, 2013.

[12] N. Reid, "Statistical science in the world of big data," Statistics and Probability Letters, vol.136, pp.42-45, 2018.

[13] G. V. Rossum, "Python programming language," USENIX annual technical conference, vol. 41, pp.36, 2007.

[14] K.R. Srinath, "Python–The Fastest Growing Programming Language," International Research Journal of Engineering and Technology (IRJET), vol.4, no.12, pp.354-357, 2017.

[15] I. Stančin and A. Jović, A, "An overview and comparison of free Python libraries for data mining and big data analysis," 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, pp.977-982, 2019.

[16] https://bi-insider.com/posts/data-science-overview/, last access 21.12.2020.

[17] H. Nizam and S. S. Akın, "Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması," XIX. Türkiye'de İnternet Konferansı, 2014.

[18] D. Moher, L. Shamseer, M. Clarke, D. Ghersi, A. Liberati, M. Petticrew and L. A. Stewart, "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P)," Systematic reviews, vol.4, no.1, 2015.

[19] M. Chen, Y. Hao, K. Hwang and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," IEEE Access, vol. 5, pp.8869-8879, 2017.

[20] P. P. Gallego, J. Gagoand M. Landín, "Artificial neural networks technology to model and predict plant biology process," Artificial neural networks methodological advances and biomedical applications, pp.197-216, 2011.

[21] S. Sharma, S, "Markov chain monte carlo methods for bayesian data analysis in astronomy," Annual Review of Astronomy and Astrophysics, vol. 55, pp.213-259, 2017.

[22] R. J. Jadhav and U. U. Pawar, "Churn prediction in telecommunication using data mining technology," International Journal of Advanced Computer Science and Applications, vol.2, no.2, 2011.

[23] C. O. Hendren, X. Mesnard, J. Dröge and M. R.Wiesner,"Estimating production data for five engineered nanomaterials as a basis for exposure assessment", 2011.

[24] G. S. Swales and Y. Yoon, "Applying artificial neural networks to investment analysis," Financial Analysts Journal, vol.48, no.5, pp.78-80, 1992.

[25] R. A. Harshman, P. E. Green, Y. Wind and M. E. Lundy, "A model for the analysis of asymmetric data in marketing research," Marketing Science, vol.1, no.2, pp.205-242, 1982.

[26] K. L. Moin and D. Q. B. Ahmed, "Use of data mining in banking," International Journal of Engineering Research and Applications, vol.2, no.2, pp.738-742, 2012.

[27] W.Lin, Z. Wu, L. Lin, A. Wen and J. Li, "An ensemble random forest algorithm for insurance big data analysis," IEEE access, vol.5, pp. 16568-16575, 2017.

[28] J. Feng, L. D. A. Barbosa and V. Torres, U.S. Patent No. 9,262,517. Washington, DC: U.S. Patent and Trademark Office, 2016.

[29] B. W. Ang, F. L. Liu, E. P. Chew, "Perfect decomposition techniques in energy and environmental analysis," Energy Policy, vol.31, no.14, pp.1561-1566, 2003.

[30] M. P. Maloney, J. M. Suit, R. Rubel and F. M. Woodus, U.S. Patent No. 6,269,447. Washington, DC: U.S. Patent and Trademark Office, 2001.

[31] H. Chen, W. Chung, J. J. Xu, G.Wang, Y. Qin and M. Chau, "Crime data mining: a general framework and some examples," Computer, vol. 37, no.4, pp.50-56, 2004.

[32] V. Shapoval, M. C. Wang, T. Hara and H. Shioya, "Data mining in tourism data analysis: inbound visitors to Japan," Journal of Travel Research, vol.57, no.3, 310-323, 2018.

[33] K. Hardy and A. Maurushat, "Opening up government data for Big Data analysis and public benefit," Computer law & security review, vol.33, no.1, pp.30-37, 2017.

[34] J. Han, E. Haihong, G. Le and J. Du, "Survey on NoSQL database," In 2011 6th international conference on pervasive computing and applications," pp. 363-366, 2011.

[35] R. Mikut and M. Reischl, "Data mining tools," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol.1, no.5, pp.431-443, 2011.

[36] E. Seyyarer, F. Ayata, T. Uçkan and A. Karcı, "Derin Öğrenmede Kullanilan Optimizasyon Algoritmalarinin Uygulanmasi Ve Kıyaslanması," Bilgisayar Bilimleri, vol.5, no.2, pp.90-98, 2020.

[37] A. Mathew, P. Amudha and S. A. Sivakumari, "A Review on Finger Vein Recognition Using Deep Learning Techniques," National Conference on Machine Learning and Its Applications, 2020.

[38] A. Parvat, J. Chavan, S. Kadam, S. Dev and V. Pathak, "A survey of deep-learning frameworks," International Conference on Inventive Systems and Control (ICISC), pp.1-7, 2017.

[39] C. W. Tsai, C. F. Lai, H. C. Chao and A. V. Vasilakos, "Big data analytics: a survey," Journal of Big data, vol.2, no.1, pp.1-32, 2015.

Comparative Analysis of Modified-P&O and Modified-PSO Based MPPTs for Partial Shading Conditions

Semih ÇAM
ECE Department
National Defense University,
Turkish Military Academy
Ankara, Turkey
secam@kho.edu.tr

Haluk GÖZDE
ECE Department
National Defense University,
Turkish Military Academy
Ankara, Turkey
hgozde@kho.edu.tr

Mustafa AKTAŞ
EEE Department
Ondokuz Mayis University
Samsun, Turkey
mustafa.aktas@omu.edu.tr

***Abstract*— Partial shading conditions are one of the most important problems of photovoltaic systems today. It not only reduces efficiency, but also damages photovoltaic panels. Various methods have been proposed in the literature to solve this problem. Line power synchronization with the switching matrix is particularly effective. However, the panels are difficult and expensive to apply in terms of both connection and control. In this study, modified Perturb and Observe (P&O) and Particle Swarm Optimization (PSO) algorithms were developed with additional algorithms and compared with each other to increase Maximum Power Point Tracker (MPPT) efficiency in a less complicated and cheaper way in the case of partial shadowing. In the developed P&O algorithm, the scanning frequency providing the maximum power for the solar radiation values coming to the photovoltaic panel was determined. For the PSO algorithm, two development methods have been applied. In the first method, corrections are made by taking the average of the errors that occur in Duty Cycle. In the second method, curve fitting was developed from the power differences of panels connected in series and Duty Cycle differences. From the results obtained, step response analysis, it was seen that the modified PSO algorithm significantly increases the MPPT process.**

Keywords*— *Photovoltaic System, MPPT, P&O Algorithm, PSO Algorithm, Partial Shading Condition.

I. INTRODUCTION

In the recent years, energy production from clean, renewable, efficient and environmentally friendly sources has become one of the important research areas. Both the carbon emission caused by fossil resources and the decreasing of these resources increase the need for renewable energy day by day. Solar energy, one of the renewable energy sources, finds more application areas due to its easier and cheaper availability. The efficiency of the power of the photovoltaic panel obtained from solar energy is important. Problems such as solar radiation change of solar energy, temperature change, partial shade on the photovoltaic panel, clouding situation, sudden change of solar radiation on the panel, events occurring in the atmosphere, humidity, reduce the efficiency of the photovoltaic panel. Traditional algorithms, smart algorithms and heuristic algorithms have been proposed to solve these problems. Traditional methods are generally techniques such as Perturb & Observe (P&O), incremental conductance (INC), open circuit voltage technique, short circuit current technique. Intelligent algorithms are methods such as fuzzy logic, artificial neural network, and genetic algorithm. Heuristic algorithms are methods that mimic events occurring in nature such as particle swarm optimization, cuckoo algorithm, artificial bee colony algorithm, firefly algorithm, flower pollination and mixed frog splash algorithm. Classical algorithms show good results in terms of efficiency under constant sunlight conditions in maximum power tracking. However, under partial shading conditions, problems such as the formation of more than one peak of the P&O algorithm and the jamming of the algorithm at local maximum points were observed [1]. On the other hand, it was observed that the PSO method reached a global maximum with low fluctuations and gave good results against the same problems [2]. In the literature, comparison of PSO method with methods such as P&O [3], incremental conductance [4], hill climbing and Fibonacci search [5], fuzzy logic [6], cuckoo algorithm [7] has been made. In addition, the cuckoo and PSO methods were compared with P&O [8], and in a later study, the classical P&O method was developed and compared with the cuckoo and PSO methods [9]. In addition, hybrid studies have been performed by combining PSO method with other methods. The PSO method was compared to standard PSO using natural selection [10], VPSO-LF based on Levy flight speed [11], voltage window search and P&O [12]. In this study, classical P&O and heuristic PSO algorithms were used in Maximum Power Point Tracking (MPPT) design in order to increase efficiency in the case of partial shading in an 83 W solar energy system, and then software improvements were made in the algorithms. For the case of shading conditions, their performances were reviewed were presented.

In Section II, problems are mentioned for partial shading. In Section III, the developed P&O algorithm is discussed. In Section IV, the contents of the PSO and two different development methods are given. In Section V, the study for two different cases under partial shading conditions is mentioned. In Section VI, the result is given and discussed.

II. PARTIAL SHADING PROBLEM

Photovoltaic panels are connected in series or in parallel to obtain the desired voltage and power. Shading occurs on some panels due to events occurring in the atmosphere or the presence of an object between the panel and the sun. In this case, while the solar radiation value on one panel is 800 W/m^2, the solar radiation value on another panel can be 200 W/m^2. As a result, the panel, which starts to act as a load, can overheat and deform. The shading condition that occurs in the panels or the difference in the value of solar radiation in each panel is called partial shading.

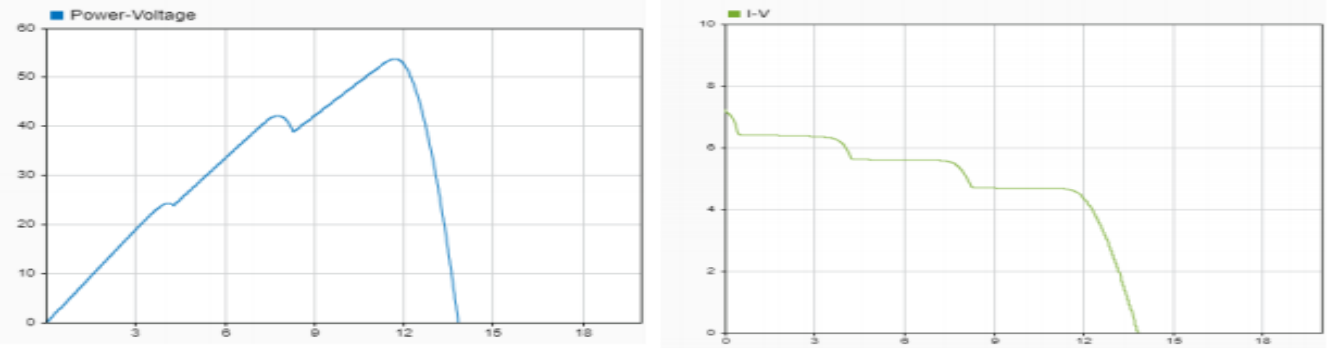


Fig. 1. Power-Voltage Curve (Left), Current-Voltage Curve (Right).

The increase in the level of the peaks increases with the variability of sunlight. While the peak value with the highest power value from more than one curve in the power curve at the output of the panel is the global peak value, the other peaks are called the local peak value as presented in Fig.1.

While all algorithms can work with high efficiency under fixed solar radiation values, algorithms operating under partial shading conditions are smarter and must distinguish these peaks. Therefore, it has been observed that classical algorithms do not give very successful results under these irradiation conditions. In this study, both the classical P&O algorithm and the PSO algorithm were software modified and developed according to the PV-system presented in Fig.2.

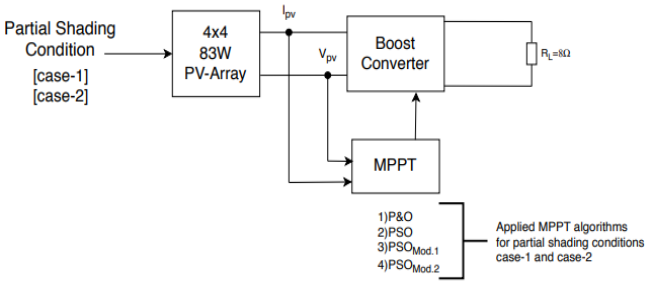


Fig. 2. Improved MPPT with P&O and PSO.

III. Improved P&O Algorithm

The P&O algorithm is one of the most used methods in the literature because it is easy to apply. In this algorithm, by looking at the output power of the photovoltaic panel, the Duty Cycle value is changed and the value of the power is changed. In classical P&O algorithms, Duty Cycle change is increased or decreased with a fixed DPWM value.

However, in this method, the efficiency of the system changes according to the solar radiation coming to the solar panel. For example, a set DPWM value can do 1000 W/m2 MPPT, but 500 W/m2 MPPT cannot. To solve this problem, the algorithm has been modified according to the flow diagram in Fig.3.

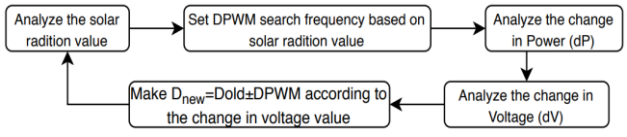


Fig. 3. Improved P&O Algorithm Flow Diagram.

IV. Improving PSO Algorithm

Particle swarm optimization is a heuristic optimization algorithm commonly used in MPPT applications. Details of the algorithm can be viewed [13]. In this study, the PSO algorithm was run according to the parameters in Table.1. It was observed that the results obtained with the standard PSO algorithm were insufficient in the case of partial shadowing and the following modifications were made.

TABLE I. PSO Parameters.

Parameters	*Values*
$x_{_min}$	0.1
$x_{_max}$	0.69
C_1	1.2
C_2	1.6
$w_{_min}$	0.1
$w_{_max}$	1.8

A. *Modification - 1*

In the PSO algorithm, it has been observed that there is a different amount of deviation from the maximum power value for each partial shading situation with the normal algorithm. In order to make a common correction, in the intuitive PSO method, the average value of the Duty Cycles obtained in each case with the Duty Cycle, where the theoretical maximum power is reached, is calculated as follows. Later, this value was used as the correction value according to the algorithm in Fig.4.

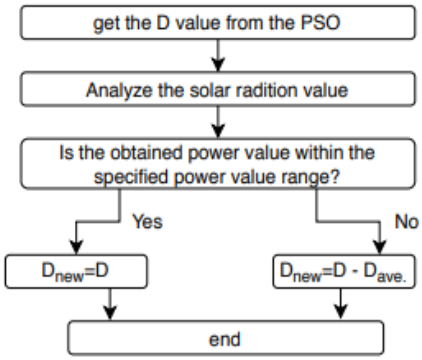


Fig. 4. Flowchart of the first modification for PSO Algorithm.

B. *Modification – 2*

In a 4x4 solar panel, if all panels come with the same irradiance value, the power values of the panels in series will be the same, but if the same irradiance value is not present,

the power value of each of the serial panels will be different. In this correction, it is aimed to reduce this difference to 0 W as presented in Fig.5.

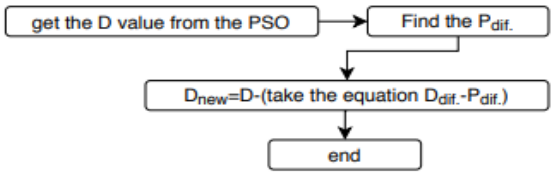


Fig. 5. Flowchart of the second modification for PSO Algorithm.

V. Using the Template

An 83 W array with 4x4 = 16 TCT connections was used to examine partial shading conditions. In the study, 2 different partial shading conditions for different mixed shading of the photovoltaic panel were examined.

A. *Case 1*

In Case-1, it has been observed that the P&O algorithm, PSOMod.2 and PSOMod.1 are very close to the theoretical maximum value, and the heuristic PSO works with 89% efficiency. When the settling time comparison is made, it is seen that the P&O algorithm is constantly scanning the ideal Duty Cycle value because it has multiple peak values, so it sits at a much higher time. For Case 1, PSOMod.2 and PSOMod.1 were found to work more efficiently. The classical P&O algorithm, where the improved PSO algorithms and heuristic PSO algorithms see higher values due to the software complexity and number of iterations, higher values has been observed due to the stop time set to 2.

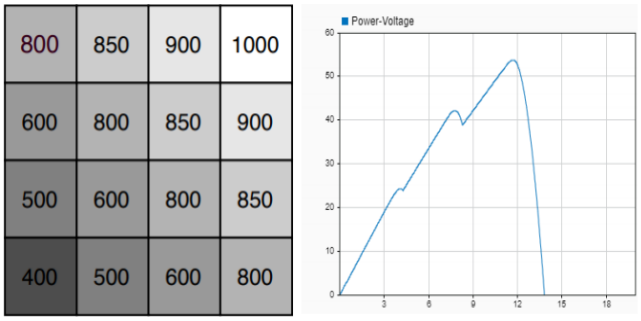


Fig. 6. Shading condition for Case-1 (Left), P-V curve for Case-1 (Right).

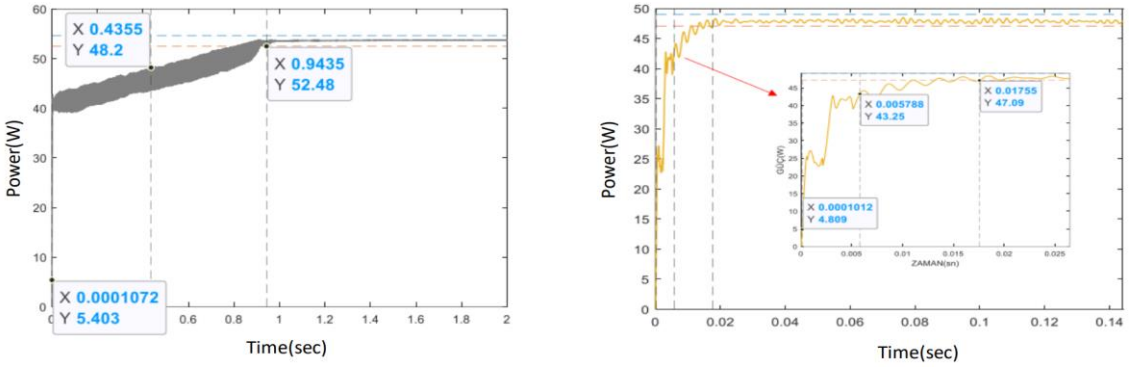


Fig. 7. Power curve for P&O (Left), Power curve for the PSO (Right).

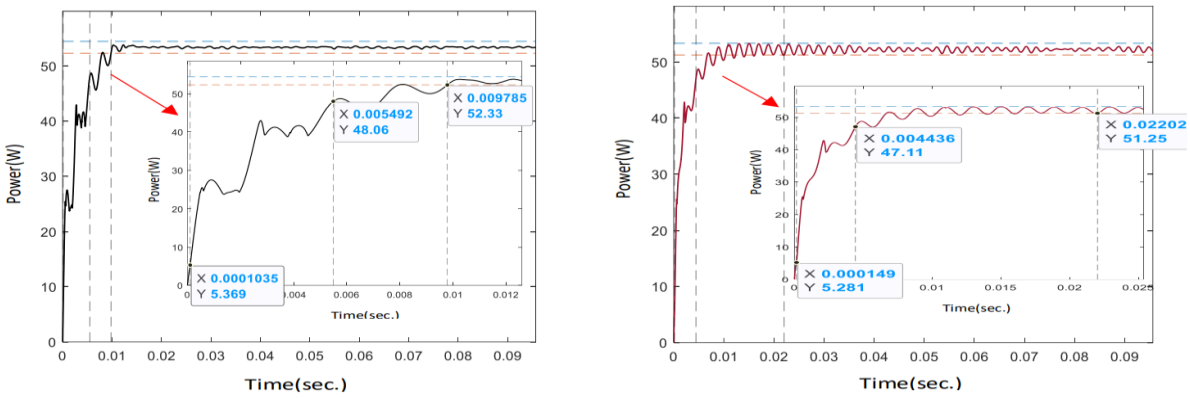


Fig. 8. Power curve for the $PSO_{Mod.1}$ (Left), Power curve for $PSO_{Mod.2}$ (Right).

TABLE II. P&O, PSO, $PSO_{MOD.1}$, $PSO_{MOD.2}$ VALUES FOR CASE-1.

	P&O	***PSO***	***PSOMod.1***	***PSOMod.2***
Theoretical Power Value	53.70	53.70	53.70	53.70
The Obtained Power Value	53.55	48.05	53.41	52.34
Efficiency (%)	99.74	89.38	99.44	97.91
Settling Time (s)	0.94	0.01	0.01	0.022
Rising Time (s)	0.43540	0.00569	0.00538	0.00427
Real Simulation Time (CPU Time)	473.03	589.22	693.00	680.25

B. *Case 2*

In Case 2, it has been observed that P&O, PSO, PSOMod.2 and PSOMod.1 are very close to the theoretical maximum value, but due to the length of the scan time of the P&O algorithm, the settling time is 100 times larger than the other algorithms. For Case-2, the P&O algorithm has been found to work less efficiently. The fact that the real simulation time of all 4 methods are close to each other is due to the fact that the stop time of the P&O algorithm is 2 times compared to the others. It has been observed that if the stop time of the P&O algorithm is the same as the others, it does not fall within the specified power range for the residence time.

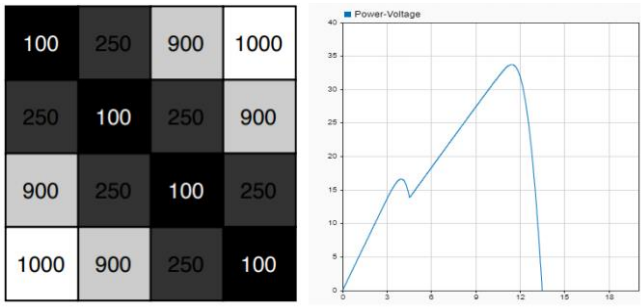


Fig. 9. (Left) Shading condition for Case-2, (Right) P-V curve for Case-2.

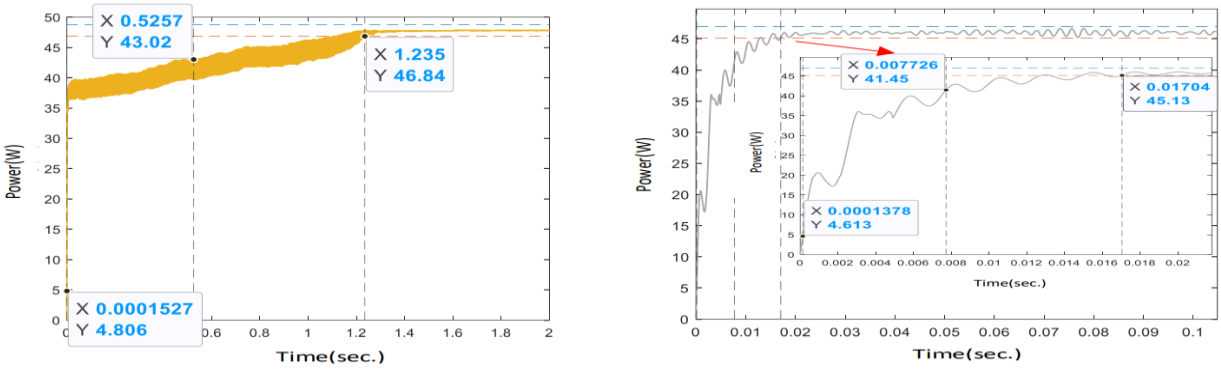


Fig. 10. Power curve for P&O (Left), Power curve for the PSO (Right).

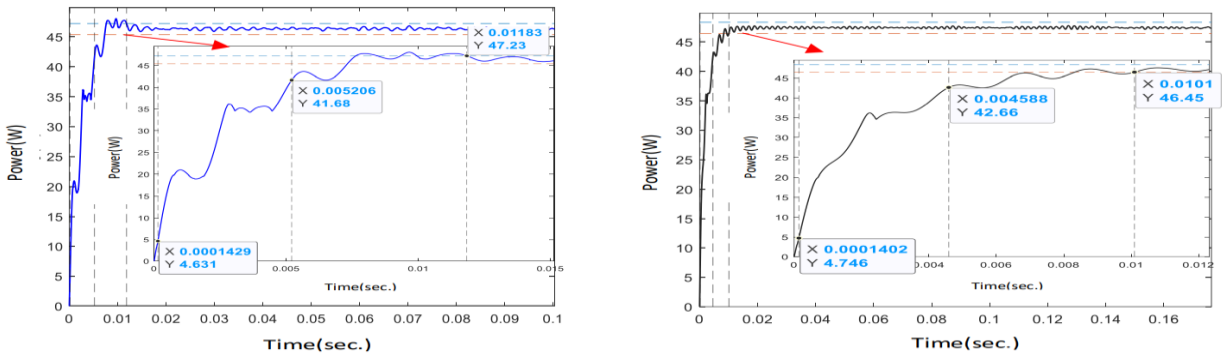


Fig. 11. Power curve for the $PSO_{Mod.1}$ (Left), Power curve for $PSO_{Mod.2}$ (Right)

TABLE III. P&O, PSO, $PSO_{MOD.1}$, $PSO_{MOD.2}$ VALUES FOR CASE-2.

	P&O	***PSO***	***PSOMod.1***	***PSOMod.2***
Theoretical Power Value	47.85	47.85	47.85	47.85
The Obtained Power Value	47.81	46.05	46.3	47.41
Efficiency (%)	99.81	96.22	96.74	99.01
Settling Time (s)	1.23	0.02	0.01	0.01
Rising Time (s)	0.52550	0.00760	0.00504	0.00445
Real Simulation Time (CPU Time)	473.73	589.37	584.42	670.51

VI. CONCLUSION

Partial shading is a major problem for photovoltaic systems, this not only reduces efficiency but also damages photovoltaic panels. In this study, it is suggested that MPPT methods based on modified P&O and PSO algorithms improve MPPT efficiency in a less complex and cheaper way in the case of partial shadowing. At the end of the simulations performed for the two partial shading conditions called State-1 and Case-2, it was seen that the PSO algorithm operates with 10% and 3% lower efficiency for both cases. On the other hand, it was observed that the sitting time of the P&O algorithm was approximately 94 times and 123 times longer for both cases. Although the modified PSO methods provide approximately 99% efficiency and output power performances, the method obtained with PSOMod.2 significantly increases the transient response of the MPPT system. In the future study, the authors plan to apply these methods and then analyze and compare various insolation conditions. On the other hand, the authors suggest that the

performance of the proposed methods can be improved by using newer and more powerful heuristic optimization algorithms.

REFERENCES

[1] P. R. L, S. Sekhar Dash, ve R. K. Dwibedi, "Design and Implementation of Perturb & Observe MPPT Algorithm under Partial Shading Conditions (PSC) for DC-DC Boost Converter by Simulation analysis", içinde 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE), Keonjhar, Odisha, India, Tem. 2020, ss. 1-4

[2] O. Kircioglu, M. Unlu, ve S. Camur, "The PSO Based Global Maximum Power Point Tracker", içinde 2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, Haz. 2019, ss. 1-4

[3] H. Chaieb ve A. Sakly, "Comparison between P&O and P.S.O methods based MPPT algorithm for photovoltaic systems", içinde 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, Ara. 2015, ss. 694-699

[4] H. Rezk, A. Mera, ve M. A. Tolba, "Performance Analysis of Solar PV System under Shading Condition", içinde 2020 International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russia, Mar. 2020, ss. 1-5

[5] M. Miyatake, M. Veerachary, F. Toriumi, N. Fujii, ve H. Ko, "Maximum Power Point Tracking of Multiple Photovoltaic Arrays: A PSO Approach", IEEE Trans. Aerosp. Electron. Syst., c. 47, sy 1, ss. 367-380, Oca. 2011

[6] O. B. Belghith, L. Sbita, ve F. Bettaher, "MPPT Design Using PSO Technique for Photovoltaic System Control Comparing to Fuzzy Logic and P&O Controllers", EPE, c. 08, sy 11, ss. 349-366, 2016

[7] Z. Gümüş ve M. Demirtaş, "Fotovoltaik Sistemlerde Maksimum Güç Noktası Takibinde Kullanılan Algoritmaların Kısmi Gölgeleme Koşulları Altında Karşılaştırılması", Journal of Polytechnic, May. 2020

[8] S. Pant ve R. P. Saini, "Comparative Study of MPPT Techniques for Solar Photovoltaic System", içinde 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON), ALIGARH, India, Kas. 2019, ss. 1-6

[9] B. S. Varun Sai, S. A. Khadtare, ve D. Chatterjee, "An Improved MPPT Technique Under Partial Shading Condition Using Simple P&O Algorithm", içinde 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE), Keonjhar, Odisha, India, Tem. 2020, ss. 1-6

[10] Y. Hu, J. Lu, Y. Deng, ve Z. Zhang, "Mppt Algorithm Based On Particle Swarm Optimization With Natural Selection", program adı: 2015 3rd International Conference on Machinery, Materials and Information Technology Applications, Qingdao, China, 2015

[11] R. Motamarri ve B. Nagu, "GMPPT by using PSO based on Lévy flight for photovoltaic system under partial shading conditions", IET Renewable Power Generation, c. 14, sy 7, ss. 1143-1155, May. 2020

[12] G. Liu, J. Zhu, H. Tao, W. Wang, ve F. Blaabjerg, "A MPPT Algorithm based on PSO for PV Array Under Partially Shaded Condition", s. 5, 2019.

[13] R. Eberhart ve J. Kennedy, "A new optimizer using particle swarm theory", içinde MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, ss. 39-43

Analysis Of Cracks In Photovoltaic Module Cells From Electroluminescence Images By Deep Learning

Miktat AKTAŞ
R&D Department
GTC Gunes Sanayi ve Ticaret A.S.
Adiyaman, Turkey
0000-0002-0731-5668

Ferdi DOĞAN
Head Of IT Department
Adıyaman University
Adiyaman, Turkey
0000-0002-9203-697X

İbrahim TÜRKOĞLU
Software Engineering Department
Fırat University
Elazığ,Turkey
0000-0003-4938-4167

***Abstract*— With the spread of solar power plants and investors becoming more conscious, the demand for quality and efficient solar panels is increasing day by day. Creating quality products is possible by minimizing the errors in production processes. The efficiency and quality of solar panels is directly proportional to the efficiency and quality of the solar cell used in the panel.In this study, it aims to provide useful contributions to 3 different steps in the solar panel production process: firstly, the quality control of the solar cell to be used before production, secondly, the detection and replacement of cells having cracks in the production process, and the classification of the panels, finally the detection of performance losses and cause after the panel production.In this study, the deep learning network models and datasets used in the literature were first examined and then the images obtained from the Electroluminescence devices were analyzed with deep learning. Alexnet model, one of the deep learning networks, was used. As a data set, a total of 876 60-cell solar panels with a minimum of 1 and a maximum of 28 cracks were obtained with a resolution of 4730x2883. As a result, 79.4 percent crack detection rate was achieved with the proposed method.**

Keywords—Solar Cell Cracks, Deep Learning, crack detection, Alexnet

I. INTRODUCTION

In our study, small or large cracks and fractures in the PV solar panel can be detected by in-depth learning networks from the PV solar panel images obtained with EL, and these can be made faster, and identical to the detection by experts in this field. It is thought that the working time will benefit 3 steps for solar panel production. At the beginnig of the PV panel production; quality control of the solar cells to be used in the panel. During PV panel production; detection and replacement of broken and cracked cells. Visual inspection and classification of panels. After PV panel production; can be used in determining performance losses and related reasons. Apart from these three benefits, for solar cell manufacturers it is thought that important contributions will be made in the process of quality-control and classification of produced cells.

In this study, we think that we have put forward an exemplary study in the fields of autonomous cell crack detection, deep learning and image processing for PV solar cell manufacturers, PV solar panel manufacturers, research centers, institutes and consulting companies operating in this field.

GTC GUNES SANAYI VE TICARET A.S. TURKEY/ADIYAMAN

II. RELATED WORKS

In the study of Alexander Bartler et al. (2019) with VGG-16 and Resnet, cells having 120x120 px resolution were studied, and cells are a classified as as damaged cell or good cell in this study. The error rate of 50% was reduced to 12.96% with the studies (no oversampling, augmentation). [2]

Julen Balzategui et al. Studied with the same data set in 2019 and 2020. A classification was made as a damaged cell or a good cell, they used 4 different CNN algorithms in their study in 2019 and stated that they could detect damaged cells at a maximum of 85%, they stated that the same data set could be worked with FCNN under the heading of future studies in 2019, and in their work published in 2020, FCNN were worked with. According to the results of the 2020 study, the rate of detecting cracks as low as 28% was achieved. [3] [4]

Mathis Hoffmann et al. (2019) studied using classical image processing and feature extraction techniques on detecting modules and cells from EL photos. A result that could detect the cells accurately was obtained at a rate of 82.5%. [5]

Din-Chang Tseng et al. (2015) studied using image processing techniques to detect finger interruptions in polycrystalline cells obtained with EL. Two classes, interrupted finger and noninterrupted finger, were studied, and an accuracy rate of 99.07% and 99.58 was achieved. [6]

John et al. (2016) compared the features of (EL) and Infrared Thermal Camera (IRT), it was stated that IRT-based scanning was more useful and efficient, in the study, the locations of the problematic panels or cells could be determined by correlating the images taken with the aid of drones with Gps data. [7]

Mahmoud Abdelhamid et al. (2014) conducted an extensive literature study on techniques for detecting fractures in silicon-based cells and made comparisons, claiming that the most successful studies could be done with PL imaging, according to the price of the equipment used, image quality and the rate of finding small cracks. [1]

Said Anvar and Mohd Zaid Abdullah (2014) carried out a comparative study on the division of cracks in solar cells using edge detection algorithms. [8]

Sergiu Deitscha et al. (2018) conducted a study on dividing PV Module cells from EL images. They were able to perform segmentation with a success rate of 97.23% [9], this study constituted a step for the later (2019) classification studies of damaged cells. In their study in

2019, they used both SVM and CNN. They achieved the rate with an average accuracy of 82.44% with SVM and 88.42% with CNN. At the same time, they were able to calculate the percentage effect of the cracks on the energy production performance of the panel. [10]

Unlike Sergiu et al. (2019), Haiyong Chen et al. (2019) studied images taken with high-resolution cameras, not from EL photos, into RGB channels and split the cell image into pieces. They claimed that they designed and optimized the deep learning network based on their own experiences, without being dependent on any model, and as a result, Sergiu and his friends achieved an accuracy rate of 94.30% with an accuracy rate of 6 points exceeding the 88.42%. Considering that the types and sizes of the entrance photographs are different, we can say that such a comparison is not rational. In addition, they divided the classifications into 7 different classes as well, broken finger, scratch, thick line, dirty cell, color difference coating stains, apart from making them as damaged and smooth cells.[11]

Mantel Claire et al. (2019) tried to detect damages in solar panels (finger outage and fractures as A, B and C according to their size) using machine learning approaches. They stated that SVM and Random Forest showed almost the same success, but the number of false detections in RF was higher. They claim that they achieved a very successful result with SVM with an accuracy rate of 99.7%. [13]

III. METHODOLOGY

A. *Requirement*

Cracked or damaged solar cells cause efficiency loss in the production and consequently an increase in production costs. Microcracks and defects not only reduce cell productivity in that area, but also reduce cell reliability. [1]

A damaged cell or group of cells can cause hot-spot heating problems that occur when the operating current in a panel exceeds the reduced short circuit current of the fault cell. Here, the cell is forced into reverse current and must dissipate this accumulated power. Indeed, if the dissipation force is large enough, this reverse-polar cell can overheat and melt the solder or cause the back-sheet to deteriorate (Figure 1). Hot-spot cells show low parallel resistance when reverse current performance is limited by current, or high parallel resistance when reverse current performance is limited by voltage. In either case, the cell may experience hot-spot problems, but in different ways. [14]

PV solar cell cracks and production errors cause performance loss up to 60% depending on the location and size and number of cracks [1]

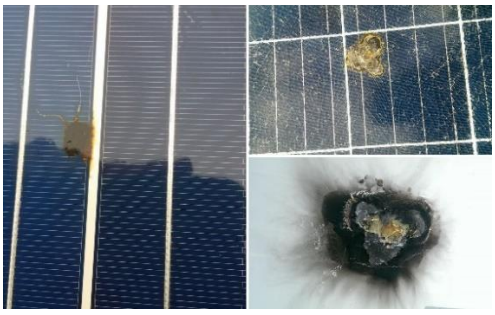


Figure 1. Hotspot's effect on Solar Panel [14]

B. *Dataset*

The photos are taken from the images obtained from the EL device used in the production line of GTC Gunes Sanayi ve Ticaret A.S. Among the 82456 solar panel images, 7500 images were selected randomly, among which the broken cells were confirmed by expert operators during production.As can be seen in Table I, a total of 876 pieces of 4730x2883 pixel images with broken images were selected. Since each solar panel contains 60 cells, a total of 52560 cells were studied.689 EL images (2222 cracks) were used for training, and 187 EL images (451 cracks) randomly selected were used for testing.

TABLE I. DATASET FEATURES

Training	Test
689 Solar Panels	187 Solar Panels
41340 Solar Cells	11220 Solar Cells
2222 Cracks	451 Cracks

C. *Material*

Training and tests were carried out in Matlab in 2019b version. Image Labeler and Deep Network Designer tools were used. Since Deep Learning networks requires a high-spec and high-capacity GPU, a computer have features like Win-10 64-Bit OS, i7-2.6 GHz CPU,24 GB Ram and NVIDIA Geforce GTX 960M graphic card with dedicated 4 GB Memory.

D. *Method*

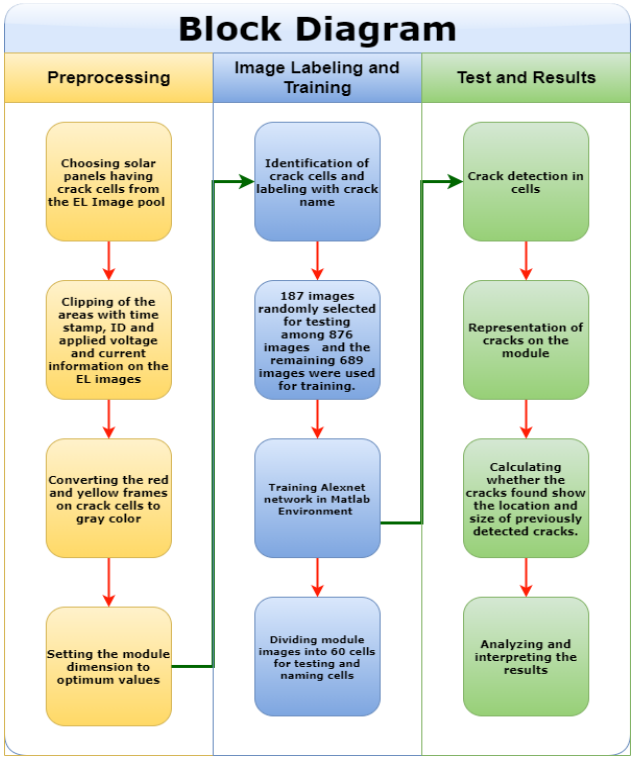


Figure 2. Block Diagram

The block diagram given in Figure 2 is detailed in the following items.

1) Pre-processing

Before the images were processed, the Module ID numbers on the image were clipped with the help of the Matlab program due to the company's data privacy policy.

The red or yellow warning lines drawn by the software used in the EL device in cell cracks were turned into gray with the help of the Matlab program to avoid any problems in training or testing.

Numbers indicating row and column numbers in the module are clipped from the image. The names of the image files have been changed due to the company's data privacy policy.

Although the dimensions of the EL images were very close to each other, the width or height values differed by a few pixels, all of the images were adjusted to the optimum height and width values to eliminate this difference. This value was determined as 4730x2883.

Preprocessing

2) Image Labeling and Training

With the Matlab Image Labeler tool, the solar panel images with a minimum of 1 and a maximum of 28 broken images were tagged. While labeling, as shown in Figure 3 in the3first study, if it includes the broken ribbons, we made the labeling in the size of a red frame, but when we found that the success rate was very low, such as 30% in the test results, we made these types of cracks in the form of two labeling as in the blue frame and We observed that the success rate in the tests increased.

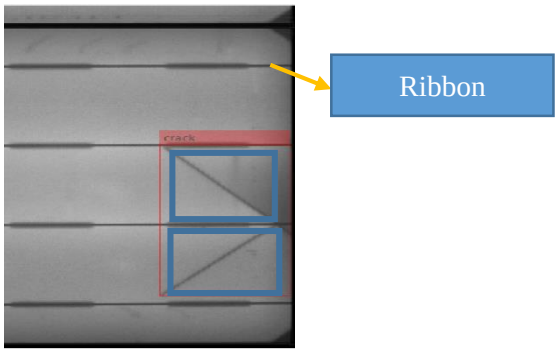


Figure 3. Labeled Solar Cell

79% of the images were used for training. These images contain a total number of 2222 broken cells. As can be seen in Table I, 689 images were used. The following parameters' values in Table II were used for training dataset:

TABLE II. PARAMETERS VALUES

MaxEpochs	4
MiniBatchSize	16
Classes	'crack‘,’background’
NegativeOverlapRange	[0 0.3]
PositiveOverlapRange	[0.85 1]

In our study, we arranged the number of classes from 1000 in the Softmax layer to 2.

Alexnet, one of the deep learning networks, was used in the training and testing. In the Imagenet LSVRC-2010 competition attended by Alex Krizhevsky and colleagues, they classified 1.2 million high-resolution images in 1000 different classes and achieved error rates of 37.5% and 17.0%, significantly better than the previous technology. Alexnette used a neural network with 60 million parameters and 650,000 neurons. They used the GPU application to make the training faster. They also entered the ILSVRC-2012 competition with a different variant of this model and reached a top-5 test error rate of 15.3% compared to 26.2% for the second best entry. [16]

In the current architecture of Alex-net, it consists of 25 layers in total as shown in Figure 4 and Figure 5. In the input layer, 2227x227x3 images are used, after which 5 convulation, 3 pooling (maxpool), 7 Relu, and 3 fully connected layers, 2 nomalization, softmax and output layers are formed.

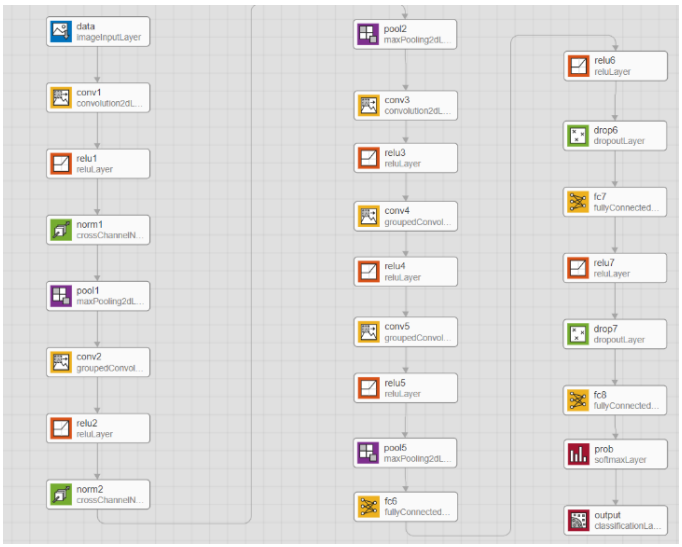


Figure 4. AlexNet Diagram [12]

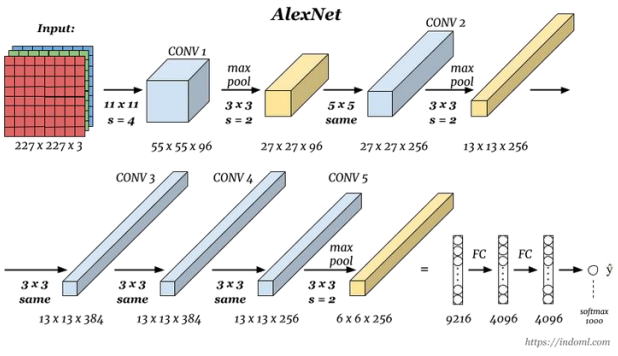


Figure 5. Alexnet Layers in Matlab

3) Test

21% of the images were used for training. As can be seen in Table I, 187 images were used. There are a total of 451 cracks in these panel images. During the test process, when the panel images were first used as a single input, then it

was seen that the rate of cracks was low, to increase this ratio the panel image was divided into 60 pieces with the name of "ModuleName_row_column.jpg" as shown in Figure 6 and tested in that way. Image dimensions were 470x 475x 3 pixels. Then, coding was done in Matlab program to display the test results on a 60-cell image and it was successful. In this way, a higher rate of fracture detection was achieved.

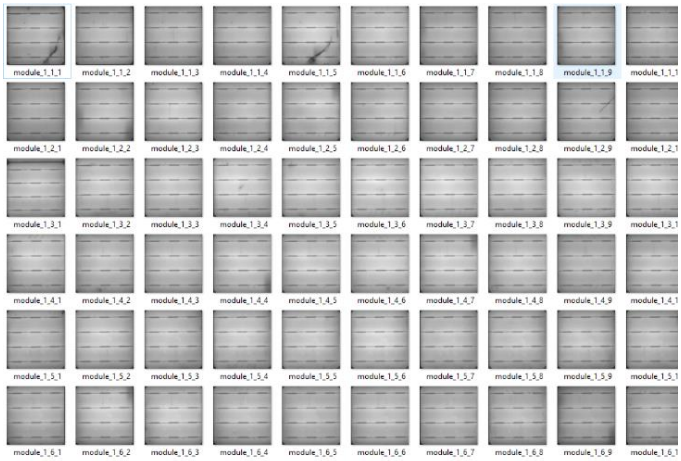


Figure 6. Cell-based fragmented images of a 60-cell solar panel

4) Calculation

The confusion matrix was used for performance calculations. A method as in Table III was followed and the results were calculated based on the precision, sensitivity and F-Score formulas. The formulas are detailed and explained below.

a) Precision

Precision is the ratio of the number of True Positive samples estimated as class 1 to the total number of samples estimated as class 1. [15]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

b) Recall

It is the ratio of the number of correctly classified positive samples to the total number of positive samples. [15]

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

c) F-Score

The precision and recall criteria alone are not sufficient to make a meaningful comparison. Evaluating both criteria together gives more accurate results. For this, the f-criterion has been defined. The F-criterion is the harmonic mean of precision and recall. [15]

$$\text{F} - \text{Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Precision} + \text{Recall}}$$

IV. RESULTS AND DISCUSSION

Training duration was 12-14 hours on average, each 60 cells are tested in 2 minutes and 24 seconds.

In the result determination, it was observed that the same crack is marked 3-4 times from it's different parts. In such cases, the number of cracks found was used in the calculation as 1.

In some cell cracks with cracks, it was observed that the dimensions were not the same. In such cases, it was accepted as positive because it fulfills the criteria that the crack and its location were determined.

When looking at the confusion matrix results in Table III, it is seen that the precision is relatively high. Considering that crack or defective cells should not be overlooked in test stations in panel production line, the point that should be emphasized is that the number of TP (True Positive) should be maximized. Of course, it is an indisputable fact that the rate of false detection should also be reduced.

Considering the results in Table III , It is understood that work should be done to reduce the number of false crack detection. It has been observed that these false detections are mostly caused by the strips on the cell, called ribbon. The "PositiveOverlapRange" value, which is the most important parameter in detecting false cracks during the test process, was changed from [0.5 1] to [0.85 1]. Increasing this rate is thought to reduce the rate of detecting false cracks, but slightly causes some fractures to be undetected. The number of false detected cracks is 589 as can be seen in Table III, most of them were ribbons (Figure 7) or gap between cells. To reduce false detections, we may increase the samples in dataset for future studies.

TABLE III. CONFUSION MATRIX

		Real	
		Crack	Not Crack
Prediction	Crack	359	589
	Not Crack	92	

TABLE IV. TEST RESULTS

Calculation Method	Result
Precision	0,794
Recall	0,376
F-Score	0,510

.

Although the rate of detecting ribbons as crack was reduced, the f-score and precision rate remained low, as seen in Table IV. At the same time, the high number of cells not having cracks is thought to be effective in the high rate of false crack detection. Test duration is one of the most important parameters on production lines of solar panels, so computers with higher GPU capacity can be used to achieve shorter test times.

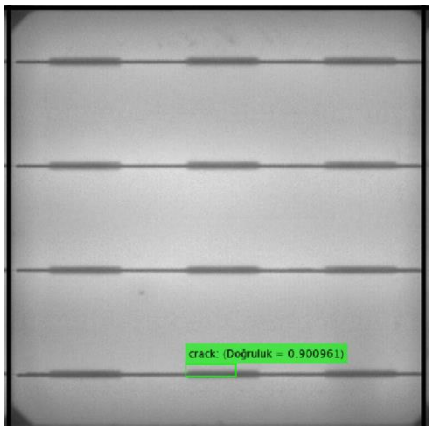


Figure 7. False Crack Detection

V. CONCLUSION

In this study, a detailed literature search was made on crack detection on PV solar cells and panels, and similar studies previously conducted on the detection of PV cell cracks were examined and a comprehensive application was carried out. Within the scope of the application, a study was carried out not only on whether there is a crack in the cell, but also the location and size of the crack.

The rate of detecting crack was 79.42%. Since it is important not to miss the cracks during PV solar panel production, it is important in this study whether the cracks can be detected or not, and a promising result has been obtained with the optimization studies.
In the next studies, training and tests in different sizes will be done in different networks to work on reducing the rate of false detections.

ACKNOWLEDGMENT

This study is funded by GTC Gunes Sanayi and Ticaret A.S. I am very grateful to CEO of GTC Gunes Sanayi ve Ticaret A.S, Ayşe Çiğdem BESEN for her permission to use required dataset and her support during my studies.

REFERENCES

[1] Abdelhamid ,M., Singh, R., Omar, M.(2014) Review of Microcrack Detection Techniques for Silicon Solar Cells. IEEE Journal Of Photovoltaics 2014,4

[2] Bartler, Alexander & Mauch, Lukas & Yang, Bin & Reuter, Michael & Stoicescu, Liviu. (2018). Automated Detection of Solar Cell Defects with Deep Learning. 2035-2039. 10.23919/EUSIPCO.2018.8553025.

[3] Balzategui, J., Eciolaza, L., Arana-Arexolaleiba, N., Altube, J., Aguerre, J.-P., Legarda-Ereno, I., & Apraiz, A. (2019). Semi-automatic quality inspection of solar cell based on Convolutional Neural Networks. 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). doi:10.1109/etfa.2019.8869359

[4] Balzategui, J., Eciolaza, L., & Arana-Arexolaleiba, N. (2020, January). Defect detection on Polycrystalline solar cells using Electroluminescence and Fully Convolutional Neural Networks. In 2020 IEEE/SICE International Symposium on System Integration (SII) (pp. 949-953). IEEE.

[5] Hoffmann, Mathis & Doll, Bernd & Talkenberg, Florian & Brabec, Christoph & Maier, Andreas & Christlein, Vincent. (2019). Fast and Robust Detection of Solar Modules in Electroluminescence Images. 519-531. 10.1007/978-3-030-29891-3_46.

[6] Tseng, D. C., Liu, Y. S., & Chou, C. M. (2015). Automatic finger interruption detection in electroluminescence images of multicrystalline solar cells. Mathematical Problems in Engineering, 2015.

[7] Tsanakas, Ioannis (John) & Ha, Long & Al Shakarchi, Franck. (2016). Advanced inspection of photovoltaic installations by aerial triangulation and terrestrial georeferencing of thermal/visual imagery. Renewable Energy. 102 (Part A). 224–233. 10.1016/j.renene.2016.10.046.

[8] Anwar, Said & Abdullah, Mohd. (2014). Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique. EURASIP Journal on Image and Video Processing. 2014. 15. 10.1186/1687-5281-2014-15.

[9] Deitsch, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., & Riess, C. Segmentation of Photovoltaic Module Cells in Electroluminescence Images, arXiv (2018).

[10] Deitsch, S., Christlein, V., Berger, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., & Riess, C. (2018). Automatic classification of defective photovoltaic module cells in electroluminescence images. arXiv preprint arXiv:1807.02894.

[11] Chen, H., Pang, Y., Hu, Q., & Liu, K. (2018). Solar cell surface defect inspection based on multispectral convolutional neural network. Journal of Intelligent Manufacturing, 1-16.

[12] Balasan, T., Classic Network: AlexNet, erişildi 15 Haziran 2020. https://indoml.files.wordpress.com/2018/03/alexnet.png?w=736

[13] Mantel, C., Villebro, F., dos Reis Benatto, G. A., Parikh, H. R., Wendlandt, S., Hossain, K., ... & Forchhammer, S. (2019, September). Machine learning prediction of defect types for electroluminescence images of photovoltaic panels. In Applications of Machine Learning (Vol. 11139, p. 1113904). International Society for Optics and Photonics.

[14] J. Wohlgemuth and W. Herrmann, Hot spot tests for crystalline silicon modules, in Proc. IEEE 31st Photovolt. Spec. Conf. Rec., Jan. 3–7, 2005, pp. 1062–1063.

[15] Coşkun, C., & Baykal, A. (2011). *Veri madenciliğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması*. Akademik Bilişim, 2011, 1-8.

[16] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

A Study on Automatic Counting of Steel Bars with Image Processing

Ali Apalı
R&D Product Development
Proton Automation
Denizli, Turkey
aa@apali.co

Ahmet Yavuz
R&D Manager
Proton Automation
Denizli, Turkey
ay@apali.co

Murat Demirtaş
R&D Project Design and Planning
Proton Automation
Denizli, Turkey
md@apali.co

Burcu Ceren Sarıoğlu
R&D Project Coordination and Management
Proton Automation
Denizli, Turkey
bcs@apali.co

Abstract—**Computer vision and image processing occupy an important place in industrial control systems nowadays. The computer vision technology used in quality control systems is of great importance in terms of controlling the products produced in a production line quickly, smoothly and accurately. The breakdowns and errors that occur in production with the controls made with classical methods reduce productivity and create additional costs. In this study, a research was conducted on the automatic counting and grouping of steel rods of different diameters by using image processing technology with the machine whose prototype was designed.**

Keywords— rebar counting, image processing, design, industry

I. INTRODUCTION

The iron and steel industry has played a major role in the development of many industries and the development of societies since ancient times. Looking at the developing countries, the Iron and Steel industry has pioneered other sectors and continues to do so. Iron and steel industry which is an important sector for Turkey and the globe; has been the leading sector in the global economy, national economies and industrialization. Iron and Steel Industry also constitutes the basis for automotive, machine building, construction, infrastructure, defense, etc. sectors. When the data were analyzed between the years 2017 -2020, Turkey is seen as having an important position among the steel producing countries in the world [1]. Turkey's transition to Industry 4.0 practices will give a competitive advantage in the Iron and steel industry.

With the increase in the use of computers, there is almost no branch of science in the world that does not benefit from computer systems and studies in this field. One of the studies in this field is image processing, which has a very wide application area. Image processing, which is one of the computer methods and widely used in many fields, is an auxiliary field found in almost every sector. Image processing consists of a series of processes. These processes start with the capture of the image and continue with the use of different techniques depending on the intended use. These processes, which include mathematics and computer science, are used in areas such as machinery, architecture, design, electronics, manufacturing, security and medical.

It was tried to determine the surface area of the golden delicious apple variety with image processing technique. The image obtained as a result of image processing with three different images taken from the apple was found to be the general area with the sum of their areas. They peeled the apple peels and compared the general area found [2].

Due to changes in people's lifestyle and needs, new challenges are arising for farmers in meeting the demands. Accordingly, separating agricultural products according to their size and quality will provide convenience to customers as well as farmers. They classified bananas according to their size, shape, color and taste. They used the Image Processing toolbox in MATLAB to determine the size and maturity of the banana. The maturity percentage was determined by evaluating the size of the banana and the pixels of the image separately. Edge detection and color changes helped in determining the quality of the banana [3].

It was performed a study for determining the size of Napoleon cherries grown in Turkey. First of all, photographs of cherries were taken and the image was transferred to MATLAB environment and it was tried to determine the caliber of each cherry, in other words how many millimeters its diameter, from the images taken using image processing. Thus, it is aimed to classify the products more accurately. As a result of the study, they determined that the success rates were influenced by many factors such as the distance the photograph was taken, the amount of light in the environment, the angle of the photograph taken, and the area definitions of each caliber used in image processing. In addition, considering that the success rates were lower in small cherries, they thought that a study to be done after Napoleon cherries reached full maturity would yield more successful results [4].

Many products produced in our country need to be classified before they are marketed for commercial purposes. They worked on automation systems working on the principle of image processing. They have done their work in collaboration with software and electronic hardware, using the properties of apple and peach. Classification was carried out with electronic hardware and software, taking as reference TS42 and TS100, which were created by TSE in their studies. They designed a classification system needed in the agricultural field with the Load Cell, A / D converter, USB camera elements as hardware, and the application they developed in the MATLAB environment as software. In the past, while the classification of apples and peaches was done in a slow and challenging way with the eye, they managed to do it faster than before with the automation system they designed in a practical and quality way [5].

While apples are distinguished by color and size in cold storage, no classification is made about the spots on the apple. Another study on the classification of apples was made. In their work, they used the image processing method to identify and classify color, size and blemishes on apples. With the software they made, they were able to make stain

classification without the need for additional equipment such as filters [6].

They calculated the food portions by using image processing in their study. They tried to calculate the food and calorie values by evaluating the portion volumes of the food images with the "image processing" algorithms they developed [7].

In another sector, defense and security fields, image processing method on object recognition and resolution is also used. They worked on an automatic fingerprint recognition system with image processing. By applying "image processing" to the fingerprint images taken by the fingerprint reader, it was divided into small pieces and the area to be processed was separated from the background. A software was developed by comparing the latest data obtained with comparison algorithms [8].

In another study in the field of security, They developed a software that can recognize moving targets by processing images from the camera. The target recognition system they developed makes a pixel comparison between the target images from the camera and the target images previously recorded on the computer [9]. The target object was detected with the filtering processes they performed.

Image processing is frequently used in tissue analysis, pathological research and detection of cancerous cells in the medical field. Medical studies, They developed a system that protects the edge details of the image while suppressing the noises that occur in the MRI [10]. In their work, they developed the "rule-based fuzzy adaptive averaging filter (KTBAOF)" which will detect the pixel values of an MRI with 128x128 resolution with fuzzy logic rules, delete the noise pixel from the image and assign the closest value to the image pixel. In this filter they used, they saw that the impact image was suppressed at appropriate values in noisy images and that the edge details of the image were protected in the best way after this process and with these studies, they proved that by increasing the number of blurry logic rules, impact noises will be suppressed and better than the existing image details in edge details.

They worked on an automatic image processing system that would provide decision support to the physician in the diagnosis of "melanoma" (a type of skin cancer) [11]. In their study, images were converted into digital data using image processing technique and classified for diagnosis. With these studies, they have accelerated the decision-making process of physicians in the field of medical.

In this study, with prototyping of designed machine, a research was carried out on the automatic counting and grouping of steel rods of different diameters by using image processing technology.

II. Material And Method

Round steel bars are one of the most produced and consumed iron and steel products. In recent years, it has become widespread to sell round steel bars of different diameters in units. For this reason, the products coming out of the rolling mills must be exactly counted and sized before packaging and grouped in desired quantities. Since the current steel bar counting and sizing methods are done manually by the workers, it slows down the continuous production line and decreases the production efficiency. Manual counting and sizing process not only increases the labor cost but also creates huge counting errors. In addition, the production line must be paused from time to time to group a certain number of steel bars for packaging. This situation reduces line efficiency. Production cost increases due to the high labor costs. As a result, companies need systems that automatically counts in order to increase efficiency and provide competitive advantage by reducing costs.

A. Mechanical Desing

A prototype machine has been designed to count steel bars automatically. The prototype machine design that to be used in image processing is shown in Fig.1. Prototype machine consist of; an energy and automation panel (Fig.1.1), an "asynchronous motor" that moves the conveyor (Fig.1.2), a "LED light source" to illuminate the area where the count will be made (Fig.1.3), an "industrial camera" for image processing and sizing (Fig.1.4), an "industrial computer" to run image processing and printouts (Fig.1.5), a "guiding arm" that moves according to the coordinate coming from the processor and performs the separation process after reaching the desired number in image processing (Fig.1.6), "separator arms" that move according to guiding arms (Fig.1.7), and steel rods that enter the conveyor for sizing (Fig.1.8).

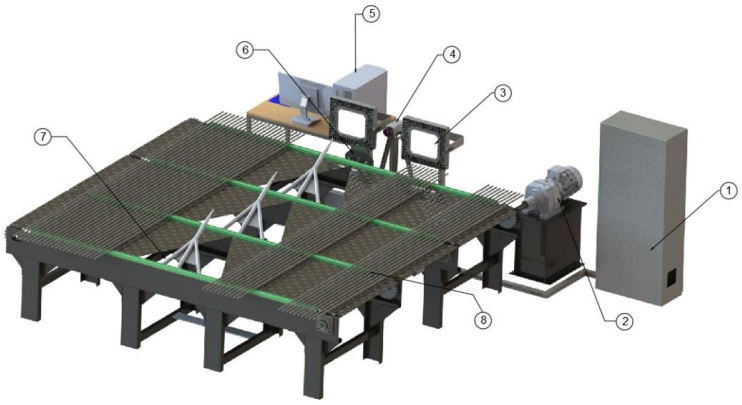


Fig. 1. Prototype Machine Design to be used in Image Processing

On the moving band shown in Fig.1, many steel bars (Fig.1.8) are seen differing in shape, color and especially size. Each and every product on the band are captured with the camera and their data are processed. For image processing, a digital camera is used with following features: Autofocus, USB 3.0, Area Scan Camera, 3.2 MP resolution, Monochrome, CMOS Sony IMX265, 1/1.8", 3.45 µm, 55 fps. Strong light sources will be placed on both sides of the camera and the exposure time of the camera will be set to the optimum time so that the image can be taken by the camera in a distinctive way. Thus, the process of removing the background from the captured image will be easier.

B. Imagine Processing

Image processing is an expression that includes all the operations performed with the aid of a computer on any image file recorded in electronic environment. The operations performed on the image files can be changing the features such as light, color, contrast, sharpness or clarity in the picture, as well as removing a noise, roughness or distortion in the picture. Apart from these, extracting meaningful data from image files and obtaining different information about the objects in that image is also called image processing.

Image processing techniques are used in many fields such as medical and biology, geographic sciences, repairing damaged images, space sciences, defense industry, industrial applications, security systems, agricultural fields. Image processing consists of two stages.

1) Pre-processing (filtering, noise removal, histogram analysis)

2) Enhancement-Visualization (thresholding, labeling, segmentation).

This study, which is based on image processing, is based on calculating the area and length measurement of the objects to be measured and grouping the objects of the same size as a result of pre-processing and enhancement-imaging processes.

In existing systems, the lengths of the final products (profile, flat bar, angle bar, pipe, construction steel, etc.) are manually measured by personal and working personnel. The separation of the final products that cannot be produced in the desired size due to production is made manually by the working personnel. This situation not only decreases the production speed but also increases the production cost.

When the image processing method is used, the production speed will increase and the number of defective final products caused by the human will be minimized since the sizes of the final products will be measured automatically. The whole system will be instantly controlled by the software that run on PLC (Programmable Logic Control) or industrial PC (computer). All desired changes will be instantly applied to the system. In this way, production will not need to be interrupted for any adjustment. In addition, as the malfunctions that occur will be detected and recorded by the automation system, both the troubleshooting process will be shortened and will help in future maintenance planning.

When processing images, the quality of the image can be increased. This is done by changing the color of the image, ie by darkening or lightening it, using filters. While the image is taken, there may be distortion or noise due to external factors. However, this situation can be remedied by using different filters. The image can be reversed or enlarged. The image can be divided into parts and objects by a process called "segmentation". With "object recognition", a very effective step is achieved on dimension detection. Shape information of objects can be accessed with "object tracking". At the same time, when similar or identical objects come across, a relationship is established between "image matching" and those objects.

Image processing flow diagram is shown in Fig.2.

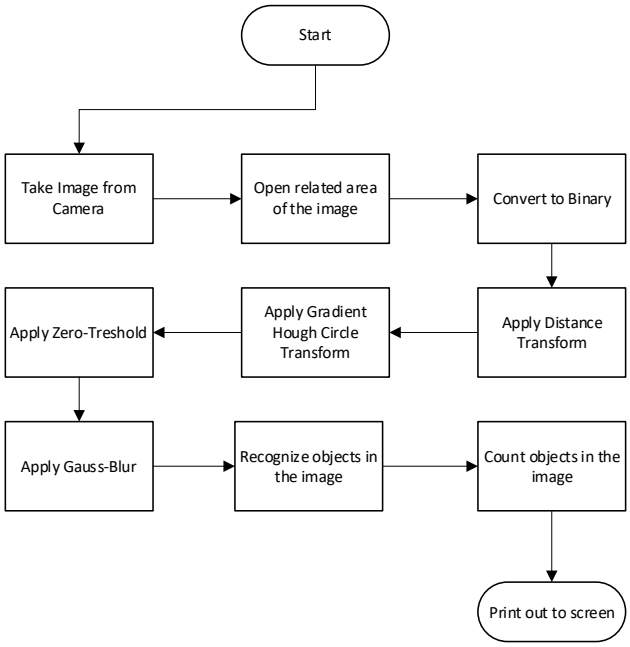


Fig 2. Image Processing Flow Diagram

As it can be seen in Fig. 2, firstly, the image is taken from the camera and converted into binary format. In order to increase the performance, instead of the whole image taken from the camera, the relevant part with the image of steel bars is separated. By applying "Distance Transform" on the separated image, the centers and diameters of the steel bars are found. With "Gradient Huugh Circle Transform", round objects are detected by extracting features in the image. In order to remove the noises on the obtained image, the image is passed through the "Zero-Threshold" operator and the "gauss-blur" operator is applied to make it softer. After these processes, the desired images will be clearly separated from unwanted images. After these processes, the counting of the steel bars is performed by comparing the calculated diameters with the diameter value entered by the user on the screen.

III. RESULTS

When the world's steel production data are examined, it is an undeniable fact that our country is in a very important position for the world. Being in such an important position creates an inevitable competition among companies in the iron and steel sector in our country. Today, it is seen that only financial power is not enough in the competition between companies and technology should be followed closely with Industry 4.0 applications.

Looking at the rolling mills in our country, it is seen that the counting process is done manually, which requires labor and is far from technology, with the sale of steel bars in units instead of weight. The negativity of the increase in labor costs causes disruptions in production speed and interruptions in the production line. In the competitive market, these disruptions and additional costs affect companies negatively.

Image processing is very important among industrial control systems today. Computer vision technology used in quality control systems is very important in terms of accurately controlling the products in the production line. With this prototype designed machine to integrate the image processing system, automatic counting of steel bars will be

performed and will contribute to the increase in production speed. With the increasing production speed, productivity will increase, labor costs will decrease and the company will be ahead of other competitors in the national and international market.

IV. CONCLUSION

This system, designed for the iron and steel industry, will contribute to the need for fast and accurate counting in production in many sectors that make mass production in the country and abroad. Considering the studies in this field in our country and in the world, it is seen that there are a limited number of companies in the world and there is no work in this field in our country. This study is a study on automatic counting of steel bars using the image processing method in the first stage, and only the design and method were determined in this study. Experiments have not yet been conducted. Results from the experiments will be presented later.

REFERENCES

[1] https://www.worldsteel.org/

[2] Işık E., Güler T., "Elma Yüzey Alanlarının Görüntü İşleme Tekniği Yöntemiyle Saptanması", Uludağ Üniversitesi, Ziraat Fakültesi Dergisi, Bursa, 17(1), 59-64, 2003

[3] Mustafa A., Nur B., Nurashikin A. F., Syed K. A., Aidil A.A., Zaipatimah A., Wong B.Y., and Zainul A., "Determination of Size and Ripeness of a Banana", IEEE, 06-08, 2008

[4] Balcı M., Altun A.A, Taşdemir Ş., "Görüntü İşleme Teknikleri Kullanılarak Napolyon Tipi Kirazların Sınıflandırılması" Selçuk Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Konya, 2016.

[5] Sert E., Taşkın D., ve Uçsuz N., "Görüntü İşleme Teknikleri İle Şeftali Ve Elma Sınıflandırma", Trakya Univ J Sci, vol. 11, no. (2), pp: 82-88, 2010

[6] Sofu M., Er O. Kayacan M.C. ve Çetişli B., "Elmaların Görüntü İşleme Yöntemi İle Sınıflandırılması Ve Leke Tespiti".*Gıda Teknolojileri Elektronik Dergisi*, vol. 8, no. 1, pp:12-25, 2013.

[7] Sun M., Lin Q., Schmidt K., Yang J., Ya N., Fernstrom J.D., Fernstrom M. H., DeLany J. P., Sclabassi R. J., "Determination of Food Portion Size by Image Processing" , 30th Annual International IEEE EMBS Conference (pp. 871-874), 2008.

[8] Özkaya N., Sağıroğlu Ş., Beşdok E., "Genel Amaçlı Otomatik Parmakizi Tanıma Sistemi Tasarımı ve Gerçekleştirilmesi", Politeknik Dergisi, 8(3), 239-247, 2005

[9] Yağımlı M., Varol S., "Renk Bileşenleri Yardımıyla Hareketli Hedeflerin Gerçek Zamanlı Tespiti", Journal of Naval Science and Engineering, 5 (2), 89-97, 2009.

[10] Toprak A., Güler İ., "Bulanık Uyarlamalı Ortalama Filtresi Kullanarak MR Görüntülerindeki Darbe Gürültüsünün Bastırılması", 5. Uluslararası İleri Teknolojiler Sempozyumu, 363-367, Karabük, 2009.

[11] Borlu M., Yüksel M.E., "Melanom Otomatik Teşhisi İçin Dermoskopik Görüntülerden Bir Görüntü İşleme Sistemi Geliştirilmesi: Ön çalışma", Türk Dermatoloji Dergisi, 2, 111-115, 2008

Machine Learning Methods for Land Cover Classification from Multi-Spectral Images

Fatma Kirac
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
kirac.fatma@izu.edu.tr

Akhtar Jamil
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-2592-1039

Alaa Ali Hameed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-8514-9255

Jawad Rasheed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-3761-1641

Mirsat Yesiltepe
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
0000-0003-4433-5606

Bulent Bayram
Department of Geomatic Engineering
Yildiz Technical University
Istanbul, Turkey
bayram@yildiz.edu.tr

***Abstract*— Remote sensing data has played vital role in land-use/land-cover applications. Many machine learning methods have been proposed to obtain different land cover classes. In this paper, we investigated the capabilities of two classifiers with object-based segmentation for land cover classification from high resolution multi-spectral images. First, graph-based minimal spanning tree segmentation was applied to segment the original image pixels into objects. From each object a set of spectral, spatial and texture features were extracted. These features were then used to train and test the artificial neural network (ANN) and support vector machine (SVM). The proposed method was evaluated on a dataset consisting of high resolution multi-spectral images with four classes (tea area, other trees, roads and builds, bare land). The experiments showed that ANN was more accuracy as it scored average accuracy of 82.60% while SVM produced 73.66%. Moreover, when postprocessing using majority analysis was applied, the average accuracy improved to 86.18%.**

Keywords— land cover classification, support vector machine, artificial neural networks, graph-based segmentation

I. INTRODUCTION

Today, digital image processing and machine learning approaches are combined to derive useful information from images. One of these research areas where these are used is the extraction of land cover from remotely sensed images. Land cover are physical areas on the soil surface forests, wetlands, streams, bare areas, impermeable surfaces on the soil, areas form the land cover. Various methods are used to extract land cover from images obtained by remote sensing systems. NDVI (Normalized Vegetation Difference) and classification methods are commonly used to vegetation information. NDVI is a measurement that uses the plant's viability by exploiting its greenness information. This measurement is made by the difference between near infrared (NIR) reflected by vegetation and red light absorbed by vegetation. NDVI is always between -1 and +1. The other objects such as water body, roads, bare soil etc. can also be distinguished using the NADI information as it approaches -1. The value of NDVI closer to + 1indidates more dense, green and healthy vegetation.

The main objective of land cover classification is to group objects with similar spectral properties. The classification process can be generally be pixel-based or object-based. Pixel-based classification is performs analysis using each pixel. This classification method has been used extensively until the 2000s. Image resolution has increased with the development of remote image sensing systems. Accordingly, the object-oriented classification method has been developed. In this classification method, the segmentation (image segmentation) process is applied to the pixels where pixels' color, frequency, brightness, neighborhood etc. are used to group similar pixels into objects. Thus, instead of individual pixels, these objects are taken into account. SLIC (Simple Linear Iterative Clustering), Mean-Shift, K-Means are among the main algorithms used for segmentation [1].

The most commonly used methods for classification are based on machine learning approaches. Machine learning techniques can be generally divided into two main categories as supervised and unsupervised learning. In supervised learning, a certain number of pixels in the image are tagged (labeled) and trained, and then these trained data are used for classification. Support vector machines (SVM), artificial neural networks (ANN), decision trees, maximum likelihood, random forests are among the main algorithms used for classification. In unsupervised learning, no labeling process is applied to the data. The system automatically tries to find the relationship between the data.

Images taken with remote image sensing systems can be used in various areas after being classified. For example, land cover maps obtained after the classification process can be used in areas such as geomorphology, Geographic Information Systems (GIS). Thanks to these maps, scientists can track changes on the earth and produce solutions to any problem that may arise. Therefore, these maps are currently needed.

In our study, we investigated two well-known supervised approaches for land cover classification: SVM and ANN. Both spectral and spatial features were derived from the high resolution images and fed into the classifier to obtain four different land cover classes: tea area, other trees, roads and builds, bare land.

II. LITERATURE REVIEW

This section highlights some of the developments made in the field of remote sensing for land cover classification using various artificial intelligence techniques.

In [2], authors examined the accuracy, configuration, speed and capacity ratios of some supervised learning algorithms (SVM, Random Forest, Logistic Regression, etc.)

used to classify spectral data on hyperspectral data. It has been observed that SVM is successful on hyperspectral data.

[3] studied the extraction of hazelnut trees from high resolution orthophoto maps. They compared object and pixel-based classification techniques. The SVM algorithm used for object-based classification produced more successful results than the maximum likelihood algorithm used for pixel-based classification (Overall accuracy SVM: 85.99%, ML: 75.83%).

Similarly, in [4] supervised learning algorithms SVM, artificial neural networks (ANN) and random forests were used for classification of ground cover (tea trees, other trees, bare areas, impermeable surfaces). Accuracy rates for tea trees were 87% for SVM, 89% for YSA and 86% for RF.

Chen et al. made a land cover classification with object-oriented super resolution mapping (OSRM) method for the mixed pixel problem (edge pixels of areas where areas differ in the images). As a result of their experiment, it has been explained that OSRM produces more land cover details for mixed objects [5].

Pipaud et al. discussed the classification of alluvial fans using mean-shift method for segmentation and SVM for classifier in object-oriented classification. As a result of the study, they concluded that mean-shift and SVM-based classification is an effective method for the description and classification of a certain place shape [1].

In their study, Zhu et al. Classified the hyperspectral image using the General Adverserial Network (GAN) method, which basically consists of two neural networks. As a result, it has been found that GANs give better results than traditional neural networks [6].

Junior et al., using eCognition and WEKA software, classified soybean plantations using geographic object-oriented image analysis (GEOBIA) and data mining, and an accuracy rate of 76% was achieved [7].

Ruiz and authors developed the Iterative K-Nearest Neighbors (IKNN) technique to classify images obtained by unmanned aerial vehicles (UAVs). This technique gave 90% accuracy compared to SVM and traditional KNN [8].

Shi and authors SVM conducted a study on the mapping of remote sensor images. To better examine the effectiveness of SVM, the Gwinnett County area, which is a complex land use and composed of different land covers, was used as the study area. In the study, SVM and MLC, one of the traditional classifiers, were compared for land cover classification. It has been observed that both methods make correct classification for the classification process in certain land cover categories. However, it has been observed that the classification accuracy of SVM method exceeds MLC in classes with complex pixels and classes with similar spectral properties. As a result, they confirmed that SVM performed better than MLC, one of the classifiers widely used in the remote sensing community [9].

Rudrapal and authors Samson performed a classification process on the hyperspectral data set. In order to better understand the data, clustering was first performed with K-Means, one of the unsupervised learning techniques. Then, the classification process was made with SVM. In the classification made on a total of 4 classes as soil, water, plant and human structures, the overall accuracy rate was found to be more than 90%. It has also been observed that SVM gives good results on a poorly trained hyperspectral data [10].

Kalkan and authors compared pixel-based and object-based classification methods using IKONOS imagery. ERDAS image software was used for pixel-based classification and e-Cognition software for object-based classification. They obtained an overall accuracy rate of 92.91% for pixel-based classifier and 98.39% for object-based classification [11].

Gürcan and authors made a land classification using Göktürk-2 satellite images. Comparing the Least Squares method and the Maximum likelihood algorithm, they obtained an average accuracy of 96.51% and 83.13%, respectively [12].

Ustüner and authors conducted a study on land cover / use classification of LANDSAT-8 satellite imagery. Within the scope of the study, SVM, random forests, KNN machine learning algorithms were used for classification process. As a result of the classification, SVM algorithm gave the highest accuracy rate (96.2%) [13].

III. DATASET AND MEHODS

A. Dataset

Rize province, where tea plants are grown extensively, was chosen as the study area. Studies have been carried out on 5 multi-spectral images, approximately 7164x9360 in size, obtained by remote image sensing systems. The images were taken using airborne UltraCam-X digital aerial camera with 30 cm ground sample distance (GSD). These data were obtained EMI Group Inc. The images shows that there are dense tea areas, other trees, uncultivated bare areas, roads and buildings (Fig. 1).

Fig. 1. A small portion of a sample image used in this work

B. Train Dataset

In order to classify in supervised learning, data must be trained first. For each class small patches were extracted consisting of a number of pixels. These patches were obtained from randomly selected two images by visual inspection. The areas were selected in such a way that they represent the respective classes without overlapping. The sample numbers of educated classes are given in Table 1.

TABLE I. TRAINING DATASET SELECTION

No	*Classes*	*Training Samples*
1	Tea Area	14
2	Other Trees	15
3	Roads & Builds	14
4	Bare Land	14

C. Classification

From each image, a set of spectral and spatial features were obtained and then these feature vectors were fed into the classifier for classification. Instead of obtaining features based on each pixel, we first employed graph-based minimal spanning tree segmentation to transform to an object-based representation and then from each object features were derived. This not only reduced the number features but also obtained more discriminating features for each object. In this study, two most widely used classifiers were investigated: SVM and ANN. These classifiers have also used commonly for the classification of remotely sensed images.

SVM is a non-parametric supervised classifier. It does not require the distribution information of the data but it needs the labels to train the data. SVMs have proven to be powerful algorithms as they can process high dimensional data with even limited number of trained data [2], [10]. SVMs are based on binary classification by creating a hyperplane at maximum distance between the members of two groups on the same plane. SVM can be applied to linear and nonlinear data. However, in data that are not separated linearly, the data is made linearly separable in a high dimensional area by using the kernel function. Polynomial kernel and Radial Basis function (RBF) are examples of kernel functions. In our study we employed RBF kernel for its efficiency and robustness and SVM tries to maximize the margin using following equation:

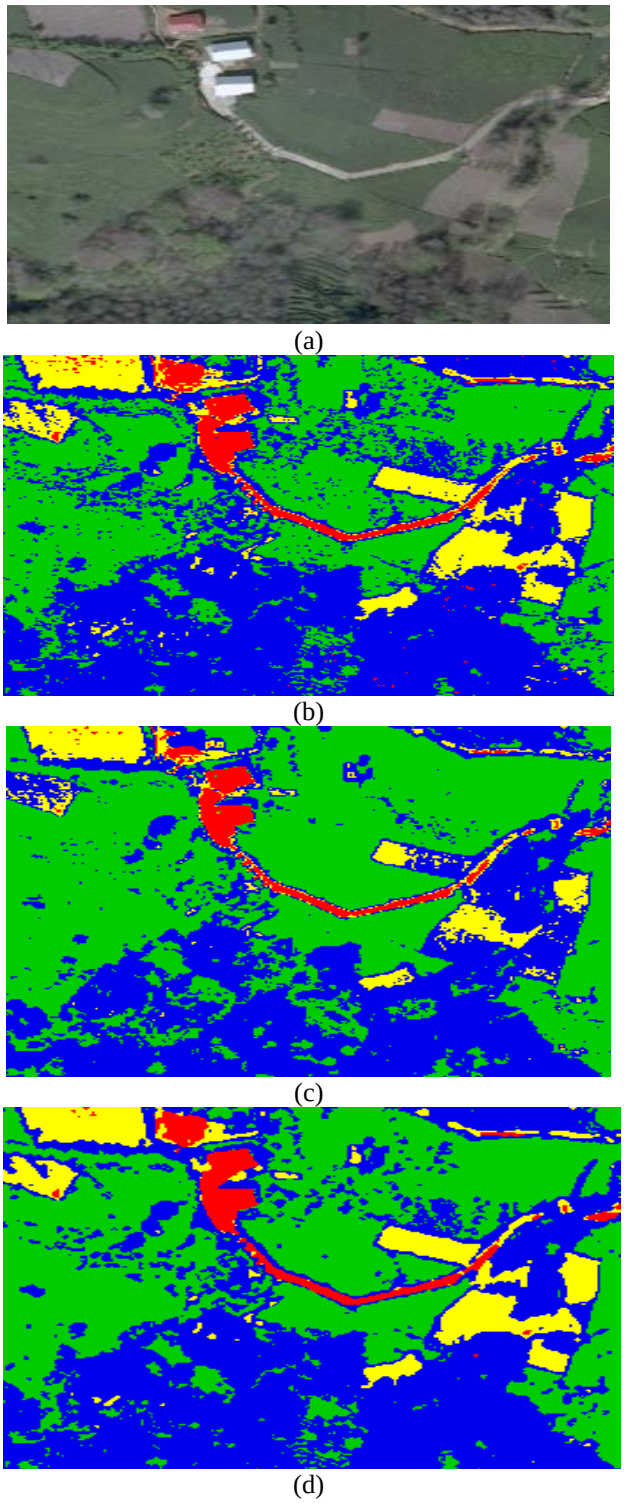
$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j)$$
$$with\ 0 \le \alpha_i \le C\ ;\ i = 1..l \qquad (1)$$
$$and \sum_{i=1}^{l} \alpha_i y_i = 0$$

where alphas are Lagrange Multiplies, K(x, y) is kernel function and C is cost.

ANN is another non-parametric supervised classifier that is also widely used in classifying remotely sensed images. This model, inspired by the human brain and nervous system, can produce solutions to complex nonlinear classification problems. In this work, a three-layered feed-forward neural network model was used with input, hidden, output neurons. The input neuron has fixed number of neurons as matching the input vector while the output layer has just 4 neurons representing each class. The number of neurons can be varied and in our case 200 neurons with single layer produced optimal results. Moreover, sigmoidal activation function was used for the neurons in the network.

IV. Experimental Results

To make land cover classification from multi-spectral images, we first prepared training examples for each class. Since the classification accuracy is based on training examples, we made our choices in clear spectral regions that are not complex. The number of samples we selected for each class are summarized in Table 1. The reason for the small number of samples is to answer the question of what classification accuracy can be achieved with little training data.



(a) (b) (c) (d)

Fig. 2. (a) Unclassified Image, (b) SVM Classification Result, (c) ANN Classification result, (d) Majority analysis map applied after the classification process.

A K-fold cross validation was applied for validation of the model. The dataset was partitioned into K sub-data sets. During training process, a single sub-data is used for validation while rest of the data is used for training. This process is repeated K times and errors were calculated for each iteration. Finally, the total errors are estimated for validation by averaging all the errors in overall all repetition. This way the model is trained using all the training examples.

The supervised classifiers generally require tuning of some parameter values. For SVM two parameters were obtained by empirical method: cost and gamma. The gamma kernel function value was set to 0.333 while cost was set to 250. Similarly, for ANN the learning rate was set at 0.2, momentum at 0.9 and number of neurons at the hidden layer were set to 200 while keeping other parameters constant.

After the models were trained, the final test images were presented for producing the final land cover maps. The results obtained for each classifier were post-processed to improve the accuracy of the classification. These techniques include morphological opening and closing to remove small holes within objects and to create a smooth boundary between different classes. Moreover, the obtained binary maps were passed through a majority analysis step to further improve the classification accuracy. The main objective of majority analysis is to assign a pixel to the dominating class in the neighborhood. The classification map generated obtained after the majority analysis is shown in Figure 2 (d).

Table II summarizes the results obtained for SVM and ANN classifiers. The average accuracy for SVM and ANN was 73.66% and 85.10% respectively. The overall results indicate that ANN was more effective for land cover classification compared to the SVM classifier. Similarly, for each class, the accuracy obtained for ANN was better than SVM. The results obtained for tea areas and other trees remained low for the SVM classifier. This can be ascribed to the spectral similarity between these two classes as both has similar vegetation index but their texture was different which was not captured by SVM in some cases. The application of majority analysis produced highly satisfactory results which produced 88.18% average accuracy for all classes. Moreover, the accuracy for each class was also higher thank both SVM and ANN classifiers. These results indicate that majority analysis as a postprocessing step is useful for obtained higher classification accuracy for land cover classification.

TABLE II. THE CLASSIFICATION ACCURACY (%) OF THE CLASSIFIERS

Classes	**Used Classifiers**		
	SVM	***ANN***	***Majority Analysis***
Tea Area	68.43	82.08	88.63
Other Trees	69.67	80.04	83.46
Roads and Buildings	71.06	79.25	82.38
Bare Land	85.51	89.01	90.25
Average	**73.66**	**82.60**	**86.18**

ACKNOWLEDGMENT

The data was provided by EMI Group Inc. Turkey for this study. The data was prepared under TEYDEP Project entitled "Development of Object Based Neural Network Image Processing System Determination of Vegetation and Forestry Boundaries" (Project Nr. 7140512). It was consulted by Prof. Dr. Bulent Bayram from Yildiz Technical University.

V. CONCLUSION

In this study, the problem of land cover classification from remotely sensed multi-spectral images is investigated. The spectral and spatial features were combined and used two commonly used supervised classifiers (SVM and ANN) for classification. Four classes of interest were defined (tea areas, other trees, road and build areas and bare land). Moreover, we selected relatively a smaller number of training samples compared to the test samples to fit the natural settings of the environment. The experimental results showed that ANN was more effective than SVM in terms of accuracy for each class. Moreover, the postprocessing using majority analysis increased the overall accuracy of the classification.

No doubt, the proposed method has certain limitations such as there were misclassifications between tea and other types of trees due to spectral similarity. As a future study, we will focus on applying automatic features extraction using deep learning-based approach, such as convolutional neural networks (CNNs). This approach will help obtained highly discriminative features that will ultimately help increase the classification accuracy. Moreover, as deep learning requires larger training data, therefore, we will prepare more training samples with labels for each class.

REFERENCES

[1] I. Pipaud And F. Lehmkuhl, "Object-Based Delineation And Classification Of Alluvial Fans By Application Of Mean-Shift Segmentation And Support Vector Machines," Geomorphology, Vol. 293, Pp. 178–200, Sep. 2017.

[2] P. Ghamisi, J. Plaza, Y. Chen, J. Li, And A. J. Plaza, "Advanced Spectral Classifiers For Hyperspectral Images: A Review," Ieee Geoscience And Remote Sensing Magazine, Vol. 5, No. 1. Institute Of Electrical And Electronics Engineers Inc., Pp. 8–32, 01-Mar-2017.

[3] A. Jamil, B. Bayram, And D. Z. Seker, "Mapping Hazelnut Trees From High Resolution Digital Orthophoto Maps: A Comparative Analysis Of An Object And Pixel-Based Approach Disaster Management View Project Automatic 3d Shoreline Extraction And Analysis From Uav and Uav-Lidar Data For Sustaniable S."

[4] A. Jamil And B. Bayram, "Tree Species Extraction And Land Use/Cover Classification From High-Resolution Digital Orthophoto Maps," Ieee J. Sel. Top. Appl. Earth Obs. Remote Sens., Vol. 11, No. 1, Pp. 89–94, Jan. 2018.

[5] Y. Chen, Y. Ge, G. B. M. Heuvelink, R. An, And Y. Chan, "Object-Based Superresolution Land-Cover Mapping From Remotely Sensed Imagery," Ieee Trans. Geosci. Remote Sens., Vol. 56, No. 1, Pp. 328–340, Jan. 2018.

[6] L. Zhu, Y. Chen, P. Ghamisi, And J. A. Benediktsson, "Generative Adversarial Networks For Hyperspectral Image Classification," Ieee Trans. Geosci. Remote Sens., Vol. 56, No. 9, Pp. 5046–5063, Sep. 2018.

[7] C. A. Da Silva Junior Et Al., "Object-Based Image Analysis Supported By Data Mining To Discriminate Large Areas Of Soybean," Int. J. Digit. Earth, Vol. 12, No. 3, Pp. 270–292, Mar. 2019.

[8] L. F. C. Ruiz, L. A. Guasselli, A. Ten Caten, And D. C. Zanotta, "Iterative K–Nearest Neighbors Algorithm (Iknn) For Submeter Spatial Resolution Image Classification Obtained By Unmanned Aerial Vehicle (Uav)," Int. J. Remote Sens., Vol. 39, No. 15–16, Pp. 5043–5058, Aug. 2018.

[9] D. Shi And X. Yang, "Support Vector Machines For Land Cover Mapping From Remote Sensor Imagery," Pp. 265–279, 2015.

[10] Dhriti Rudrapal And Mansi Subhedar, "Land Cover Classification Using Support Vector Machine," Int. J. Eng. Res., Vol. V4, No. 09, Pp. 584–588, 2015.

[11] K. Kalkan And D. Maktav, "Nesne Tabanl I Ve Piksel Tabanl I S I N I Fland I Rma Yöntemlerinin Kar Şı La Ş T I R I Lmas I (Ikonos Örne Ğ I) Nesne Tabanli Ve Piksel Tabanli Siniflandirma Yöntemlerinin Karşilaştirilmasi (Ikonos Örneği)," No. January, 2010.

[12] I. Gürcan, M. Teke, And U. M. Leloğlu, "Göktürk - 2 Uydusu Için Arazi Sınıflandırması Land Use / Land Cover Classification For G Öktü Rk-2 Satellite," No. May, Pp. 1–4, 2016.

[13] C. Paper, T. Changes, R. Using, And M. Satellite, "Landsat-8 Uydu Görüntüsü Ile Arazi Örtüsü S I N I Fland I Rmas I Nda Makine Ö Ğ Renme Algoritmalar I N I N Kullan I M I (The Use Of Ma ....," No. July, 2017.

Smart Home Automation System Design Based on IoT Device Cloud

Muhammad Ilyas
Electrical and Electronics Engineering
Altinbas University
Istanbul-Turkey
muhammad.ilyas@altinbas.edu.tr

Osman Nuri UÇAN
Electrical and Electronics Engineering
Altinbas University
Istanbul-Turkey
osman.ucan@altinbas.edu.tr

Yehya El Mohamad
Electrical and Computer Engineering
Altinbas university
yehya.elmohamad@ogr.altinbas.edu.tr

***Abstract* - With the advancement and development of communication technology, the Internet of Things (IoT) has become the focus of attention and attraction for many in terms of home automation. The Internet of Things is used to control and monitor devices and keep an eye on the home environment through the home automation system. Therefore, this paper provides a flexible, low-value and energy-efficient environmental monitoring tool with an intuitive interface based on the Internet of Things (IoT). The network server built into NodeMCU is used to access and control tools remotely either locally or globally. The tools are managed online or through a smartphone app that supports WIFI and mobile network (LTE / 4G) through access to a cloud hosting server. Provides a unique protocol for viewing and controlling the home environment over more than one switch. To demonstrate the safety and effectiveness of this device, the instruments are combined with medium switches, power components, temperature and humidity sensor, gas sensor and notification system with the proposed processing device. The proposed system can easily and efficiently control IoT based devices for home automation and support home safety through autonomous operation, energy saving and ensuring the required comfort and safety for residents.**

Index terms - Home Automation System, IOT Device Cloud, NodeMCU (ESP8266), Web Page, Voice Control, Smart Phone.

I. INTRODUCTION

Modern technology in the 21st century has provided a huge leap in the world of telecommunications and the Internet. This facilitated communication and the transfer of information freely and contributed to the expansion of communication between individuals and groups. All thanks to the smartphone and the Internet, which provided all the requirements to achieve the field of smart control. With the help of these two mechanisms and automatic control of the house, it was possible. Now we can control all electrical, electronic and other devices using smart phones. he human continued to seek comfort requirements for mankind, Smart home has become the focus of attention of scientists and engineers to work on building an integrated house in order to provide comfort and safety necessary for humanity.

As technology advances and develops, machine learning has acquired a very important pathway to discover human behavior, activity, and its broad potential in computing systems, including smart home environments. Which works on algorithms created by many researchers who study human activity and its environment inside the smart home, but these algorithms suffer from a lack of accuracy and knowledge of possibilities. This results in a lower resolution rate with less noticeable samples. Implementing a learning algorithm and feature selection is a major challenge for the many parameters that depend on it, as well as the activities to be defined and the nature of the database[1].

Internet of Thing (iots) is a major topic of our time and in our daily life. This led to a historic shift in technology, industry and engineering in all respects. This technology is embodied in a wide range of tools, sensors and systems that intelligently connect across the network. This overall advancement in iots has set new standards for technology and data systems. Iots have changed our daily lives and the way we live in many ways. New iots products, such as internet devices, power management devices, and automation components, take us to the vision of a "smart home" that aims to provide intelligence, comfort and safety inside the home. Iots further development and progress in the health field, which it has promised to provide within the smart home for the elderly and the disabled to provide the necessary health services at a low cost. Access to the Internet of Things has become widely available to newspapers, articles and the Internet, as it raises many issues and challenges that must be addressed in order to achieve the desired goal [2].

Some technologies have supported the smart home in many respects, which helped its progress and development. As an internal localization technology that enables us to find and track people through the Universal Mobile Telecommunications System (UMTS) [3].

Green energy sources are a great alternative to energy in our daily life, such as rechargeable batteries, photovoltaic (PV) energy, fuel cell (FC) and wind turbine (WT) [4]. The advantages of these technologies are numerous, such as saving energy, lowering electricity bills, and reducing pollution. This technology helped the smart home by securing the permanent power of the system to operate it, storing energy and providing it for use when necessary.

The iot platform is an all-in-one service that provides users with the things they want to bring online. It can assist millions of simultaneous device connections without difficulty and allowing the user to configure devices for machine-to-machine communication [5]. With the widespread and development of electricity and electronics inside the home and the great advancement in information technology, many different home automation technologies have been used to control remote devices such as fans, televisions, music players, air conditioners, lighting, etc. It is controlled by mobile devices using a short-range interface such as, Bluetooth, WIFI, ZigBee and GSM module [6] [7] [8]. These technologies provide communication and control of home appliances within a certain range and do not allow users to monitor their homes from outside. Although these systems provide comfort, energy efficiency, and safety, they have limitations in the scope of communications and functions.

The Internet of Things enables people and things to communicate anywhere, anytime, and with anyone using any network or service. Automation is the focus of Internet of Things technologies. Through environmental control and monitoring of energy consumption in buildings, schools, museums and offices using different types of sensors and motors that control lighting, temperature and humidity [9].

The main division of the home automation system is the microcontroller. Wi-Fi based node controller (NodeMCU) is an open source platform that is used here as the main controller. The NodeMCU is mainly used to collect information and data obtained by sensors and upload the data to the IoT server. Also, this unit receives orders from users via computers / mobile smartphones to perform tasks [10].

The NodeMCU (ESP8266) is like any other development board like Raspberry Pi or Arduino. It is handled and programmed on the Arduino program, which supports an Integrated Development Environment (IDE) for coding and uploading to the NodeMCU. With the help of this unit and the connection to the cloud computing servers, we can control and monitor the home from anywhere in the world, here it does not matter the distance, location or country you are in, we can simply access the home control interface from anywhere and see the developments, the system contains smart features that alert us By sending notifications if there is anything suspicious in the home such as fire, gas leak or theft, we can know the status of all devices connected to the system.

In this paper, we present a low-cost wireless controller to control and track the home environment. The wireless NodeMCU and the built-in internet server are used to access and control all devices through only one interface, which is used from any device, be it a smartphone or laptop. Through this interface, all devices and sensors are controlled and monitored. The interface requires you to create your own account for security and to gain acceptance into the device when opening the platform. Voice activation of switch software has also been integrated to help mainly to the elderly and the handicapped.

II. BACKGROUNDS AND RELATED WORKS

A. MOTIVATION AND PROBLEM STATEMENT

Home automation provides many benefits such as ease of access, security, comfort, safety, entertainment and low power. Numerous studies and projects have been conducted to solve problems of home automation systems. Most of the current systems are not suitable for users due to their high cost and difficulty in maintaining. Current systems also lack many technologies and unfriendly accesses. And there are some systems that do not consider the safety and security, it is a very important aspect of any smart home to avoid risks. Most of the systems currently available have limited local connectivity and insufficient functionality and features like ZigBee, Bluetooth, and Wi-Fi.

The used systems are classified into two main categories in the market, which are local and global control systems, each with its own characteristics and advantages, as the local systems rely on limited wireless communication within a certain range and limits for smart home monitoring and control. As for global control systems, they provide great advantages for controlling and monitoring over the Internet, whether internationally or locally, via a smartphone or laptop.

In order to control and monitor the smart home, an efficient and easy user interface is required. To address the above problems and reduce limitations

on home automation, the current study provides a cost-effective and efficient integrated control system for remote and proximity control over the Internet.

Home automation makes our life easier and more convenient and allows for reduced energy consumption.

- Turn on appliances remotely (fan, water boiler, etc.)
- Schedule devices to run at specific times (TV, lights, air conditioner, etc.)
- Monitor your energy consumption.

B. RESEARCH CONTRIBUTIONS

The main contribution of this study includes developing and improving smart home automation systems, introducing a low-cost system and making it more flexible and efficient by providing the best services to the user. The system monitors home conditions and controls home appliances through an easy and comfortable way over the Internet at any place and time. Our study adopted the following goals and contributions:

1. Designing a miniature model for the smart home to facilitate access to devices, monitoring and controlling them through the Internet of Things gateway, and using the NodeMCU unit as a precise unit to connect the devices to each other and enable us to connect the system to the Internet.
2. An easy and flexible webpage and application created to access and control home appliances and the ability to continuously monitor environment.
3. Providing the system with an e-mail feature to alert us when risks appear.
4. Adding the super user feature that allows us to add new users to the system.
5. A full study of the proposed system has been made in terms of speed of response to commands and rate of energy consumption.

There is a fundamental aspect of the Internet of Things that increases the reliability of devices to manage and control them. Like any other system, hardware sometimes fails. In our present system we regularly maintain and monitor devices and in case something goes wrong, it gets fixed perfectly.

The biggest challenge is designing a machine learning algorithm that will study human behavior and activities within the smart home environment to provide the best service to the user and give him the most appropriate choice upon request and the necessity of automatic response to the system. This is all done by automating activity recognition by placing unobtrusive sensors and other devices to monitor users and study their behaviors within the smart home. Implementation of this algorithm is very difficult nowadays due to the difficulty of human behavior, the complexity of human life, and the constant noise inside the home.

Future work that I aim to work on is to create a machine learning algorithm that will study human behaviors and activities in order to provide the most appropriate options and services and provide an automatic response to some issues. The ability to distinguish the system for the best option and the need for a database that fully supports this work.

The proposed model is based on the Internet of Things. The system supports Wi-Fi calling for local control and iot cloud for remote control from anywhere in the world, monitoring the home environment via the web platform, and the ability to receive notifications in case of danger. Connecting to the system is either through Wi-Fi or cellular communication to access and control the user interface through a smartphone or laptop. NodeMCU is used as a microcontroller and Wi-Fi as the communication protocol. NodeMCU was programmed using (IDE) software. This program writes the codes and uploads them to the console. Any device and sensor can be combined with a smart home automation system to track and monitor the home environment and activities. The system works wirelessly, whether through Wi-Fi or cellular connection to send and receive data. The system provides safety and comfort to users, especially the elderly and the disabled.

C. RELATED WORKS

The smart home system is a very important historical and qualitative leap in human life, as it moved humans from one reality to another, allowing them to control and monitor all devices and the surrounding environment through only one interface. The smart home has many features such as voice control of home appliances and control via specific platform through a smart phone or laptop, the smart home can improve comfort, safety and energy management. The smart home helps the elderly, people with chronic diseases and the handicapped by providing them with a comfortable and safe environment.

The smart home collects all the existing devices and sensors in a smart way to facilitate dealing with them and providing the services required of them in the most efficient and flexible way, which is accessed through a comprehensive platform for all devices that we want to control through a local or global wireless connection through an Internet connection. This platform is accessed by smartphones and laptops in an easy way by logging in to the user for complete privacy and protection. The smart home has many features such as perimeter

monitoring, safety, comfort, remote control, and alarm in case of danger such as fire, theft or gas leakage. The Internet of Things is a relatively new type of development, based on the computing and advanced communication technologies of the Internet.

It is expected in the near future to obtain an integrated smart home that secures all the user requirements quickly and proficiently, and here we mention some previous studies related to this field, and a large number of studies dealing with the topic of smart home. Hence, these are some studies that are relevant to our topic.

In [11] [12] Here, the Bluetooth module and the GSM module are used to control the sensors and devices at home using the smartphone through a specific application. The Bluetooth module was used by connecting it to the Arduino unit to access and control devices within the range and limits of the house. The GSM module was used either by connecting it to the Arduino module or a microcontroller to access and control devices by receiving and sending text messages informing us of the condition of the house and its operating environment. This feature supports remote control within the country.

In [13] [14] Here, WIFI has been used as the communication protocol to access devices and provide required information And control it using a smartphone and the Internet, this system relies on the Internet of things for safety, energy management and security. The devices are accessed and controlled via a specific platform.

In this reference [15], The Raspberry Pi unit was used to control the devices by sending and receiving email via Gmail to start and stop the devices, this process is done by processing and reading mail to carry out commands.

In this reference [16], A ZigBee Module was used to interact and communicate with network devices to control and monitor them. This technology works through several elements as the moderator is responsible for initiating and controlling the work on the network. It also stores information about the network, which includes information about protection and approved broadcast centers. As for the routers, they are responsible for dynamically expanding the network, providing a copy of the router settings, and providing Fault Tolerance technology, which means that other devices do not stop when one of the devices stops. As for the devices, they are only the devices that receive and transmit only.

In this reference [17], Here the NodeMcu module was used to control the devices through Wi-Fi connectivity via smartphone using the Adafruit platform and with the help of IFTTT which provides the web-based service. The audio recording feature was used to control and execute the information by voice control.

Some studies related to smart home systems have been reviewed and summarized, and the limitations and barriers affecting home automation systems in general have been reviewed. Our current system aims to bypass the current limitations and develop the system in terms of efficiency and flexibility.

The table below shows the comparisons between our system and other smart home automation systems.

Home Automation Systems With IOT	Control Units and Systems Requirements											
	Local Control	Universal Control	Security and Privacy	Safety	Control Interface	Energy Saving and Management	Wireless Controller	Communication and Processing Unit	Design and Real Implementation	Smart Phone or Laptop	Web-Based	Voice Control
M. A. E.-L. Mowad, A. Fathy, and A. Hafez, [10]	✓			✓	✓		RF Wireless and Bluetooth	Microcontroller and Arduino UNO		✓		
J. Bangali and A. Shaligram, [11]	✓			✓			GSM	Atmega644p Microcontroller	✓	✓		
N. David, A. Chima, A. Ugochukwu, and E. Obinna, [12]	✓		✓	✓	✓		WIFI and Bluetooth	Arduino Mega 2560 and Arduino Ethernet Shield		✓	✓	
G. Mahalakshmi and M. Vigneshwaran, [13]	✓			✓	✓	✓	WIFI	Arduino UNO	✓	✓	✓	
P. B. Rao and S. K. Uma, [14]	✓		✓		✓		Ethernet	RASPBERRY	✓	✓		
K. Gill, S.-H. Yang, F. Yao, and X. Lu, [15]	✓			✓	✓		WIFI	Zigbee	✓	✓		
S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, [16]	✓		✓		✓	✓	WIFI	NodeMCU (ESP8266)	✓	✓	✓	✓
IOT DEVICE CLOUD (MY RESARCH)	✓	✓	✓	✓	✓	✓	WIFI Or Cellular Connectivity	NodeMCU (ESP8266)	✓	✓	✓	✓

Fig. 1 Comparisons with Internet of Things system

III. SYSTEM DESIGN AND IMPLEMENTATION

The proposed system uses WIFI and telephone services to connect to the Internet and access the control platform. The system includes a smart home web page, network server and NodeMcu as the main console that connects and controls devices by receiving and sending information and data from the user platform. The proposed smart home system includes various capabilities such as user authentication, fire and safety system with siren indicators, alarm notifications, and automatic home appliance management. The main objective of the system is to control and monitor the proposed devices from anywhere in the world and from any country. In our proposed system, voice control is used to access and control devices through specific voice commands. Voice control is one of the most important achievements of our time to facilitate access and assist users with special needs, the elderly and those with severe illnesses. The system is supported by voice control along with the proposed web platform to provide more powerful features for the proposed home automation system, and the NodeMcu small unit automatically communicates with the assigned server by programming this unit, to receive commands and send them for execution.

Fig. 2 Smart home automation system architecture

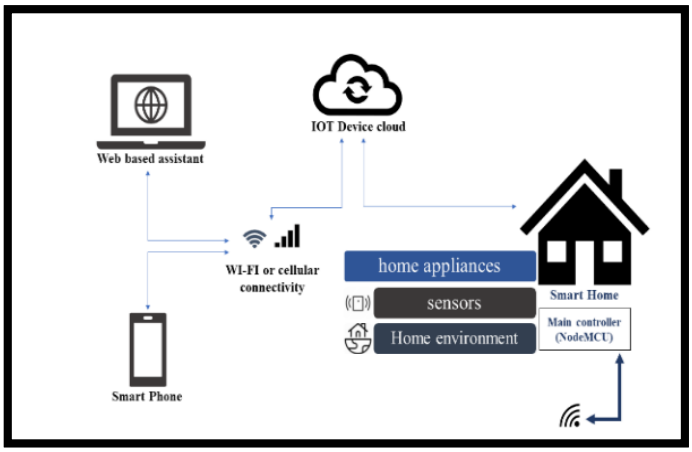
A. System Requirement

- NodeMcu (ESP8266).
- Web Page (PHP, MYSQL, HTML, jQuery).
- Arduino Software (IDE).
- Android application (Android Studio).

NodeMcu (ESP8266) provides flexibility for building an IoT based application which is open source firmware [17]. The NodeMcu Module (ESP8266) is the first choice for IoT applications in recent times, for several distinct reasons, the lower cost, and its features that support Wi-Fi. And it saves Nodejs, which takes little time in the account to perform the tasks.

The web page is programmed and shaped using the following applications and software (PHP, MYSQL, HTML, jQuery). An Android application was created via (Android Studio) linked to the web page for ease of access and control. The page was protected by logging in to the user to avoid piracy problems in order to provide the necessary privacy for the user, and the page was provided with voice controller to make the system more flexible and efficient with manual control.The information is executed and processed by this page via a global server. The Arduino IDE program was used to compile the code.

The system is equipped with a feature to alert you in the event of something suspicious in the house, such as a fire, gas leak, or theft. The user is notified



by sending an electronic notification of the existing problem.

B. Working Models

Figure 3 shows how smart home automation works. As explained below, internet connection is required as a prerequisite for accessing and controlling a smart home. The user accesses his smart home through the created web page.

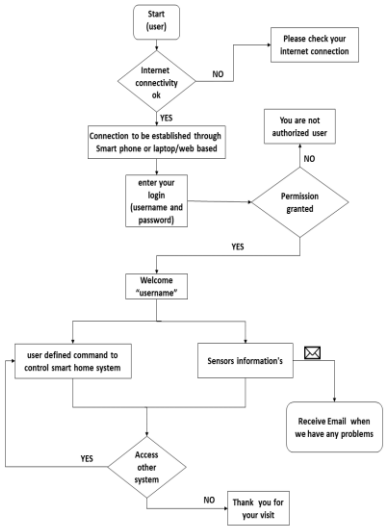


Fig. 3 System flow on the smart home automation system

First, the smart home is controlled via the web page or application that has been linked to the page to facilitate user access to control and monitor the smart home. Controlled here by voice control and manual control of the user page. The user has been assigned an access code for security purposes, which is required by the web page for security and privacy, and to prevent unauthorized access to the smart home.

After verifying the success of the connection, the user will be able to access and control his devices through the web page or the application. Here a connection is established between the web page and the NodeMCU smart home automation main console. The page control functions include ON/OFF switches for home appliances and the voice control to control by giving voice commands and monitoring the environment through sensors that display what you read such as temperature, humidity, movement, fire detection, gas leaks, and others. Which is displayed directly on the web page. The user is notified through the arrival of an email when a problem occurs within the home to solve it and provide safety and protection for the smart home.

A new feature called Unique Username has been added. This feature allows us to add a new user who can access and control smart home systems by logging in with a username and access code to access the web page or app for control and monitoring. Here the super user is added by the system owner or the project founder. The advantage of this addition is that it allows many people to become users of the system within the same house, such as wife and children, through their own phones. If anyone is busy, others stay aware of what is happening in the home and control and monitor devices. Users also receive an email alerting them if there is a danger inside the home.

In this paper, a small example of a main controller designed for a smart home is covered. Existing devices (led and dc motor) and sensors (temperature, humidity and gas sensor) were connected to the NodeMcu main control unit, and then were linked to an IoT server to access and execute commands. The figure 5 shows the connection of devices and sensors to NodeMcu.

The user controls the proposed home automation systems through the web page or the application. The user can control the devices manually through the ON/OFF buttons and through the voice control by executing voice commands. Sensors alert us to the occurrence of a danger by sending an e-mail to the user. We can augment and connect any devices and sensors in this proposed design and control them.

Through Relay, we can connect any electrical device in the house in order to control it through the control interface and continuously monitor devices such as the refrigerator, heater, air conditioner and fans, etc.

The system has been studied and followed in terms of speed of response to specific commands and rate of energy consumption. The system responded successfully and completely in terms of quick execution of orders, and the power consumption was very low. Most of the information and studies that have been submitted to the system through the data sheet for each device and through immediate monitoring by calculating the time that the system takes to execute the orders. Hardware response times to commands are very approximate. The figure 4 shows power consumption and system response.

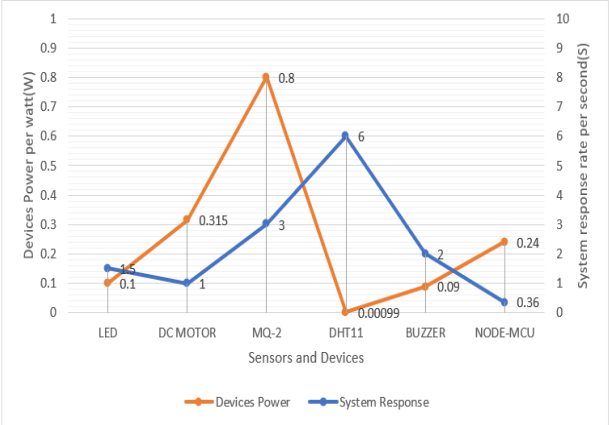


Fig. 4 The rate of energy consumption and response to commands in the system

The information provided to the devices and sensors and their energy consumption in the system as shown in the data sheet is as follows:

1. Temperature and humidity sensor response rate (MQ-2).

Relative Humidity: Response time → 1/e (63%)25°C 6s

1m/s Air 6s

Temperature: Response time → 1/e (63%) 10S

Electrical Characteristics:

Power supply: DC 3.3～5.5V

Supply current: Measure 0.3mA Standby 60μA

Sampling period: Secondary Greater than 2 seconds

2. Gaz sensor (DHT11).

Response time → 3 second

Electrical Characteristics:

Power supply: DC 4.9～5.1V

Supply current: Measure 0.3mA Standby 60μA

3. NodeMCU (ESP8266).

Response time→ jumps from 0.03 to 1 second if incoming POST payload is >1024 bytes

Electrical Characteristics:

Operating Voltage: DC 2.5 V ~ 3.6 V

Operating Current Average value: 80 mA

4. Active Buzzer (LTE12)

Response time → from 2 to 3 second

Electrical Characteristics:

Rated Voltage: DC 3～5V

Rated Current (MAX): 30 mA

5. Led (T-1 3/4).

Response time → from 1 to 2 second

Electrical Characteristics:

Rated Voltage: DC 5V

Rated Current: 30 mA

6. Toy DC Motor

Response time → from 1 to 2 second

Electrical Characteristics:

Operating Voltage: 4.5V to 9V

Current at No load: 70mA (max)

Loaded current: 250mA (approx)

No-load Speed: 9000 rpm

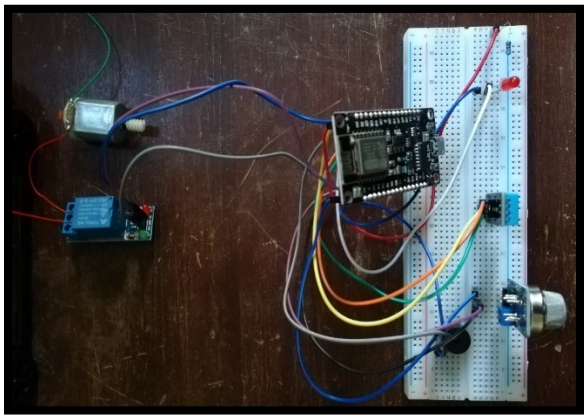


Fig. 5 Internal architecture of the controller unit

IV. EXPERIMENTAL RESULTS AND VALIDATION

In this section, system functionality results are displayed and verified from the applicable model for home automation systems. Thus, the NodeMcu module (ESP8266) successfully controls the devices and sensors (LED, DC Motor, temperature humidity sensor and gas sensor) through the created web page which enables us to access and control the devices either by manual control ON/OFF or via Voice control.

The proposed system allows us to access and monitor devices from anywhere in the world using an Internet of Things device cloud. The system was tested with manual and voice ON/OFF commands to operate the devices from another country via the user interface (web page) or the application. The system works successfully and efficiently, the system displays on the user interface the temperature and humidity inside the house through the sensor (DHT11), and the gas and smoke sensor (MQ2) senses the risks in the event of a gas leak or a fire inside the house and works efficiently through the experience by setting the alarm (BUZZER) when we are at home and if we are outside the house an email is sent to alert us of the dangers.

The superuser feature has been verified and has fully worked by adding new users to the system to access and control the devices through their own username and password. Only the owner of the system or program creator can add a distinguished user when he wants to.

The project presented a completely successful business in terms of efficiency and flexibility of use. The commands were executed through manual control and voice control to access the system and ease of control through the web page or the application. The system provides assistance and comfort in controlling and monitoring the home environment for the elderly and the disabled.

V. CONCLUSIONS AND FUTURE WORK

This study presents a flexible and cost-effective IoT home automation system. The actuators and sensors were connected to the NodeMcu to continuously control and monitor it through the web interface, or the application created to access and control devices from anywhere in the world, which in turn updates the data to the IoT server. The home environment is monitored through the data that is updated on the web page via a smartphone or laptop. An email is also sent to inform us if there is a danger inside the home for safety and security purposes. Devices are controlled easily and efficiently through the web page, either by manual operation ON/OFF or via voice control. The results of this study are promising, and additional actuators and sensors can be added to the system, and the security, safety, intelligence and comfort of the user at home can be increased through the developed system, and we can benefit from developing this model to market it in the future due to its low cost and ease of use. And saving the energy source through green energy which is a great alternative. Such as rechargeable batteries, photovoltaic (PV) power, fuel cell (FC) and wind turbine (WT).

VI. ACKNOWLEDGEMENTS

Firstly, I would like to thank Supervisor Osman Nuri UÇAN, Lecturer at ALTINBAS University of Electrical and Electronic Engineering and Co-Supervisor Muhammad Ilyas, Lecturer at ALTINBAS University of Electrical and Electronics Engineering for their supportive guidance and comments on thesis completion. Second, I would like to thank my family, relatives and friends for the support and guidance in completing this project. I am grateful to all the other individuals who helped complete this project.

REFRENCES

[1] N. *Oukrich*, "Daily Human Activity Recognition in Smart Home based on Feature Selection, Neural Network and Load Signature of Appliances," p. 131.

[2] S. Kumar, "Ubiquitous Smart Home System Using Android Application," *Int. J. Comput. Netw. Commun.*, vol. 6, no. 1, pp. 33–43, Jan. 2014, doi: 10.5121/ijcnc.2014.6103.

[3] "M. Ilyas, O. Bayat and O. Ileri, 'Indoor location estimation by using MLE based algorithm on smallcell networks,' 2015 2.".

[4] "G. M. Alshabbani, M. K. Abd, M. Ilyas and O. Bayat, "Management of Micro-grid with (SM) to Decrease Electricity Bills by.".

[5] A. M. Nitu, J. Hasan, and S. Alom, "Wireless Home Automation System Using IoT and PaaS," p. 6.

[6] Md. I. Husain, M. Alam, Md. G. Rashed, Md. E. Haque, M. A. F. M. Rashidul Hasan, and D. Das, "Bluetooth Network Based Remote Controlled Home Automation System," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, Dhaka, Bangladesh, May 2019, pp. 1–6, doi: 10.1109/ICASERT.2019.8934500.

[7] "Internet of Things, Smart Home, Home Automation, Android Smartphone, Arduino," *Internet Things*, p. 7, 2013.

[8] R. Piyare and S. R. Lee, "Smart Home-Control and Monitoring System Using Smart Phone," p. 5.

[9] V. S. Gunge, "Smart Home Automation: A Literature Review," *Int. J. Comput. Appl.*, p. 5.

[10] W. A. Jabbar *et al.*, "Design and Fabrication of Smart Home with Internet of Things Enabled Automation System," *IEEE Access*, vol. 7, pp. 144059–144074, 2019, doi: 10.1109/ACCESS.2019.2942846.

[11] M. A. E.-L. Mowad, A. Fathy, and A. Hafez, "Smart Home Automated Control System Using Android Application and Microcontroller," vol. 5, no. 5, p. 5, 2014.

[12] J. Bangali and A. Shaligram, "Design and Implementation of Security Systems for Smart Home based on GSM technology," *Int. J. Smart Home*, vol. 7, no. 6, pp. 201–208, Nov. 2013, doi: 10.14257/ijsh.2013.7.6.19.

[13] N. David, A. Chima, A. Ugochukwu, and E. Obinna, "Design of a Home Automation System Using Arduino," vol. 6, no. 6, p. 7, 2015.

[14] G. Mahalakshmi and M. Vigneshwaran, "IOT Based Home Automation Using Arduino," vol. 3, no. 8, p. 5, 2017.

[15] P. B. Rao and S. K. Uma, "RASPBERRY PI HOME AUTOMATION WITH WIRELESS SENSORS USING SMART PHONE," p. 7, 2015.

[16] K. Gill, S.-H. Yang, F. Yao, and X. Lu, "A ZigBee-Based Home Automation System," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, p. 10, 2009.

[17] S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, "Smart Energy Efficient Home Automation System Using IoT," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Ghaziabad, India, Apr. 2019, pp. 1–4, doi: 10.1109/IoT-SIU.2019.8777607.

Recyclable Material Detection in Video Streams using Neural Networks

Enes Bayturk
Mirsis Information Technologies Inc.
Istanbul, Turkey
enes.bayturk@mirsis.com.tr

Abdullah Emre Agca
Mirsis Information Technologies Inc.
Istanbul, Turkey
abdullah.agca@mirsis.com.tr

Halil Balamur
Mirsis Information Technologies Inc.
Istanbul, Turkey
halil.balamur@mirsis.com.tr

Levent Kutlucan
Mirsis Information Technologies Inc.
Istanbul, Turkey
levent.kutlucan@mirsis.com.tr

Ulas Vural *
Visus Artificial Vision Systems Ltd.
Kocaeli, Turkey
vural@visustek.com.tr

***Abstract*—Waste management is one of the major problems of the modern cities. Increasing the effectiveness of the recycling processes helps to reduce the amount of the waste needed to store, to improve the quality of life in the city, and to preserve economical values. Modern neural network-based object detection methods are used to automatize the recyclable waste segregation. In this paper, we proposed a novel waste localization and classification method that works on video streams of unconditional environments. The method uses a variant of Inception architecture and faster R-CNN algorithm to detect the recyclable items. Spatio-temporal information is used to enhance the classification accuracy on varying illumination and backgrounds by adopting an object tracking algorithm. The experiments showed that the proposed method achieves promising results on difficult scenes and different kinds of recyclable objects.**

***Index Terms*—Waste Management, Material Classification, Object Detection, Object Tracking, CNN, Deep Learning**

I. Introduction

The amount of garbage produced by cities has increased significantly in recent years. Waste management has become the biggest problem for many municipalities. Today's effective waste management approaches aim to reduce the amount of waste that needs to be stored while recovering materials with the highest economic value. Correct and efficient sorting of recyclable materials is an important and difficult part of this process. Generally, segregation of recyclable materials is done manually, and this requires a large number of people to work in unhealthy conditions.

It is known that modern artificial neural network-based methods are quite successful for classification and detection of objects. These methods, which can work efficiently on commonly used hardware, have rapidly become widespread and have found themselves many different areas of application [1]. Some of these applications aim to solve the problems of smart cities such as waste management. A comprehensive review of recent automated waste segregation methods is given in [2]. In [3], ResNet50, VGG16, and ResNet18 architectures have been compared on ThrashNet dataset and they have achieved the best results while using the Resnet18. For 4-class classification problem, the reported accuracy of the Resnet18 model is around 87%. To improve the accuracy of the ResNet architecture, it is combined with the Inception architecture and the accuracy of 6-class classification is improved to 88.6% [4]. For the same dataset and the classification problem, Aral et.al. have given a comparison of fine-tuned models based on DenseNet121, DesneNet169, InceptionResNetV2, MobileNet, and Xception architectures [5]. They have stated that the DenseNet121 architecture has been the best with an accuracy of 95%. Despite its high accuracy, one main disadvantage of DenseNet121 is a high prediction time. Changing the connection patterns in the dense blocks increases the model efficiency [6]. The models of all these studies are based on simple image datasets which cannot mimic the natural scenes for the waste segregation tasks. So, they require some special devices to form plain and controlled backgrounds. Another limitation of the mentioned studies is that they only deal with a single object classification task on still images.

Waste localization and detection is generally a more complex problem than the classification. Faster R-CNN (Region Based Convolutional Neural Network) architecture has been used to detect three different classes (landfill, paper, recycling) on a single still image [7]. The method has achieved the mean average precision of 0.682. It has not been tested on natural images as it only works with images that have white plain backgrounds. The method is based on ZF-Net which is relatively simple so its results are limited to a three-class detector. In a recent study, a solution for detecting garbage in video streams with natural background is presented [8]. The solution uses YoloV3 as a convolutional neural network to detect four different types of waste. However, these types are garbage bags, dumpsters, garbage bins, and garbage heaps instead of recyclable materials. Another shortage of the method is that it has been neglecting the temporal information in the video streams.

In this paper, we proposed a novel method to detect recyclable materials in the videos that have natural backgrounds

* Corresponding author: Ulas Vural, vural@visustek.com.tr .

Fig. 1. Sample frames from our dataset with different backgrounds and lighting conditions

and varying illumination. Working on the challenging scenes require to select a robust object detection method. Thus, faster R-CNN object detection algorithm with Inception v2 architecture is preferred as the base object detector. Then, to improve the classification accuracy of the selected method, the detector is supported by a temporal voting window that is obtained by a multi-object tracking algorithm. Experimental validations show that the temporal voting considerably increases the accuracy of the detector.

The rest of the paper is organized as follows: In Section II, properties of our dataset is given in details. Section III describes the proposed method including the explanation of object detection model and the multi-object tracking algorithm. Experimental validations and their results are presented in Section IV and the paper concludes in Section V.

II. Dataset

Current public datasets for the waste detection are mostly limited to still images with plain backgrounds. A method which works on natural scenes needs to be able to cope with complex background textures as well as with the variations on illimunation. Because of this, a custom video dataset that well reflects the natural environments has been created to train and test the proposed method.

The dataset contains 50 videos from four scenes with distinct backgrounds such as plastic rug, wooden, grass, and concrete. In Figure 1, some sample frames from the dataset are given. The videos in the set are at 10 fps and around 10 seconds long. The original resolutions of the videos are 3840x2160. All the videos have been downscaled to the resolution of 960x540 for higher processing speeds and better resembling the resolutions of commonly used surveillance cameras.

The objects in the dataset are formed of four different types of recyclable materials. These are cardboard, glass, metal, and plastic. All the recyclable objects in the videos were manually classified and their bounding boxes were marked. In Table I, you can find the quantities of these objects and their material types.

TABLE I
Description of the Original Dataset

Material Type	Original Data	Augmented Data	Total
Cardboard	2112	7458	9570
Glass	1197	5028	6225
Metal	3116	8941	12057
Plastic	1827	6573	8400

Data augmentation approach is adopted to increase the number of samples. Objects obtained from the original video frames have been synthetically located to the new frames with some variations on their rotations, scales, and translations. After the augmentation, the total number of objects are increased to 28,000 while preserving the distributions of the material classes.

III. Method

We will first give an overview of the proposed method here and leave the details of the method to the subsections.

The method has two main parts. The first one is a neural network based object detector and the second is a multi-object tracker. For each video frame, the object detector finds all recyclable items in that frame. Then for each detected item, an object tracking algorithm decides whether the item is a continuation of an existing object track or a view of a new object. The tracker algorithm starts a new tracker routine for each of the new item. The method decides the final estimation of the object material types according to the frequencies of the estimated labels for single frames inside a given time window.

A. Object Detector

Today's state-of-the-art object classifiers are based on the CNN (Convolutional Neural Network) architectures. These architectures efficiently use GPUs, reduce complex and time-consuming feature engineering efforts and achieve high accurate classification results. In our method, we prefer to use the Inception v2 architecture. This architecture reduces representational bottleneck problem of the basic batch normalized

Inception architecture and works more efficiently by using factorization techniques [9].

Faster R-CNN algorithm offers high performance object detection by using region proposals [10]. These proposals are efficiently produced by a neural network and it replaces slow selective search algorithm. In the proposed method, we use faster R-CNN algorithm based on the Inception v2 architecture. This multi-object detector works on each video frames independently and supplies intermediate information to the object tracker. This information contains single frame material estimations and the bounding boxes of the detected objects. The object tracker uses this input to improve the estimation quality.

B. Object Tracker

The proposed object tracker uses outputs of the object detector as inputs. For each video frame, detector marks the bounding boxes of detected objects. The tracker processes these detected bounding boxes to extract temporal relations of multiple objects. The tracking algorithm checks if there is an overlapping between any detected bounding boxes in two consecutive frames (Fig. 2). Separating axis test method is used to determine the intersections between two rectangular bounding boxes. If there is an intersection then the tracker marks the current candidate as a continuation of a previously detected object. Otherwise, the tracker starts to process the candidate as a new standalone object.

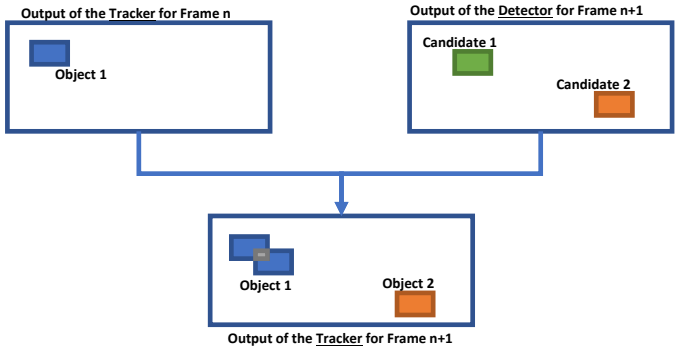


Fig. 2. The tracker checks if there is an intersection between the bounding boxes in two consecutive frames.

The tracker includes a voting mechanism to reduce the effect of noisy classifications. For each standalone object, the tracker forms a circular linked-list with the size of the temporal tracking window. For each new frame, the voting list member is filled with the estimation of the object classifier. Once all the voting list is completely filled, the tracker presents a estimation according to the estimation label with the highest frequency. New estimations overwrite the oldest estimation in the voting list if there is no free space.

IV. Experiments

We have realized two different experiments to show the effectiveness of our method. The first experiment aims to show the relationship between the time window size and the classification accuracy. In this experiment, we tried 6 different window sizes. The results show that increasing the window size always increases F1 scores and the precisions. In Table II, the size of time windows and the achieved metrics are given. When a window size is 1 than it means that the classifier decides the final estimation by only considering the current frame. Larger window sizes indicate that the final estimation uses intermediate estimations of the previous frames throughout the given time window. The last row of the table means that the method uses as many intermediate estimations as possible for an object to decide its final label. In other words, for this situation the window size varies for the objects according to their total tracking durations.

TABLE II
Effects of the tracking window size on the accuracy metrics

Window Size	Precision	Recall	F1
1	0.877	0.974	0.923
5	0.927	0.974	0.950
10	0.936	0.971	0.953
15	0.950	0.967	0.959
20	0.969	0.962	0.966
*	0.940	1.000	0.970

In the second experiment, we wanted to show the robustness of our method against the variations in the training and the test sets. Five-fold cross validation is used in this experiment. For each of the validations, it took around 17 hours to train a four-class classifier on a standard PC with a GTX 860M graphics card. The hyper-parameters of the models have been kept constant and not fine-tuned. During the training phase, momentum optimizer is used with the learning rate of 0.00002. The number of steps is set as 200,000 and the batch size is 1. COCO pretrained weights were used while training the model.

Fig. 3. Sample frames where the proposed method successfully detects and tracks the objects.

Some visual samples of the system output is presented in Figure 3 and Figure 4 .In Figure 3, succesfully detected and

TABLE III
RESULTS OF 5-FOLD CROSS VALIDATION

Window Size	1			10			20			*		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Train-1	0.95	0.98	0.96	0.98	0.98	0.98	1.00	0.97	0.99	0.98	1.00	0.99
Train-2	0.95	0.97	0.96	0.98	0.97	0.98	0.99	0.96	0.97	0.98	0.98	0.98
Train-3	0.92	0.98	0.95	0.96	0.98	0.97	0.99	0.97	0.98	0.98	1.00	0.99
Train-4	0.95	0.99	0.97	0.97	0.98	0.98	0.99	0.98	0.99	0.98	1.00	0.99
Train-5	0.94	0.97	0.96	0.99	0.97	0.98	1.00	0.96	0.98	0.98	1.00	0.99
Average	0.94	0.98	0.96	0.98	0.98	0.98	0.99	0.97	0.98	0.98	1.00	0.99

(a) (b)

Fig. 4. (a) The proposed method detects the object but misclassifies it. (b) The system cannot detect the object.

classified objects are given. You can also see the tracking path of the objects in these images. In Figure 4 (a), the method finds and tracks the object but misclassifies it and in (b), you can see the unqiue object which our tracker cannot find and track. This is a transparent object on a complex grass texture.

In Table III, the result metrics are given for each of the cross-validation folds and for different tracking window sizes. The average F1 score hits to 0.98 for window sizes larger than 10. The results show that the method performs similarly on different training and test sets.

V. CONCLUSION

A clean and healthy environment is an important asset for all countries. Therefore, increasingly large budgets are allocated for waste management every year. Effective waste management will reduce costs and support the protection of the environment and the health of living beings. In particular, the correct collection of recyclable waste will make a significant economic contribution. In the near future, it is expected that wastes will be collected and separated by autonomous machines. This may only be possible by accurately detecting and classifying waste in natural environment.

In this paper, a system using modern artificial neural network approaches and detecting recyclable objects in natural scenes is proposed. The proposed method detects multiple recyclable objects and offers high classification accuracy by using the temporal tracking information. This method may help automated devices to use motion based visual information. Thus, a solution with high detection and classification accuracy and low computational cost will be presented.

As future work, it is planned to test different models and architectures for object detection and tracking. In addition, studies will be carried out on the detection of wastes by UAVs.

REFERENCES

[1] Zou Z, Shi Z, Guo Y, Ye J. Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055. 2019 May 13.

[2] Flores MG, Tan Jr JB, Fandino P, Guia J, Limpiada RL, Bermudez R, Manalang JO, Gerardo BD, Iii BT. Literature Review of Automated Waste Segregation System using Machine Learning: A Comprehensive Analysis. International Journal of Simulation: Systems, Science & Technology. 2018.

[3] Gyawali D, Regmi A, Shakya A, Gautam A, Shrestha S. Comparative Analysis of Multiple Deep CNN Models for Waste Classification. arXiv preprint arXiv:2004.02168. 2020 Apr 5.

[4] Ruiz V, Sánchez Á, Vélez JF, Raducanu B. Automatic image-based waste classification. InInternational Work-Conference on the Interplay Between Natural and Artificial Computation 2019 Jun 3 (pp. 422-431). Springer, Cham.

[5] Aral RA, Keskin ŞR, Kaya M, Hacıömeroğlu M. Classification of trashnet dataset based on deep learning models. In2018 IEEE International Conference on Big Data (Big Data) 2018 Dec 10 (pp. 2058-2062). IEEE.

[6] Bircanoğlu, Cenk, Meltem Atay, Fuat Beşer, Özgün Genç, and Merve Ayyüce Kızrak. "Recyclenet: Intelligent waste sorting using deep neural networks." In 2018 Innovations in Intelligent Systems and Applications (INISTA), pp. 1-7. IEEE, 2018.

[7] Awe O, Mengistu R, Sreedhar V. Smart trash net: Waste localization and classification. arXiv preprint. 2017 Dec 15.

[8] De Carolis B, Ladogana F, Macchiarulo N. YOLO TrashNet: Garbage Detection in Video Streams. In2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) 2020 May 27 (pp. 1-7). IEEE.

[9] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 2818-2826).

[10] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence. 2016 Jun 6;39(6):1137-49.

Tuberculosis and Lung Cancer Prediction using Machine Learning Methods and Over-Sampling Technique

Amani YAHYAOUI
Department of Software Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
amani.yahyaoui@izu.edu.tr

Amir Karaj
Department of Software Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
amirkaraj02@gmail.com

Merve Hamzaoğlu
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
mervehamzaogluu@gmail.com

Akhtar Jamil
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-2592-1039

Nejat YUMUŞAK
Department of Computer Engineering
SAKARYA University
Sakarya, Turkey
nyumusak@sakarya.edu.tr

***Abstract*—With the continuous advancement of technology, people and machines can complement their specific skills to achieve effective results. In this sense, with the inevitable increase in the number of diseases that threaten human health, Decision Support Systems (DSS) are widely used in the medical field to help doctors making better clinical decisions. Among these diseases, such as tuberculosis and lung cancer are considered potentially serious infectious and are among the top 10 causes of death in the world. This paper presents a medical DSS for tuberculosis and lung cancer diagnosis by using machine learning algorithms, such as the Support Vector Machines (SVM) and Artificial Neural Network (ANN). Moreover, Borderline Synthetic Minority Over-Sampling Technique (Borderline - SMOTE) was also employed to increase the number of minor sample size. The experimental dataset used is taken from Diyarbakir chest diseases hospital. The obtained results proved the efficiency of the proposed system in helping doctors making the right decision and improving the quality of health care.**

Keywords—Decision Support Systems (DSS), chest diseases, tuberculosis, lung cancer, machine learning, Support Vector Machines, Artificial Neural Network.

I. INTRODUCTION

Machine learning (ML) is a branch of artificial intelligence which is able to create intelligent machines that think like humans and make some human tasks and decisions [1]. The machine learning techniques are used in many research area such as industry, medicine, banking, statistics to name just a few.

Lungs are vital organs in the human body and vulnerable by many diseases, namely tuberculosis and lung cancer. In fact, Tuberculosis is a contagious infectious disease that, primarily attacks the lungs and can damage other organs such as the brain [1]. This disease is caused by bacteria called Mycobacterium tuberculosis that transmits the disease from person to person by air [1]. According to the World Health Organization, tuberculosis is one of the ten leading causes of death worldwide. Indeed, it caused the deaths of 1.3 million people in 2017 [2]. Tuberculosis develops slowly and can be manifested by several symptoms like fever with night sweats, cough sometimes with a few bloodstreams, shortness of breath, pains in the chest, a state of fatigue, loss of appetite, weight loss, headache, presence of large ganglia [3].

Another disease that attack lings is the lung cancer, which is the most dangerous of all cancer types [4] that initially starts in the lungs and, then can spread throughout the human body. It consists of uncontrolled cell division in the lungs. Doctors confirm that smoking is the principal cause of lung cancer but also passive tobacco exposure can also cause lung cancer for nonsmokers [4]. According to the World Health Organization last statistics, lung cancer caused the deaths of 2.09 million people in 2018 [5]. Unlike other cancers, the lung cancer symptoms appear when the disease is in an advanced stage. Among these symptoms, the appetite loss, the voice changes, the frequent chest infections such as bronchitis or pneumonia, the breath shortness, unexplained headaches, the weight loss, the wheezing [6] Lung cancer treatments depend on three main factors, namely the tumor location, its stage and the person health state. Generally, surgery, radiation and chemotherapy are the main lung cancer treatment [6]. In medicine, doctors confirm that the earlier the disease is discovered, the greater the recovery chances. The different tuberculosis and lung cancer symptoms mentioned in this first section help doctors to make the correct diagnosis of these diseases and to start the necessary treatment. Despite this, the mortality rate in worldwide of patients suffering from chest disease is continually increasing. This situation can be explained by the possible diagnosis errors, doctor's competence, the continuous appearance of new complicated diseases or the lack of means that can help doctors to make the right decisions.

As a solution, in today's technologically advanced world, researchers are focusing on the use of artificial intelligence techniques to help doctors making the right decision in medical diagnosis on the right time. Among these techniques, ANN and SVM are used in the present paper due to their popularity. To increase the samples, Borderline-SMOTE was also used to generate new samples. The main

objective was to remove the imbalance issue and increase the performance of the classifier.

This paper is organized as fellow: Section 2 presents some previous works that focus on lung cancer and tuberculosis diagnosis by using some of machine learning techniques. Section 3 describes the machine learning algorithms used in this paper. The dataset, the proposed methods, the application developed and the obtained results used are detailed in Section 4. Finally, the conclusions are summarized in section 5.

II. Related Work

In the literature, many researchers have proposed various methods for lung diseases diagnosis. For instance, Udayakumar E. et. al [7] have designed an automated approach for predicting tuberculosis by using the SVM method. The authors employed two datasets have been used from Shenzhen Hospital, China and from Tuberculosis clinic of Montgomery County (USA). As result, the SVM has shown good performance by giving 82% as classification accuracy [7].

In addition, Rehana Rajan and K. G. Satheesh Kumar [8] have proposed a hybrid classification system composed by two classification methods, which are the SVM and the Multi-layer Perceptron (MLP) to identify tuberculosis. The results have shown that the hybrid system is able to identify tuberculosis with an accuracy rate of 83.42% for the SVM method and with an accuracy rate of 74.61% for the MLP method [8].

P.JohnVivek, and Swathika.S.R [9] have presented a method for tuberculosis identification by using the most used classifiers in disease identification, which are the K-Nearest Neighbors KNN , the Binary-SVM and the Multi-class SVM. The Multi-class SVM gave the best results 10 with an accuracy of 92%, followed by the Binary SVM with 89% and 75% by the KNN [9].

Gayatri S Mahajan and Dr. S. R. Ganorkar [10] have proposed a new method based on SVM for the tuberculosis detection in chest radiography. In this research paper, authors have included image processing field to extract feature from chest images, have applied the SVM method to classify the extracted features, than checked whether the patient is affected with tuberculosis or not. In this research paper, the authors have shown that the proposed system can efficiently verify the presence or not of tuberculosis with an accuracy of 88% [10].

Wafaa Alakwaa, Mohammad Nassef and Amr Badr have suggested a computer- aided diagnosis (CAD) system for lung cancer diagnosis by using Neural Network method [11]. The dataset used in this research was taken from Kaggles Data Science Bowl (DSB) and composed with 1397 patients computed tomography (CT) scan. The classification accuracy from this research has reached 86 %[11].

Moreover, in [12], Raviprakash S. Shriwas proposed a system for lung cancer prediction in its earliest stage by using Artificial Neural Network ML technique. The performance of the proposed system was very good and has reached an accuracy of 96%.

In addition, in [13], Abdelwadood M. Mesleh have proposed a Computer-Aided Design (CAD) hybrid system that allow to detect the lung cancer by using three different algorithms which are Multi-Layer (ML), Neural Networks (NNs) and the Independent Component Analysis (ICA). The performance of the proposed system has achieved an accuracy of 91%. Based on this literature review, it can be said that SVM and ANN algorithms have good performance in the medical field and specially in lung cancer and tuberculosis diagnosis.

III. Materials and Method

A. Data Set

The dataset used in our research was taken from Diyarbakir chest diseases hospital. The dataset 27 include 250 real record from patients distributed as follows: 100 patient suffering from 28 tuberculosis, 100 patient suffering from lung cancer and 50 healthy patients. The dataset include the most 38 relevant attributes that can help doctors to easily identify the disease. For some attributes, the maximum and the minimum values are given to indicates the interval that can be accepted for non-infected patient. For example, the value of White Blood Cell (WBC) must be between 4-11. For other attributes, the value can be 0 which indicate the non-existence of the attribute and 1 mean the existence of the attribute, for 36 example the smoking addiction attributes is 0 or 1.

B. SVM

SVM is basically a binary classification method, developed by Cortes and Vapnik in 1995 [14]. The main idea of SVM is to classify the membership of an entry into one of two classes separated by a hyperplane (Fig. 1). If the data are linearly separable, the hyperplane separates the two classes by maximizing the minimum distance between the data and the hyperplane.

However, the majority of classification tasks are not linearly separable. To apply the SVM on such tasks, the data is first transformed into a higher dimensional space in which the data is supposed be linearly separable by using a kernel function. There are various kernel function, such as the radial basic kernel, polynomial kernel, linear kernel etc. Among these kernel functions radial basis and polynomial kernel are widely used. For more detail about the SVM refer to [15] and [16].

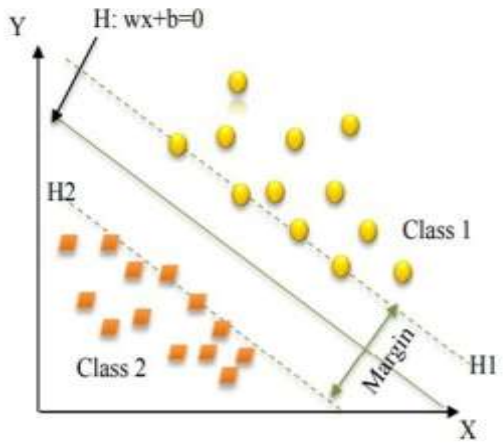


Fig. 1. Hyperplane construction using SVM method for classification [17]

C. ANN

Motivated from the working of the human brain, the ANN algorithms tries to mimic the similar behavior of the human brain. The first mathematical modeling tests of the human brain by the notion of formal neurons was carried out by W. M. McCulloch and W. Pit in 1943[18]. This concept was then developed into a full-fledged neural network method known as perceptron system in 1957 by Franck Rosenblatt, which consisted of a set of artificial neurons arranged in layers and interconnected by synaptic weight [19]. Such architecture of the neurons learns the patterns in the data similar to the way the human brain works.

As shown in Fig. 2, the perceptron is organized in three layers: the input, hidden and output layers. The input layer is a set of neurons carrying the input signal. The hidden layer(s) tries to extract the hidden relations between the variables. The number hidden layers and the number of neurons on these layers can be varied. The output layer consists of the neurons according to the number of the output classes for the target problem.

The working of the neural network system is based on two phases: the training phase and the operation phase [19]. In the first phase, based on the knowledge extracted from the learning data, the synaptic weights of each neuron will be adapted to solve a particular problem. Once the training phase is completed, the ANN system produces the results based on knowledge gained from the training phase.

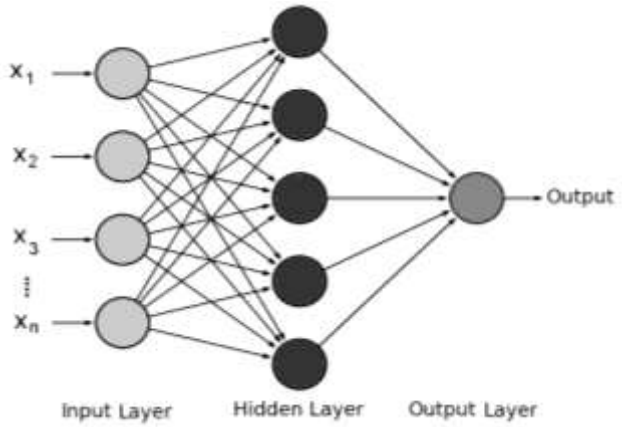


Fig. 2. Multilayer Perceptron organization in ANN [18]

D. Classification

In our study, two classifiers which are the SVM and the ANN have been investigated. The overall workflow of the proposed method is shown in Fig. 3. Any supervised classifier requires training and testing data. Therefore, first we divided the data into test and train datasets. In fact, the train data represent two-thirds of the total data, and the test data represent one-third of the total data. Furthermore, analysis was performed to replace any missing attributes or invalid values. In such case, the attributes were replaced with the mean value of all the attributes in each set. Moreover, the data is labelled as shown in Table I.

Finally, both classifiers were trained using the training data. For each classifier, we employed K-fold cross-validation to use all the available data for training. The trained models were saved and then feed then with the test data. They model produced labels for all three possible classes which were then compared with real ground truth for evaluations.

TABLE-I CLASSES AND LABELS

Case	Abbreviation	Label
Tuberculosis	TB	1
Lung Cancer	LC	2
Normal	NR	3

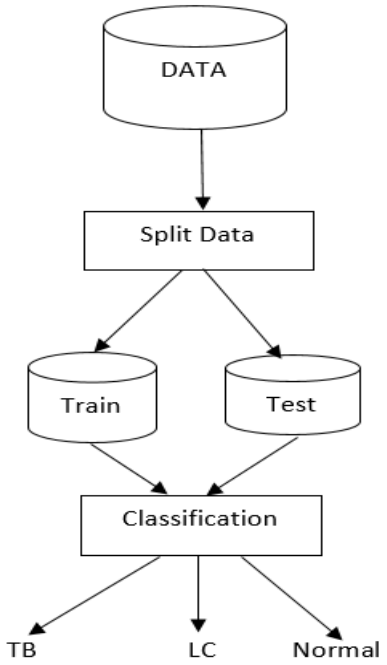


Fig. 3. Tuberculosis and lung cancer diagnosis using ANN and SVM

IV. RESULTS

Several experiments were performed to evaluate the performance of the used classifiers. The following sections provide a detailed description.

A. Experimental Setup

Since, the supervised classifiers depend on several parameters, obtaining the optimal values for these parameters is crucial as they effect the classification accuracy of the classifier. Therefore, before applying the model, the optimal parameters for each classifier were obtained by applying an exhaustive grid search. The parameters that produced highest validation accuracy were used for final classification. In this study, SVM with radial basis function was used. It requires tuning of two most important parameters, the cost (C) and gamma (γ). The optimal values for these parameters were obtained by searching in a predefined range: $C \in \{ e^{-5} - e^{5} \}$ while $\gamma \in \{e^{-3} - e^{0}\}$. The final values obtained for C was 275 while γ was 0.15. Similarly, for ANN both momentum and learning rate were fine-tuned. Learning rate was empirically set to 0.1 and momentum was set to 0.7. Moreover, the maximum number of iterations were 1000 and two hidden layers each with 200 neurons were used.

The method were conducted on Intel® Xeon® CPU E3-1231 with 2.40GHz processing power and 32GB RAM, using with Matlab © Environment.

B. Evaluation

Performance of the proposed methods were evaluated using the overall accuracy, precision, recall and f-measure metrics. These metrics were derived using True Positive (TP) which is the correct predictions, True Negative (TN) which is correctly predicted for wrong event, False Positive (FP) is incorrectly predicted and False Negative (FN) correctly predicted for wrong event. These measures were obtained using following formulas:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The obtained results for SVM and ANN classifiers are summarized in Table II and Table III respectively. As shown that both classifiers produced good classification accuracy for each class. The classification results obtained by using the ANN model is relatively higher than the SVM for each class. The classification results for tuberculosis disease by using ANN is 97.68% while the SVM is 94.11%. The classification results for lung cancer disease by using ANN is 96.51% while the SVM presented 83.37%. Similarly, for normal cases both ANN and SVM produced same result (97.68%).

SVM produced the lowest accuracy for the LC class which achieved highest accuracy for normal patient class. While ANN produced consistently similar results for all three classes. In terms of processing time, SVM took 20 seconds for training while ANN took 68 minutes for training. The testing time was same for both and was negligible therefore, we did not account for the testing time.

TABLE- II CLASSIFICATION ACCURACY (%) FOR SVM CLASSIFIER

Class	OA	PR	RC	FM
TB	94.11	94.65	91.36	92.97
LC	83.37	85.36	84.3	84.82
NR	97.68	98.36	96.36	97.34
Mean	91.72	92.79	90.73	91.71

*OA: Overall Accuracy, PR: Precision, RC: Recall, FM: f-score

TABLE- III CLASSIFICATION ACCURACY (%) FOR ANN CLASSIFIER

Class	OA	PR	RC	FM
TB	97.68	98.36	95.77	97.04
LC	96.51	94.65	95.05	94.84
NR	97.68	99.02	96.36	97.67
Mean	97.29	97.34	95.72	96.52

*OA: Overall Accuracy, PR: Precision, RC: Recall, FM: f-score

V. CONCLUSION

In this paper, a comparative analysis was performed using SVM and ANN classifiers for detection and diagnosis of tuberculosis and lung cancer diseases. The experimental results indicated that both techniques were effective to detect these diseases with high accuracy. These can be incorporated in a decision make system that can help the doctors to diagnosis tuberculosis and lung cancer with higher accuracy.

Our results are in line with the state-of-the-art method for lungs cancer and tuberculosis detection. However, in future, we would like to further enhance the accuracy of the proposed method by integrating a deep learning-based approach such as convolutional neural networks.

REFERENCES

[1] A. T, S. Natarajan, and K. N. B. Murthy, "A Data Mining Approach to the Diagnosis of Tuberculosis by Cascading Clustering and Classificatio,"
no. December 2014, 2011.

[2] World health Organization, global tuberculosis report. 2018.

[3] A. Yahiaoui, O. Er, and N. Yumusak, "A new method of automatic recognition for tuberculosis disease diagnosis using support vector machines, " Biomed. Res., vol. 28, no. 9, pp. 4208–4212, 2017.

[4] M. A. Hussain, T. M. Ansari, P. S. Gawas, and N. N. Chowdhury, "Lung Cancer Detection Using Artificial Neural Network & Fuzzy Clustering,
" Ijarcce, vol. 4, no. 3, pp. 360–363, 2015.

[5] A. Yahyaoui and N. Yumuşak, "Decision support system based on the support
vector machines and the adaptive support vector machines algorithm for solving chest
disease 46 diagnosis problems," Biomed. Res., vol. 29, no. 7, pp. 1474–1480, 2018.

[6] U. E., S. S., and V. P., "TB screening using SVM and CBC techniques," Curr.
Pediatr. 48 Res., vol. 21, no. 2, pp. 338–342, 2017.

[7] K. Patel, Brijeshkumar; Chavda, "Hybrid SVM for Automatic Detection of 50
Tuberculosis," Int. J. Adv. Res. 1Computer Sci. Manag. Stud., vol. 3, no. 11, pp. 44–53,
2013.

[8] M. P. JohnVivek and S. S.R, "Accurate TB manifestation using multi class SVM
54 classifier," Iarjset, vol. 2, no. 1, pp. 37–44, 2015.

[9] G. S. Mahajan, "Detection of Tuberculosis Using Chest Cardiograph," Int. J. Innov. Res.
Sci. Eng. Technol., vol. 6, no. 5, pp. 9858–9865, 2017

[10] W. Alakwaa, M. Nassef, and A. Badr, "Lung Cancer Detection and Classification with
3D Convolutional Neural Network (3D-CNN)," Int. J. Adv. Comput. Sci. Appl., vol. 8,
no. 8, 2017.

[11] R. S. Shriwas and A. D. Dikondawar, "Lung Cancer Detection and Prediction By Using 1
Neural," IPASJ Int. J. Electron. Commun., vol. 3, no. 1, pp. 17–21, 2015.

[12] A. M. Mesleh, "Lung cancer detection using multi-layer neural networks with 3 independent component analysis: A comparative study of training algorithms," Jordan
4 J. Biol. Sci., vol. 10, no. 4, pp. 239–249, 2017.

[13] T. Evgeniou and M. Pontil, "Support Vector Machines: Theory and Applications," in 7 Advanced Course on Artificial Intelligence, Springer,Berlin, Heidelberg., 2001, pp. 249–257.

[14] A. Kowalczyk, Support Vector Machines succınctly. 2017.

[15] C. Z. Deng, Naiyang, Yingjie Tian, Support vector machines: optimization based theory, algorithms, and extensions. 2012.

[16] L. H. Lee, R. Rajkumar, L. H. Lo, C. H. Wan, and D. Isa, "Oil and gas pipeline failure prediction system using long range ultrasonic transducers and Euclidean-Support Vector Machines classification approach," Expert Syst. Appl., vol. 40, no. 6, pp. 1925–1934, 2013.

[17] R. M. Cesar and L. da Fontoura Costa, An introduction to neural networks. CRC press, 2014.

[18] K. Suzuki, Artificial Neural Networks – architectures and applications, no. August. 21 2013.

A Recipe for Social Media Analysis

Shahid Alam
Department of Computer Engineering
Adana Alparsalan Turkes Science and Technology University
Adana, Turkey
salam@atu.edu.tr

Juvariya Khan
Department of Management Sciences
Virtual University of Pakistan
Lahore, Pakistan
juvariya.mak93@gmail.com

***Abstract* — The Ubiquitous nature of smartphones has significantly increased the use of social media platforms, such as Facebook, Twitter, TikTok, and LinkedIn, etc., among the public, government, and businesses. Facebook generated ~70 billion USD in 2019 in advertisement revenues alone, a ~27% increase from the previous year. Social media has also played a strong role in outbreaks of social protests responsible for political changes in different countries. As we can see from the above examples, social media plays a big role in business intelligence and international politics. In this paper, we present and discuss a high-level functional intelligence model (recipe) of Social Media Analysis (SMA). This model synthesizes the input data and uses operational intelligence to provide actionable recommendations. In addition, it also matches the synthesized function of the experiences and learning gained from the environment. The SMA model presented is independent of the application domain, and can be applied to different domains, such as Education, Healthcare and Government, etc. Finally, we also present some of the challenges faced by SMA and how the SMA model presented in this paper solves them.**

Keywords — Social media analysis, Natural language processing, Machine learning, Information retrieval, Data visualization.

I. Introduction

Merriam-Webster dictionary defines social media [1] as, *a form of electronic communication, such as websites and applications used for social networking and microblogging, where users create online communities and groups to share information, ideas, personal messages, and other contents*. Some examples of these platforms are Twitter, Facebook, YouTube, WhatsApp, WeChat, TikTok, LinkedIn, Wikipedia, blogs, and news. The ubiquitous nature of smartphones has significantly increased the use of these social media platforms among the general public. In addition to the general public these platforms are also, used by governments: to interact with citizens, analyze/monitor public opinion and activities, etc. [2]; and use by businesses: for e-commerce, marketing research, communication, sales promotions and discounts, and employee learning, etc.

Modern businesses consider social media as a promising platform to effectively communicate with the targeted customers for conducting marketing and other promotional activities. As an example, Facebook (some of its popular social media subsidiaries also include WhatsApp and Instagram) generated ~70 billion USD in 2019 in advertisement revenues alone, a ~27% increase from the previous year [3]. This indicates, that social media platforms have become a significant portion of digital marketing and will play a major role in the coming years as well.

The way people use and perceive social media defines their powers and practicalities. The use of social media can empower ordinary citizens to conceive ways to sustain the crisis. Coronavirus has restricted face to face social interactions, but due to the innovative use of social media (Facebook and Twitter), farmers in India can connect directly with their end customers, and are finding a way out of this crisis [4]. Social media has also played a strong role in outbreaks of social protests responsible for political changes in different countries [5].

As we can see from the above examples, social media plays a big role in business intelligence and international politics. Although social media contains text, images, and videos, for this paper we restrict the contents to only textual data and discuss the recipes (methods and procedures) only for analyzing the textual data.

Social media analysis (SMA) is a set of techniques and tools to accumulate, aggregate, and systematize social media data to discover and provide significant patterns. The content found on social media platforms typically yields semi-structured or unstructured text. This type of text lacks metadata and is difficult to analyze directly. It typically contains emails, instant messages, and documents, that are user-generated. Since the bulk of text on social media is semi-structured or unstructured, we need to first convert this to a structured data, to manage and analyze and to make sense out of this text. To make decisions based on this data and its analysis (data-driven decisions) we want to present the analysis in a form that makes it easy to interpret. Techniques from different technical areas are used for this purpose, such as natural language processing (NLP), machine learning (ML), information retrieval (IR), statistics and data visualization (DV).

II. Ingredients

Before giving the recipe for SMA, we first discuss the core ingredients (major technical areas), their terminologies and list some of their techniques to better understand the recipe. Here we just give a brief introduction to these areas. For a detailed discussion, review, and explanation, the reader is referred to the respective references for each of these technical areas given in this paper.

A. Natural language Processing

NLP [6], [7] is a branch of artificial intelligence and computational linguistics that deals with the processing of a natural language, such as English, Spanish, French and Chinese, etc. NLP is used to extract important information from natural language input and/or produce natural language output. Different levels/types of language processing in NLP are: phonology – interpreting speech sounds; morphology – words are composed of morphemes, the smallest units of meaning, such as preregistration is composed of the prefix pre, the root registration and the suffixation; lexical – meaning of words; syntactic – uncover the grammatical structure; semantic – meaning of sentences; discourse, meaning of text longer than a sentence; pragmatic – uncover extra meaning, e.g., the context in which the text is used. Different sub-techniques of NLP are used for this purpose. Some of them are, lemmatization, part-of-speech tagging, parsing, morphological and word segmentation, stemming, and lexical semantics. Some of the uses of NLP are, named entity recognition, semantic analysis, opinion mining, topic segmentation, automatic summarization, and text-to-speech.

There is a strong and natural relation between NLP and SMA. Therefore, techniques from NLP form the most important ingredients of SMA.

B. Machine Learning

ML [8], [9] is a branch of artificial intelligence that automatically makes predictions based on a mathematical model build from sample data also called training data. Different approaches used for learning from the training data are supervised learning – the learning is performed on labeled sample data; some of the algorithms used are, support vector machines, linear regression, Naive Bayes and decision trees; unsupervised learning – the learning is performed on unlabeled sample data; some of the algorithms used are, clustering and anomaly detection.

The success of a learning algorithm depends on the data used. Therefore, ML techniques are data-driven and combine basic concepts from computer science, statistics, probability ad optimization. Some of the problems solved by ML are text or document classification, spam detection, network intrusion, fraud detection, and optical character recognition.

Some of the terminologies commonly used in ML are:

- Dataset – Items or instances of data that will be used for learning and evaluation.
- Features – The set of attributes/properties associated with each sample in the dataset.
- Labels – Classes or categories assigned to each sample in the dataset.
- Hyperparameters – Parameters that are specified as inputs to the learning algorithm.
- Training sample – Samples from the dataset used for learning.
- Validation sample – Samples from the dataset used for selecting appropriate hyperparameters. They can be part of the training sample.
- Testing sample – Samples from the dataset used for evaluating the performance of a learning algorithm.
- Hypothesis set – A set of functions (algorithms) mapping features to the set of labels.

The first step in ML is to randomly divide the dataset into training, validation, and testing sets. Next comes the association (selection) of features to samples in the dataset. The selected features are used to train the learning algorithm and tune the hyperparameters. For each value (set) of these parameters, a different hypothesis is selected from the set of hypotheses. In the end, the best performing hypothesis on the validation sample is selected. Finally, the prediction is made on the testing sample using the selected hypothesis. The performance of the algorithm is evaluated using different metrics. Techniques from ML also form the most important ingredients of SMA.

C. Information Retrieval

IR [10] is defined as a set of techniques that are used for finding information in the form of data (usually text) of an unstructured or semi structured nature that satisfies an information requirement from within large sets of data (documents). This large set of data is referred to as the collection or a corpus (a body of text). Some of the important IR models are Boolean, Vector Space, and Latent Semantic Indexing.

A Boolean Model is a model in which documents are set of terms and queries are Boolean expressions. A typical Boolean model contains: a set of words, i.e., the indexing terms either present (1) or absent (0); a Boolean expression; Boolean algebra; a prediction, a document is predicted as relevant if it satisfies the query. Some of the properties of a Boolean model are: it only retrieves exact matches; it is a very simple model based on sets and is easy to implement; it does not provide the ranking of the retrieved documents.

A Vector Space Model represents a set of documents as vectors in a common vector space. The queries are considered as vectors in a high dimensional Euclidean space. The similarity of vectors (a document vector and a query vector) is computed using the cosine similarity (cosine of the angle between them). Different techniques are used to rank the retrieved document, such as term frequency, document frequency, and collection frequency. Some of the properties of a vector space model are: it is a simple model based on linear algebra; it allows partial matching; it provides the ranking of the retrieved documents.

A Latent Semantic Indexing Model is an extension of the vector space model and analyzes the relationship between a set of documents and the terms (words) they contain. Latent semantic indexing assumes that words that are close in meaning will occur in semantically similar texts. A term document matrix is used to store these similarity values, which can be queried to retrieve specific documents. A term document matrix is a 2D matrix that lists the frequency that each term occurs in the documents. The terms are assigned weights using TF-IDF, which assigns more weight to the rare terms. To lower the dimensions and noise, a low-rank approximation [10] is applied to the term-document matrix. This new low-dimensional matrix is then queried to retrieve specific documents.

IR models select and(or) rank the document that is retrieved by a query. Therefore, techniques from IR are mostly used for accumulating the data and form an important part of the ingredients of SMA.

D. Data Visualization

The ability of humans to remember pictures (visuals) far better than words/texts [11], [12] makes humans more visually inclined. It is easy for them to understand and consume certain information in a visual form than words/text. Therefore in today's data-driven world, DV [13] is very critical to efficiently and aptly communicate with humans. DV deals with representing the data graphically. A graphic mark represents a data value or a set of data values. This mapping helps communicate information clearly and efficiently to the users. We can divide these quantitative mappings and relationships into seven types [14]:

- Time-Series – Instances of one or more measures at equidistant points. A line chart may be used to represent such a relationship.
- Correlation – Comparison between two variables to see if they tend to move in the same or opposite direction. A scatter plot may be used to represent such a relationship.

- Ranking – Data values ordered by size or intensity. A bar chart or heat map may be used to represent such a relationship.
- Part-to-Whole – Data values representing parts of a whole. A pie or bar chart may be used to represent such a relationship.
- Deviation – Data values compared against a reference. A bar chart may be used to represent such a relationship.
- Distribution – Count of a variable in a given interval. A histogram or a boxplot chart may be used to represent such a relationship.
- Geospatial – Comparison of data values across a map or location. A network chart may be used to represent such a relationship.
- Nominal Comparison – Comparison of discrete values with no particular order. A bar chart may be used to represent such a relationship.

A typical data visualization process includes: Data Import – extract data from the source; Data Preparation – prepare the data for visualization, e.g., normalizing values and interpolating missing values; Data Manipulation – select the data to visualize, e.g., filtering, joining and grouping; Mapping – map the data to geometric primitives, e.g., points and lines; Rendering – transform the geometric data into visual depiction. Techniques from DV form the final critical ingredients of SMA.

Figure 1 lists some of the techniques used in each of these areas. We only listed the core techniques here. The reader interested in getting more details and explanations about these and other such techniques is referred to the respective references given in this paper for each of the technical areas.

III. Recipe (Model)

We present in Figure 2 a high-level functional intelligence model (recipe) of SMA that synthesizes the input data and uses operational intelligence to provide actionable recommendations. This model also matches the synthesized function of the experiences/learning gained from the environment. The model/process of SMA is divided into three steps (components).

1. Data Identification — Data analysis is as good as the data we are searching in. Therefore it is very important to identify the proper data sources to evaluate. In this step we are basically answering the questions about whose opinions or ideas/thoughts we are interested in, where the conversations are happening, and do we need to just look in the current or past (when) conversations.
2. Data Analysis — After collecting the data, now we want to answer some questions related to the data, such as what are the opinions of customers about a certain product of a company, or how to rate the evaluations/ratings of public about a politician running for the presidency of a country. As mentioned before social media platforms yield semi-structured or unstructured text data. We first convert this to structured data and then perform analysis making sense out of this data and trying to answer the questions related to the data.
3. Information Interpretation — So now we have performed some analysis on the data. How to interpret this analysis, or present it in a form that helps people make some important decisions. For example when, where and what information to disseminate to increase the exposure of the prospective customers to a company's products.

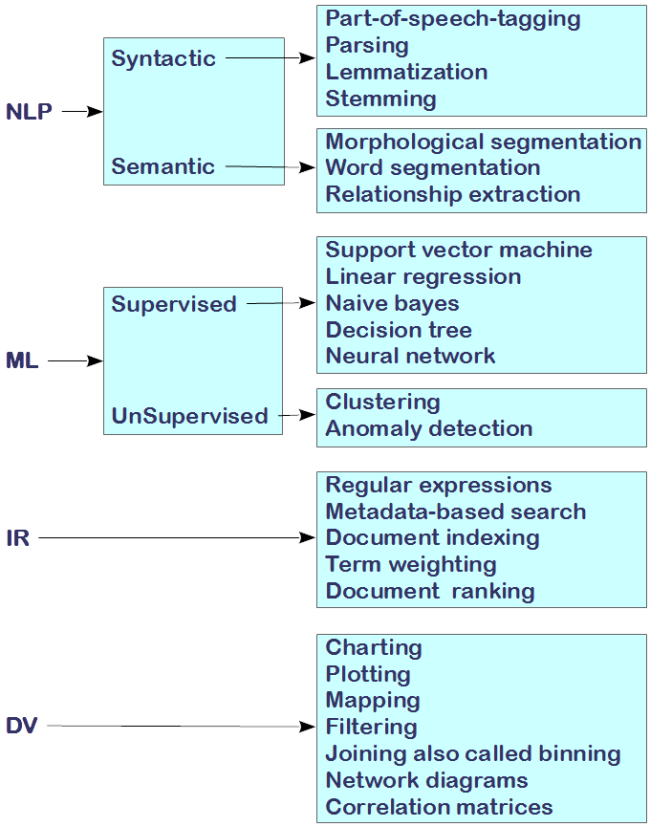


Fig 1. List of some of the core techniques used in each of the technical areas (NLP = Natural language processing, ML = Machine learning, IR = Information retrieval and DV = Data visualization) discussed in section II. Reader interested in getting more details and explanation about these and other such techniques is referred to the respective references given in this paper for each of the technical areas.

Depending on the type of analysis, the SMA model shown in Figure 2 uses one or a combination of techniques from the technical areas discussed in section II. Some of these core techniques are listed in Figure 1. A complete use case of this SMA model is given below.

For example, to increase the sales of a company we would like to analyze and evaluate the customers' and consumers' views about the company's products and services. As the first step in the SMA model presented in this paper, we will identify the data. Depending on the presence of the company on different social media platforms, this step may include retrieving relevant information from Twitter and Facebook, etc. In the second step we will perform data analysis. It may involve performing opinion mining and trend analysis using different (syntactic and semantic) NLP techniques. Different ML techniques will be used to learn from the opinions and trends of the customers and consumers. As a result of this, we will get the most important opinions and trends, which will be used by the company's executives and managers to make important decisions to increase the sales of the company. As

the last step, the results will be visualized, using different visualization techniques, to ease the comprehension and better decisions by executives and managers. These decisions may involve changing, updating, and removing some of the company's products.

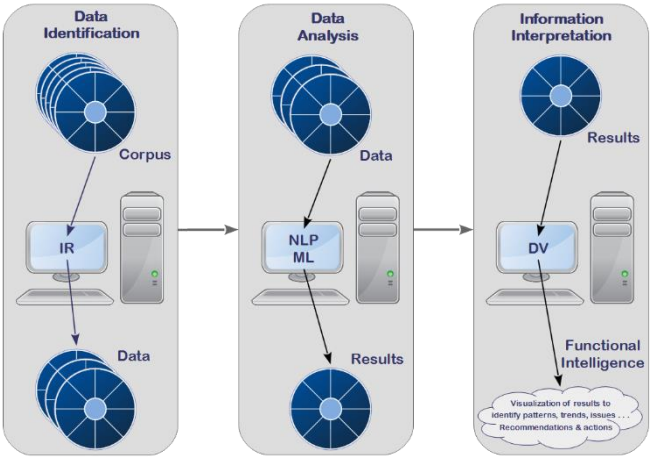


Fig 2. A functional intelligence model of Social Media Analysis divided into three major components, data identification, data analysis and information interpretation. IR = Information Retrieval, NLP = Natural Language Processing, ML = Machine Learning and DV = Data Visualization.

A. *Applications*

In this section, we give some applications of the SMA model, i.e., how this model can be used in some of the different domains. Here, we just describe three of them.

a) Education: For data identification and retrieval, we would like to take observations, interviews, and the analysis of a participant's products such as articles, blogs, and writeups on social sites, opinions shared on, for example, Twitter, Facebook, and Instagram. For instance, if a school administration wants to find the difficulties the teachers and students are facing in the old curriculum, what are their pain points and which topics are causing worry. Also, we can target the audience from specific groups such as students from schools, colleges, or universities. We can analyze this data using different techniques as described in Figure 1. This way, at a much quicker rate we might get the feedback than the traditional process of adapting a lot of evaluations. As a result, we will get opinions for those topics which are causing frustration for students and teachers. These opinions and issues will be used by higher authorities in the administrative department to visualize whether all the questions and considerations have come to a productive point before planning new strategies. Similarly, SMA can also help in determining the topic of interests that can help educators to develop more personalized learning programs.

b) Healthcare: SMA can be applied for enhancing the health and safety of the public and to support important issues regarding healthcare, welfare, and social safety of the society. As an example, for data identification and retrieval, public health organizations can track posts that talk about the health of people and can get a sense of the severity of the symptoms in real-time. In another case, we can track conversations to know how the public is responding to emergent health issues. To get the general public thinking we would use social data while applying SMA. By monitoring the public talks, we can positively respond to a situation for instance, if we are facing an outbreak, so by putting more resources to deal with spreading health problems. As a final step, social data gives a track to collaborate and finalize our findings and to measure perception about the illness. We can analyze this data using different techniques as described in Figure 1 and get a real-time response from the health authorities. SMA can be used by health professionals to connect and engage their audience in real-time. Visualizing this analysis, they can influence and build strong relationships with their audience.

c) Government: SMA has been successfully applied by governments to improve governance and to create a strong relationship with its citizens. For data identification retrieval, governments can build social communities such as a page on Facebook and a handle on Twitter, etc. This way conversations can be monitored to create better relationships between the government and citizens. As an example, a government put forwards a proposal and take a referendum approach where voters are asked to vote. For this purpose government would provide a questionnaire or at least two responses for yes or no just to analyze how intelligible the question is and how people would respond to these referendum questions. Easy to understand, clear, and to the point. The proposal will be analyzed and measured, such as considering a time frame, unit of measure, and what factors should be included. Finally, gathered data would be sorted out in a different way such as plotting or finding correlations or creating tables on an Excel sheet. There are other visualizing tools that are helpful in this context. Lastly, the researcher would interpret the findings whether the collected data answered the required question, or helped in defending objections and how. These questions would be of great help for the higher authorities for decision making, approaching better strategies, and to decide the best course of action.

IV. Challenges Facing SMA

There are various applications and many advantages of SMA as discussed in this paper, but implementing a successful SMA is full of challenges. In this section, we briefly discuss some of these challenges and their solutions.

A. *Unstructured Data*

Unstructured data [15] is the raw data that is not categorized and is in different formats. It is commonly found in text messages. Making sense out of text (words not numbers) is very challenging. For example the word bad may mean good depending on the context, relationship and other variables. A correct semantic analysis, including word segmentation and relationship extraction, is required to properly understand the meaning of a word in such a sentence. Other techniques such as sentiment analysis, pattern recognition, and cognitive analysis are very helpful in solving

such problems. To solve some of the problems with unstructured data the SMA model presented in this paper uses such techniques from NLP and ML technical areas.

B. Real-time Analytics

One of the main purposes of SMA is to provide quick (real-time) meaningful insights into the data that percolates into an organization to affect strategic planning. Moreover, some of the data on social media changes instantaneously, especially when an event is unfolding. So to be effective, organizations, governments, and businesses should base their plans on real-time SMA [16], [17]. A high signal to noise ratio in social media data makes it difficult to perform and provide real-time analysis. Some of the solutions to this problem are, to use high-end machines, distributed systems, such as cloud computing, and parallelize the analysis as much as possible. The SMA model presented in this paper can be easily ported to a cloud computing environment, and the techniques used can also be parallelized to improve the speed of computation. Moreover, some of the social media platforms provide APIs, such as Streaming API from Twitter, to retrieve data in real-time.

C. Big Data

The data on social media shares some of the same characteristics as of big data. Some of the common challenges of big data are [18]: volume – the space required for storage; velocity – the speed of data creation; variety – taking many different forms; veracity – data integrity and authenticity; The solution to some of these challenges are already provided by big data analysis techniques [19] and can also be applied to data on social media.

D. Data Quality

Evaluating the quality of the data on social media is always a concern for SMA. A lot of times the information provided is fake. Also, social media is cluttered with fake and duplicate accounts/profiles. Furthermore, the data may not reliable because of some exaggerations or extra information added (to make it more interesting for the readers) to the data. We can use similar techniques for evaluating the quality (believability, validity, and relevancy, etc.) of data on social media as applied to big data [20].

E. Visualization

It is very crucial to visualize the data properly when correct decisions need to be taken quickly and efficiently. For effective disaster recoveries, a clear and concise representation of the data needs to be presented to the decision-makers in emergency management services. Traditional visual analysis methods only deal with structured, low volume, and uniform data. The volume, variety (different formats such as textual and geo-data), and fast-changing rate of social media data make it difficult to visualize. Recently different interactive and multimedia techniques [21]–[23] have been proposed to solve this problem. Social media data is noisy and lacks quality. Therefore, visual analytics methods [24] that help provide information about uncertainty and other quality issues will be in high demand.

V. CONCLUSION

In this paper, we have highlighted the importance of SMA and presented a functional intelligence model to successfully implement SMA in any domain. We have also described its applications in different domains and some of the challenges faced by SMA. Currently, we are working on implementing this model. In the future, we will carry out an empirical study to apply this model in different domains and evaluate its effectiveness and performance using various metrics.

REFERENCES

[1] Merriam-Webster, "Definition of social media," 2021.

[2] G. F. Khan, Social media for Government: A Practical Guide to Understanding, Implementing, and Managing Social Media Tools in the Public Sphere. Springer Nature Singapore Pte Ltd, 2017.

[3] J. Clement, "Facebook's advertising revenue worldwide from 2009 to 2019," 2020.

[4] M. Newton, How Social Networks Are Helping Indian Farmers Hit By the Coronavirus Lockdowns. https://en.reset.org/blog/howsocial-networks-are-helping-indian-farmers-hit-coronavirus-lockdowns-06102020, 10 July 2020.

[5] P. N. Howard, A. Duffy, D. Freelon, M. M. Hussain, W. Mari, and M. Maziad, "Opening closed regimes: what was the role of social media during the arab spring?," Available at SSRN 2595096, 2011.

[6] C. D. Manning and H. Sch¨utze, Foundations of statistical natural language processing. MIT press, 1999.

[7] E. D. Liddy, "Natural language processing. In encyclopedia of library and information science," Marcel Decker Inc, NY, 2001.

[8] E. Alpaydin, Introduction to machine learning, fourth edition. MIT press, 2020.

[9] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning-Book, second edition. The MIT Press, 2018.

[10] C. D. Manning, P. Raghavan, and H. Sch¨utze, Introduction to information retrieval. Cambridge university press, 2008.

[11] A. Paivio, T. B. Rogers, and P. C. Smythe, "Why are pictures easier to recall than words?," Psychonomic Science, vol. 11, no. 4, pp. 137–138, 1968.

[12] D. L. Nelson, V. S. Reed, and J. R. Walling, "Pictorial superiority effect.," Journal of experimental psychology: Human learning and memory, vol. 2, no. 5, p. 523, 1976.

[13] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: A survey," The VLDB Journal, vol. 29, no. 1, pp. 93–117, 2020.

[14] S. Few, "Selecting the right graph for your message," 2018.

[15] J. Cote, J. H. Salmela, A. Baria, and S. J. Russell, "Organizing and interpreting unstructured qualitative data," The sport psychologist, vol. 7, no. 2, pp. 127–137, 1993.

[16] I. Lee, "Social media analytics for enterprises: Typology, methods, and processes," Business Horizons, vol. 61, no. 2, pp. 199–210, 2018.

[17] B. Yadranjiaghdam, N. Pool, and N. Tabrizi, "A survey on real-time big data analytics: applications and tools," in 2016 international conference on computational science and computational intelligence (CSCI), pp. 404–409, IEEE, 2016.

[18] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger, "Social media analytics–challenges in topic discovery, data collection, and data preparation," International journal of information management, vol. 39, pp. 156–168, 2018.

[19] C. P. Chen and C.Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," Information sciences, vol. 275, pp. 314–347, 2014.

[20] A. Immonen, P. Paakkonen, and E. Ovaska, "Evaluating the quality of social media data in big data architecture," Ieee Access, vol. 3, pp. 2028–2043, 2015.

[21] J. Chae, D. Thom, Y. Jang, S. Kim, T. Ertl, and D. S. Ebert, "Public behavior response analysis in disaster events utilizing visual analytics of microblog data," Computers & Graphics, vol. 38, pp. 51–60, 2014.

[22] S. Chen, X. Yuan, Z. Wang, C. Guo, J. Liang, Z. Wang, X. Zhang, and J. Zhang, "Interactive visual discovering of movement patterns from sparsely sampled geo-tagged social media data," IEEE transactions on visualization and computer graphics, vol. 22, no. 1, pp. 270–279, 2015.

[23] T. Mei, Y. Rui, S. Li, and Q. Tian, "Multimedia search reranking: A literature survey," ACM Computing Surveys (CSUR), vol. 46, no. 3, pp. 1–38, 2014.

[24] Y. Wu, N. Cao, D. Gotz, Y.-P. Tan, and D. A. Keim, "A survey on visual analytics of social media data," IEEE Transactions on Multimedia, vol. 18, no. 11, pp. 2135–2148, 2016.

Self HARMing Malware - An Adaptive IoT Honeypot for Automated, Repetitive Malware

Seamus Dowling
Department of Computing, Mayo Campus, Galway Mayo Institute of Technology, Mayo, Ireland
seamus.dowling@gmit.ie

***Abstract*—Internet-of-Things (IoT) contains physically constrained devices which impacts on the security of IoT deployments. This can make them vulnerable to dedicated IoT attack software such as Mirai and increases the attack surface available to IoT malware developers. This article describes the functionality, deployment and assessment of an adaptive IoT Honeypot for Automated, Repetitive Malware (HARM). It presents findings from the live deployment of HARM that uses reinforcement learning to learn from attack interactions. It also demonstrates that dedicated IoT honeypot datasets contain attack information that is repetitive and automated. Standard honeypot deployments have scripted responses which can terminate attack interactions. This article enhances standard honeypots by exploiting malware characteristics of automation and repetition. It identifies that the learning evolution of HARM correlates with the discovery of the entire attack sequence. It also uses the captured dataset to assess reinforcement learning policies so as to optimise further HARM deployments.**

***Index Terms*—Honeypot, Reinforcement Learning, Adaptive, Internet of Things**

I. Introduction

The security of data collected and transmitted by Internet-of-Things (IoT) devices depends on the physical resources available to those devices. Full function devices (FFD) and reduced function devices (RFD) will have constrained resources when implementing services such as security. This leaves them vulnerable to compromise and increases the attack surface for malware [1]. It is estimated that there will be over 20 billion IoT devices connected to the Internet in 2020 [2]. Large volumes of data processed by these devices give rise to new challenges and privacy concerns [3]. Already attack code specifically targeting IoT devices has been made available, spawning new variants and using new attack vectors and methods [4]. Honeypots can play a reactive role in analysing the methods used by new malware [5] but are typically deployed to gather information for retrospective analysis. This can make them redundant for rapid detection of new malware variants as the data collection process is longitudinal [6]. New honeypot methods need to be considered to adapt to the emerging threat posed by increased IoT deployments. This article first describes an IoT honeypot deployment. The resultant dataset contains predominately repetitive data of automated malware methods [7]. The article then describes the integration of reinforcement learning into the existing IoT honeypot technology to exploit these characteristics, namely automation and repetition. The resultant adaptive IoT honeypot, a **H**oneypot for **A**utomated, **R**epetitive **M**alware (HARM) is deployed to attract and capture attack sequences targeting IoT devices. It demonstrates that the adaptive IoT HARM uses these malware characteristics to learn the best actions to take when interacting with a known IoT bot. By doing so it expeditiously determines the entire attack command sequence. The attack dataset is then used to evaluate reinforcement learning policies so as to assess the performance of HARM and optimise it for future deployments. Our proposed state action space formalism is designed to target automated and repetitive malware. Thus we propose the following contributions over existing work:

- Demonstrate that the learning evolution of HARM, correlates with key increases in attack commands and culminates with the realisation of the entire attack sequence.
- The captured automated and repetitive dataset can be used in a controlled environment to evaluate reinforcement learning policies so as to optimise future adaptive IoT HARM deployments.
- Q-Learning outperformed SARSA as a learning agent on HARM, in controlled experiments.

II. Evolving IoT Threats

As the Internet evolves to include IoT, society will be connected in more valuable and relevant ways. A network of combined devices can impact society with the deployment of sensors to monitor societal ecosystems. Optimising power grids, controlling traffic flows, monitoring pollution and urban environmental ecosystems create smart cities for the betterment of its population. Heterogeneous technologies will require integration at a physical level. Data collected and collated will require transmission in a compatible format and then collated in a manner that ensures relevant mining. Information from IoT systems will come from a multitude of sources. As well as smart objects from wireless sensor networks (WSN) and industrial IoT installations, society itself will contribute to the collection of data. Data captured will require a communications stack built onto the electronics, to transmit and receive as part of a network [8] and provide reachability to all connected devices. Data, security and communications all differ for smart objects within these components. IEEE defines two device types that can participate in an 802.15.4 network; a full-function device (FFD) and a reduced-function device (RFD). An FFD is capable of serving as a personal area network (PAN) coordinator or a coordinator. An RFD is not capable of serving as a PAN coordinator, has minimal resources and memory capacity, and is intended

for simple applications. Communications capabilities for smart objects will range from very basic to highly complex. Security measures implemented will depend on the objects processing capabilities, leaving some objects more vulnerable than others. Security of communications and data, and access and participation are critical components to IoT deployments. Encryption techniques such as PKI are often too power and resource hungry for IoT devices therefore alternatives need to be considered [9]. Previous research on IoT honeypots highlight the diversity of IoT objects [10]. Concepts such as smart cities give rise to issues concerning the privacy of their citizens and the use of data [3]. Information stored and communicated about location, social activity and profiling need to be secure to engender trust and promote the use of smart city applications. The diversity of data from objects require secure storage and meaningful integration [11]. All smart objects and user end devices require the capabilities to securely exchange data between each other, coordinating devices, gateways and subsequent storage locations. All objects and communication channels are open to probing and compromise. The US Department of Homeland Security produced a risk assessment for cyber-physical systems (CPS) in a smart city [12], highlighting potential vulnerabilities for CPS as integral parts of smart city infrastructure. The Industrial IoT (IIoT) Consortium published a security framework and an approach to assess cybersecurity in IIoT systems [13].

III. Evolving Honeypots

A. Standard Honeypots

A honeypot is an analytical tool and its role is to deceive and collect attack information. This information can be analysed retrospectively to determine the modus operandi of attackers. Honeypots can have low or high interaction. Provos [14] in 2003 presented Honeyd, an easy to deploy, low risk honeypot. It details how to deploy virtual honeypots with different IPs safely. Honeyd acted as a catalyst for the development of further low interaction honeypots. Nepenthes [15] and Argos [16] became very popular global honeypot tools. Honeypots such as Kippo, provide backend databases to collect all activity such as IP addresses, timestamp, attempts, interactions, commands, downloads and executions [17]. Downloaded files can be sandboxed and analysed [18]. After an attacker has compromised a honeypot, it will attempt to interact in a structured manner [19]. The initial engagement for an attack, post compromise, is to examine the hardware and software to determine if progression is relevant. On a live production system, this will return the underlying architecture, CPU, uptime, operating system, user privileges and further relevant information. An attack sequence may then attempt to modify the host system. On a honeypot, engaging the attack sequence at this point prolongs activity. It then attempts to download, install and run malware to complete the compromise. There are often legal and ethical issues associated with operating honeypots [20]. In a desire to gather as much information as possible on attacker behaviour, a honeypot could allow the execution of malicious code [21].

B. Adaptive Honeypots

The operation of standard honeypots highlights the need for creating honeypots that prolong attacker interaction for an optimum time period, without breaching legal or ethical responsibilities. Machine learning techniques have previously been applied to honeypots datasets for retrospective analysis of captured datasets. Supervised and unsupervised methods are used to model malware interaction and to classify attacks [22], [23]. To be truly adaptive, a honeypot needs to proactively engage with an attack sequence in real-time. The application of machine learning for proactive engagement has not been fully explored. Wagener [24] and Pauna [25] used reinforcement learning to extract as much information as possible about the intruder. Their honeypots were developed to engage a human attacker by blocking commands, returning error messages and issuing insults.

C. IoT Honeypots

The popularity of IoT deployments has attracted malware targeting end devices [26]. Recently, the Mirai botnet spawned multiple variants targeting IoT devices through various attack vectors [4]. New IoT versions of honeypots are being designed continuously to capture new IoT malware on different attack vectors. IoTPot is a bespoke honeypot designed to analyze malware attacks targeting IoT devices [27]. ConPot is a SCADA honeypot developed for critical IIoT architectures [28]. The repetitive and automated nature of the malware is visible on a smaller dataset from a IoT honeypot [7]. ThingPot [29], simulated vulnerable end devices. IoTCandyJar presented as a range of industrial, commercial and end devices which could be accessed across multiple attack vectors [30].

IV. Automation and Repetition

Honeypot deployments are longitudinal and the resultant dataset contains large amounts of repetitive data. This allows researchers to ascertain patterns in the behaviour over a period of time. Global honeypots often operate for long periods with a view to collecting large datasets. Temporal variances and IP blocks provide interesting insight into the propagation methods used by bots and botnets. This diurnal pattern was evident in the captured dataset from an IoT honeypot deployed over 3 months [7]. The IoT honeypot simulated a ZigBee gateway on a SSH attack vector. The attack types can be identified by sandboxing the downloaded files, observing shell interactions and consulting threat advisories. Attack types were examined and collated and 99.6% of the traffic encountered on the honeypot was automated. They were identified as *Dictionary*, *Recon*, *Failed*, *Launch* commands, *XOR DDoS* and *BillGates*. They demonstrated automated methods to gather information on variables such as compilers, CPU and operating systems.

V. HARM Methodology

The development of HARM involves 2 stages:

- **Implementation:** Integrating reinforcement learning into an IoT honeypot

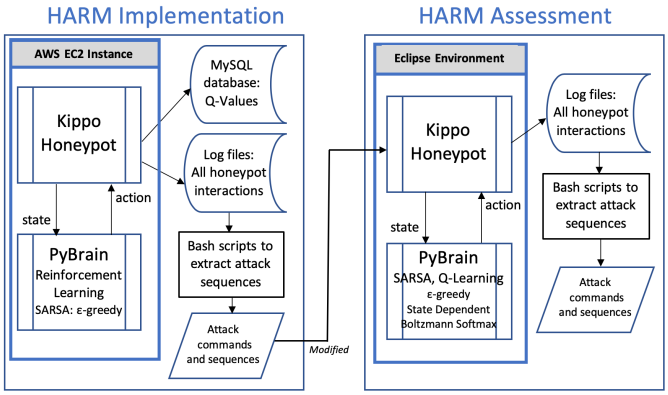


Fig. 1. Implementing and Assessing HARM

- **Assessment:** Deploying and assessing the performance of HARM

Together these two stages use the very characteristics of malware, automation and repetition to assess the performance of HARM in order to improve subsequent deployments. Fig. 1 presents the 2 stages as a concept.

A. Implementation

Our proposed **H**oneypot for **A**utomated, **R**epetitive **M**alware involves the integration of reinforcement learning into the existing IoT honeypot technology to exploit the automated and repetitive characteristics of malware. The reinforcement learning state/action space and reward function is designed to increase the number of commands from the attack sequence.

Reinforcement learning is a machine learning technique in which a learning agent learns from its environment, through trial and error interactions. Rather than being instructed as to which action it should take given a specific set of inputs, it instead learns based on previous experiences as to which action it should take in the current circumstance. Throughout its deployment, HARM is considered to be a server environment with integrated reinforcement learning and SSH as an access point. Within this environment, the server has states that are examined and changed with bash scripts. Examples are *iptables*, *wget* and *sudo*. The reinforcement learning agent can perform actions on these states such as to allow, block or substitute the execution of the scripts. The environment issues a reward to the agent for performing that action. The agent learns from this process as the honeypot is attacked and over time learns the optimum policy π*, mapping the optimal action to be taken each time, for each state s. The learning process will eventually converge as the honeypot is rewarded for each attack episode. This temporal difference method for on-policy learning uses the transition from one state/action pair to the next state/action pair, to derive the reward. *State, Action, Reward, State, Action* also known as SARSA, is a common implementation of on-policy reinforcement learning (3). The reward policy Q is estimated for a given state s_t and a given action a_t. The environment is explored using a random component ϵ or exploited using learned Q values. The estimated Q value is expanded with a received reward r_t plus an estimated future reward $Q(s_{t+1}, a_{t+1})$, that is discounted (γ). A learning rate parameter is also applied (α).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (1)$$

As each attack is considered an episode, the policy is evaluated at the end of each episode. SARSA is implemented with ϵ-greedy policy = 0.10, discount factor γ=1 and step size α=0.5. More detailed reward functionality and comparison with previous research for HARM is available [31].

B. Assessment

The captured dataset can be used to evaluate the policies of reinforcement learning. The command sequences in the dataset can function as an input stream into the adaptive honeypot within a controlled offline Eclipse development environment [1]. The input stream is parsed by a Python action daemon to reflect malware interactions captured by the online deployment. HARM was deployed on the Internet using SARSA (1). Sutton and Barto [32] presents algorithms for both SARSA and Q-Learning. Algorithm 1 shows that SARSA chooses s_{t+1} and a_{t+1} prior to updating the Q function. Algorithm 2 shows Q-Learning which first updates the Q function and then chooses the next action based on the updated Q function.

Algorithm 1 Reinforcement Learning SARSA

Initialise $Q(s,a)$ arbitrarily
repeat (for each episode):
Initialize s
Choose a from s using policy derived from Q (e.g., ε-greedy)
repeat
Take action a, observe r, s_{t+1}
Choose a_{t+1} from s_{t+1} using policy derived from Q (e.g., ε-greedy)
$Q(s_t, a_t) \leftarrow Q(s_t,a_t)+\alpha[r + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t, a_t)]$
$s \leftarrow s_{t+1}$, $a \leftarrow a_{t+1}$;
until s is terminal

Algorithm 2 Reinforcement Learning Q-Learning

Initialise $Q(s,a)$ arbitrarily
repeat (for each episode):
Initialize s
repeat
Choose a from s using policy derived from Q (e.g., ε-greedy)
Take action a, observe r, s_{t+1}
$Q(s_t, a_t) \leftarrow Q(s_t,a_t)+\alpha[r + \gamma \max_{a+1} Q(s_{t+1},a_{t+1}) - Q(s_t, a_t)]$
$s \leftarrow s_{t+1}$;
until s is terminal

HARM's implementation of SARSA is provided by PyBrain [33]. PyBrain is a machine learning library which provides for the definition of the state/action space. Within PyBrain, the implementation of Algorithms 1 and 2 are facilitated with Python code as follows:

[1] www.eclipse.org

```
from pybrain.rl.agents import LearningAgent
from pybrain.rl.learners import Q,SARSA
from pybrain.rl.explorers import EpsilonGreedyExplorer

import threading

class RL:
    def __init__(self):
    learner = SARSA()
    learner._setExplorer(EpsilonGreedyExplorer(0.1))
```

Fig. 2. Adaptive Honeypot Reinforcement Learning Selection

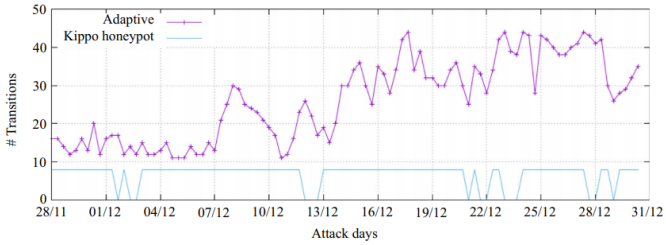


Fig. 3. Attack Command Sequence Realisation

- **Algorithm 1:** *self.module.updatevalue(self.laststate, self.lastaction, qvalue + self.alpha * (self.lastreward + self.gamma * qnext - qvalue))*
- **Algorithm 2:** *self.module.updatevalue(self.laststate, self.lastaction, qvalue + self.alpha * (self.lastreward + self.gamma * maxnext - qvalue))*

For this code to function, the adaptive honeypot stores variables for the current state, action and reward. These are passed to PyBrain to select the best action. This action is subsequently passed back to HARM to perform that action and update the Q function. As part of the assessment process, PyBrain also provides for learning policies ϵ-*greedy*, *Boltzmann Softmax* and *State Dependent*

HARM can import and use SARSA or Q-Learning from PyBrain. It can also specify the ϵ-greedy explorer component. These learning policies can be coded as *pybrain.rl.explorers*, as seen in fig. 2. This flexibility within the controlled environment provides for the assessment of the performance of HARM. By modifying the variables (SARSA, Q-Learning, ϵ-greedy, ϵ-greedy values, Boltzmann Softmax, State dependent) within a controlled environment and streaming a captured dataset modified to mirror malware methods, the performance of HARM can be assessed.

VI. Results

The adaptive IoT HARM was developed to generate rewards on 75 states and is freely available [34]. Amazon Web Services (AWS) EC2 was used to facilitate the Internet facing honeypots. Kippo, PyBrain, MySQL and other dependencies were installed on the adaptive IoT honeypot EC2 instance. It was accessible through SSH for a period of 30 days (Nov/Dec) and immediately started to record repetitive malware activity. A standard Kippo honeypot was deployed simultaneously for comparison purposes.

A. Learning Evolution and Attack Sequence Realisation

Initially HARM logged dictionary, bruteforce and failed attempts. These are excluded as they represent pre-compromise interactions. Thereafter it captured other malware traffic including a Mirai bot variant which became the dominant automated and repetitive attacking tool, averaging over 5 attacks per day.

HARM's reward function is designed to prolong interaction and by default, increase the number of commands in an attack sequence (1). It calculated the reward values as it encountered each command in the attack sequence. These reward values were extracted and collated for 100 distinct attack events. The log files capturing the interactions as seen on fig. 1, highlighted key points in the malware interaction. On the 8th, 12th and 17th December for example, the adaptive IoT HARM substituted a result for *cat* command, blocked a *mount* command or blocked an *echo* command causing the interaction with the attack sequence to increase. The honeypot continued to learn until attack 60 when it allowed all commands in the sequence to be executed (17th December). The adaptive HARM learning was also affected by the other infrequent malware types also encountered by HARM. This impacted upon the realisation of the entire attack sequence for the dominant Mirai variant. Fig. 3 presents the learning evolution and realisation of the command sequence for this Mirai variant. A standard Kippo honeypot was deployed at the same time and also encountered the Mirai variant. It only ever executed the first 8 commands in the sequence before interaction terminated. It is included in fig. 3 for comparison purposes and demonstrates the improved functionality of HARM in prolonging interaction, realising entire attack sequences and ultimately capturing more relevant datasets for analysis.

B. HARM Assessment

To compare the performance of SARSA and Q-Learning, the dataset from section VI-A was streamed twice through HARM in a controlled, offline Eclipse environment. The input stream is parsed by a Python action daemon to reflect malware interactions captured by the online deployment. For the first streaming, the learner variable in fig. 2 was set to SARSA, the second time with it set to Q-Learning. ϵ-greedy remained constant at 0.1 for both. The honeypot captured the interactions, actions and number of commands in the attack sequence that represent transitions from s to s_{t+1}. These transitions were collated allowing for the comparison of SARSA and Q-Learning performance on the adaptive honeypot. Fig. 4 demonstrates the result of using two sets of variables (ϵ-greedy=0.1 for SARSA and Q-Learning). SARSA performed better initially having more interactions with the data stream. However Q-Learning was the first to realise the entire 44 command attack sequence at attack 63 compared with SARSA at attack 66.

The assessment process uses SARSA and Q-Learning with a set ϵ-greedy value. This process could be expanded to create combinations of SARSA, Q-Learning, ϵ-greedy, ϵ-

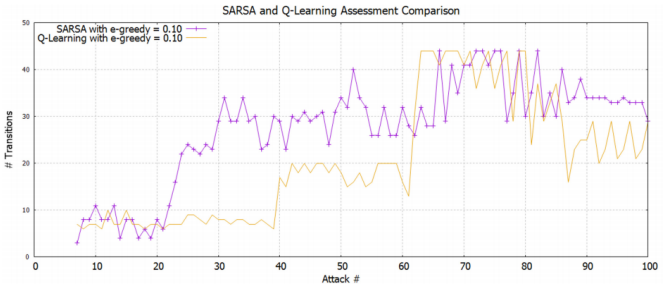


Fig. 4. Performance of SARSA and Q-Learning

greedy values, Boltzmann Softmax and State dependent. This could be performed quickly in the controlled environment to determine the optimum combination against a trending variant. Subsequent timely deployments of HARM are then optimised to adapt and learn from current trending automated and repetitive malware.

VII. Conclusions

Malware operations are predominately automated and repetitive. Bots provide a mechanism to propagate, compromise hosts and communicate with command and control (C&C). Human social engineering may play a part in facilitating infection of the end devices and botmaster communication. But the global process of generating a botnet of compromised hosts is highly automated and highly repetitive.

The design, deployment and findings of HARM presented in this paper turn those very characteristics of automation and repetition upon the malware itself. It identifies that these traits can be used to facilitate learning on the adaptive IoT honeypot and quickly realise an entire attack stream. There are many tools that act as early warning systems for new malware such as zero-day attacks. Intrusion detection and prevention systems can flag and defend against anomalies. Honeypots have proven efficacy in capturing and analysing emerging malware threats. It has been pointed out that honeypots played a role in originally identifying the Mirai bot. As more IoT devices become connected to the Internet, new attack vectors emerge. Constrained resources on intermediate and end devices provide malware developers with a fertile ground for exploitation. To counter this, new methods of detection need to be considered. Methods that can expedite the analysis and identification of attacks, will improve overall security of IoT deployments and the Internet.

Further deployment and assessment are required to observe normal malware interactions, to determine if HARM demonstrates improved functionality against all malware types. Future work on HARM would also have it identify attack types by initial command sequences, running separate reinforcement learning processes for each type and maintaining multiple reward datasets. The assessment of HARM identified Q-learning as being more efficient when realising the entire attack sequence, within a controlled environment. It would be relevant for future work to see if this is specific to a particular bot command stream or if this is the case for all malware attacks.

References

[1] Xu, L., Jiang, C., Wang, J., Yuan, J., Ren, Y., 2014. Information security in big data: privacy and data mining. Ieee Access 2, 1149–1176

[2] van der Meulen, R., 2017. 6.4 Billion Connected Things Will Be in Use in 2016. Technical Report. Gartner. https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016. Accessed July 27, 2018.

[3] Bello-Orgaz, G., Jung, J.J., Camacho, D., 2016. Social big data: Recent achievements and new challenges. Information Fusion 28, 45–59.

[4] Antonakakis, M et al., 2017. Understanding the mirai botnet, in: 26th USENIX Security Symposium (USENIX Security 17), pp. 1093–1110.

[5] Kolias, C., Kambourakis, G., Stavrou, A., Voas, J., 2017. Ddos in the iot: Mirai and other botnets. Computer 50, 80–84.

[6] Watson, D., Riden, J., 2008. The honeynet project: Data collection tools, infrastructure, archives and analysis, in: 2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing, IEEE. pp. 24–30

[7] Dowling, S., Schukat, M., Melvin, H., 2017. A zigbee honeypot to assess iot cyberattack behaviour, in: Signals and Systems Conference (ISSC), 2017 28th Irish, IEEE. pp. 1–6.

[8] Mottola, L., Picco, G.P., 2011. Programming wireless sensor networks: Fundamental concepts and state of the art. ACM Computing Surveys (CSUR) 43, 19.

[9] Yan, T., Wen, Q., 2012. A trust-third-party based key management protocol for secure mobile rfid service based on the internet of things, in: Knowledge Discovery and Data Mining. Springer, pp. 201–208

[10] Dowling, S., Schukat, M., Melvin, H., 2017a. Data-centric framework for adaptive smart city honeynets, in: Smart City Symposium Prague (SCSP), IEEE. pp. 1–7.

[11] Su, K., Li, J., Fu, H., 2011. Smart city and the applications, in: International conference on electronics, communications and control (ICECC), IEEE. pp. 1028–1031.

[12] Dept. of Homeland Security, 2015. The Future of Smart Cities: Cyper-Physical Infrastructure Risk. Technical Report. Department of Homeland Security. https://ics-cert.us-cert.gov/Future-Smart-Cities-CyberPhysical-Infrastructure-Risk. Accessed June 10, 2020.

[13] Consortium, I.I., 2016. Industrial Internet of Things Volume G4: Security Framework. Technical Report. Industrial Internet Consortium. http://www.iiconsortium.org/IISF.htm. Accessed May 3, 2019.

[14] Provos, N., 2003. Honeyd-a virtual honeypot daemon, in: 10th DFN-CERT Workshop, Hamburg, Germany, p. 4.

[15] Baecher, P., Koetter, M., Holz, T., Dornseif, M., Freiling, F., 2006. The nepenthes platform: An efficient approach to collect malware, in: International Workshop on Recent Advances in Intrusion Detection, Springer. pp. 165–184.

[16] Portokalidis, G., Slowinska, A., Bos, H., 2006. Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation, in: ACM SIGOPS Operating Systems Review, ACM. pp. 15–27.

[17] Valli, C., Rabadia, P., Woodward, A., 2013. Patterns and patter-an investigation into ssh activity using kippo honeypots pp. 1–6.

[18] Provataki, A., Katos, V., 2013. Differential malware forensics. Digital Investigation 10, 311–322.

[19] Ramsbrock, D., Berthier, R., Cukier, M., 2007. Profiling attacker behavior following ssh compromises, in: 37th Annual IEEE/IFIP international conference on dependable systems and networks (DSN'07), IEEE. pp. 119–124.

[20] Rubin, B.S., Cheung, D., 2006. Computer security education and research: handle with care. IEEE security & privacy 4, 56–59

[21] McCarty, B., 2003. The honeynet arms race. IEEE Security & Privacy 99, 79–82

[22] Hayatle, O., Otrok, H., Youssef, A., 2013. A markov decision process model for high interaction honeypots. Information Security Journal: A Global Perspective 22, 159–170.

[23] Ghourabi, A., Abbes, T., Bouhoula, A., 2014. Characterization of attacks collected from the deployment of web service honeypot. Security and Communication Networks 7, 338–351.

[24] Wagener, G., Dulaunoy, A., Engel, T., et al., 2011. Heliza: talking dirty to the attackers. Journal in computer virology 7, 221–232

[25] Pauna, A., Bica, I., 2014. Rassh-reinforced adaptive ssh honeypot, in: (COMM), 2014 10th International Conference on Communications
[26] Kapersky, 2018. New iot-malware grew three-fold in h1 2018. https://www.kaspersky.com/about/press-releases/2018 new-iotmalware-grew-three-fold-in-h1-2018. Accessed October 03 2019
[27] Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C., 2015. Iotpot: analysing the rise of iot compromises. EMU 9, 1.
[28] Jicha, A., Patton, M., Chen, H., 2016. Scada honeypots: An in-depth analysis of conpot, in: 2016 IEEE conference on intelligence and security informatics (ISI), IEEE. pp. 196–198
[29] Wang, M., Santillan, J., Kuipers, F., 2018. Thingpot: an interactive internet-of-things honeypot. arXiv preprint arXiv:1807.04114.
[30] Luo, T., Xu, Z., Jin, X., Jia, Y., Ouyang, X., 2017. Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices. Black Hat.
[31] Dowling, S., Schukat, M., Enda, B., 2018. Improving adaptive honeypot functionality with efficient reinforcement learning parameters for automated malware. Journal of Cyber Security Technology 2, 75–91.
[32] Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.
[33] Schaul, T., 2010. Pybrain. Journal of Machine Learning Research 11, 743–746
[34] Dowling, S., 2017. An adaptive honeypot using reinforcement learning implementation. Repository: https://github.com/sosdow/RLHPot. Accessed December 12 2018.

Predict the Match Outcome in One Day International Cricket Matches Using Machine Learning

Zia Ur Rehman
Department of Computer Science
University of Engineering and Technology
Taxila, Pakistan
rajazia96@gmail.com

Muhammad Munawar Iqbal
Department of Computer Science
University of Engineering and Technology
Taxila, Pakistan
munwar.iq@uettaxila.edu.pk

Hamza Safwan
Department of Computer Science
University of Engineering and Technology
Taxila, Pakistan
hamza.ciit135@gmail.com

Abstract*— In sports, cricket is a very famous game that being watched everywhere in the world. It has tremendous spectators everywhere in the world. One-day International (ODI) cricket matches are popular globally. The impact of ODI is increasing financial and sponsorship every year. Betters and bookmakers are interested in match outcomes. Our focus is to predict the ODI Matches. ODI matches have several features like pregame, in-game, and post-game that affect the match outcomes: the venue, pitch condition, toss winner, toss decision, ground, host country, and dynamic strategies. We use the 2000 to 2019 ODI matches dataset. The dataset introduced some new features like Pitch Condition, Venue Familiarity, Winner Inning, Inning First, Inning Second. This study investigates machine learning classifiers to predict the ODI match outcomes. We applied machine learning algorithms such as Naïve Bayes, SVM, KNN, Random Forest, and Decision tree to generate predictive models. The experiment demonstrates that the model performs well in both instances, Result and Winner. In the decision tree and Random Forest result accuracy up to 98 %, SVM, Naïve Bayes, and Logistic Regression accuracy up to 87%,65%, and 67% respectively. It shows that the Random Forest and decision tree perform well in accuracy, precision, and recall compared to other Statistical Models. This predictive model will help the Gambling Industry, Sports Analysts, Reporting Media, and Cricketing boards to analyze the team performance.

Keywords*— Cricket Prediction, Data Mining, Sports Analytics, Supervised Machine Learning, Weka, Classification

I. INTRODUCTION

In Sports, Statistical Analysis is very important for analyzing the performance of the games. Many games are played in world tennis, football, cricket, and hockey, but cricket is the most crucial game in almost 106 countries [1]. Cricket is the 2nd most favourite game that is played after football matches. International Cricket Council (ICC) manages all cricket-related problems. The report shows 106 countries is a member of ICC. ICC cricket fans range is 1.5 billion across the globe.

There are mainly three played formats, which are Test, ODI, and T-20 matches. T-20 matches consist of 20 overs, and this format is also famous in today's world[2].In ODI, we have 50 overs matches between the two countries, and the result decides which team won or lose the match. ICC also manages the World Cup for ODI after five years. In the World Cup, only 10 top countries were allowed to participate in the event. Match Prediction is a very complex task because many factors change the match outcomes. Cricket Prediction helps the coach, bookmakers, bettors, and media for defining the strategy. One of the most important techniques used for the prediction of matches is Machine Learning [3].In Machine Learning, we have several independent variables such as pitch conditions, venue of the team, toss winner, toss decision, and winner innings. This whole process uses all independent features to train the model and then model evaluation on future matches to know the model effectiveness. Machine Learning[3] Models Effectiveness measured by Confusion Metrics, accuracy, precision, and recall score [4, 5].

The main objective of this research is solving the question

Can Machine Learning Models provide accurate results regarding cricket matches related to ODI matches? If so, which classifier is best in accuracy, confusion metrics, precision, and recall. To answer the research question defined above, we use classification methods such as Naïve Bayes, Random Forest, SVM, Logistic Regression, KNN, and decision Tree [6] to predict ODI matches. This research uses a 20 years dataset(2000 to 2019) that consists of many independent features that highly impact matches. Another important aspect of this research introduces new features like pitch condition (dry, dust, dead, green, bouncy), Venue, Toss, Winner Inning, and host country. All these features belong to prematch, in the match, and post-match attributes. The main reason for adding new features to our dataset is that most researchers use limited features, and their results do not provide higher accuracy concerning the state-of-the-art attributes.

In the dataset, we have an attribute name ground, that the match has played. Basically, in this attribute, we find one more attribute pitch condition that contains 5 different labels. This attribute finds from different websites, ICC, Quora, and Google, that provide all 157 international ground pitch conditions; these pitch conditions have the highest impact on cricket matches. We use two datasets for prediction. Our target class is Result (Won/Lost) in the first dataset, and in the second dataset, our target class is the winner (14 labels). Previous researchers mainly focus on T20 leagues, support different franchises. But in this research, our focus is on finding those attributes that have the highest impact on ODI matches. This research mostly benefits Cricket fans, Coach,

Cricket Manager, the gambling industry, bookmakers, Sports analytics, and scholars who work in the sports area.

This paper is mainly organized into five sections. In section 2, we define the cricket game rules and the terms that involve in cricket. Also, discuss previous research work in cricket and Machine Learning Algorithm that apply in cricket matches. Section 3 defines the methodology of the research and discusses the new attributes added manually to the dataset. In section 4, we discuss the results and comparison between the techniques discuss in the Framework. In the 5. section, we discuss the conclusions

II. Literature Review

In Sports, statistical analysis is very important for future predictions. Cricket is a very important game that was played in almost 102 countries in the world. Most researchers use data mining and Machine learning techniques to predict matches, but there is inconsistent data that the match-winner not correctly predicts. In sports, Analytics prediction is significant because everyone involves coaches, captains, fans, and the gambling industry. They propose a new method for the Pakistan team winning percentage against any team in the world. This paper's main point introduces a new attribute known as a consecutive win before matches [1]. This attribute increases the prediction outcome to 80%, But they mainly focus on the Pakistan team [1]. They use different Machine Learning Algorithms to Predict the Winner of IPL(Indian Premier League) and dataset use from 2008 to 2017. They use the Decision Tree, Random Forest, and XGboost.The results they acquire from the Decision tree is 76% Accuracy, and XGboost accuracy will be 91%, but this paper mainly focuses on the (IPL) [6].

Kaluarachchi, Amal, and S. Varde Aparna primarily focus on Classification techniques such as (Bagging, Boosting, Decision tree, Naïve Bayes). They also provide the Cric AI tools that pick some of the factors that affect the ODI match outcomes and also use 8 different classifiers for prediction. It uses Weka tools for Knowledge analysis and prediction [7, 8]. The main issue of this work is they use only four attributes and do not provide the results.

This paper uses a new methodology of the Indian premier league winner[6, 9]. They propose a Multivariate Linear Regression model for predicting the match; they also use new attributes like team strength. In team strength, they calculate all factors that affect the team or do not affect the team; these factors are considered in this paper [10]. Another research carried by Aburas, Abdurazzag A., Muhammed Mehtab is to predict the winner of the ODI Cricket world cup 2019 using Big data and the KNN Machine learning Algorithm [3] [10]. They use different datasets to predict the world cup winner, they use Player stats in England, and all-rounder players, Best Bowlers, and collectively relate using R Language.

In this research, the author Anik, Aminul Islam, Sakif [11] predicts individual player performance in cricket matches. They predict Tamim Iqbal scores in upcoming matches. They use different Machine learning Algorithms and feature selection techniques for determining the score of the player. This paper predicts the ongoing match prediction like predicting the match's score using Linear Regression, and they include some new factors like venue team, wicket left, and batting team. And the second most important prediction predicts the match outcome using Gaussian Naïve Bayes, and their accuracy is 91% They use the Weka tool for dataset Analysis[12]. In this research, the author Passi, K., and Pandey predict the player performance (batsmen and bowler). In the batsmen perspective, they predict how many runs in future games; in the bowler perspective, they predict how many wickets were taken and how many runs they solve using classification[13]. In this research, Kumar, Jalaz, Rajeev Kumar, and Pushpender Kumar predict ODI cricket matches outcome using the Decision Tree and Multilayer Perceptron Neural Network [11]. In the decision tree, they use only in-game attributes for the prediction of the match outcome.

For measuring performances, these values were used accuracy, precision, and f1 score [14]. This paper uses a limited dataset for 2013 to 2017 for prediction and only four attributes considered, and accuracy is 76%. In this research, Yasir, M., Chen, L., Shah, S.A., Akbar proposes a method for T-20 matches prediction they use some new features like ground statistics, team statistics, and player statistic. Firstly, predict before the match, secondly predict when the match is ongoing. It shows match prediction before the match gives correct result compared to the ongoing match its accuracy is 85 % and 89%[15]. They show KNN is the best Learning Algorithm, and its accuracy is 71%,[16-18].

In previous research, we see that mostly work on T-20 and ODI matches. They mostly use pre matches attributes and in-game attributes. This research introduces some new features that impact matches like pitch condition (dry, dead, dusty, green, bouncy), team venue, and winner inning. In this research, we use feature selection techniques to combine machine learning models for evaluating the attributes that impact ODI matches. We use the combination of all attributes external, internal, and after a match (Winner Inning) use to predict ODI matches. We use five classifiers to predict matches and finally evaluate all models, and all results are compared using graphically in the next sections.

III. Research Methodology

In this chapter, we define the terms and proposed methodology for the prediction of ODI matches. This research uses Machine Learning Approaches to predict ODI matches and evaluates, which classifier best in terms of accuracy. We build a model that uses 14 features pitch (Dusty/Dead/Green/Bouncy/flat), Toss, Venue, Batting (First/second, Winner Inning) in our prediction model. Our main contribution is introducing new features like pitch conditions that impact on ODI matches. We study all types of pitches and describe the next section in detail. We use two target classes for Prediction (Winner and Results) separately. We have 14 labels; these 14 labels consist of the first 14 teams of ICC rankings. In the Result class, we use two labels (Won or lost). Won define the first Team won the match and the lost define the first Team loss the match.

Figure 1 describes the Framework of our research. Initially, data is crawl from Cricinfo. In Cricinfo, we have huge numbers of filters or queries that provide demand data. We firstly choose the first 14 teams of ICC Rankings in 2020. This dataset (2000 to 2019) contains only 6 features used in ODI matches, and all other features were added manually. We apply pre-processing techniques for data cleaning. We have some features that have no impact on matches we define detail in the next chapter when we discuss the dataset. We remove these features from our dataset (match id, date, Margin, Ground). Some features have missing values, i.e We have matches with no result or tied matches; we exclude those matches from our dataset. The third phase of our proposed method is attribute Generation. In attribute generation, we apply exhaustive search or use ICC websites or Quora that support new features such as team1Inning, team2Inning,winnerInning,pitchcondition(dry/dust/dead/green/bounce) , Toss, Bat (first or second), and Winner of the match. These all features manually add to our dataset.

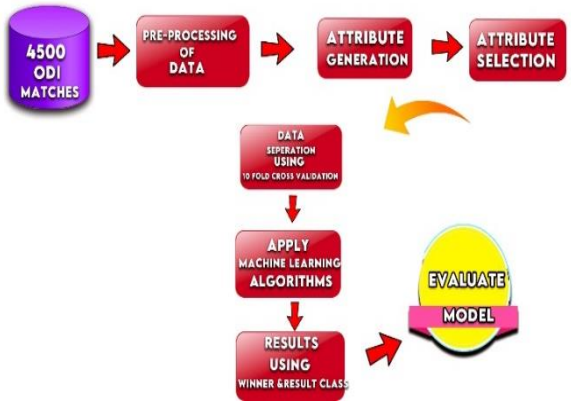


***Figure 1**:* A proposed methodology for predicting the match outcome

In 10-Fold cross-validation, all datasets are divided into equal subsets. Then training and validation using these subsets. In 10 cross-validation, we use 80 % testing data, and 20 % data use for validation. Various Machine Learning Algorithms use in this Research Naïve Bayes, Logistic Regression Nearest Neighbor, Decision Tree, and Random Forest. By applying various ML algorithms separately to different Target Classes (Results and Winner). In the first case, our target class is Result with two outcomes (Won or Lost), and Secondly, our target class is the winner, which defines 14 different teams according to ICC Rankings 2020. In the final phase, all ML Algorithm Classifiers are compared to each other, and then we decide which ML Algorithm future use for predicting matches

The fourth phase is attribute selection techniques that discuss in the next section. These attribute selection Strategies (Wrapper, Ranker, Subset Evaluation) choose our model as the best-suited attributes. We use the Weka tool for Attribute Selection Strategies that provide the library of all modules. Then we apply the number of different Machine Learning classifiers that provide the prediction Model of match results. The training and testing method used for all classifiers are 10-Fold cross-validation.

IV. Data and Result Analysis

A. *Dataset Description*

We use ODI data for a predictive model. This dataset consists of 4500 instances and 18 features. Only the first 14 teams used for prediction. This research uses (2000 to 2019) ODI matches that help our predictive model concerning today's cricket behaviour. Firstly, when the dataset is crawl from Cricinfo, we have only six features, then we read different cricket websites for adding more features to the dataset.

Table 1 Detailed descriptions of dataset

Attribute	**Description**
Team 1	First Team that played a match
Result	Two outcomes Won OR Lost
Team 2	2nd Team that played the match
Toss	Toss have two outcomes (lost or won)
Bat	Bat has two outcomes batting first or second
Ground	Defines which venue teams Playing
Start Date	Define the which date match Played
Host Country	Defines which country
Venue_Team1	Team 1 venue defines (Home/Away, Neutral)
Venue_Team2	Team 1 venue defines (Home/Away, Neutral)
Inning_Team1	Defines the two outcomes (first or second)
Inning_Team2	Defines the two outcomes (first or second)
Margin	Defines how many run or wickets the Team
Team1_Toss	Defines the first team toss won or lost
Team2_Toss	Defines the second team toss won or lost
Winner Inning	Defines which team Winner first inning winner or second inning winner
pitch	Defines the behaviour of wicket Dry pitches (Flat): Dusty pitches (Slow/Low) Green pitches (Bouncy/Fast/Swing) Fast pitches (Bouncy) Dead pitches(flat)

This dataset consists of 18 features (2000 to 2019) discuss in Table 1.Some of those features have a high impact on matches. In a previous literature study, we see that the pitch condition attribute is not highlighted, but our research shows that the pitch attribute has the highest impact on ODI using Feature selection techniques described in the below section. We have five attributes that define the pitch behaviour slow, dry, dusty, green, and bouncy. Two more things change pitch behaviour to every country, grass on the pitch, and crack on the pitch. In the cricket world, we have a total of 157 international grounds registered in ICC. We manually

assign the labels and fill the 4500 instances using ICC pitch Rating, and Quora describes them in the (below section). In some cases, two different countries have the same pitch behaviour [7, 8]. For example, we have Pakistan, and India is a country almost all the pitch conditions are the same so assign the label (flat/dead). These are the main types of pitches that help in assigning the labels of the previous discussion:

B. Types of Pitches

1. DRY PITCHES

In this type of pitches, mostly some moistures exist and a lot of cracks on them; that is why these types of conditions are supported for spinners. These are the various countries that supported dry pitches in Pakistan, India, and Srilanka.

2. DUSTY PITCHES/SLOW/LOW

These pitches are also known as soft pitches because these pitches clay is very soft; that is why not rolled in these pitches. These types of pitches are found in Srilanka, West Indies, Zimbabwe, and Bangladesh.

3. GREEN PITCH/BOUNCY/FAST

In green wickets, a considerable amount of grass helps the fast bowler swinging the ball both ways. These pitches are mainly found in western countries like England, South Africa, and New Zealand and mostly use for test matches.

4. FAST PITCHES

In this type of pitches, mostly high cracks and bounce are very high. These are the mostly Perth and WACA stadium in Australia.

5. DEAD PITCHES/FLAT

This type of pitches mostly consists of the flat; there is no grass on it. There is no swing and seam on it. In these types of pitches, mostly high run scores match. The main countries are India and Pakistan.

C. Feature Selection

Different feature selection techniques were available in Weka for the selection of relevant attributes. These are the main Evaluator Algorithms provide by Weka for Feature selection and Ranking of Attributes. The wrapper, Information Gain, Relief and Correlation-based feature (CFS) All these types of Evaluator (Wrapper)provide the Classification methods (Random Forest, Naïve Bayes, Logistic Regression and decision tree) to evaluate the attributes. The Information gain and Relief filter method uses KNN for evaluating the features. These are the two tables (Table 2 &Table 3) showing which attribute was selected by which classification method. In the case of Table 2, the target class of our predictive Model is Result (Win or lost). We use a wrapper with Naïve Bayes to select only four attributes, and in the case of Information Gain, select the 15 attributes and define the ranking of every attribute. Team 1 and Team 2 have the highest rank using information gain. We clearly see in Table 2 we have four cases that the pitch attribute is selected Wrapper with Logistic, Random Forest and Information Gain, Relief. After clearly check all attributes and ranking, we choose the 14 characteristics for our classification problems.

In the case of Table 3, the target class of our predictive Model is Winner (14 labels). We use a wrapper with Naïve Bayes to select only six attributes, and in the case of Information Gain, select the 15 attributes and define every attribute ranking. Team 1 and Team 2 have the highest rank using information gain. We clearly see in table 2 we have two cases that the pitch attribute is selected Information Gain, Relief. In the Information gain and Relief filter methods, we see that pitch attribute ranking is 4. We see in Table 2 & Table 3 that pitch behaviour is significant for matches.

Table 2 Attributes chose by Wrapper and Ranker Methods Target Class is Result (won/lost)

Wrapper with Naïve Bayes	Information Gain+ Ranker	Relief+ Ranker	Wrapper with Random Forest	Wrapper with KNN	Wrapper with Logistic
Team Winner Team 2 Venue Team 2 Winner Inning	1 Team 1 6 Team 2 3 Winner 9 Venue_Team2 8 Venue_Team1 13 Margin 7Host_Country 16 pitch 5 Bat 10Inning_Team1 11Inning_Team2 14 Team2_Toss 12 Team1_Toss 4 Toss 15Winner_Inning	15Winner_Inning 3 Winner 13 Margin 1 Team 1 6 Team 2 5 Bat 10 Inning_Team1 11 Inning_Team2 12 Team1_Toss 4 Toss 14 Team2_Toss 9 Venue Team 2 8 Venue Team 1 16 pitch 7 Host Country	Team 1 Winner Toss pitch	Team 1 Winner Team 2	Team 1 Winner pitch

Table 3 Attributes chose by Wrapper and Ranker Methods Target Class is Winner (14 teams)

Wrapper with Naïve Bayes	Information Gain+ Ranker	Relief+ Ranker	CFS Evaluator	Wrapper with J48
Team 1 Result Toss Team 2 Team1_Toss Margin	6 Team 2 1 Team 1 7 Host Country 16 pitch 13 Margin 8 Venue_Team1 9 Venue_Team2 2 Result 15Winner_Inning 10Inning_Team1 5 Bat 11 Inning_Team2 4 Toss 14 Team2_Toss 12 Team1_Toss	6 Team 2 1 Team 1 7Host_Country 16 pitch 2 Result 15Winner_Inn 9Venue_Team2 8Venue_Team1 13 Margin 11Inning_Te2 10Inning_Te1 5 Bat -4 Toss 14Team2_Toss 12Team1_Toss	Team 1 Team 2 Host Country Winner Inning	Team 1 Result Toss Bat Team 2 Inning_Team1

D. Experimental settings and Evaluation Measures

We use Weka (Waikato Environment for Knowledge Analysis) tool used for this research. Weka is an open-source tool or intelligent data mining software that provides different machine learning algorithms for analyzing the data. Also, python use for Exploratory Analysis and visualizing the ODI matches. This research processing machine is an Intel Core i3 5th generation processor with 4 GB Ram on Windows 8 and 64-bit operating system. All these types of algorithms have different kinds of

hypermeters. Machine Learning Algorithms, including Naïve Bayes, Random Forest, Decision tree, K-Nearest Neighbor, Logistic Regression, and SVM, was available in Weka. We use various metrics precision, recall, and confusion matrix for the selection of the best classifier. We use 10- fold cross-validation techniques for derivation of the model for training and testing of models. This technique distributes the dataset into 10folds with stratified cross-validation. We were using 9 folds for training the model and the last one-fold for testing purposes. This whole process takes ten times to repeat and finally generates the average prediction accuracy.

V. Result Analysis

A. Case 1: Results derived Using Target class is Result (Won/Lost) in Feature set

In the first case, our target attribute is the result; this attribute consists of two labels, won or lost. We evaluate several attributes like home team advantage, toss winner, batting decision, venue of the team, host country, pitch condition (dusty, dry, dead, green, bounce), and winner Inning. All teams have several benefits in-home ground, no pressure when bowling and batting, crowd support, pitch conditions, and no travelling. In this experiment, we observed that Pakistan has no home ground available since 2009. After a terrorist attack on Srilanka teams, we use UAE as a home ground for Pakistan, and these pitches behaviour well known to the Pakistani coaches. The Machine Learning classifiers are used for the prediction of cricket matches. Figure 2 shows a comparison between different classifiers.

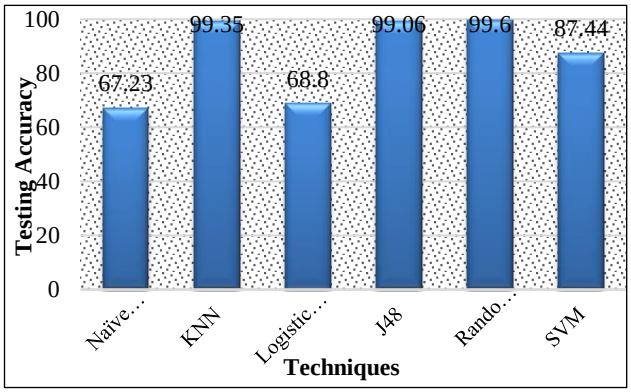


Fig 2 Accuracy derived by Target class Result (Won/Lost) Using Machine Learning Algorithms

It indicates that Random Forest and Decision Tress accuracy almost 99%. In the case of Naïve Bayes accuracy up to 67%, In the case of Logistic Regression accuracy up to 68%, SVM accuracy up to 87%. This means that the Random Forest Decision Tree and SVM provide better results as compared to the literature. But Logistic Regression and Naïve Bayes provide the lowest accuracy. We also define the Confusion matric of all classifier that helps in precision-recall and f-Measures. In Figure 3, we observed that Precision, Recall, and F-Measure are consistent in Random Forest and Decision Tree. Still, in the case of Logistic Regression and Naïve Bayes, we have low precision, recall, and f-Measure. Table 4,5,6,7,8 shows the confusion Matric of all Machine Learning Techniques.

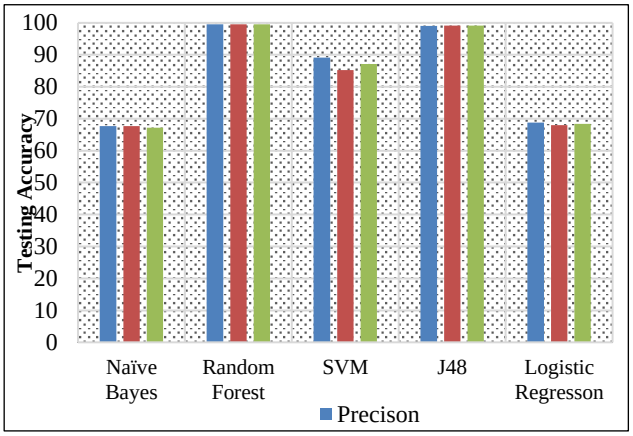


Fig 3: Precision-Recall and F-Measures of Machine Learning Algorithms from Result (Won/Lost) is Target Class

Naïve Bayes misclassified 1482 instances of 4507, whereas Decision Tree, Random Forest, SVM, and Logistic Regression misclassified 18,24,583,1406, respectively. These misclassified instances increased the False Positive Rates. As all classifiers are discussed, we see Random Forest, Decision Tree, and SVM perform well in the case of Precision, Recall, and F-Measures.

Table 4 confusion matrix of naïve Bayes results shows that 735 instances should belong to the lost class, but they misclassified to win the class. This result generates a low precision and recall rate because they misclassified 747 instances correctly; they were classified as a lost class. In this case, our False Positive and False Negative rate is 33%and 32%, respectively. All confusion Matric of ML classifier were explained in Table 4,5,6,7,8. Table 6 shows that logistic regression results show that 735 instances should belong to a lost class, but they misclassified to win the class. This result generates a low precision, and the recall rate is low because they misclassified 690 instances correctly; they are classified as a lost class. So, their False Negative Rate is 31%.

Table 4 Confusion matric for Result Class using Naïve Bayes

	Won	**Lost**	**Total**
Won	1505	735	2240
Lost	747	1520	2267
Total	2252	2255	4507

Table 5 Confusion matrix for Result Class using SVM

	Won	**Lost**	**Total**
Won	1899	341	2240
Lost	242	2025	2267
Total	214	2366	4507

Table 6 Confusion matrix for Result class using Logistic Regression

	Won	Lost	Total
Won	1524	716	2240
Lost	690	1577	2267
Total	2214	2293	4507

Table 7 Confusion Matric for Result class Using Decision Tree

	Won	Lost	Total
Won	2231	09	2240
Lost	09	2258	2267
Total	2240	2267	4507

Table 8 Confusion Matric for Result class Using Random Forest

	Won	Lost	Total
Won	2227	13	2240
Lost	12	2255	2267
Total	2239	2268	4507

B. Case 2: Results derived Using Target Class is Winner (14 labels) in Feature set

As discussed in the previous section, our target Attribute is the result and checks other attributes performance. Still, in this case, our target attribute is the winner, which consists of 14 labels.

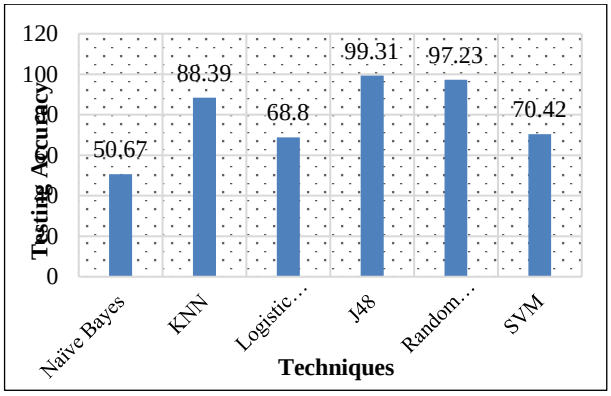


Fig 4: Accuracy derived by Target class Winner (14 labels) Using Machine Learning Algorithms

These labels provide the result of which team wins the match. For example, we have two countries that played the match, India and West Indies. Our model decides which Team wins the match according to the name; in this case, India wins the match, so India's label. These are important features that affect the matches (toss, batting decision, pitch condition (dust, dry, dead, green, bounce), venue, home support, and Winner innings). Machine Learning Techniques are used for the prediction of matches. Figure 4 depicts the performance of all models. Figure4 define the accuracy of all models Naïve Bayes, SVM, Random Forest, Decision tree and KNN,50%,70%,97%,99%,88%.Figure 5 defines the Precision, Accuracy, and F-Measures of all Models. We see that Random Forest, Decision Tree almost the same 99%, but in Naïve Bayes, SVM, KNN gives the 50%,68%,83%. Table 9 shows the comparative analysis of Precision-Recall and F-Measures of different classifiers. This table shows that precision-recall and f-Measure of Random Forest and Decision Tree perform well-concerning SVM and Naïve Bayes.

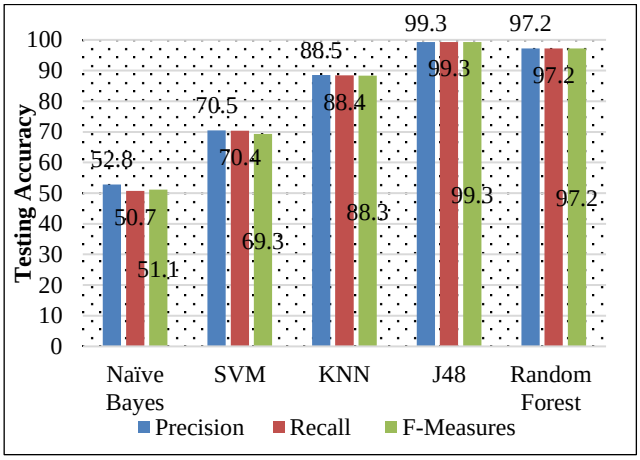


Fig 5 Precision-Recall and F-Measures of Machine Learning Algorithms from Winner (14 labels) is Target Class

Table 9 Comparative Analysis of MLClassifiers

Classifier	Performance Measure	India	Pakistan	Australia	South Africa	England
Random forest	**Precision**	0.982	0.941	0.988	0.982	0.976
	Recall	0.994	0.947	0.995	0.988	0.993
	F-Measure	0.988	0.944	0.992	0.985	0.985
Decision Tree	**Precision**	0.998	0.977	0.994	0.996	0.998
	Recall	0.997	0.990	0.992	0.995	0.998
	F-Measure	0.998	0.983	0.993	0.996	0.998
Support Vector Machine	**Precision**	0.700	0.806	0.713	0.755	0.726
	Recall	0.869	0.757	0.928	0.873	0.485
	F-Measure	0.776	0.781	0.806	0.810	0.581
Naïve Bayes	**Precision**	0.568	0.519	0.573	0.616	0.480
	Recall	0.438	0.551	0.494	0.530	0.502
	F-Measure	0.494	0.535	0.531	0.570	0.491

VI. CONCLUSION

This paper mainly investigates the ODI matches outcomes and feature importance. We use previous matches for predicting new matches that no played so far. In this paper, our main contribution is we introduce new features like Pitch behaviour (dusty/dry/dead/green/bouncy), Inning of Team, Venue (home, away, neutral), Familiarity, Winner Inning, toss Winner, toss decision. And check the importance of every attribute using Filter based Methods and Wrapper. We see that our attributes have a high impact on cricket matches. In literature, we see that no one uses Pitch behaviour in the match. We perform an exhaustive search to check the ground's behaviour and check the rules that define the ICC. Using Filter based and wrapper method, we use only 14 features for the predictive model. We generate two predictive Models; first, predict the Winner (14

labels) name of the team and then predict the Result (won/Lost) of the Match. We use a Machine Learning Classifier for Predictive Model. In these two cases, we use five Machine Learning classifiers Naïve Bayes, Logistic Regression, KNN, Random Forest, and Decision tree, and we find that Decision tree and Random Forest(97%,99%) perform well in terms of Accuracy, Precision, Recall.

There is also a lot of enhancement in this research. We mainly focus on won or lost ignore tied or rain-interrupted matches. Also specific to already matches played, not focus on ongoing matches**.** In future perspectives, we check which players perform well in Asia, Africa, and other continents. Another perspective is we check which player man of the match incoming matches or world cup.

REFERENCES

[1] W. Ahmed, "A Multivariate Data Mining Approach to Predict Match Outcome in One-Day International Cricket," Karachi Institute of Economics and Technology, 2015.

[2] K. Kapadia, H. Abdel-Jaber, F. Thabtah, and W. Hadi, "Sport analytics for cricket game results using machine learning: An experimental study," *Applied Computing and Informatics*, 2019.

[3] J. Kumar, R. Kumar, and P. Kumar, "Outcome prediction of ODI cricket matches using decision trees and MLP networks," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 343-347: IEEE.

[4] X.-B. Fu, S.-L. Yue, and D.-Y. Pan, "Camera-based Basketball Scoring Detection Using Convolutional Neural Network," *International Journal of Automation and Computing*, pp. 1-11, 2020.

[5] K. Alpan and G. S. İlgi, "Classification of Diabetes Dataset with Data Mining Techniques by Using WEKA Approach," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2020, pp. 1-7: IEEE.

[6] A. A. Aburas, M. Mehtab, and Y. Mehtab, "Cricket World Cup Predictions Using KNN Intelligent Bigdata Approach," in *Proceedings of the 2018 International Conference on Computing and Big Data*, 2018, pp. 18-22.

[7] A. Kaluarachchi and S. V. Aparna, "CricAI: A classification based tool to predict the outcome in ODI cricket," in *2010 Fifth International Conference on Information and Automation for Sustainability*, 2010, pp. 250-255: IEEE.

[8] A. I. Anik, S. Yeaser, A. I. Hossain, and A. Chakrabarty, "Player's Performance Prediction in ODI Cricket Using Machine Learning Algorithms," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 500-505: IEEE.

[9] U. Ujwal, P. Antony, and D. Sachin, "Predictive analysis of sports data using google prediction API," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 2814-2816, 2018.

[10] T. Singh, V. Singla, and P. Bhatia, "Score and winning prediction in cricket through data mining," in *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*, 2015, pp. 60-66: IEEE.

[11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.

[12] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3-24, 2007.

[13] T. Ngo, "Data mining: practical machine learning tools and technique, by ian h. witten, eibe frank, mark a. hell," *ACM SIGSOFT Software Engineering Notes*, vol. 36, no. 5, pp. 51-52, 2011.

[14] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine learning*, vol. 59, no. 1-2, pp. 161-205, 2005.

[15] S. Chowdhury, K. A. Islam, M. M. Rahman, T. S. Raisa, and N. M. Zayed, "One Day International (ODI) Cricket Match Prediction in Logistic Analysis: India VS. Pakistan," *Journal of Human Movement and Sports Sciences*, vol. 8, no. 6, pp. 543-548, 2020.

[16] A. Balasundaram, S. Ashokkumar, D. Jayashree, and S. M. Kumar, "Data mining based Classification of Players in Game of Cricket," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 271-275: IEEE.

[17] W. De Vazelhes, C. Carey, Y. Tang, N. Vauquier, and A. Bellet, "metric-learn: Metric learning algorithms in python," *Journal of Machine Learning Research*, vol. 21, no. 138, pp. 1-6, 2020.

[18] S. G. Taj, T. Bhavana, E. Pravallika, P. S. Reddy, and G. Swarnalatha, "Machine Learning Model for Predicting the Performance Analysis of Cricket Players'," *Sustainable Humanosphere*, vol. 16, no. 2, pp. 17-23, 2020.

Rumor Detection Based on Temporal Dimensions

Arfa Hussain
FAST NUCES, Lahore
arfahussaina1@gmail.com

Saira Karim
FAST NUCES, Lahore
saira.karim@nu.edu.pk

***Abstract*— The speed at which data spreads on Twitter means that a lot of information shared in the early stages of an incident is not monitored. This makes it difficult to find true news from various unverified news. Machine learning techniques help the researchers to detect fake news/rumors on time. This research aims to detect rumors from collection of tweets shared on twitter. The proposed scheme presents a novel idea of rumor detection using event detection techniques. This system first identify important words that can be related a particular incident/event using temporal features, cluster related words based on their co-occurrence and then identify rumor tweets. Our work investigates the problem of rumor detection using both supervised and unsupervised learning. This idea also uses language processing and modeling of contextual and social features of users sharing those tweets to detect rumors and non-rumors. The proposed research is compared with the state of the art method and give better results.**

Keywords*— Rumor detection, entropy, agglomerative clustering

I. INTRODUCTION

[1] Microblogging was launched in 2007. That's when Twitter exploded on stage and offered people a platform in 140 characters or less to express their thoughts and views. Twitter is regarded as a microblogging site, more than any other social networking site, due to its character count constraints. Through tweets and retweets, Twitter reaches a large number of people easily, keeps up to date with the latest news and trends, and immediately shares them with others, reaches new viewers, and contributes to the discussion of events. Twitter is as important as a medium of interaction as it is a tool of broadcast. Twitter facilitates and promotes interaction in a public forum between multiple parties, enabling you to receive immediate feedback from customers. This promotes accountability in interactions because the content on Twitter is available to everyone else to track.

People are increasingly using social media platforms to track newsworthy events and to collect news. The pace at which information spreads on Twitter, however, ensures that much of the information posted in an event's early stages is unchecked. In all categories of information, false stories spread significantly faster and wider than true stories. [2] It becomes more difficult for reporters or journalists to separate verified data from speculation and report the news.

Misinformation also produces a confusing and unpredictable situation, leading to poor decision-making and inhumane effects. [3] For example, a 2013 story about the bombing of the White House, injuring Barack Obama made U.S. stock markets terrified and made a huge loss to them. Rumors have also played a crucial role in the deadly riots in London in 2011. Consequently, it is very important to assess the validity of spreading false information (rumors) promptly.

In this research the answers to the following research questions are explored and examined:

How to find the suitable features for rumor detection in case of supervised and unsupervised learning?

How these features can be used to detect rumors on Twitter?

What is the effectiveness of these methods in terms of precision, recall and F1 scores?

The main contribution of this paper is as follows.

For supervised learning, features are identified to find important (topical) words. Information related to hot topics/events spread very fast. Therefore, temporal feature are very important. Term document frequency, entropy of words in a certain time period and number of hashtags in a certain time period are used to determine important terms. Further clustering and classification tasks are performed using the true labels. Finally precision, recall and F1 scores are computed.

For unsupervised model, contextual and social features are extracted from the tweets. Clustering is applied on these features and then classify these tweets into rumors and non-rumors.

The rest of this paper is organized as follows. Section II discusses the background and literature available for rumor detection on twitter. Section III explains the proposed methodology, experimental results and their analysis. Section IV presents the conclusion and references are mentioned in section V.

II. LITERATURE

A lot of work has been done to identify rumors. In [5], rumors are detected and find the source of rumor. Data is collected from Twitter. 20 features were selected based on content of tweet and accounts of users. They identified that contextual features are more important than user features for rumor detection. In [6] social and contextual features from tweets are extracted and conditional random fields are applied to determine the verification of tweets. They used the who follows whom social graph of Twitter users and injected a number of monitor nodes that report the received data. They concluded that rumors and their sources can be predicted correctly if sufficient number of monitor nodes are present in the network. A novel idea of using random conditional fields learned from dynamic of social media posts is explored in [4]. The classifier learns the context of the event that helps in better accuracy and F1 score. In [8] a novel pattern matching algorithm is introduced that matches rumor patterns from social media streaming. These patterns combine structural and behavioral properties of rumors. From a stream of posts, a graph is extracted that is labeled and directed. If the posts contain promoted relationships, then these were converted to edges and assigned as an input to the algorithm. Algorithm operates data stream and lists matched patterns. In order to find distinct patterns, that are more important and related in rumor events, evaluation was done through TF-IDF. The trends frequently occur are supposed to separate rumors from non-rumors. Rumors are detected and find the source of rumor. The research in [9] studies how credibility perceived from different features of tweets can help in evaluating the truthfulness of tweets in conjunction with fast-paced events. Surveys were conducted regarding pictures uploaded on twitter to analyze the credibility perception of users. Some of

their key observations include the fact that while specifics of the author that are not readily available on Twitter feeds are important for making tweets easy to validate. Displaying multiple tweets that confirm a claim misleads people to believe what is really a fake story. The analysis highlights the accuracy of tweet credits and finding.

In rumor detection, labeled data collection is also an important factor to study as manual annotation is very expensive and tedious. Few researches focused on this type of study. In [10], the main challenge was to collect extensive training data. A weakly supervised approach is presented, which collects wider but very error prone data comprises of hundreds of thousands of tweets. During collection, tweets were automatically tagged by their source (trustworthy or untrustworthy) and a classifier is trained on this dataset. With this unclean dataset, detection of fake news was done by 0.9 F1 score. Jain et al. [11] strategy relied on the fact that verified News Channel accounts provide more reliable data as compared to the user's general public account. They also suggested a way to use semantic analysis to separate rumors from non-rumors. This approach works in four steps. First they extract tweets related to trending topics and cluster them according to the topic. Next, in each cluster, tweets are segregated according to their source (news channel or public account). In the third step, contextual and sentiment mismatch is calculated for tweets from verified and unverified sources for each topic. The topics where mismatch ratio is higher are labeled as rumors.

[12] Zhao indicated that rumors would provoke skeptical users inquiring about their truthfulness. A piece of information has a number of inquiring tweets that could then imply that the information is rumored. Firstly, search for the enquiry phrases (that — this — it) is performed. Then similar posts are clustered together. Next, posts related to each cluster that do not include these phrases are collected. Finally, clusters are ranked based on the likelihood of containing a disputed factual claim.

[19] In this study rumors are considered as anomalies. Factor analysis of mixed data is performed on the proposed features to detect anomalies. Deviation degree is described in terms of Euclidian distance and cosine similarity. Rumors are detected by assigning ranks based on deviation degree.

III. Research Methodology

We proposed two different methodologies, supervised and unsupervised for labeled and unlabeled data. For labeled data, we extended the idea of event detection [13] and for unlabeled data, we have expanded A. Zubiaga's[4] research.

The proposed method for supervised Learning extends the idea of event detection [13] to detect rumors in twitter data set. The basic idea of this study was to group tweets into clusters where each cluster represents an event. It first detects events from daily microblog posts. Next, related events are tracked to find change in event and finally a summary of these events is generated. The main contribution in [13] is the detection of event clusters. Since rumors are mostly about some events, the same idea can be used to group tweets related to an event and then classify them into rumor/non-rumor.

Nonetheless, the method in [13] has not been applied directly to detect rumors and it is therefore important to further research their applicability with an effective rumor dataset. Events that are posted by many people are typically hot topics or a controversial affair. They are shared & retweeted by many users and spread within no time. Similar is the case for rumors. Rumor is a story of uncertain fact and spread like fire. So we can use event detection methods for rumor detection also. The proposed method is different from event detection method because [13] does not perform any classification. To differentiate the rumors and non-rumors, we have used an index of terms.

A. Dataset Collection and Processing

In this research, PHEME dataset for Rumor Detection and Veracity Classification [15] is used. This set contains a collection of Twitter rumors and non-rumors that were posted during news breaks. We have used this dataset because most of the rumor detection researches [4] used this dataset which helps to compare our research with others.

1) *Structure of Dataset:* There is a collection of tweets for each scenario/event, including both rumors and non-rumors. The tweet are classified into source and reaction tweets (tweets responding to source tweets). Each tweet also has a rumor veracity information.

2) *Information about Tweet Threads:* Every thread contains the following information:

Source tweet, hashtags, favorite count, followers count, time of tweet, user mentions, URLs, retweet count, followings count, all user profile information

3) *Number of Incidents:* The dataset includes reports of nine incidents and each of the rumors is interpreted with its veracity meaning either true or false. In this research, we are using five events that are as follows:

Ferguson: On August 9, 2014, 18-year-old, African American Michael Brown shot by a Darren Wilson, a 28-year-old white Ferguson police officer.

Ottawa Shooting: On October 22, 2014, a shooting at Parliament Hill in Ottawa, Canada, resulted in the death of two, including the perpetrator.

Charlie Hebdo: On January 7, 2015, two armed gunmen killed 12 people injuring 11 others due to hurting their religious sentiments at the French satirical weekly newspaper Charlie Hebdo in Paris.

Sydney Siege: In a cafe in Sydney, Australia, on 15 December 2014, a lone gunman held 18 people hostages.

Data is first preprocessed. The preprocessing steps include: stop word removal, conversion to lower case, stemming and lemmatization. NLTK is used for stop word removal and Porter algorithm is used for stemming.

B. Supervised Rumor Detection

The primary objective of this work is to detect rumors from the tweets. The rumor detection task is divided into following subtasks:

1) Select salient words from tweets

2) Cluster these words to detect events using co-occurrence graph

3) Detect rumors from the events

Salient Words Selection: The "Occurrence of words in Hashtag" is used as a unique measure to calculate a word's likelihood of being topical. Similarly, sstatistical research reveals that conceptual and temporal characteristics

differentiate rumors over a long-term window from non-rumors. [14].

In this research we used three different methods to get salient words that are defined as follows:

Document frequency of words, for temporal dimensions we used the method of words occurrence in hash tags in a specific interval of time and entropy of words in a specific interval of time.

Document Frequency of Words - Document frequency is the number of times a word occurs within specific document. Document frequency is an important measure to determine how relevant a particular document is for a particular word. It is also important to understand the significance of terms within and across the documents. The formula of document frequency is:

df_i = Number of documents with term t in it / Total number of documents.

Word Occurring in Hash Tags in a Specific Interval of Time - Hashtags show the user's intentions and are associated with rumors. Therefore, the hashtags appearing again and again are more likely to be a salient word. By using the following formula we evaluate the word occurrence in hash tags:

hashtags = Number of words with hashtags / Total number of hashtags during certain time period.

Entropy of Words in a Specific Interval of Time

The salient words are not uniformly distributed so we use the concept of entropy. The formula of entropy is given as:

$$\text{entropy} = -\sum_{i=1}^{k} \frac{TF_i}{TF} * \log(\frac{TF_i}{TF})$$

Where TF_i is the word frequency appearing in the i_{th} hour and TF is the words' total frequency. We have performed experiments with different values of TF.

C. Grouping Salient Words into Cluster

If two salient words co-occur frequently, then they will be related to each other to show different exposures of a rumor or non-rumor.

Co-occurrence of Words: To find co-occurrence of words, we choose one salient word at a time and find their co-occurring words. Then select only those co-occurring words that have a certain threshold. After this, build a co-occurrence graph from the picked salient words. Co-occurrence graph comprises of vertices where each vertex represents a topical word. If two topical words (vertices) co-occur in at least one tweet then an edge is drawn between them. Each edge has a weight/count that is the frequency of co-occurrence.

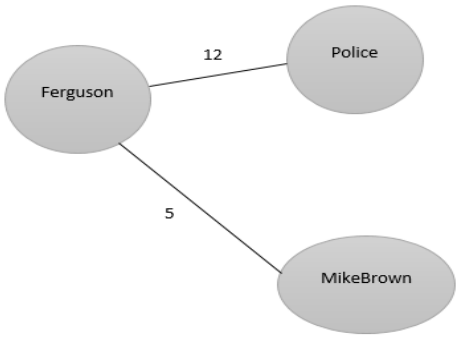


Figure 1: Co-occurrence Graph

Clustering of Co-occurrence Graph

After constructing the co-occurrence graph, we make clusters. Clustering will be done on the basis of context as shown in figure 1. The idea of performing clustering by taking contextual information into account can be traced back in experimental research [16] as well as in modelling works [17]. The topical words that come often with each other will make a cluster by using hierarchical divisive clustering algorithm using Ward's linkage method [18]. Since we have five events in our dataset so it will make five different clusters separating each event. Every cluster contains the topical words of one event. This step will help to differentiate events from each other. It searches for the very rare but very informative phrases and yields surprisingly good performance.

An inverted index is maintained for all the words in cluster containing the information of tweet and its label (rumored or non-rumored). If a test tweet is given, pass it from all the steps i.e. preprocessing and finding of salient words. Now find the count of each word that is already maintained in the inverted index of training phase. If the word has more rumored count than non-rumored, assign that word as a rumor and vice versa. If a word has similar counts for rumors and non-rumors, then we will find its bigrams. Again, find counts of rumors and non-rumors for all the bigrams of that word as shown in Figure 3. Also find out if the bigrams having similar counts for rumors and non-rumors then we will move towards the trigrams, but it is a rare case. In the end, considering the whole tweet if more words are rumored, we assign the whole tweet as a rumor and vice versa.

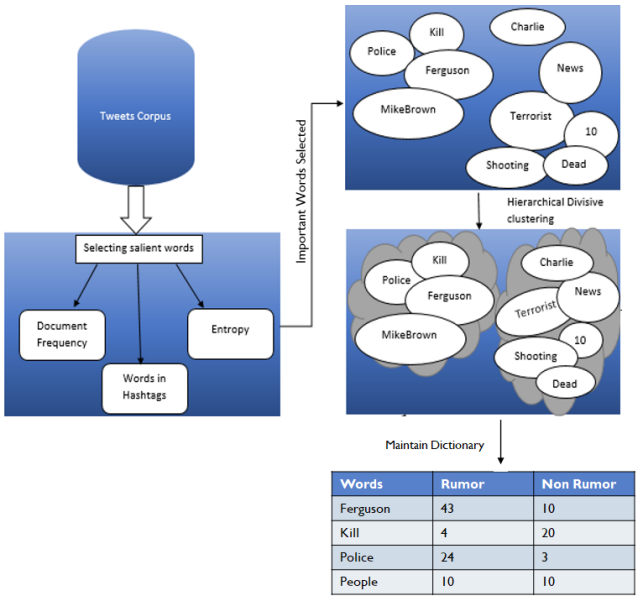


Figure 2: Workflow of Rumor Detection for Supervised Learning

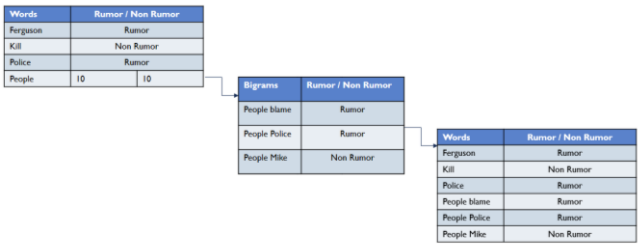


Figure 3: Bigrams Addition in Dictionary

D. Evaluation of Supervised Rumor Detection

Two factors can determine whether or not a cluster represents a rumor: features used for salient word selection and clustering algorithms. So, we split salient words selection into three levels.

In first level, we take only document frequency of words. If the document frequency of a word is greater than 0.6 then we select that word as an important word.

In second level, we take both document frequency and entropy. We have collected words by using different time frames as a threshold i.e. for 3600, 86400, 604800, 1,296,000, 2,592,000 seconds that is one hour, one day, seven days, fifteen days and one month. For example, if the word come within one hour then it will be selected as a topical word.

In third level, we take all features i.e. document frequency, entropy and hashtags. Results came out better with third level and considering the time frame of 86400 i.e. one day. The precision, recall and F1 scores of [4] and proposed methodology are given in Table1 and Table2 respectively.

Events	Precision	Recall	F1 Score
GermanWings Crash	74.3%	66.8%	70.4%
Charlie Hebdo	54.5%	76.2%	63.6%
Ottawa Shooting	84.1%	58.5%	69.0%
Sydney Siege	76.4%	38.5%	51.2%
Ferguson	56.6%	39.4%	46.5%

Table 1: State of the art results [4]

Events	Precision	Recall	F1 Score
GermanWings Crash	78.35%	81.25%	81.01%
Charlie Hebdo	74.98%	75.61%	75.24%
Ottawa Shooting	75.09%	79.75%	79.96%
Sydney Siege	74.33%	75.1%	65.99%
Ferguson	75.43%	82.5%	82.4%

Table 2: Proposed Methodology results for Labeled Data

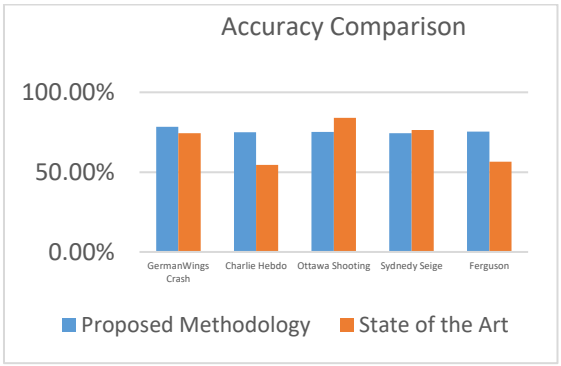


Figure 4: Comparison of Proposed Methodology with State-of-the-Art Method

In figure 4, we made the comparison between proposed methodology and state of the art method [4], it shows we have almost the same results for two events i.e. Germanwings and Sydney siege while our method worked well for two events i.e. Charlie Hebdo and Ferguson.

E. Unsupervised Rumor Detection

For unlabeled data, we have used contextual and behavioral features. Most of the times when some information is not confirmed, it includes inquiry phrases.

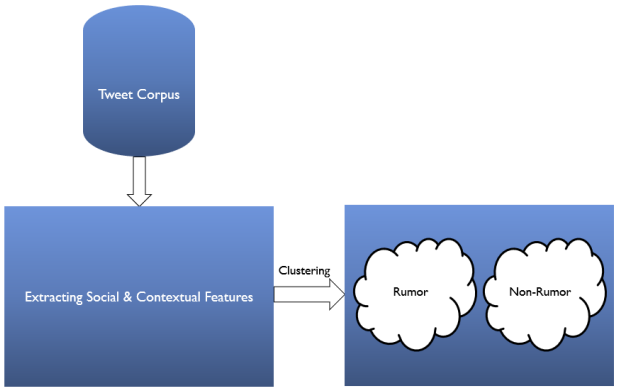


Figure 5: Workflow of Rumor Detection for Unlabeled Data

For every single event, we have used the features explained in [4], so we will take out the following features:

Enquiry Words

Within a tweet it is likely that the post would be speculated because the user is attempting to ask regarding the incident. So we calculated the number of enquiry words in a tweet. Enquiry words can be: 'What', 'where', 'how', 'who', 'why', '?' etc.

Time of Tweet

Time of tweet can be a very important point to figure out a rumor.

Retweet Count

How many times a tweet is retweeted can be a helpful clue for rumor detection. The speculated tweets will probably contain more retweet counts, because rumors spread faster.

Followers Count

If you have more follower counts, a user is authenticated. This is the way followers count assist in identifying rumors.

Listed Count

Listed count indicates the number of people who placed you in a list.

Vector of Tweets

Make vectors of tweets using doc2vec will help to get the contextual details of a tweet. Doc2Vec is a commonly used method that, regardless of its length, produces an embedded text. Although Word2Vec calculates a feature vector for every word in the corpus, similarly for every document in the corpus, Doc2Vec calculates a feature vector. The Doc2vec model is Word2Vec-based, with only another vector (paragraph ID) added to the data. It helps us to get the features of tweets for every incident in one document. This is how we will get five documents of features for every incident.

After selecting features, now perform K-means clustering based on these contextual and social features. One of the cluster will denote rumors and others will denote non rumors. Results are shown in Table 3.

Events	Accuracy	Precision	Recall	F1 Score
GermanWings Crash	65.04%	75.65%	65.31%	69.7%
Charlie Hebdo	60.23%	59.88%	79.61%	68.34%
Ottawa Shooting	59.88%	81.09%	66.75%	70.06%
Sydney Siege	62.45%	75.93%	60.01%	60.01%
Ferguson	66.22%	57.28%	64.62%	64.29%

Table 3: Proposed Methodology results for Unlabeled Data

IV. CONCLUSION

In this research, we proposed a workflow for rumor detection both for supervised and unsupervised learning. In case of supervised learning, three features are used to select the salient words. co-occurrence graph is constructed to make clusters. Finally classification is performed based on n-gram counts being rumored or non-rumored. In case of unlabeled data, rumors are detected by finding contextual and social features, and grouping them into rumor and non-rumor. Our proposed method gave better results than the state-of-the-art method of rumor detection.

V. REFERENCES

[1] Kate Starbird, Chris Schenk, "Promoting structured data in citizen communications during disaster response: an account of strategies for diffusion of the 'Tweak the Tweet' syntax", in Crisis Information Management, 2012.

[2] Á Figueira, L Oliveira, "The current state of fake news: challenges and opportunities", Procedia Computer Science, 2017.

[3] CR Sunstein, "On rumors: How falsehoods spread, why we believe them, and what can be done", 2014.

[4] A. Zubiaga, M. Liakata, and R. Procter, “Exploiting context for rumour detection in social media,” in International Conference on Social Informatics, 2017.

[5] V. P. Sahana, A. R. Pias, R. Shastri, and S. Mandloi, “Automatic detection of rumoured tweets and finding its origin,” *International Conference on Computing and Network Communications (CoCoNet)*, Trivandrum, 2015.

[6] E. Seo, P. Mohapatra, and T. Abdelzaher, “Identifying rumors and their sources in social networks,” in Proceedings of SPIE - The International Society for Optical Engineering, 2012.

[7] D. Westerman, P. R. Spence, and B. Van Der Heide, “Social Media as Information Source: Recency of Updates and Credibility of Information,” in Journal of Computer Mediated Communication, 2014.

[8] S. Wang and T. Terano, “Detecting rumor patterns in streaming social media,” in Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data, 2015.

[9] A. Zubiaga and H. Ji, “Tweet, but verify: epistemic study of information verification on Twitter,” in Social Network Analysis and Mining, 2014.

[10] S. Helmstetter and H. Paulheim, “Weakly supervised learning for fake news detection on Twitter,” in Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2018.

[11] S. Jain, V. Sharma, and R. Kaushal, “Towards automated real-time detection of misinformation on Twitter,” in International Conference on Advances in Computing, Communications and Informatics, ICACCI, 2016.

[12] Z. Zhao, P. Resnick, and Q. Mei, “Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts Categories and Subject Descriptors Detection Problems in Social Media,” in Proceedings of the 24th International Conference on World Wide Web, 2015.

[13] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu, “Towards effective event detection, tracking and summarization on microblog data,” in International Conference on Web-Age Information Management, 2011.

[14] S. Kwon, M. Cha, and K. Jung, “Rumor Detection over Varying Time Windows,” in PLoS One, 2017.

[15] “PHEME dataset for Rumour Detection and Veracity Classification,” https://figshare.com/articles/PHEME dataset for Rumour Detection and Veracity Classification/6392078.”

[16] O.H. Hamid, A. Wendemuth, and J. Braun, “Temporal context and conditional associative learning,” in BMC neuroscience, 2010.

[17] O.H., Hamid, “The role of temporal statistics in the transfer of experience in context-dependent reinforcement learning,” in 14th International Conference on Hybrid Intelligent Systems. IEEE, 2014

[18] Satyam Kumar, Hierarchical Clustering: Agglomerative and Divisive, 2019.

[19] W. Chen, C. K. Yeo, C. T. Lau and B. S. Lee, "Behavior deviation: An anomaly detection view of rumor preemption," IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016.

ARM Based Development of Embedded System for an Energy and Harmonic Analyzer

Ugur POLAT
Electrical and Electronics Engineering Department, Graduate School of Natural&Applied Sciences
Gaziantep University
Gaziantep,TURKEY
ORCID: 0000-0002-9526-1945

Ergun ERCELEBI
Electrical and Electronics Engineering Department, Graduate School of Natural&Applied Sciences
Gaziantep University
Gaziantep,TURKEY
ercelebi@gantep.edu.tr

Abstract—**In the recent decades, the topics on environmentally friendly, energy saving, renewable energy resources, that scientists are acting on them and new equipment that are developing day by day in technology invite us to safeguard our future in a very better way. The rise in energy consumption and therefore the gradual decrease in limited energy resources raise fundamental questions like what proportion energy is produced, what's its efficiency, how it distributes to the end users. Not only on an individual basis, but also on a worldwide basis research and development are disbursed on this subject. Additionally, required equipment usage in energy generation, distribution and utilization is increasing every day, too. Within the common systems for monitoring power circuits, you need to put in individual measuring devices e.g., Voltmeter, Ammeter, Oscilloscope Wattmeter etc. separately thanks to the increasing harmonic currents and voltages within the grids.**
Our design yields an industrial analyzer that sense the current and voltage at the input then designed circuit transmits signals proportional levels to the MCU which is able to be analyzed based on some standards such as IEEE Std 1459TM-2010 [1]. An A/D converter, converts the voltage and current signals to the digital representation at regular intervals. MCU periodically samples ($f_{sampling} = 1600$ Hz, $N = 32$, A/D resolution 12 − bits) then analyze the voltage and current signals at regular intervals of time with the approximately *0,4 % error rate. After that, stores the data of electrical variables which is analyzed. e.g., these electrical variables could also be frequency, true RMS volts, true RMS amps, cos(φ) and even harmonic values, including energy variables. For the project ARM Cortex family as STM-32F405 and LCD-TFT screen which part number is HY-32D are one among the foremost important components. The features of this device completely provide our requirements with isolation circuits, fuses and another safety equipments. According to another devices, this project will be cheap, easy to carry and includes much more features. e.g., Algorithm structures known as message queues were used in the system, unnecessary transaction crowd was eliminated and even small values were measured.
As a summary, it is an objective of the thesis to develop an industrial analyzer which a single analyzer device that analyzes the variable parameters in a grid by eliminating the require of multiple measuring devices. Thanks to the LCD, we are able to see all data only in a screen and it's completely mobilized.
***The error rate depends on the $f_{sampling}$, N, and A/D resolution.**

Keywords—***Embedded Systems, Arm Cortex, Energy Analyzer, Harmonic Analyzer, Single Phase Analyzer, STM-32f429, Digital Signal Processing, Real-time Operating Systems)***

I. INTRODUCTION

A. Literature Summary

The measurement in electrical systems is generally a numerical expression of an electrical quantity. There are lots of data in the electrical networks. In alternative current systems, generally voltage, current, frequency, phase difference, power factor, active power, reactive power, energy values and harmonical calculations are numerically measured values.[1]

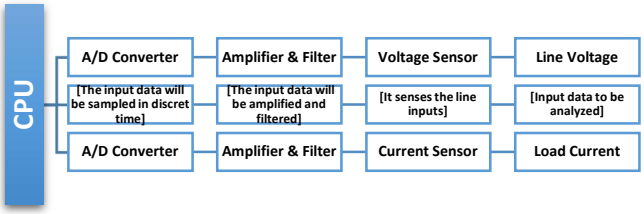


Figure 1: Equivalent system block diagram for analyzer

1) Effective Value in AC Systems

Voltage and current magnitudes change according to time and have a certain amplitude of frequency. For this reason, it is usually expressed with its effective values. Effective value is the value of an alternating current or voltage equal to the square root of the arithmetic mean of the squares of the instantaneous values taken throughout one complete cycle [1]. The effective value can be represented as the rms (root mean square) value consisting of the effective value or the initials of its English equivalent. The active power expended on a load is obtained by (2) in alternating and direct current systems [1]. According to this equation, the effective value can be expressed as the square root of the mean of the square of the voltage (or current), the integral over a period. In digital systems, the sum of the squares of instantaneous samples of the signal is averaged. The square root of this value is calculated and the effective value is found.

$$P = \frac{v^2}{R} \qquad (1)$$

$$\frac{v_{dc}^2}{R} = \frac{1}{T} \cdot \int_0^T \frac{v(t)^2}{R} \cdot dt \qquad (2)$$

[1]Merriam-Webmaster-Dictionary
https://www.merriam-webster.com/dictionary/effective%20value

$$\sqrt{v_{dc}^2} = \sqrt{\frac{1}{T} \cdot \int_0^T v(t)^2 \, dt} \quad (3)$$

$$v_{rms} = \sqrt{\frac{1}{T} \cdot \int_0^T v(t)^2 \, dt} \quad (4)$$

The effective (effective) value of the grid voltage in sine form can be written as in (6).

$$v(t) = v_m \cdot sin(\omega t) \quad (5)$$

$$v_{rms} = \sqrt{\frac{1}{T} \cdot \int_0^T \{v_m \cdot sin(\omega t)\}^2 \cdot d\omega\, t} \quad (6)$$

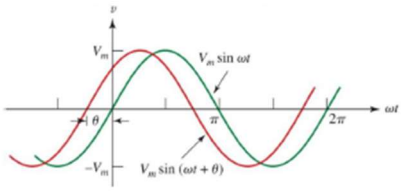


Figure 2:Value of the grid voltage in sine form

In the figure (2),

- The wave in red is said to *lead* the wave in green by θ.
- The wave in green *sin(ωt)* is said to *lag* the wave in red by θ.
- The units of θ and *ωt* must be consistent when computing the sine function.

The effective value of the network voltage in the sine form in the range of [0, 2π] can be mathematically calculated by (7) with the implementing of v_m to (6) where v_m as the peak value.

$$v_{rms} = \sqrt{\frac{v_m^2}{2\pi} \cdot \int_0^{2\pi} sin(\omega t)^2 \cdot d\omega\, t} \quad (7)$$

$$v_{rms} = \sqrt{\frac{v_m^2}{4\pi} \cdot \int_0^{2\pi} [1 - cos(2\omega t)] \cdot d\omega\, t} \quad (8)$$

$$v_{rms} = \sqrt{\frac{v_m^2}{4\pi} \cdot \left[\int_0^{2\pi} (1 \cdot d\omega t) - \int_0^{2\pi} cos(2\omega t) \cdot d\omega\, t\right]} \quad (9)$$

$$v_{rms} = \sqrt{\frac{v_m^2}{4\pi} \cdot [2\pi - 0]} \quad (10)$$

$$v_{rms} = \frac{v_m}{\sqrt{2}} \quad (11)$$

2) Phase Difference Between AC Voltage and Current

Inductance and capacitors in AC systems cause phase difference between voltage and current. If we take the V1 signal as reference in Figure 3, the V2 signal is in reverse phase according to the V1 signal, and the V3 signal is in forward phase according to the V1 signal. Even if the amplitudes are not equal, the frequencies must be equal in order to be able to talk about the phase difference. If there is a DC component on the signal, this component can be reset and the phase difference can be checked.

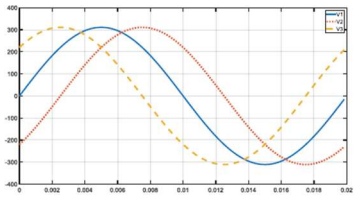


Figure 3: Sinusoidal signals that has the phase difference

If the load is pure ohmic, there is no phase difference between voltage and current. In case of inductive load, the current is in phase back from the voltage, in case of capacitive load the voltage is in phase back from the current.

This phase difference between voltage and current is denoted by "φ" and is called the phase angle. Since the phase angle close to zero is the most ideal situation for the network, the reactive power in the systems is compensated and the phase difference between the current drawn from the network and the voltage is tried to be reset. In this case, the reactive power required by the system is supplied from the compensation utility box that installed in the system.

Current and voltage signals with a phase difference of *57.3* ° is shown in Figure 4.

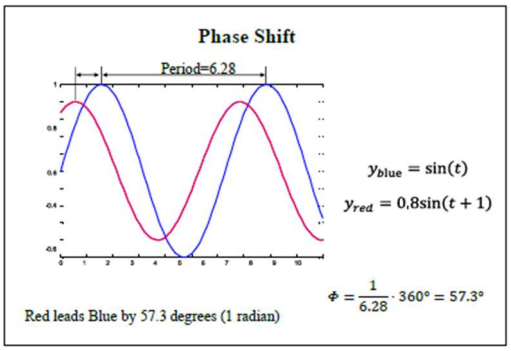


Figure 4:Sinusoidal signals that has phase difference

3) Power Calculation in AC Systems

Active power (P) in alternating current systems is defined as useful and usable power. Reactive power (Q) is defined as the power that does not work actively but is withdrawn from the network and then transferred back to the network. Apparent power (S) is the total resultant power formed by active and reactive power components. If the system has a pure ohmic load, there will be no phase difference between the current and voltage, so there will be no reactive power. Therefore, the active power value will be equal to the apparent power value. If there is a reactive power in the system, the apparent power will be equal to the combination of active and reactive powers [1].

$$S^2 = P^2 + Q^2 \quad (12)$$

$$P = S \cdot cos\varphi \quad (13)$$

$$Q = S \cdot sin\varphi \quad (14)$$

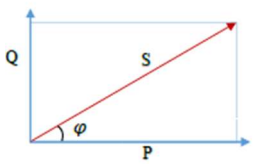


Figure 5: Power Triangle, Relationship between Power factors

In electrical systems, instantaneous power is expressed by (15), where $v(t)$ and $i(t)$ are instantaneous values of voltage and current.

$$P(t) = v(t) \cdot i(t) \quad (15)$$

Devices that measure electrical power measure the average value of power [1]. In this case, if the instantaneous power expression is integrated over a period, (17) can be written for the average power value (active power). In (5) we defined $v(t)$, similarly, we can define $i(t)$ as shown in (16). Then when we can implement (5) and (16) into (15) we get the active power(P).

$$i(t) = i_m \cdot sin(\omega t - \varphi) \quad (16)$$

$$P = \frac{1}{2\pi} \cdot \int_0^{2\pi} [v_m \cdot i_m \cdot sin(\omega t) \cdot sin(\omega t - \varphi)] \cdot d\omega t \quad (17)$$

$$P = \frac{v_m \cdot i_m}{2\pi} \cdot \int_0^{2\pi} [sin(\omega t) \cdot sin(\omega t - \varphi)] \cdot d\omega t \quad (18)$$

$$P = \frac{v_m \cdot i_m}{4\pi} \cdot \int_0^{2\pi} [cos(\varphi) - cos(2\omega t - \varphi)] \cdot d\omega t \quad (19)$$

$$P = 0{,}5 \cdot v_m \cdot i_m \cdot cos(\varphi) \quad (20)$$

$$P = v_{rms} \cdot i_{rms} \cdot cos(\varphi) \quad (21)$$

The reactive power value can be calculated by the product of the voltage component and the *90°* phase-shifted state of the current component or by the product of the current component and the *90°* phase-shifted state of the voltage component. If the *90°* phase shifted state of the current signal is expressed as Iq, (23) can be written for the reactive power. If we implement (5) and phase-shifted current all together;

$$i_q(t) = i_m \cdot sin\left(\omega t - \varphi + \frac{\pi}{2}\right) = i_m \cdot cos(\omega t - \varphi) \quad (22)$$

$$Q = \frac{1}{2\pi} \cdot \int_0^{2\pi} [v_m \cdot i_m \cdot sin(\omega t) \cdot cos(\omega t - \varphi)] \cdot d\omega t \quad (23)$$

$$Q = \frac{v_m \cdot i_m}{2\pi} \cdot \int_0^{2\pi} [sin(\omega t) \cdot cos(\omega t - \varphi)] \cdot d\omega t \quad (24)$$

$$Q = \frac{v_m \cdot i_m}{4\pi} \cdot \int_0^{2\pi} [sin(2\omega t - \varphi) + sin(\varphi)] \cdot d\omega t \quad (25)$$

$$Q = 0{,}5 \cdot v_m \cdot i_m \cdot sin(\varphi) \quad (26)$$

$$Q = v_{rms} \cdot i_{rms} \cdot sin(\varphi) \quad (27)$$

e.g., For figure (6), assume that $v_{rms} = 230\ V, i_{rms} = 12\ A$

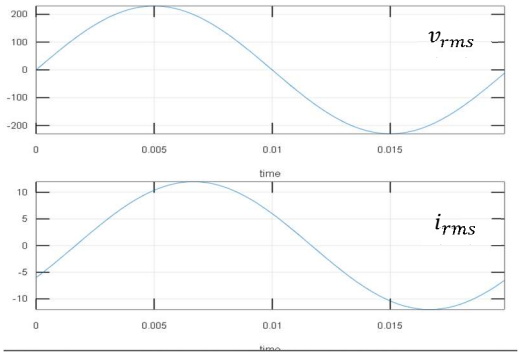


Figure 6:Voltage and current graph for the load that has 30-degree phase difference.

-Active Power → $P = v_{rms} \cdot i_{rms} \cdot \cos(\varphi) = 230 * 12 * \cos(30) = 2{,}390.230\ watt$

-Reactive Power → $Q = v_{rms} \cdot i_{rms} \cdot \sin(\varphi) = 230 * 12 * \sin(30) = 1{,}380.000\ VAr$

-Apparent Power → $S = v_{rms} \cdot i_{rms} = 230 * 12 = 2760{,}000\ VA$

will be calculated as above. In digital systems, it is possible to calculate the phase difference and power values based on the instantaneous values of voltage and current.

In MATLAB environment, instantaneous values of current and voltage and power values can be calculated by using averaging command.

```
t = [0: 1e-6: 0.02];                    active_power = mean (v.*i);
v = 230*sqrt (2) *sin(2*pi*50*t);       reactive_power = mean (v.*iq);
i = 12*sqrt (2) *sin (2*pi*50*t – (pi/6));  apparent_power=230*12;
iq = circshift (i, [5e3, -5e3]);
```

4) Power Factor and Cos (φ) in Power Systems

In power systems, the ratio of active power to apparent power is defined as the power factor [1]. If the network voltage and current are in sinusoidal form, that is, if the system does not contain a non-linear load, *cos (φ)* is equal to the power factor (PF).

In the equation (21) and (27) we get (P) and (Q) then we can use them to find (S) value as in (28)

$$S = \sqrt{P^2 + Q^2} \quad (28)$$

$$cos(\varphi) = PF = P/S \quad (29)$$

In addition to that, if there are non-linear loads such as rectifiers, uninterruptible power supplies, electronic ballasts in power systems, current and voltage signal shapes move away from the sinusoidal form and harmonics occur in the system. In a harmonic system, power expressions such as the effective value of the voltage fundamental component v_1 and the effective value of the current fundamental component i_1 can be written as follows.[1]

$$S_1 = \sqrt{{P_1}^2 + {Q_1}^2} \quad (30)$$

$$P_1 = v_1 \cdot i_1 \cdot cos(\varphi) \quad (31)$$

$$Q_1 = v_1 \cdot i_1 \cdot sin(\varphi) \quad (32)$$

$$PF = \frac{v_1}{v_{rms}} \cdot \frac{i_1}{i_{rms}} \cdot cos(\varphi) \quad (33)$$

$$PF = K_{v_distortion} \cdot K_{i_distortion} \cdot cos(\varphi) \quad (34)$$

In (34), $K_{distortion}$ coefficients are defined as distortion factors for current and voltage, and cos(φ) is defined as the displacement factor. If it is assumed that there is no harmonic in the network voltage, the $K_{v_distortion}$ coefficient will have no effect and the power factor value will be as in (35).

$$PF = 1 \cdot K_{i_distortion} \cdot cos(\varphi) = K_{i_distortion} \cdot cos(\varphi) \quad (35)$$

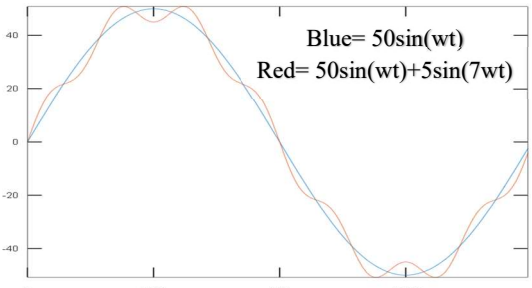


Figure 7: A current signal with the 7th harmonic in it

5) Harmonic Distortion

In electrical power systems containing harmonics, the effective values of current and voltage can be expressed in the following equations.

$$v_{rms} = \sqrt{v_1^2 + v_2^2 + v_3^2 + \cdots + v_n^2} \quad (36)$$

$$v_{rms} = \sqrt{v_1^2 + \sum_{h=2}^{n} v_n^2} \quad (37)$$

$$i_{rms} = \sqrt{i_1^2 + i_2^2 + i_3^2 + \cdots + i_n^2} \quad (38)$$

$$i_{rms} = \sqrt{i_1^2 + \sum_{h=2}^{n} i_n^2} \quad (39)$$

Total harmonic distortion value can be defined as the ratio of harmonic components to fundamental components. This value can be written separately for current and voltage.

$$THD_v = \frac{\sqrt{v_2^2+v_3^2+\cdots+v_n^2}}{v_1} \quad (40)$$

$$THD_i = \frac{\sqrt{i_2^2+i_3^2+\cdots+i_n^2}}{i_1} \quad (41)$$

B. Purpose of the Experimentation

Today, digital measurement systems are generally preferred in order to be able to measure in alternating current systems and to have healthy data about energy quality. The accuracy of electrical quantities measured in these systems depends on certain parameters. [1]

It is aimed to design an embedded system by using various methods to investigate and compare the effects of these parameters on the measurement results and to reduce the errors that may occur.

C. Key Findings

The duty cycle frequency of the MCU to be used in the designed digital analyzer system, the resolution of the analog-digital converter, the sampling rate and the number of channels that can operate simultaneously are compatible. The analyze can be performed without losing time in sampling. In addition, if the temperature dependence of the sensors and electronic materials used is calibrated with an external temperature measurement, a fault-free measurement can be obtained.

II. PHASE LOCKED LOOP (PLL) DESIGN

In grid-connected applications, phase angle must be calculated accurately in order to work simultaneously with the grid [8]. Phase angle is determined by a software phase locked loop (PLL) without being affected by distortions such as frequency shifts, voltage changes or harmonics that may occur from time to time in the network.

In general, the phase locked loop is a closed loop control system with an internal oscillator to be able to lock onto the phase of an external signal using a feedback structure. It can be simply expressed as a control system that minimizes the phase difference between the reference signal and the output signal. In devices connected to the network, there may be situations such as frequency change, absence of phase or imbalance [8]. In such cases, a phase locked loop design should be designed to minimize these errors for the stability of measurement systems. As seen in Figure 8, a simple phase locked loop (PLL) which consists of a phase detect (PD), a loop filter (LPF), and a voltage-controlled oscillator (VCO).

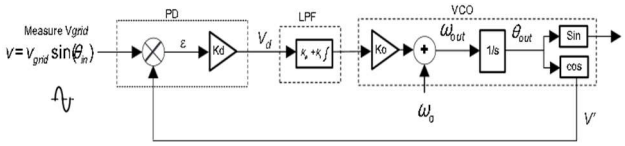


Figure 8:Phase Locked Loop Block Diagram

The discrete time block diagram of the classical PLL model is as follows.

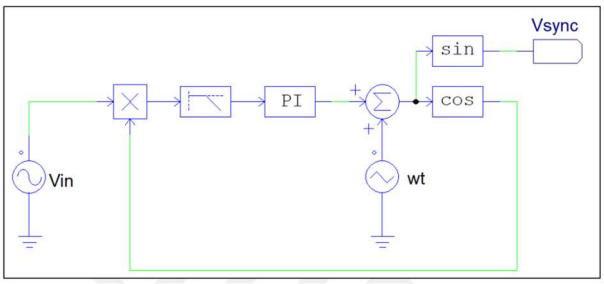


Figure 9: Classic PLL PSIM model

Thanks to the synchronization signal obtained by the PLL model, synchronization with the network will be ensured in case of disturbances in the network and frequency shifts. As an example of the discrete time block diagram of the PLL model obtained in Figure 9, it is seen in Figure 10 that it is properly locked into phase at the output against the sinusoidal input signal with phase difference. Filter coefficients for the sample are as follows.

Second order low pass filter coefficients;
b_0 = 0,00015515946
b_1 = 0,00031031892
b_2 = 0,00015515946
a_1 = -1,9648072
a_2 = 0,96542785
PI controller coefficients;
b_0= 10,5
b_1= -9,95

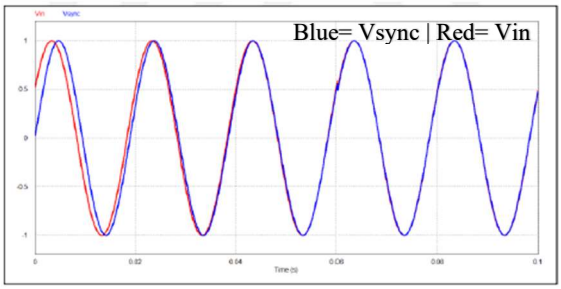


Figure 10:Response of designed PLL to sinusoidal input signal

III. HARDWARE AND SOFTWARE DESIGN

The energy requirement of the equipment to be designed is reduced gradually to 12V DC voltage with the converter circuit within the system with 230V AC voltage. All digital elements on the circuit (Microcontroller, OPAMP, Communication Peripherals etc.) except the TFT LCD screen, operate with 3.3V DC voltage. Only TFT LCD screen operates with 5V DC voltage. 12V DC voltage taken from the converter circuit is converted into 5V DC and 3.3V DC voltage by linear voltage regulators step by step. LM7805 is employed for 5V conversion and 78M33 linear regulator is employed for 3.3V conversion. additionally, diodes are added to the input sections for linear current control. USB,

Bluetooth, RS485 for communication within the designed hardware and such the pins of CPU for debugging BOOT0 and CLK pins for serial communication are given from the PCB via headers. it's hospitable coding as desired. Altium Designer, MATLAB, Proteus, PSIM and other engineering programs were used for the planning.

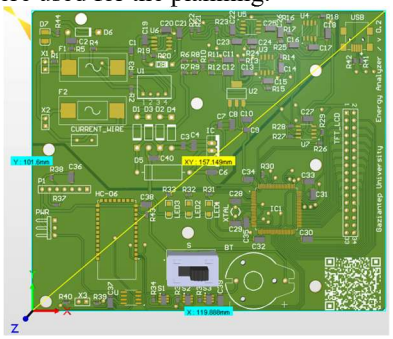


Figure 11: Designed v0.2 PCB in Altium Designer

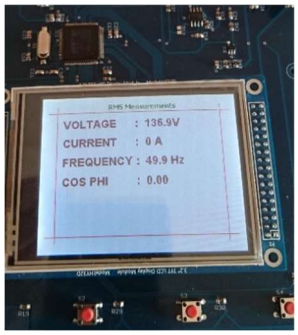


Figure 12:Assemblied Designed PCB v.0.1

IV. EXPERERIMENTAL RESULTS

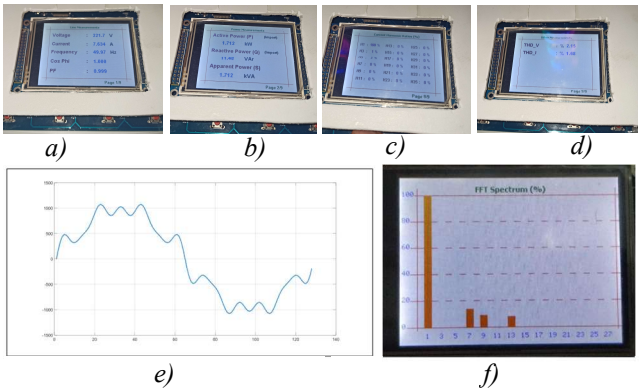


Figure 13: a) Grid Magnitudes b) Power Magnitudes c) Harmonics on the current d) Total distortion on the grid e) 1000sin(wt) + 150sin(7wt) + 100sin(9wt) + 80sin(13wt) signal f) FFT spectroscopy of (e).

V. CONCLUSION AND FUTURE WORKS

Within the scope of this thesis, an embedded system has been developed so as to make numerical measurements in electrical grid systems and therefore the factors which will affect the measurement are investigated by making experiments on various systems supported IEEE 1459-2010 Standards [10].

Furthermore, it's proved that various factors directly affect the measurement quality. Temperature changes of electronic components used are one in all the important factors. thanks to the optimal working environment of the components are different, so as to attenuate the negative impact of this case, the materials used is preferred with higher sensitivity and lower temperature coefficient. Also, the field effect sensor utilized in the current input is tormented by the external magnetic field. To avoid negative effect of this case, it's going to be possible to produce magnetic isolation of the area where the sensor is located on the hardware or standard transformer like rated 250/5A is also accustomed get far better results with the high precision. However, a current transformer requires much space. Besides, sensitivity of the operational amplifiers utilized in voltage and current inputs is critical. a low precision operational amplifier will cause irrelevant results, especially when making measurements too close to zero. To avoid this case, choosing a high quality and high precision operational amplifier will influence positively affect the results. Additionally, chosen MCU is vital. If a microcontroller with high performance and decimal processing capability isn't preferred, it'll influence negatively the results in order that it'll cause loss of sensitivity during the mathematical operations. For the remainder, the second revolution of PCB also generated and everyone necessary revisions are upgraded. thanks to its open source, you'll be able to modify the PCB software as you desire. The software is sort of the identical but the screenshotted PCB is modified.

REFERENCES

[1] IEEE-Std 1459-2010, "IEEE Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or unbalanced Conditions", 2010.

[2] C. K. Alexander and M.N.O. Sadiku, "Fundamentals of Electric Circuits", 5th Edition, Mc. Graw Hill, 2013.

[3] T.M. Blooming and D.J. Carnovale, "Application of IEEE STD 519-1992 Harmonic Limits", Annual Pulp and Paper Industry Technical Conference, Appleton, WI, June 18- 23 2006.

[4] Texas Instruments, "Software Phase Locked Loop Design Using C2000 Microcontrollers for Single Phase Grid Connected Inverter", Application Report, (2013-rev.2017).

[5] ST Microelectronics, "STM32F405xx STM32F407xx Microcontroller Datasheet", 2020.

[6] A. Özdemir and M. Taştan, "PLL Based Digital Adaptive Filter fosr Detecting Interharmonics", Hindawi Publishing Corporation Mathematical Problems in Engineering, Article ID 501781, 2014.

[7] J. W. Nilsson and S. Riedel, "Electric Circuits", Chapter 9, 9th Edition, NILSSON RIEDEL, January 13, 2010.

[8] D. Stojic, N. Georgijevic, M. Rivera and S. D. Milic, "Novel Ortogonal Signal Generator for Single Phase PLL Applications", IET Power Electronics, Volume:11, Issue:3, page: 427-433, March 01, 2018.

[9] LSIS Co. Ltd., "Electric power measuring system", JUSTIA Patents, Patent: 9863986, July 13, 2016.

[10] P. Wattanayingcharoen, A. Detchrat and S. Chitwong, "Developing Harmonic Power Analyzer based on IEEE 1459-2010 Standard", Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol II, IMECS 2012, Hong Kong, March 14-16, 2012.

New Geometric Based Features for Facial Expression Recognition

Nuri ÖZBEY
R&D Department
Arçelik A.Ş. Refrigerator Plant
Eskisehir, Turkey
nuri.ozbey@arcelik.com
ORCID: 0000-0002-5980-9985

M. Bilginer GÜLMEZOĞLU
Electrical and Electronics Engineering
Eskişehir Osmangazi University
Eskisehir, Turkey
bgulmez@ogu.edu.tr
ORCID: 0000-0002-1570-4989

***Abstract*—Facial expression recognition is significantly beneficial to computer vision due to its potential applications. Facial expression recognition includes machine and human interaction, security, psychology, and changing appearance in social media applications. In this paper, new geometric based features are proposed for the recognition of seven facial expressions. The number of features is reduced by using feature selection methods. Obtained features are applied to Support Vector Machines (SVM) classifier. In the experimental studies, the extended Cohn-Kanade (CK+) dataset is used and facial expressions are classified by using the 10-fold cross-validation method. Classification accuracy of 93.5% is achieved in CK+ dataset with the features selected by the Sequential Backward Feature Selection method.**

***Index Terms*—facial expression recognition, geometric based features, feature selection, support vector machine**

I. Introduction

Facial expression is one of the most effective, natural, and important tools for people to show their emotions and objectives [1]. Facial expressions are commonly used in behavioral understanding of emotions, cerebral science, and social collaborations [2]. There are two common methods to obtain facial features: geometric feature-based methods and appearance-based methods [1]. The geometric features introduce the shape and positions of facial components such as mouth, nose, eyes, and eyebrows that are extracted to create a feature vector representing facial geometry. In appearance-based methods, image filters such as Gabor wavelets are applied to the entire face or specific facial areas to remove facial appearance differences, including wrinkles, bumps and grooves

In the last decade, automatic facial expression recognition has been gaining tremendous attention and has become a crucial topic in scientific society because facial expressions are one of the most robust, natural, and immediate tools for people to show their emotions and intentions [3]. The recognition rate of individual facial expressions is not high, which decreases the success rate of real-time facial expression recognition as well. The main complexity of independent facial expression recognition is that the appearance of the human face affects the correct acquisition of expressive features [4]. In recent years, researchers have implemented many enhancements to the expression recognition systems to improve the facial expression recognition rate.

By taking the median of the landmark tracking results from facial expression sequences training, the classic expression sequence is created for each facial expression class. [2]. Bartlett et al. presented results on an user independent fully automatic system for real time recognition of basic facial expressions from video [5]. They presented an approach to obtain a further speed advantage by combining feature selection based on Adaboost with feature integration based on SVM.

Statistical local features, local binary pattern (LBP), are widely used in facial expression recognition especially with the SVM [3], [6]. Appearance and geometric based features are also used with SVM in facial expression recognition [7], [8]. Li et al. proposed a recognition algorithm of person independent facial expression based on improved LBP and HOSVD (Higher-Order Singular Value Decomposition) [4]. In the phase of facial expression classification and recognition, the conventional nearest neighbor classification is modified into k- nearest neighbor pre-classification, and the local energy obtained by HOSVD is utilized to specify the resemblance of two images for secondary classification.

The Convolutional Neural Network (CNN) is another popular method used for facial expression recognition [5], [9]. Liu et al. proposed a FER model based on improved CNN for Sobel edge detection and fused SVM [10]. To solve the problem of facial recognition in face closure, Feng and Shao proposed a human eye facial expression recognition model for transfer learning [11]. Bartlett et al. introduced a methodical comparison of machine learning methods employed to the automatic recognition problem of facial expressions [12]. They demonstrated their results on a sequence of experiments by comparing recognition methods, including AdaBoost, SVM, linear discriminant analysis.

In this paper, seven facial expressions in the CK+ dataset are classified. For this purpose, new geometrical-based features are extracted from the landmark points on the facial images. These features and the features selected by three selection methods are applied to the SVM classifier.

The paper is organized as follows: the feature extraction

(a) (b) (c) (d)

Fig. 1: (a) and (b) denotes feature landmark points, (c) and (d) denotes selected 28 features from SBFS.

and feature selection are given in Sections II and III respectively. The classifier and experimental study are presented in Sections IV and V respectively. Finally, Section VI includes the conclusion.

II. Feature Extraction

Feature extraction is the first phase of the facial expression recognition that is mainly composed of three stages: face detection, facial landmark tracking, extracting features from the landmark tracking result. We used the Viola-Jones face detection algorithm in the face detection phase [13]. Facial landmarks are taken from the dlib's facial landmark detector [14]. 68 facial landmarks are obtained by using the same detector. In literature, most of the researchers use both neutral state and expressions state in the feature extraction. However, in our work, the first group of extracted features corresponds to the Euclidean distances between two odd-numbered landmarks from 1 to 67, 3 to 67, ... etc. The second group of extracted features corresponds to the Euclidean distances between two even-numbered landmarks from 2 to 68, 4 to 68, ... etc. Feature vectors were created by combining these two groups and a total of 1122 features were extracted.

III. Feature Selection

Most of the feature selection methods can be divided into 3 main categories. These are filter methods, wrapper methods, and tree-based models. In this study, three feature selection methods are used to reduce the size of feature vectors. These methods are two Wrapper-Methods(Sequential Forward Feature Selection and Sequential Backward Feature Selection) and Principle Component Analysis (PCA), which is very common in dimension reduction. They are briefly described in the subsections below.

A. Sequential Forward Feature Selection (SFFS)

In SFFS, first, the best single feature is selected, then two pairs of features are formed using one of the remaining features and this best feature, and the best feature pair is selected. In the third step, triplets of features are taken using one of the remaining features, and these two best features and the best triplet is selected. These steps are repeated until the best representative subset of features is selected. After the application of SFFS, we obtain the best 60 features.

B. Sequential Backward Feature Selection (SBFS)

The second feature selection method is SBFS. In this method, first, the classification function is computed for all features. Then, each feature is deleted once at a time, the classification function is computed for all subsets with all minus deleted one, and one feature having the worst classification rate is discarded. These steps are repeated until the best subset of features is selected. After the application of SBFS, we have the best 28 features

C. Feature Selection with Principal Component Analysis

Principal Component Analysis (PCA) is a technique that is mostly applied for dimension reduction. The main purpose of PCA is to keep the data set with the highest variance in high dimensional data while applying dimension reduction. It provides a lower dimension by finding the general properties of the given dimension. Certain features will be lost with size reduction; but intended, these disappearing have little informational characteristics about the classification. In this study, we applied PCA as a feature selection method. In the CK+ dataset, the number of eigenvalues is tuned from 2 features to all features. Best recognition results are obtained by the selection of 268 features.

IV. SVM Classifier

SVM is a supervised learning method which is commonly used for classification, regression, and outliers detection [15]. In this classifier, a datum item is indicated as a point into the n-dimensional space along with the value of each feature corresponding to a specific coordinate. Also, the classification is implemented by finding the hyper-plane that discriminates the classes. In SVM, if there is no linear hyper-plane between two or more classes, a method called the kernel trick is applied. In this study, the RBF kernel is applied to classify 7 facial expression classes.

V. Experimental Study

In the experimental study, the CK+ dataset is used, and test results are demonstrated in the following subsections.

A. Dataset

In the Facial Expression Recognition study, the Extended Cohn-Kanade (CK+) dataset [16] was used. This dataset was widely used by many researchers [1]–[9], [11], [12], [17]. The dataset contains facial images from 123 subjects with 7 emotions (i.e., anger, contempt, disgust, fear, happiness, sadness, and surprise). In our study, the first of frame sequence belonging to each emotion is chosen as a neutral frame. Also, the last two or three emotional frames have been chosen to increase the number of samples on each emotion in the dataset. This leads to a total of 989 images of 8 classes. For a fair comparison with other many studies, we use 7 facial expression classes without a neutral class in the recognition process. Finally, we have 866 images whose distributions are given based on their classes in Table-I.

TABLE I: The number of samples for each class in CK+ dataset

Anger	Contempt	Disgust	Fear	Happiness	Sadness	Suprise	**Total**
90	82	158	121	138	111	166	**866**

B. Results and Discussions

During the classification process, SVM is used together with the RBF kernel. 10-fold cross-validation is used to obtain average accuracy results. In this method, the dataset is divided into 3 parts as training, testing, and verification. A recognition rate of 88.2% is obtained from the RBF kernel of SVM using features without any selection. The confusion matrix for this case is given in Figure 2. 89.5% and 93.5% recognition rates are obtained using the RBF kernel of SVM for the features selected by SFFS (60 features) and SBFS (28 features) respectively. The confusion matrices for these cases are given in Figures 3 and 4 respectively. When the 268 features that

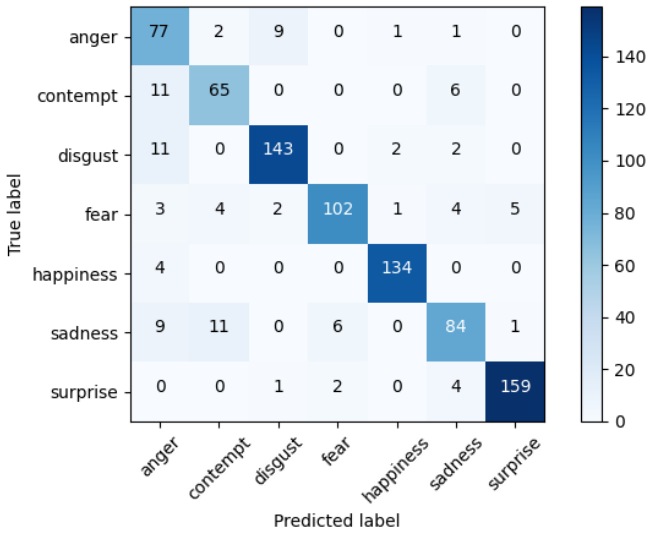


Fig. 2: Confusion matrix obtained for the features without selection.

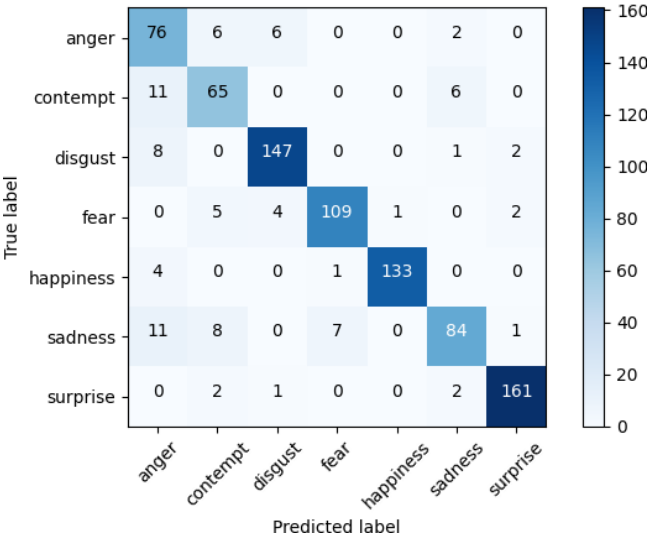


Fig. 3: Confusion matrix obtained for the features selected by SFFS.

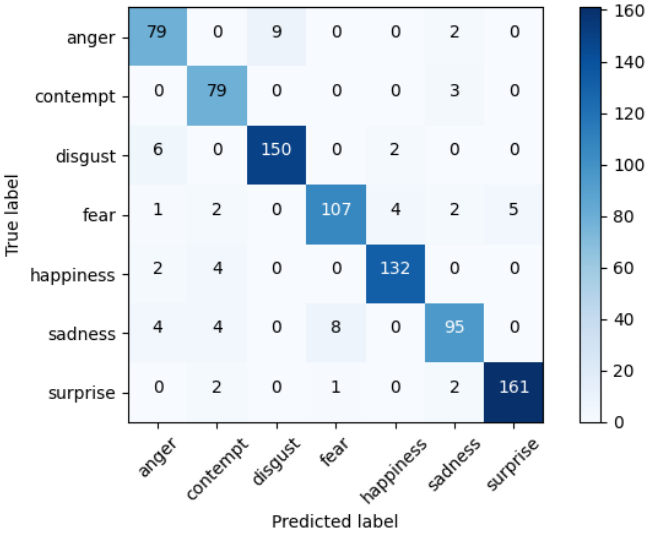


Fig. 4: Confusion matrix obtained for the features selected by SBFS.

TABLE II: The Classification Results on the CK+ Dataset

Feature Selection Method	Accuracy on CK+ Dataset in Percentage				
	Proposed Method	RBF Kernel SVM [7]	Linear Kernel SVM [12]	*LDA [12]	RBF Kernel SVM [17]
None	**88.2**	84.7	88.0	44.4	81.1
SFFS	**89.5**	88.7	–	–	–
SBFS	93.5	–	–	–	–
PCA	**90.8**	–	75.5	80.1	–

LDA : Linear Discriminant Analysis

are selected by PCA are used with RBF Kernel of SVM the recognition rate of 90.8% is obtained. The confusion matrix of RBF kernel SVM classifier for feature selection with PCA is given in Figure 5.

The feature vectors which are obtained by Euclidean distances are classified with RBF kernel SVM by using 10-fold cross-validation. The results of the feature selection methods are given in Table-II. The best classification accuracy of 93.5% is achieved in the CK+ dataset with 28 features selected by SBFS in the proposed method. This result is also higher than the results given in references [7] and [17] when the RBF kernel SVM is used as a classifier.

When linear kernel SVM and Linear Discriminant Analysis (LDA) are used as in [12], the classification accuracies are less than those of the proposed method. On the other hand, the study in [12] has inspired our study in order to apply feature selection methods on the geometric features. It has been proven that facial expression recognition accuracy will increase by using PCA and Adaboost when selecting features. The proposed method has also achieved more successful facial expression recognition when compared with the results given in [7] and [12]. The results show better classification accuracy than [17] when only the features are used.

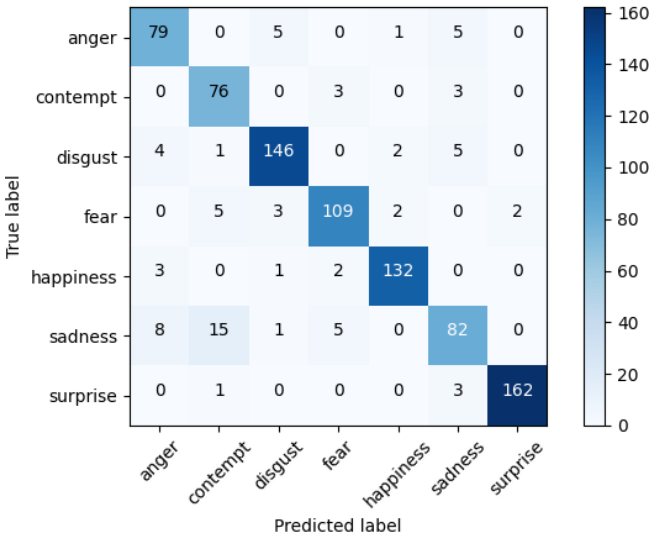


Fig. 5: Confusion matrix obtained for the features selected by PCA.

VI. Conclusion

In this study, different geometric features of landmark points on the facial images are proposed in facial expression recognition rather than using usual landmarks. The Euclidean distances between two of the landmark points by skipping one landmark point are used when obtaining geometric features. When these features are used to classify seven facial expressions with the SVM classifier,the best results are obtained with the proposed method compared with the results given in the literature.

Three different feature selection methods are applied during the facial expression classification stages due to the negative impact of some features. When SBFS is used 28 features are selected and the classification accuracy of 93.5% is obtained. When SFFS and PCA are used 60 and 268 features are selected and the classification accuracies of 89.5% and 90.8% are obtained respectively. These results are also higher than the results given in [7], [12], [17].

In feature studies, it is aimed to achieve facial expression recognition with a higher success rate by applying geometric and appearance-based features together. In addition, it is planned to expand the work with deep learning methods to increase the classification performance.

Acknowledgment

We would like to thank Prof. Jeffery Cohn and Megan Ritter for the use of the Extended Cohn–Kanade dataset.

References

[1] C. Shan, S. Gong and P. W. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study”, Image and Vision Computing, Vol. 27, pp. 803–816, 2009.

[2] D. Ghimire and J. Lee, “Geometric Feature-Based Facial Expression Recognition in Image Sequences Using Multi-Class AdaBoost and Support Vector Machines”, Sensors, Vol. 13, pp. 7714-7734, 2013; doi:10.3390/s130607714.

[3] X. Zhao and S. Zhang,” Facial Expression Recognition Based on Local Binary Patterns and Kernel Discriminant Isomap”, Sensors, Vol. 11, pp. 9573-9588, 2011; doi:10.3390/s111009573.

[4] Y. He and S. Chen, “Person-Independent Facial Expression Recognition Based on Improved Local Binary Pattern and Higher-Order Singular Value Decomposition”, IEEE Access, Vol. (, pp. 190184-190193, 2020; doi: 10.1109/ACCESS.2020.3032406.

[5] M. S. Bartlett, G. Littlewort, I. Fasel and J. R. Movellan, “Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction”, Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW03), 16-22 June, Madison, Wisconsin, USA.

[6] Caifeng Shan, Shaogang Gong and P. W. McOwan, ”Robust facial expression recognition using local binary patterns,”IEEE International Conference on Image Processing 2005, Genova, 2005, pp. II-370, doi: 10.1109/ICIP.2005.1530069.
[7] C. Gacav, B. Benligiray and C. Topal, ”Greedy search for descriptive spatial face features,”2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 1497-1501, doi: 10.1109/ICASSP.2017.7952406.
[8] T. Mohammad and M. L. Ali, ”Robust facial expression recognition based on Local Monotonic Pattern (LMP),”14th International Conference on Computer and Information Technology (ICCIT 2011), Dhaka, 2011, pp. 572-576, doi: 10.1109/ICCITechn.2011.6164854.
[9] L. S. Videla and P. M. A. Kumar, ”Facial Expression Classification Using Vanilla Convolution Neural Network,” 2020 7th International Conference on Smart Structures and Systems (ICSSS), Chennai, India, 2020, pp. 1-5, doi: 10.1109/ICSSS49621.2020.9202053.
[10] S. Liu, X. Tang and D. Wang, ”Facial Expression Recognition Based on Sobel Operator and Improved CNN-SVM,” 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP), Shanghai, China, 2020, pp. 236-240, doi: 10.1109/ICICSP50920.2020.9232063.
[11] H. Feng and J. Shao, ”Facial Expression Recognition Based on Local Features of Transfer Learning,”2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 2020, pp. 71-76, doi: 10.1109/ITNEC48623.2020.9084794.
[12] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel and J. Movellan, ”Recognizing facial expression machine learning and application to spontaneous behavior,” 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 2005, pp. 568-573 vol. 2, doi: 10.1109/CVPR.2005.297.
[13] P.A. Viola and M.J. Jones, ”Robust real-time face detection”, IEEE International Conference on Computer Vision (ICCV), 2001.
[14] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009
[15] M.A. Hearst ”Support Vector Machines,” in IEEE Intelligent Systems and their Applications , 1998.
[16] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z. and Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101
[17] D. Ghimire, S. Jeong, S. Yoon, J. Choi and J. Lee, ”Facial expression recognition based on region specific appearance and geometric features,” 2015 Tenth International Conference on Digital Information Management (ICDIM), Jeju, 2015, pp. 142-147, doi: 10.1109/ICDIM.2015.7381857.

Spectrogram Images Based Identification of Bird Species Using Convolutional Neural Networks

Jutyar Awrahman
Department of Computer Engineering,
Karabuk University,
Karabuk, Turkey,
jutyarfa92@gmail.com

Hakan Kutucu
Department of Computer Engineering,
Karabuk University,
Karabuk, Turkey,
hakankutucu@karabuk.edu.tr

***Abstract*—The identification of bird species by their sounds is one field of research. This paper focuses on identifying bird species based on spectrogram images using Convolutional Neural Networks (CNN). This represents more than a challenge when speaking of advanced identification of bird species with spectrogram analysis. Different CNN architecture models and representations of the spectrum have been trained, validated, and tested on 10600 audio instances, which belongs to 437 different classes in our dataset. We had concluded that CNN allows achieving good results since it eliminates the potential modeling errors from results of the incomplete or inaccurate bird species knowledge. The models were implemented by using the Python programming language and the Librosa library. The bird sounds have been obtained from different areas in Turkey, and we performed pre-processing operations to create a spectrogram dataset. We divided the label sounds for training, testing, and validation. Then, the samples were put into 10-folds. The system trains for 300-epochs and the loss is 1.0207 and the overall training accuracy stands at 0.91.**

Keywords—Birds sound dataset, Classification, CNN, Spectrogram.

I. INTRODUCTION

The monitoring of animals' populations is an important issue in ecology. The use of acoustics for classifying and monitoring animals in their environments is an interesting subject lately. Using recorded sound data for animal species classification is useful for monitoring biodiversity, breeding, and population dynamics [1].

The aim of this paper is to detect and classify bird species using a Convolutional Neural Network (CNN). The detection of bird species is a useful issue to solve, and it has been a major cause of birds' conservation for many years. One of the most common problems of bird species detection is the fact that bird species fall into several types [2]. Speech researchers are worked on imagine data in the early stages in 1960s and the spectrogram was used, Representation of information by using spectrogram produced a good feature for sound classification [3].

Audio systems are highly advanced nowadays, yet there are still a lot of false results [4]. Due to the differences between negative and positive proportions, the system always tries to classify the regions and use some of them later. To increase the accuracy of a detection system, separate models are used for classification [5].

With the improvement of the deep neural network, many new architectural models appear. Hence, it is critical to assess which provide the finest execution, with the lowest time utilization. In some cases, deep neural network systems are overkill for a specific task, whereas using less complex strategies can produce comparable results while, at the same time, sparing assets. Subsequently, working on each issue needs detailed and a lot of studies.

The organization of the paper is as follows: The related work is given in Section II. The bird sounds dataset, representation of the spectrograms, and different CNN architecture models are introduced in Section III, an explanation of the results are shown in Section IV, and finally, the conclusion of the paper in Section V.

II. RELATED WORK

In this section, we present the latest research papers and projects related to our work. Jaiswal et al. [6] used a Convolutional Neural Network as a method to classify sounds. The dataset used consisted of various urban sounds and spectrograms generated from these sounds, to be used by a CNN, and various layers were used. Although the accuracy of state-of-the-art methods was 78%, the accuracy achieved by this paper was 85%.

Bai et al. [7] worked on a system for identifying birds based on an inception model with some techniques of data augmentation, working on BirdCLEF, and extracting features using a log-mel spectrogram. They chose an inception-v3 model, which allows more extracted features and has fewer parameters. They added data augmentation to prevent overfitting and improve performance and the model achieved 0.055 of classification mean average precision (c-mAP).

Küçüktopcu et al. [8] worked on a model to recognize the sounds of birds. The model is a composite of three basic hardware elements, namely a microphone, a microcontroller, and a storage unit. First, they recorded sounds from the environment. Then, they removed noise. The final proposed system conducts feature extraction and the classification of bird species. They store the processed data on an SD card.

Xie et al. [9] worked on acoustic and visual features and used a CNN for bird sound classification. Their dataset consisted of 14 bird species, and the best achievement of their

study was an F1-score of 95.95% through the fusion of acoustic features, visual features, and deep learning.

This paper presents methods for changing sound to image (spectrogram), feature extraction from bird's sound, and classification of birds by using (CNN).

III. METHODOLOGY

The methodology has been divided into different steps. Various representations of signal and different CNN architecture models have been used. Practically, this research has been implemented using the Python programming language and the Librosa library.

A. Dataset Details

Bird sounds have been obtained from different areas in Turkey. The dataset consists of 802 bird sounds, which are in mp3 files. The files contain events from different life stages of the birds, and have different durations. All file names are in Turkish, and they have stereo channels. First, we listened to all files and found that some parts of the files contained background noises, such as humans, other animal sounds, or rain. Sound durations ranged from 3 seconds to 5.3 minutes, so we extracted the good parts of the bird sounds. Finally, we obtained 10600 .wav files with a 44.1 KHz mono channel, each 5 seconds long. The samples were put into 10-folds with 437 bird species (classes). The dataset was labeled by creating a CSV file, and all the file names, numbers of folds, and Latin names of species (Scientific names) were put in the CSV file.

B. Spectrogram

A spectrogram is the representation of a signal with different frequencies over different times. Both the vertical and horizontal axis of a spectrogram represents the frequency and the time of the signals that have been converted into the spectrogram. It contains more information than other types of time-frequency. A spectrogram can be used to show bird sounds by converting the vocals to various signals and those signals to a spectrogram. After that, the spectrogram can be used to differentiate each bird species from the others [10].

A Convolutional Neural Network (CNN) works on images; therefore, the audio files first must be converted to an image format so that they can be used by the CNN [1].

In this study, we used three time-frequency transform methods (Chroma-STFT, Mel-frequency cepstral coefficients (MFCCs), and a Mel Spectrogram) to convert the sounds to spectrograms.

a) Chroma STFT (Short-Time Fourier Transform)

We use the waveform to compute Chroma STFT. Which can be used on audio samples, and the results can be displayed as a spectrum in Chroma gram. The notes of the audio are shown on the vertical axis in the Chroma-gram. Chroma preferences are good for representing music audio since the spectrums are shown in 12 bins that indicate distinct semitones [11].

b) Mel Frequency Cepstral Coefficients (MFCCs)

It can be used as a Mel-frequency spectrogram. This technique is often used in audio-related applications. A signal is a small indication, usually between 10 to 20 bins, and shows the overall view of the spectral envelope [12].

c) Mel-Spectrogram

A Mel scaled spectrogram is computed by Mel-spectrogram. This is used to represent a signal at a different frequencies over different times, and the signals are usually divided into 128 bins [11]. Fig. 1 shows the representation of the above three spectrogram types of *Dendrocopos leucotos* which is a white-backed woodpecker.

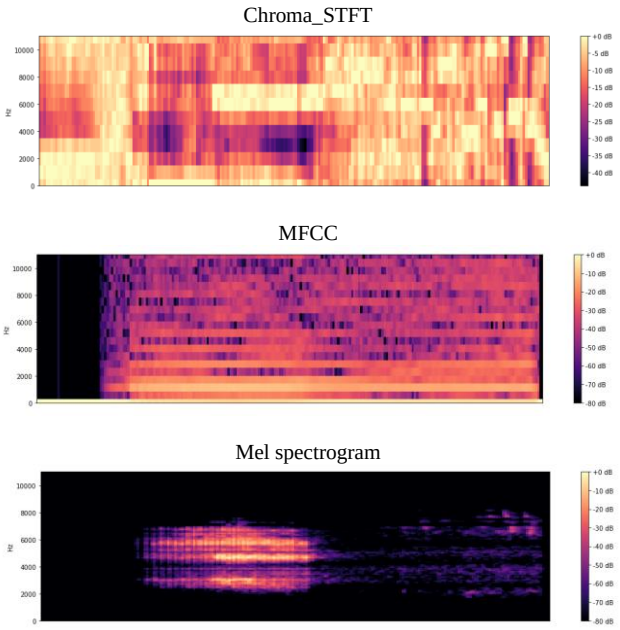


Fig 1. Spectrogram of Dendrocopos leucotos

C. Preprocessing and segmentation

The audio files are first preprocessed into a format that can be used to train the Convolutional Neural Network [1]. First, we changed the audio files from stereo to mono channel and normalized to -3 dB with a 44,1 KHz sample rate. All sounds have different durations, ranging from 3 seconds to 5.3 minutes. We split all sounds manually into 5 seconds segments.

All the sample sets are randomly divided into a training set, validation set, and testing set. The ratio of this division is 8:1:1. Using these samples, the model is well trained and validated. The flowchart of the training of identification model is illustrated in Fig.2.

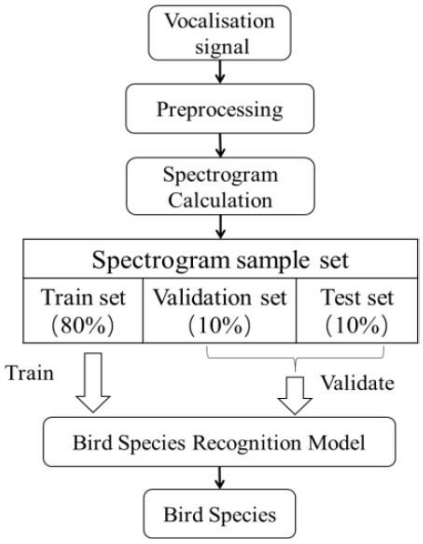


Fig. 2. Flowchart of the training of identification models.

D. System Modeling

This study uses a Convolution Neural Network, because CNNs have had considerable success in most classification tasks. The CNN is inspired by the connectivity patterns of neurons. It consists of several layers: first the input layer, then several convolutional layers and a pooling layer, then one or more fully connected layers, and finally the output layer [13].

The main portion of a CNN is the convolution operation. This is the core building block of a CNN. The main purpose of these operations is to extract features. The required preprocessing in CNN is much fewer than other traditional methods. The main advantage of CNN is that there is no need to extract the matrix and design the formula to extract the features manually. we used 2D convolution layers consisting of different sizes of filters, and the first layer in the CNN performed convolution with a spectrogram of 128 features. As explained the operation of the Convolution Network is shown in Fig. 3 [14].

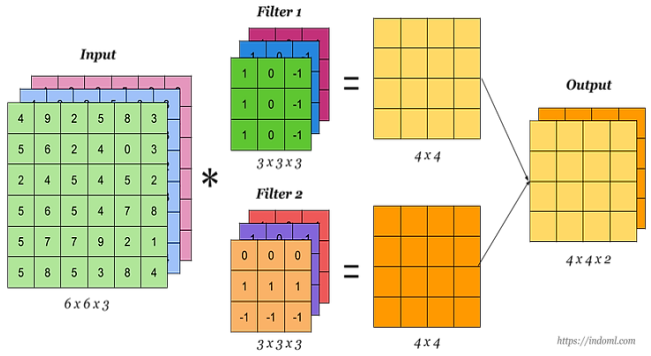


Fig. 3. The operation of the Convolution Network.

After the convolutional layer, the pooling layer is applied. This is used to decrease the number of parameters; images are shrunk down while preserving the most important information in them. We used a max pooling (2,2) filter with a stride of 2, as shown in Fig.4.

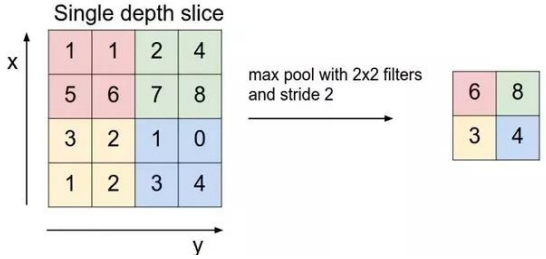
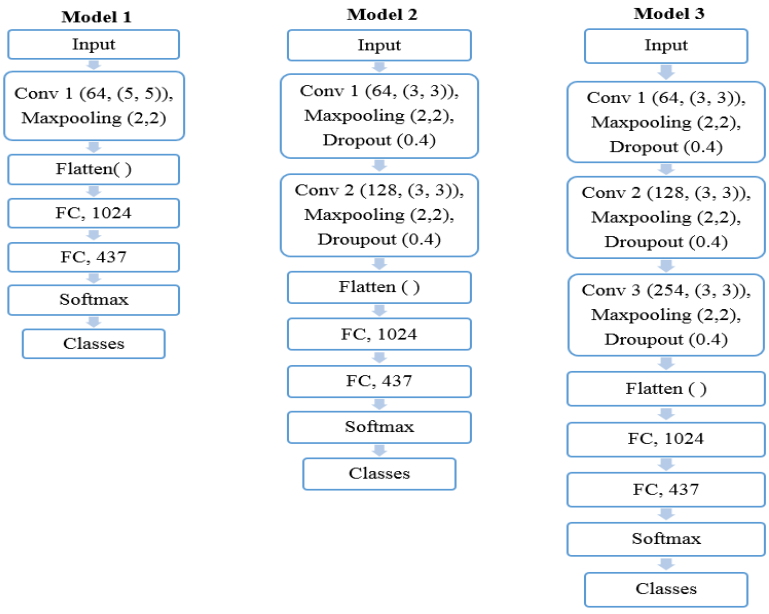


Fig. 4. The representation of the max pooling layer

The fully connected layers come after convolution and pooling layers. After training methods to change sounds into the spectrogram, we need to create CNN architectures consisting of different filters and layers so as to have better results for our system and to increase the accuracy, lower the loss, and reduce overfitting for the process. The parameters of the CNN models used in the study are given in Table 1.

Table 1. Parameters of CNN models used in the paper.



An attempt has been made to optimize our training processing time and maintain good overall performance. We evaluated different kinds of parameter settings and found the following to be very effective [15].

The dropout has been used to prevent or reduce overfitting. We used a dropout with 0.4 unit. In this work, the (Adam) optimizer algorithm which is responsible for reducing the losses and for providing the most accurate results possible has been used. Loss function was used to make our prediction to able to predict expected outcomes. We used categorical cross-entropy as a loss function because it is the best one to work with multi-class classification labels. To decrease the complexity of the models, we used L2 regularization since it is better than L1 for complex data.

E. Implementation

The models were implemented by using the Python programming language and the Librosa library. The steps of the implementation are as follows:

1) Create Sample sets

Three representation types of spectrograms with 150 epochs have been used. These are:

a) Chroma-stft

Chroma-STFT is the first method we use to transform bird sounds to spectrograms, with 12 dimensions, and the total number of parameters is 1.336.499. The results show an accuracy of 0.4 with a loss of 3.1351.

b) Mel-frequency cepstral coefficients

By using Mel-frequency cepstral coefficients (MFCCs) on our dataset to create a spectrogram with 20 dimensions, we achieve an accuracy of 0.83, and loss of 0.88316, where the number of total parameters is 1.076.403.

c) Mel spectrogram

Thirdly, we used a Mel spectrogram to form an image from sound, with 128 dimensions. We achieved an accuracy of 0.89 and a loss of 1.061098, where the total number of parameters is 1.336.499.

2) CNN models

Testing the models showed that each had different accuracy and loss values. This helps to decide between them and choose a good CNN architecture model. The results of the models used are as follows:

Net 1: CONV-POOL-FC-FC

In the first architecture, we have 4 layers which are Convolutional, Pooling and two fully connected layers. The 2D convolution layer consist of (5x5) filter with a channel size is 64, the Max Pooling layer is (2x2). The fully connected layers of size 1024, 437. The (MSE) loss function is used. We used ReLU in all activations, but the last used activation was Softmax. After 150 epochs, the model achieved a validation accuracy of 0.73 and a validation loss of 0.008, where the number of parameters is 2.547.765

Net 2: CONV-POOL-CONV-POOL-FC-FC

In this architecture, we have 6 layers, using a 2D convolution of a (3x3) filter with a channel size of 64, 128, the max pooling layer is (2x2), and the (MSE) loss function is used. We used ReLU in all activations, but the last used activation was Softmax, we have two fully connected layers of (1024,437) respectively. We use the dropout of (0.4). After 150 epochs. The model achieved a validation accuracy of 0.59 and a validation loss of 0.001.

Net 3: CONV-POOL-CONV-POOL-CONV-POOL-FC-FC

This model consists of a 2D convolution of a 3x3 filter with channel sizes of 64, 128, and 254 respectively. A (2x2) max pooling layer and dropout of 0.4 were used. There are two fully connected layers, of size 1024 and 437, respectively. Hyperbolic Tangent (Tanh) was used on three layers, ReLU on one layer, and Softmax for the final activation functions respectively, we used the Adam optimizer and categorical-crossentropy loss function, we used L2 regularization (0.0005). After 150 epochs, the model achieved a validation accuracy of 0.89 and a validation loss of 1.0611.

Net 4: CONV-POOL-CONV-POOL-CONV-POOL-FC-FC

In the last architecture, we have 8 layers, we used the techniques as in Model-3 and the same 2D convolution of 3x3 filter with channel size of 64, 128, 254 respectively. But in this architecture, we increase the number of iterations from 150 epochs to 300. The model achieved validation accuracy of 0.91 and a validation loss of 1,0207 as shown in Fig 5.

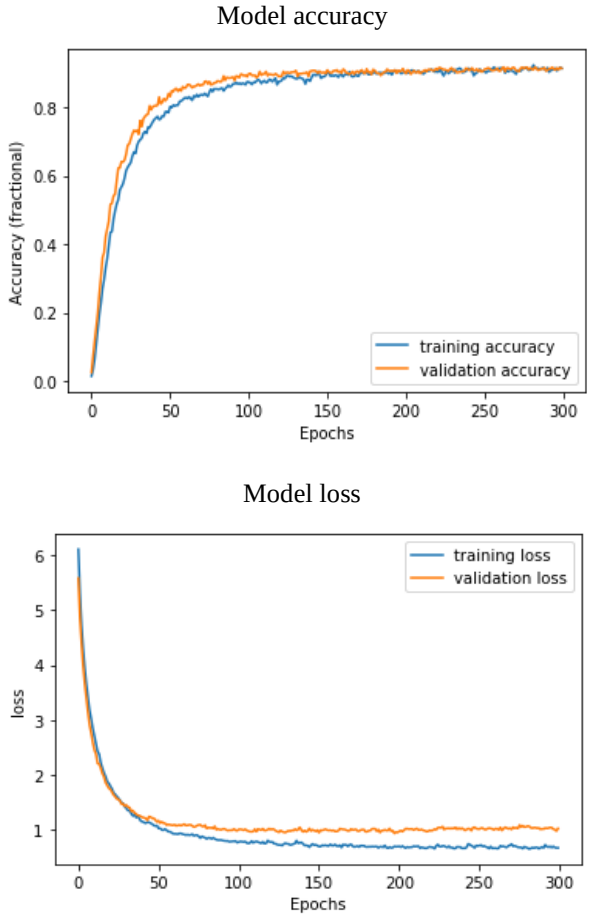


Fig. 5. Accuracy of the fourth model

IV. RESULT

The performances, after training, of Chroma-STFT, Mel-frequency cepstral coefficients (MFCCs), and Mel-Spectrogram with 150 epochs, as well as summaries of accuracy, and loss, are shown in Table 2.

Table 2. Summary of above results with accuracy and loss

representation of spectrum	accuracy	loss
Chroma-STFT	40%	3.135106
MFCC	83%	0.883159
Mel_spectrogram	89%	1.061098

The results in the Table-2 showed the Mel-spectrogram was used to transform sound to spectrogram, a good result was achieved, and the third approach is the best since it has the highest performance. Therefore, therefore, this will be chosen for our system. From the results of other researches, it is clear that a CNN, as compared to an LSTM, SVM, or RNN, has high success in some classification task, because it works well on images and speech, and because it is more powerful and includes more features.

The research papers [16, 17] took on the (DCASE 2017) challenge of classifying audio using RNN and CNN algorithm and achieved the accuracy shown in Table 3.

Table 3. Accuracy of DCASE 2017 by using RNN and CNN

CNN model	Accuracy
RNN	74.8%
CNN	83.65%

The results of the model presented in our paper can be compared to understand and choose the best model. It is clear that the CNN has been designed with hyper-parameters, so using classification methods with a CNN increases the accuracy of the models, as suggested above. It should also be noticed that the highest training efficiency among all the models was achieved by the CNN model. Some models have been trained on the papers dataset as explained in Table 4.

Table 4. Summary of our models' result

CNN models	Accuracy	Loss
Model 1	73%	0.000805
Model 2	59%	0.001164
Model 3	89%	1.061098
Model 4	91%	1.020716

After testing the CNN models on the dataset, it is obvious the fourth model is the best, since there is an increase in the number of epochs to 300, and the accuracy of the model is as high as 91%.

V. CONCLUSION

This paper focuses on identifying bird species based on spectrogram images using Convolutional Neural Networks (CNN). The labeled dataset has been created which consists of 10600 .wav files, each 5 seconds long, covering the sounds of 437 different bird species. Different representations of the spectrogram have been used, at the result with Mel-spectrogram to transform sound to spectrogram the best result on out dataset was achieved, and some Convolution Neural Network architecture models have been exposed, the fourth architecture model achieved high accuracy, with low amount of overfitting and loss. The CNN network won, with accuracy of 91%, 0.94 precision, 0.91 recall, and 0.91 F1-score.

REFERENCE

[1] Martinsson, J., "Bird Species Identification Using Convolutional Neural Networks" (Master's thesis), Gothenburg, 2017.

[2] Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., and Zhang, Z., "The Application of Two-Level Attention Models in Deep Convolutional Neural Network for Fine-Grained Image Classification". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA, pp. 842-850, 2015.

[3] Dennis, J., Tran, H. D., and Li, H., "Spectrogram image feature for sound event classification in mismatched conditions". *IEEE signal processing letters*, 18(2), 130-133, 2010.

[4] Zhao, B., Wu, X., Feng, J., Peng, Q., and Yan, S., "Diversified Visual Attention Networks for Fine-Grained Object Classification". *IEEE Transactions on Multimedia*, *19*, pp. 1245-1256, 2017.

[5] Zhang, Y., Wei, X. S., Wu, J., Cai, J., Lu, J., Nguyen, V. A., and Do, M. N., "Weakly Supervised Fine-Grained Categorization with Part-Based Image Representation". *IEEE Transactions on Image Processing*, *25*(4), 1713-1725, 2016.

[6] Jaiswal, K., and Patel, D. K, "Sound Classification Using Convolutional Neural Networks". In 2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 81-84, (2018).

[7] Bai, J., Wang, B., Chen, C., Chen, J., and Fu, Z., "Inception-V3 Based Method of Lifeclef 2019 Bird Recognition". In *CLEF (Working Notes)*, 2019.

[8] Küçüktopcu, O., Masazade, E., Ünsalan, C., and Varshney, P. K.," A Real-Time Bird Sound Recognition System Using a Low-Cost Microcontroller". *Applied Acoustics*, *148*, 194-201, 2019.

[9] Xie, J., and Zhu, M., "Handcrafted features and late fusion with deep learning for bird sound classification". *Ecological Informatics*, 52, 74-81, (2019).

[10] Xie, J. J., Ding, C. Q., Li, W. B., and Cai, C. H., "Audio-Only Bird Species Automated Identification Method with Limited Training Data Based On Multi-Channel Deep Convolutional Neural Networks". *arXiv preprint arXiv:1803.01107*, 2018.

[11] Alkhawaldeh, R. S., "DGR: Gender Recognition of Human Speech Using One-Dimensional Conventional Neural Network". *Scientific Programming*, 2019.

[12] Rajesh, S., and Nalini, N. J., "Musical Instrument Emotion Recognition Using Deep Recurrent Neural Network". *Procedia Computer Science*, *167*, 16-25, 2020.

[13] Zhou, H., Song, Y., and Shu, H., "Using Deep Convolutional Meural Network to Classify Urban Sounds". *In TENCON 2017-2017 IEEE Region 10 Conference*, (pp. 3089-3092), 2017.

[14] Valenti, M., Squartini, S., Diment, A., Parascandolo, G., and Virtanen, T., "A Convolutional Neural Network Approach for Acoustic Scene Classification". *In 2017 International Joint Conference on Neural Networks (IJCNN)*(pp. 1547-1554), (2017, May).

[15] Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, D., Ritter, M., and Eibl, M., "Large-Scale Bird Sound Classification using Convolutional Neural Networks". In *CLEF, 2017*.

[16] Hussain, K., Hussain, M., and Khan, M. G., "An Improved Acoustic Scene Classification Method Using Convolutional Neural Networks (CNNs)". *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *44*(1), 68-76, 2018.

[17] Dang, A., Vu, T. H., and Wang, J. C., "Deep Learning for DCASE2017 Challenge". In *Workshop on DCASE2017 Challenge, Tech. Rep*, Munich, Germany, *2017*.

Analysis and Investigation of Malicious DNS Queries Using CIRA-CIC-DoHBrw-2020 Dataset

Muosa Tayseer Jafar
mou20178003@std.psut.edu.jo

Mohammad Al-Fawa'reh
moh20178019@std.psut.edu.jo

Zaid Al-Hrahsheh
zaid_hrh@aabu.edu.jo

Shifa Tayseer Jafar
shifaa.tayseer@gamil.com

Princess Sumaya University for Technology
Amman, Jordan

***ABSTRACT*- Domain Name System (DNS) is one of the earliest vulnerable network protocols with various security gaps that have been exploited repeatedly over the last decades. DNS abuse is one of the most challenging threats for cybersecurity specialists. However, providing secure DNS is still a big challenging mission as attackers use complicated methodologies to inject malicious code in DNS inquiries. Many researchers have explored different machine learning (ML) techniques to encounter this challenge. However, there are still several challenges and barriers to utilizing ML. This paper introduces a systematic approach for identifying malicious and encrypted DNS queries by examining the network traffic and deriving statistical characteristics. Afterward, implementing several ML methods: (RF: Random Forest, DT: Decision Tree Classifier, GNB Gaussian Naive Bayes, KNN: k-nearest neighbor, Logistic regression, SVC: Support Vector Classifier, QDA: Quadratic Discriminant Analysis, SGD)". These models were employed to evaluate their ability to detect malicious DNS traffic using the CIRA-CIC-DoHBrw2020 data set. The Experiments revealed a good accuracy score where DT and RF models have achieved the highest accuracy, 99.99 % relative to other detection methods.**

Keywords—DNS Traffic, Malicious DNS, Zero Day attack, CIRA-CIC-DoHBrw2020, DNS tunneling.

I. INTRODUCTION

DNS is an important protocol that has a substantial role in relation to web activities such as browsing and e-mail. It represents the phonebook of the Internet. Humans access information online through domain names, like google.com or yahoo.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources. DNS allows applications to use site names like Google.com instead of the IP addresses that cannot be memorized [1].

Since DNS is not used for Data transfer, numerous organizations get less consideration and have no monitoring plans in terms of security checking compared to other protocols like Web activity where attacks often take place. Since DNS is used by everyone, everywhere and all traffic flows through it. Point your traffic to the right destination. Because of its rule and its sensitivity in your real environment, it is exposed to many threats from attackers that target to control the DNS and grant them the possibility to abuse the DNS in order to extract and infect all data from it.

Since many organizations utilize one or two DNS servers, they may wake up to the reality that they are unable to protect their DNS against massive attacks with a large amount of traffic to their website that leads to servers crashing, preventing their users from accessing the website. This is because a large amount of traffic could be vulnerable to DNS security breaches. A malicious attack can also aim to exploit security vulnerabilities on the server that runs the DNS services and extract valuable data such as passwords, usernames, and other personal information.

Previous efforts in securing the DNS have focused on protecting the validity of information coming from the DNS. Since much of the Internet's traffic is encrypted and served by large content delivery networks, in many cases, domain name systems are the only clear text sign about the specific service being accessed.

DNS Tunneling is a method of cyber attack that encodes the data of other programs or protocols in DNS queries and responses. DNS tunneling often includes data payloads that can be added to an attacked DNS server and used to control a remote server and applications.

Cyber criminals use several tunneling techniques to hide their identity. The most used techniques are FTP-DNS tunneling, HTTP-DNS, tunneling, HTTPS-DNS tunneling, and POP3-DNS tunneling.

Many traditional methods proposed to detect a malicious domain include the usual domain blacklist Names [2], Network Traffic Analysis [3] , Detailing of Web Page Content [4],DNS Traffic Analysis [5], and Analysis of salient lexical features [6]. Most of the work on malicious URLs is content-based or non-content-based; the disclosure does not take into account the domain name and DNS data for the malicious account file URL, so the results obtained lack accuracy. Hence an effective mechanism for detecting the harmful field will also help in improving the accuracy of malicious URL detection.

Nowadays, several facilities are available for tunneling over DNS, and most of these tools point to free Wi-Fi access to sites that require restricted access via HTTP [7]. However, serious threats may occur with access to free Wi-Fi. These threats can be represented as malicious activities that can be assimilated through the DNS tunnel. With a DNS tunnel, complete remote control can be performed over a channel of a compromised internet host. Furthermore, various activities can be done via the DNS tunnel, such as file transfers system commands, or even IP tunnel. Feederbot [8] and Moto [9] are examples of known DNS tunneling tools using DNS as a way to communicate .

All recently identified threats have stimulated the information security community to provide robust DNS tunnel detection methods [10]. Different types of DNS tunnel detection techniques have been proposed. These methods can be classified into two broad categories; Traffic and payload analysis. The first category aims at analyzing the overall traffic as certain important features such as DNS traffic volume, number of hostnames per domain, site, and domain record can be identified. The second category analyzes the payload of a single query to identify many features such as content, number of bytes, and domain length.

The analysis of DNS tunnel features has led the researchers to use rule-based criteria where both traffic and payload are analyzed based on certain features. Once a pre-set condition occurs, DNS tunnel determination will be triggered. However, with the complex and time-consuming task of manual rules regulation, researchers tend to use machine learning techniques (MLT).

The main feature behind machine learning lies in the statistical model that has the ability to automatically define important rules [7-9]. In addition, with the advent of annotated datasets such as JSON [11] containing network connections with predefined labels (such as Tunneled or Legitimate). The focus on machine learning has expanded because MLT requires annotated historical data. Hence, MLT is capable of training the model relying on this data. According to this training, new data will be available for testing.

In fact, there are many types of MLTs, like SVM, NB, DT, KNN, and others. With this diversity, it is difficult to determine which classifier would be more suitable, which would suit DNS tunnel discovery. This paper aims to [provide a comparative analysis of the DNS tunneling process using 9 MLTs classifiers, including NB, DT, and SVM.

This paper presents a lightweight approach leveraging ML models to detect malicious activities designed specifically to be deployed in the internal network of an enterprise. To detect malicious domains using a model trained by a machine learning algorithm using a combination of features of a domain name such as DNS data, lexical characteristics, and website reputation. We create separate data sets for benign and malicious domain names from various well-known and reliable sources and extract the above-mentioned features from those domain names and feed them to logistic regression machine learning algorithms and generate a model. We present an approach that demonstrates the simplicity, robustness, and scalability of our approach via empirical experiments on real-world data. The produced model is then experimented with a new list of domain names to classify them as benign or malicious.

The structure of this paper is as follows: section 2 includes a brief description of DNS, and the previous related works are in Section 3. Section 4discusses the methodology, the results are addressed in Section 5, and the conclusion of the work is in Section 6.

II. RELATED WORK

There has not been much research focusing on Malicious and encrypted DNS traffic. Current approaches are complex solutions and have inconsistencies during the processing phase [12]. This paper adopts a systematic approach for anomaly DNS queries that results in significant detection and less overhead in traffic processing.

Preston [13] focused on the primary domain as a filter to classify the DNS traffic rather than the queries. The features have been extracted from subdomains from multiple groups. The author used supervised machine learning for examining DNS traffic and filter benign and malicious domains. However, this approach has a limitation of the inability to detect malicious queries in the main domain. In which the sub-domain is not enough for detecting the other types of attacks.

A. Das et al. [12] presented a novel approach by focusing on semi-supervised learning for detecting DNS tunnels. Their technique learns the characteristics of normal DNS traffic and calculates the MSE between different sample classes to detect DNS tunnels. The authors focused on text query with a limited number of features to classify DNS queries using ML techniques such as k-means clustering to classify DNS concentrated on the all-TXT queries, and just used ten features whereas ML needs a lot of features for more learning to get high accuracy, however, detecting all TXT queries is time-consuming.

While Palaniappan et al. [14] used a logistic regression algorithm with lexical feature-based analysis to classify DNS queries to benign and malicious DNS domains, the authors only used four features and used the active DNS analysis as a filter. The proposed model achieved 60% accuracy. The main limitation of this model is focusing on a small dataset.

K. Shima et al. [15] collected network traffic on a 1-day basis and only focused on the reflector traffic. After that, he extracted the features on the network level. they used SVM and to classify benign and malicious DNS servers. This mechanism suffers from an inability to handle encrypted communication.

C. Liu et al. [16] focused on DNS tunneling and deep learning to detect malicious queries based on Byte-level. The model can extract all information in the entire DNS queries. However, they only focused on the sequential and structural data that was found in the initial request.

Banadaki et al. [17] examined a new dataset called CIRA-CIC-DoHBrw-2020 using several ML algorithms such as (XG Boost, Gradient Boosting, and Light Gradient Boosting Machine). In addition, they investigate the important features. However, the preprocessing and optimization phase were unclear.

III. METHODOLOGY

The proposed method consists of four main phases; Data Collection, Feature Extraction, preprocessing, and model deployment. The Data Collection phase's purpose is to identify a benchmark dataset of DNS tunneling in order to facilitate the comparison among the classifiers. The Feature Extraction phase goal to exploit some features of payload and traffic analysis. The preprocessing phase purpose of validating the data as suitable input of the ML algorithm Figure 1 show the proposed Methodology. The Model deployment consists of training and testing phases by carrying out several ML algorithms such as SVM, NB, and DT.

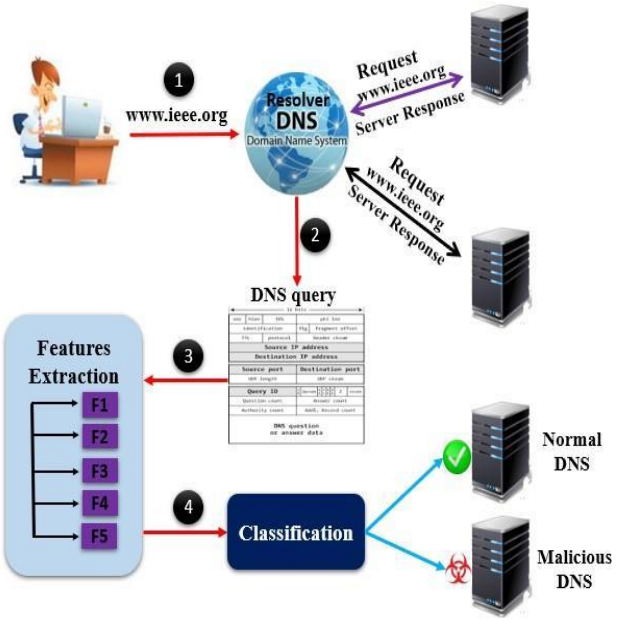


Fig. 1 The proposed Methodology

Step 1: Data collection

Finding an ideal dataset is a great challenge because it is considered private so it can be shared for privacy issues, does not reflect the current surface attack of cyber-attacks, and comes from different servers and operating systems which make it need normalization. In this paper, a realistic dataset was adopted from [11]. It is new, encrypted data and follows a systematic approach in the generation phase. In addition, it contains malicious activities and has been compiled at the packet level, which helps in deep examination processes. The dataset includes raw data of malicious DNS traffic alongside normal DNS traffic.

The data collection process performed in four scenarios. The first scenario generating Non-DoH activity by accessing different web servers. The traffic has been collected using wireshark and TCPdump to store it at the packet level, then convert the output to flow level in order to reduce the cost of processing resources. In the second scenario, Several DNS tunneling tools have been used such as DNSCat2, Iodine, and dns2tcp to generate Malicious-DoH traffic. These tools send TLS-encrypted HTTPS data in DNS queries to DoH servers (Adguard, Cloudflare, Google, Quad9). In the third scenario (Benign-DoH), Several web browsers have been used to generate Benign-DoH in the same mechanism as in scenario Non-DoH. In the fourth scenario, several browsers and DNS tunneling tools have been used to access the top 10k Alexa websites Figure 2 show the traffic distribution.

The public dataset available on the internet is usually unclean, incompatible, and sometimes suffers from several issues. Data preprocessing plays an important role in converting the unclean data into a clean and consistent format.

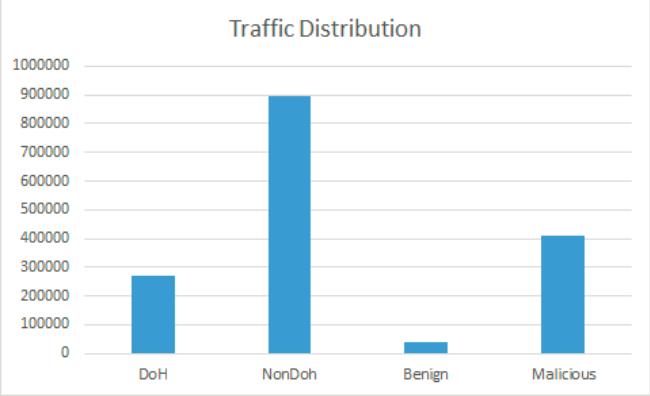


Fig. 2 Traffic Distribution

Step 2: Data cleaning

This phase includes 3 main steps: removing the duplicate flows, handing the missing and outfitter values using the median, then encoding the categorical features (Source/Destination Port number/IP address) using one-hot encoding method.

Step 3 Feature extraction

Two classes of features have been extracted from the adopted dataset using DoHMeter and CIC flow meter; statistical features such as mean, median average. and network features such as source IP, port number, flags. All extracted features are listed in table (3).

Table 1: List of extracted statistical traffic features

Parameter	Feature
F1-F4	Rate /Number of flow bytes sent or received
F5-12	Variance / Skew from median / Skew from mode /Coefficient of Variation/ Standard Deviation /Mean / Median of Packet Length
F13-F20	Standard Deviation/ Coefficient of Variation / Skew from median / Skew from mode /Variance /Mean/Median/ Mode of Packet Time
F21-F28	Median/ Standard Deviation / Coefficient of Variation / Skew from median/ Skew from mode /Variance /Mean/ Mode Request/response time difference
F29-F33	Source/Destination IP / Source/Destination Port number/ Timestamp
F34	Flow Duration

Step4: Data scaling

This phase aims to normalize all features on the same scale to prevent biasing the ML models. Many types of scaling approach have been used in the literature. While data standardization method used in this paper.

Step 5: Data over Sampling

The adapted dataset is imbalanced as shown in the figure 2 which affects the reliability of the ML models. so, we used the SMOT algorithm for dataset oversampling.

Step 6: Data splitting

The dataset has been split into training, testing, and validation. 80%, 10% and 10 % respectively. and use cross-validation with 10 folds.

Step 7: Model deployment

This paper adapted eight ML methods as follow:

Random FOREST (RF): This algorithm is used in classification and regression problems. It is a type of a supervised classification algorithm and represents a collection of randomly selected DTs. The RF called ensemble learning and used to combine multiple classifiers in order to solve a complex problem and improve the performance of the classification model.RF collects the prediction accuracy of each tree and predicts the final output based on the majority votes of predictions. RF is considered a powerful algorithm since it has a lot of features that improve the results in the random forest such as runs efficiently on large databases, handles thousands of input variables without variable deletion, offers an experimental method for detecting variable interactions.

Decision Tree (DT): is a supervised learning technique and used in different fields such as statistics, data mining, and ML. it predicts responses values using learning decision rules derived from features. it can be used for decision making in both regression and classification tasks.DT consists of several components; the root node and branch node. or sub-tree, splitting, decision node, leaf or terminal node, pruning. DT is a powerful algorithm it is easy to understand, requires little data preparation, also able to handle numerical and categorical data, and deal with multiple output problems.

Logistic Regression (LR) is another technique of ML for regression algorithms. LR finds the relationships and dependencies between variables and transforms the output using the logistic sigmoid function to return probabilistic values that can be mapped to binary classes.

Quadratic Discriminant Analysis (QDA) is widely used in classification algorithms and statistics problems. It has a closed-form solution that can be easily computed with inherent multiclass, also has proven to work well in practice with no hyperparameters to tune. QDA is an extension of Linear Discriminant Analysis (LDA) which has a common variance for each class while in QDA, each class has its own variance or covariance matrix.

Support Vector Machines (SVM) is a supervised ML technique used for solving classification and regression problems. SVM generates a hyperplane in multidimensional space in an iterative manner to minimize the classification error rate. Moreover, SVM divides the datasets into classes to find a maximum marginal hyperplane. Accordingly, it achieves a high accuracy compared to other classifier models.

Stochastic Gradient Descent (SGD): this classifier basically is a simple and efficient optimization algorithm in ML and DL used to find the values of functions parameters that minimize the classification cost. Typically, there are three types of Gradient Descent; Batch Gradient Descent, Stochastic Gradient Descent, and Mini-batch Gradient Descent

Naive Bayes: A very simple and robust model of supervised ML that focuses on the application of Bayes' theorem with independent (naive) assumptions of conditionality between traits. Every feature is categorized independently of each other; hence it will speed up the prediction of the category of unknown data. Naive Bayes requires highly scalable features in a learning problem.

k-nearest neighbors: (KNN) algorithm is simple, easy-to-implement and not only used to solve classification problems but also regression problems. However, the flawed work for the KNN starts with selecting the number K of the neighbors then calculating the Euclidean distance of K number of neighbors and subsequently taking the K nearest neighbors as per the calculated Euclidean distance, and finally counting the number of the data points in each category and based on its distance between the input and the center of every class; then classifying the input into correct class [18].

IV. EXPERIMENTS AND MODELS EVALUATION:

The ML-based model is trained to detect the attack on network traffic as presented in Figure 1. The ML algorithm should be generic to detect the unseen instance correctly. Accuracy and ROC are generated using the number of correct predictions on the test dataset to find the actual class label against the predicted class label for each category and then extract the classification metrics. The accuracy represents the total correct prediction overall the total prediction, as shown in equation 1.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

The ROC curve shows the relationship between the experimental sensitivity and the specificity for every possible cut-off. The ROC curve is a graph with the x-axis represents as shown in equations 2and 3:

$$1 - \text{specificity} (= \text{false-positive fraction} = FP/(FP+TN)) \quad (2)$$

$$\text{While} \quad \text{y-axis represents } TP/(TP+FN)) \quad (3)$$

The DT achieved the best results among all eight algorithms investigated for malicious and standard DNS queries with accuracy of 99.996% as shown in Figure 3 and Table 2. However, RF and KNN does not deviate substantially from the highest accuracy. while LR, QDA, GNB and SGD have an average accuracy less than eight percentage points among the overall tested data sets. SVM is the worst algorithm in the training and testing time and SGD is the fastest algorithm in the testing phase.

The second part of the experiments evaluates these MLT's to discover their ability to identify the DNS traffic either encrypted or not. Figure 4 and Table 3 show the RF classifier has the best results among all other algorithms with 99.9802 % accuracy. The DT achieves the second-best accuracy with 999715%, while GNB and QDA are the worst algorithm with 91.5967% and 87.4713%, respectively. The conducted Experiments show that the SVM is the slowest algorithm in the training and testing phase, while GNB is the fastest algorithm in the training and the SGD is the fastest in the testing phase. To Summarize, the best-achieved results show a classification accuracy of 99.99% for RF, DT, KNN in M and B classification.

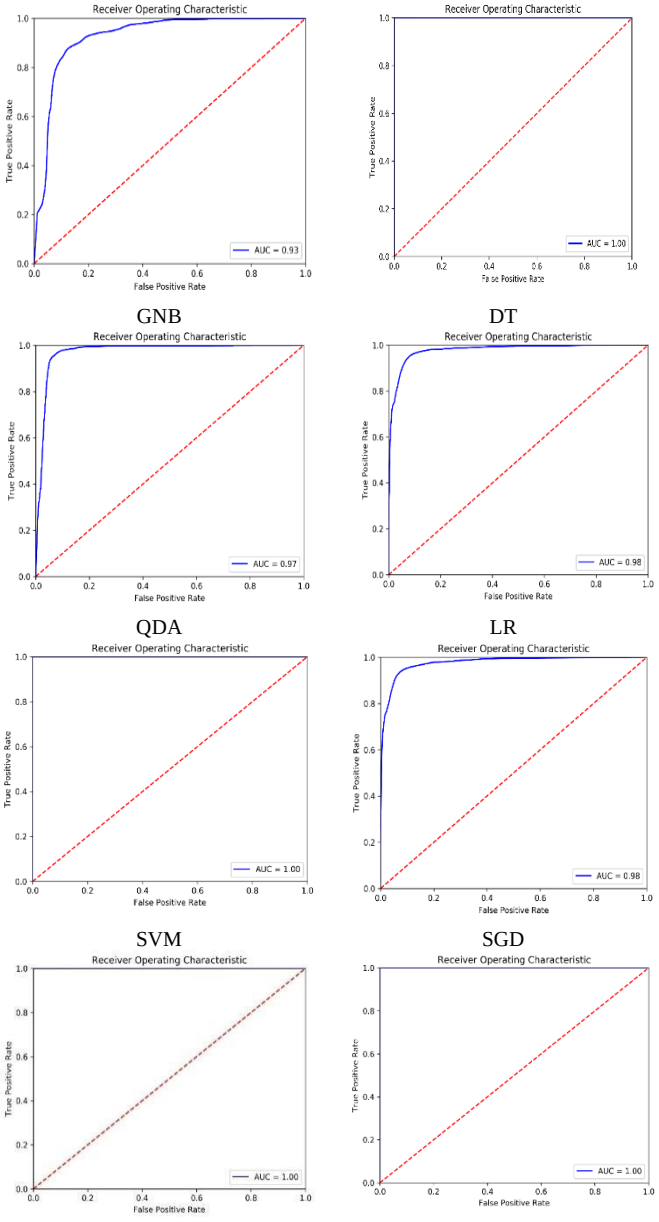


Fig. 3 ROC Curve comparing MLT based on network and statistical feature for malicious and benign traffic

Table 2: Malicious and Benign calcification Metrics

Model	Training Time	Testing Time	Accuracy
RF	118.7859	0.586187	0.999802
DT	72.24614	0.098546	0.999715
QDA	5.168157	0.573868	0.915967
GNB	1.486871	0.513123	0.874713
SGD	4.446488	0.030216	0.934754
KNN	344.022	1536.963	0.99482
LR	28.24521	0.030461	0.936506
SVM	32743.5	40657.9	0.99917

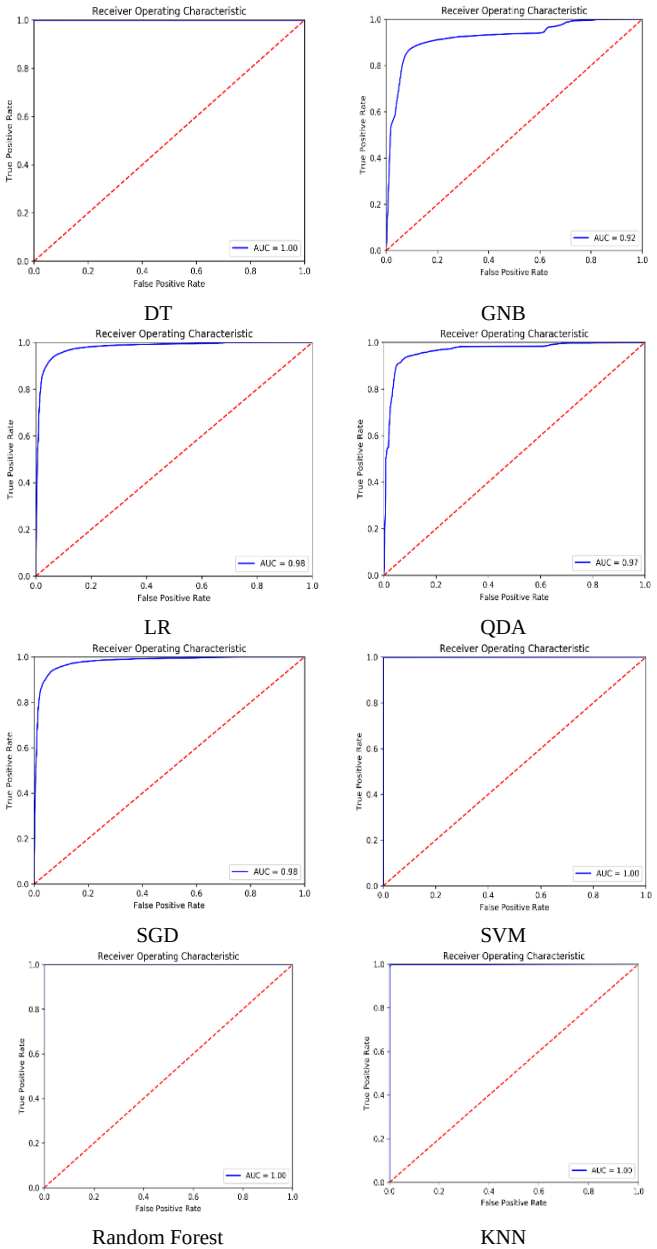


Fig. 4 ROC Curve comparing MLT based on network and statistical feature for DoH and non-DoH traffic.

Table 3: Doh and NonDoh Classification metrics

Model	Training Time	Testing Time	Accuracy
RF	31.69924	0.21619	0.999954
DT	11.90182	0.04168	0.99996
QDA	2.16018	0.26679	0.932362
GNB	0.70696	0.22751	0.799326
SGD	2.03135	0.01798	0.930879
KNN	95.52005	279.7582	0.999631
LR	11.40786	0.053378	0.935558
SVM	8185.87503	10164.49	0.99917

V. CONCLUSION

This paper has introduced a systematic approach for identifying malicious and encrypted DNS queries by examining network traffic using statistical characteristics. Several experiments have been conducted with eight ML algorithms to evaluate the efficiency and the performance of these classifiers using statistical and network features: All tests are experimented based on the CIRA-CIC-DoHBrw-2020 dataset. The attack rate of the dataset is approximately 25%. Other types of the collected traffic are distributed as 56% non-DoH traffic, 17% of Doh packets, and 2% of regular traffic. Experiments have shown that a single ML algorithm cannot effectively manage the time and the accuracy of detecting Malicious DNS queries. The accuracy of RF, SVM, DT, KNN is almost 99.9%. However, SVM and KNN are the slowest algorithms in the training phase. In contrast, GNB is the fastest algorithm for identifying traffic type but has the worst results in detection phase.

REFERENCES

[1] N. U. Aijaz, M. Misbahuddin, and S. Raziuddin, "Survey on DNS-Specific Security Issues and Solution Approaches," in Data Science and Security, Springer, 2020, pp. 79–89.

[2] M. Sammour, B. Hussin, and M. F. I. Othman, "Comparative Analysis for Detecting DNS Tunneling Using Machine Learning Techniques," Int. J. Appl. Eng. Res., vol. 12, no. 22, pp. 12762–12766, 2017.

[3] M. Aiello, M. Mongelli, and G. Papaleo, "Basic classifiers for DNS tunneling detection," in 2013 IEEE Symposium on Computers and Communications (ISCC), 2013, pp. 880–885.

[4] L. A. Trejo, V. Ferman, M. A. Medina-Pérez, F. M. A. Giacinti, R. Monroy, and J. E. Ramirez-Marquez, "DNS-ADVP: A Machine Learning Anomaly Detection and Visual Platform to Protect Top-Level Domain Name Servers Against DDoS Attacks," IEEE Access, vol. 7, pp. 116358–116369, 2019.

[5] H. Zhao, Z. Chang, G. Bao, and X. Zeng, "Malicious domain names detection algorithm based on N-gram," J. Comput. Networks Commun., vol. 2019, 2019.

[6] F. Allard, R. Dubois, P. Gompel, and M. Morel, "Tunneling activities detection using machine learning techniques," J. Telecommun. Inf. Technol., pp. 37–42, 2011.

[7] A. Nadler, A. Aminov, and A. Shabtai, "Detection of malicious and low throughput data exfiltration over the DNS protocol," Comput. Secur., vol. 80, pp. 36–53, 2019.

[8] H. Ichise, Y. Jin, K. Iida, and Y. Takai, "Detection and Blocking of Anomaly DNS Traffic by Analyzing Achieved NS Record History," in 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, pp. 1586–1590.

[9] A. Almusawi and H. Amintoosi, "DNS Tunneling detection method based on multilabel support vector machine," Secur. Commun. Networks, vol. 2018, 2018.

[10] G. Farnham and A. Atlasis, "Detecting DNS tunneling," SANS Inst. InfoSec Read. Room, vol. 9, pp. 1–32, 2013.

[11] X. Zang, A. Rastogi, S. Sunkara, R. Gupta, J. Zhang, and J. Chen, "Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines," arXiv Prepr. arXiv2007.12720, 2020.

[12] A. Das, M.-Y. Shen, M. Shashanka, and J. Wang, "Detection of Exfiltration and Tunneling over DNS," in 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, pp. 737–742.

[13] R. Preston, "DNS Tunneling Detection with Supervised Learning," in 2019 IEEE International Symposium on Technologies for Homeland Security (HST), 2019, pp. 1–6.

[14] G. Palaniappan, S. Sangeetha, B. Rajendran, S. Goyal, and B. S. Bindhumadhava, "Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features," Procedia Comput. Sci., vol. 171, pp. 654–661, 2020.

[15] K. Shima, R. Nakamura, K. Okada, T. Ishihara, D. Miyamoto, and Y. Sekiya, "Classifying DNS Servers Based on Response Message Matrix Using Machine Learning," in 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 1550–1551.

[16] C. Liu, L. Dai, W. Cui, and T. Lin, "A Byte-level CNN Method to Detect DNS Tunnels," 2019, pp. 1–8.

[17] Y. M. Banadaki, "Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers," J. Comput. Sci. Appl., vol. 8, no. 2, pp. 46–55, 2020.

[18] M. Al-Fawa'reh and M. Al-Fayoumiy, "Detecting Stealth-based Attacks in Large Campus Networks," Int. J., vol. 9, no. 4, 2020.

Deep-Immune-Network Model for Vulnerable Clone Detection

Canan BATUR ŞAHİN
Faculty of Engineering and Natural Sciences
Malatya Turgut Ozal University
Malatya, Turkey
canan.batur@ozal.edu.tr

***Abstract*— Code-clone is primarily utilized for the implementation of the software reusability concept. The behavior of code reuse generally leads to the propagation of vulnerabilities in the case of the reproduction of a piece of a vulnerable code. Such a vulnerability is called a code-clone vulnerability. This paper is the first work that proposes a novel methodology, which is capable of discovering vulnerable code-clones in open-source software programs efficiently and accurately. We use the immense potential of immune network models for understanding robust code-clone detection from vulnerable software metrics. Therefore, we propose the first deep-immune-network model that learns which software metrics are useful to discover vulnerable code-clones. We use the immune network model with two deep neural network models to discover vulnerable code-clones. To the best of our knowledge, we apply the immune-network model to the domain of vulnerable code-clone discovery for the first time. We introduce an empirical assessment of our approach and examine its usefulness in practice in five different open-source software projects. We have concluded that the suggested approach is both reasonable and valuable for the detection of the software vulnerability of code-clones.**

Keywords—Deep neural network (DNN), Immune network, Software vulnerability, Object-oriented metrics, Code-clone.

I. Introduction

Code-clone detection is crucial for enhancing the code quality and producing more effective and beneficial software. Software clones are frequently regarded to indicate the lack of software quality. Detecting code-clones in a software system is essential to reduce both significant maintenance costs and the risks of potential failures associated with clone operations.

Code-clones cause a lot of damage in large-scale software systems. In any system given, the existence of code-clones does not influence its operation. However, it leads to an increase in its complexity and maintenance costs. Furthermore, code-clones can lead to increasing possible errors, update costs, required resource size, and response time for change requests. The researches demonstrated that 5% to 20% of software systems might include a duplicated code produced by merely copying the current program code and pasting it with or without slight modifications.

It is possible to group clone types into two on the basis of the similarity of their program text and functionality Types of clones are presented as follows: **Type 1:** Exact clones are the type of clones, in which the duplication of code fragments is performed without any modification, in other words, they remain unchanged. **Type 2:** Renamed clones represent identical clones in syntactical terms, apart from the modification of identifiers, types, comments, and whitespace. **Type 3:** Restructured clones. Renamed clones are further modified in structural terms (e.g., insertion, deletion, or statement rearrangement) for the purpose of generating restructured clones. **Type 4:** Semantic clones can differ in syntactical terms but have identical functionality.

The artificial immune network model represents a network of connected recognition cells that learn using feedback mechanisms. The fundamental idea of immune network model is that, in the case of the recognition of invasive antigens by antibodies, various antibodies make up a dynamic network by interacting between themselves. In the case of a higher similarity between antibodies, inhibition will be produced by the network. In the case of a low similarity, a stimulus will be generated by the network. Thus, the network can sustain population diversity and equilibrium and finally reaches stability by consisting of various memory cells.

We focus on establishing a code-clone discovery system for the purpose of detecting vulnerable clones from the software security perspective. In this paper, we try to adopt deep learning/immune network model-based approaches for the detection of software vulnerabilities. It is possible to utilize software metrics as an indicator of the existence of software vulnerabilities. The object-oriented software metrics have been utilized with deep learning for the purpose of predicting a source code prone to include vulnerabilities. Thus, the software metrics optimized for vulnerability detection are understood, and deep learning-based detection systems are capable of learning from the software metrics. Software metrics can discriminate between vulnerable and non-vulnerable functions. However, strong relationships between the object-oriented- metrics and the vulnerabilities that are present in the studied functions cannot be revealed. Therefore, the proposed models can demonstrate which components need more attention during inspection and testing.

In this paper [1], an approach suggested for the scalable detection of vulnerable code-clones, which can detect security vulnerabilities in large-scale software programs in an efficient and accurate way. A vulnerability-preserving abstraction scheme, ensuring the discovery of 24% more unknown vulnerability variants, is adopted in VUDDY. In this paper [2], a comprehensive review of the research introduced, focusing only on traditional ML techniques for detecting vulnerabilities. In this paper [3], the DNN neural network model is utilized for the efficient vulnerability classification. The National Vulnerability Database of the United States is utilized for the validation of the efficiency of the suggested model. In [4], a hybrid approach is proposed, which combined a genetic algorithm (GA) for feature optimization and a deep neural network (DNN) for classification. Furthermore, the DNN technique is improvised by utilizing an adaptive auto-encoder that represents the chosen software features better. In this paper [5], the VGRAPH detection system, mining vulnerable and patched source code from GitHub utilized for the identification of vulnerable code-clones afterward, is developed. Using graph-based components, VGRAPH is capable of identifying vulnerable code-clones at a 98% precision and 97% recall.

The rest of this paper is organized as follows. Section 2 explains the methods. The proposed method is described in Section 3. The experimantal results and discussion are shown in

Section 4. The conclusion and future studies are presented in Section 5.

II. METHODS

a. *Deep-Learning Based Classifiers*

Deep learning (DL) represents a branch of machine learning models. It is characterized by its ability to extract hierarchical representations from input data as a result of the establishment of deep neural networks having multiple layers of nonlinear transformations.

b. *Long-Short-Term-Memory (LSTM)*

LSTM networks represent one of the most effective solutions to a sequence of prediction problems because of the recognizing patterns in data sequences. Since LSTM networks have a particular type of memory, they can selectively remember patterns for a long time. They represent quite a reasonable approach to predict the period with the unknown long delays that occur between important events. The LSTM memory block's structure is composed of three gates and a self-recurrent connection.

An input gate modulates the additional degree of the novel memory content to the memory cell. Forget gate modulates the current memory forgetting gate. The output gate manages memory content exposure. Figure 1 illustrated architecture of the LSTM recurrent neural networks.

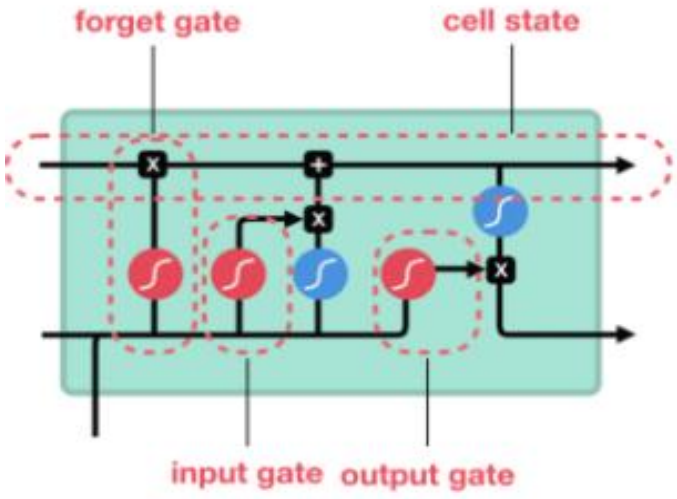


Fig. 1. Architecture of the LSTM recurrent neural networks [4].

c. *Gated Recurrent Unit (GRU)*

GRU represents an enhanced version of standard recurrent neural networks (RNNs) [6]. GRU networks have two gates: a reset gate (*r*), which performs the adjustment of the incorporation of novel input with the previous memory, and an update gate (*z*), which performs the control of preserving the previous memory.

The update gate assists the model by determining the amount of the previous information (from past time steps). The reset gate is utilized for the model to make a decision on the past information amount necessary to forget. Figure 2 shows architecture of the GRU recurrent neural networks.

I. PROPOSED METHOD

a. *Representation*

Each data was assumed to be denoted as follows:

x = {x1, x2, . . ., xN}, where N refers to the length of the data sample, and training data have a corresponding target value, vulnerable or not vulnerable, defined in software vulnerability models. Each antigenic pattern, a candidate solution, is represented as a bit string with length N, where N denotes the total number of software metrics used as features. For every class, a vector is formed for the measurement values of the software metrics to be utilized for code-clone detection. DNN methods take a sequence {x1, x2, . . ., xN} as input and construct a corresponding sequence of hidden states (or representations) {h1, h2, . . ., hN}. The antigenic pattern with the maximum values of fitness for vulnerabilities may be regarded as the most representative example of a population of vulnerabilities.

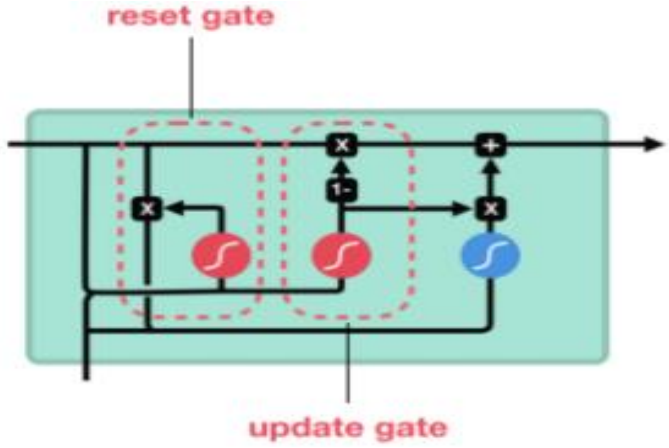


Fig. 2. Architecture of the GRU recurrent neural networks [4].

b. *Fitness Function*

The fitness of an antigenic pattern is proportional to the antibody-antigen affinity. The fitness function is used to reveal the quality of every antigenic pattern. We use the affinity function of two software metric (SM) vectors to measure their similarity. Each SM serves as an antigen.

Definition 1. Antibody-antigen affinity. Formula (12) presents a detailed description of the measurement of antibody-antibody affinity:

$$f(\mathrm{Abi}, \mathrm{Abj}) = \frac{1}{1+\| Abi, Abj \|} \qquad (12)$$

In the formula, ‖. ‖ refers to the Euclidean distance, Abi, and Abj refer to the antibody collection. The measure of the Euclidean distance is utilized to determine the distance, and thus affinity (inversely proportional) between the antibody and an antibody:

$$\mathrm{D_{i,j}} = \| Abi, Abj \| = \sqrt{\sum_{k=1}^{L} (Abi, Abj)^2} \quad , \mathrm{i}=1....\mathrm{N} \quad (13)$$

The distance between every two antibodies is found identically. The distance (affinity) between metric measurement vectors is computed for the purpose of identifying between which classes there is a similarity. For every project under evaluation, every of its classes is analyzed. A vector space of measurement values represents a class. The vector values are compared and contrasted by employing the stimulation function to find the vulnerable clone similarity.

c. *Deep-Immune-Memory Networks (Deep-IMNs)*

The proposed model is a computational modeling paradigm that depends on the immense detection and prediction capability of artificial immune neural networks. We construct a deep-immune-memory network that discovers software vulnerabilities from metrics with characteristics that are more prone to clone security.

In this paper, artificial immune networks (aiNet) overcome the detection of vulnerable code-clones due to the advantage of the memory network dynamics. The proposed model is designed to create a memory set recognizing and representing vulnerable clones.

In accordance with the aiNet learning algorithm, the detection of code-clones, software metrics are utilized with the aim of pooling the neurons into a network-level vector representation for every iNet in a separate manner. Then, the quantification of the affinity between the interactions of an antibody with an antibody and other antibodies is performed by the measurement of the similarity degree (affinity). The affinity is employed for the detection of behavioral equivalence between SM vectors, which is then generalized to the vulnerability.

Procedure: Deep-IMNs

Step 1: {InitializeAntibody Pool } //Enter software vulnerability metrics

Step 2: [Train] {1...N} (Input Size)

Step 3: **repeat**

Step 4: **For** each antibody Abj, j = 1,...,N, **do**:

Step 5: Evaluate fitness (affinity) of each Antibody (Abj)

Step 6: From C*, re-select ζ% of the antibodies with highest dk,j and put them into a matrix Mj of clonal memory;

Step 7: DNNM=*CreateDNNClonalMemory*(*Mj, t,State_id*)

Step 8: Apoptosis: eliminate all the memory clones from DNNM whose affinity Dk,j > σd:

Step 9: Determine the affinity Si,k among the DNNclonalmemory. Si,k=|| DNNMj,i − DNNM j,k ||, ∀i ,k

Step 10: Clonal suppression: eliminate those DNN memory clones whose si,k < σs:

Step 11: Concatenate the total antibody memory matrix with the resultant DNN clonal memory DNNMj* for Agj: Ab{m} ← [Ab{m};DNNMj *]

Step 12: DNNM* ←*UpdateDNNClonalMemory* (DNN*Mj, t,State_id*)

Step 13: *State_id* ← *State_id+1*

Step 14: Determine the affinity among all the DNN clonal memory DNNM* antibodies from Ab{m}: Si,k=|| Ab^i{m} – Ab^k{m} ||, ∀i ,k

Step 15: Network suppression: eliminate all the antibodies such that si,k < σs:

Step 16: **Until a termination condition is met**

Step 17: Build the total antibody matrix Ab← [Ab{m}; Ab{d}]

Step 18: List of the vulnerable metric set for robust clone detection

Step 19: **END For**

Fig. 3. Pseudocode of Deep-IMNs.

In the pseudocode of the Deep-IMNs model, Ab refers to the antibodies pool, the similarity matrix between every pair of antibody is represented by S, the vector that contains the affinities of Abj showed by C*, the vector that contains the affinity between each element from the set denoted by C*, ζ represents the percentage of the mature antibodies that should be chosen, Mj is the memory clone for antibody Abj, Mj* is the resultant clonal memory, σd is the natural death threshold, σs refers to the suppression threshold, and σcut refers to the cutting threshold. The Euclidean distance between the antibodies, capable of forming the affinity matrix, expresses Dk,j.

IV. Experimental Results and Discussion

In this section, we proceed with assessing our suggested method with software code metrics and vulnerabilities. For the evaluation, we consider five commonly employed open-source projects, such as Linux Kernel, Mozilla, Xen Hypervisor, httpd, and glibc.

We conducted experiments for the assessment of the performance of software metrics of security vulnerabilities. In the assessment of the suggested methodology, five different open-source projects were utilized. We identified a lot of potential metric-set combinations to take a decision on the correlation of vulnerable metrics sets for code-clone detection. The findings demonstrate that it is possible to use the dataset for the purpose of distinguishing which metrics are more prone to detect security vulnerabilities. It is also shown that the Deep-IMNs approach can predict more prone software metric sets for detecting the most vulnerabilities in the dataset at high accuracy. The best vulnerable software metric sets for the suggested LSTM-IMNs and GRU-IMNs are shown in Table 1 and Table 2, respectively.

The findings showed that using particular software metrics, including encapsulation, and inheritance, it was possible to achieve a high accuracy rate above 98% for LSTM-IMNs and GRU-IMNs in vulnerable code-clone detection, respectively. Furthermore, the obtained results showed that the httpd open-source project achieved the best metric-set for vulnerable code-clone detection. Figures 6-7 plot the Root Mean Square Error (RMSE)-measure of the deep-immune-network models and with respect to the five projects with a different number of hidden layers. The proposed deep-immune-network models achieved more successful results for httpd, Mozilla, Linux Kernel, Xen Superior, and glibc projects, respectively.

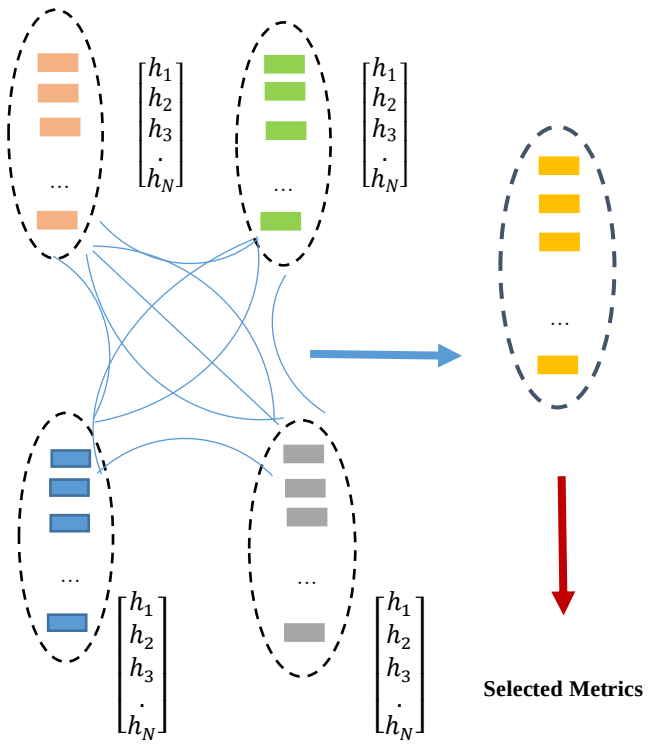


Fig. 4. The Schema of the proposed Deep-IMNs

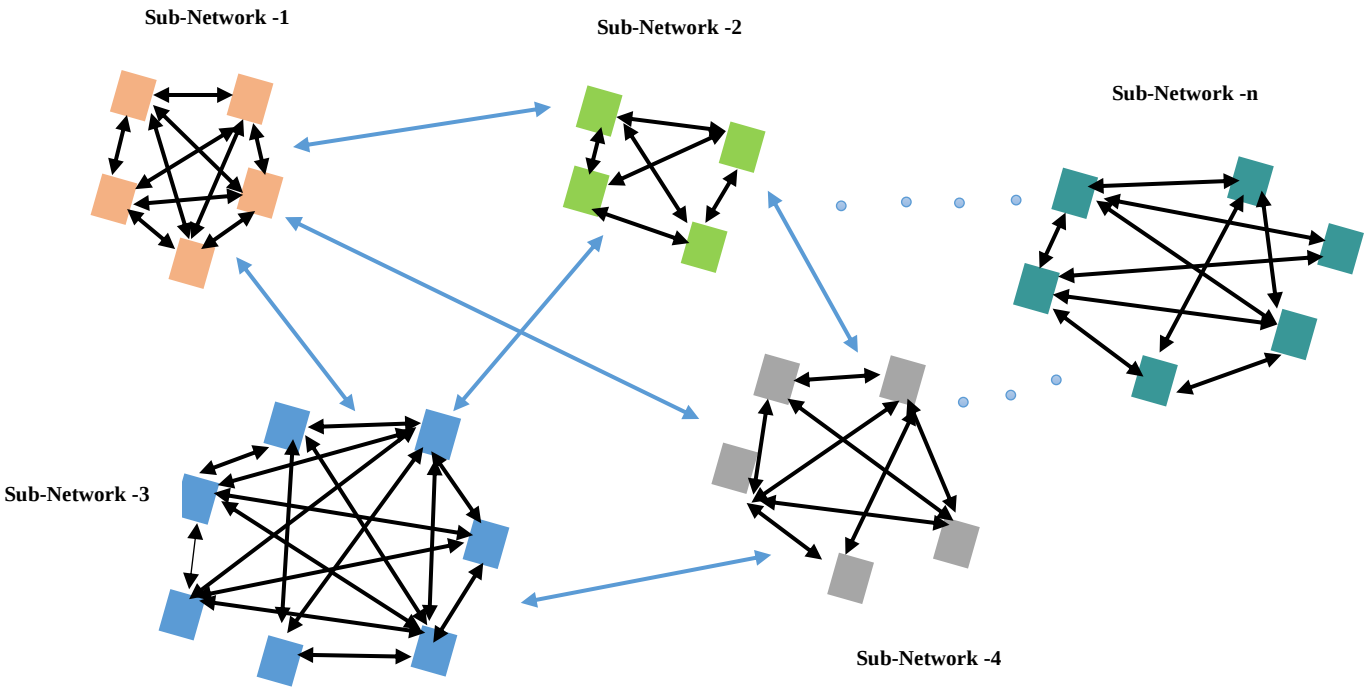


Fig. 5. Structure of Sub Immune Neural Networks

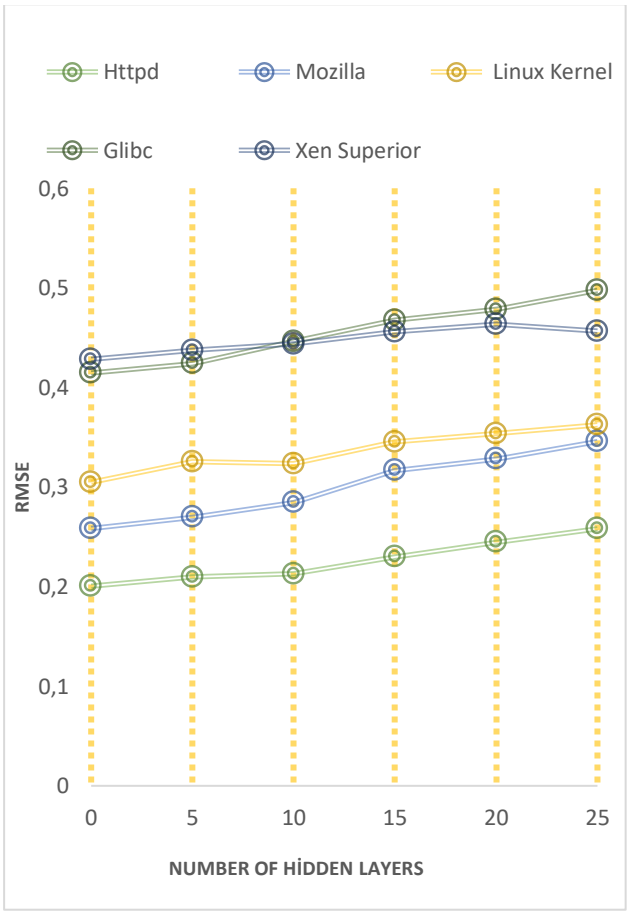


Fig. 6. The relationship between number of hidden layers and Root Mean Square Error (RMSE) of five projects based on different metric categories for LSTM-IMNs.

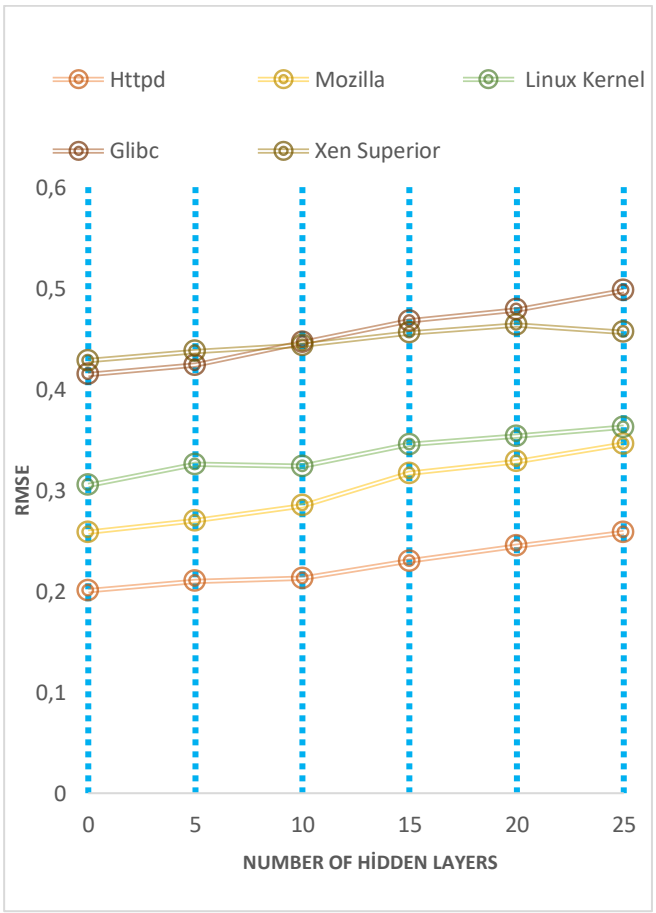


Fig. 7. The relationship between number of hidden layers and Root Mean Square Error (RMSE) of five projects based on different metric categories for GRU-IMNs.

Table I. Best vulnerable software-metris set for LSTM-IMNs

Metric Set	File-level (LSTM-IMNs)				
	Accuracy % (Linux Kernel)	Accuracy % (Mozilla)	Accuracy % (Xen Hypervisor)	Accuracy % (Httpd)	Accuracy % (Glibc)
Inheritance	95.36	94. 23	92.78	97. 65	93.75
Coupling	92.37	95. 74	90.25	96. 34	91.04
Polymorphism	90.32	95.64	88.25	96.45	90.1
Complexity	95.54	93. 43	92.56	97. 21	91.25
Size	96.76	95.79	85.67	96. 11	86.24
Encapsulation	94.67	96.93	89.55	**98. 15**	90.56

TABLE II. BEST VULNERABLE SOFTWARE-METRIS SET FOR GRU-IMNS

Metric Set	File-level (GRU-IMNs)				
	Accuracy % (Linux Kernel)	Accuracy % (Mozilla)	Accuracy % (Xen Hypervisor)	Accuracy % (Httpd)	Accuracy % (Glibc)
Inheritance	96.83	94. 79	95.08	**98.76**	94. 16
Coupling	94.75	97. 69	93.87	95.26	93. 84
Polymorphism	92.95	96.48	84.53	93.21	85.98
Complexity	97.36	94. 92	93.86	94.54	94. 06
Size	98.23	95.37	88.06	94.76	85. 31
Encapsulation	96.60	97.64	91.21	94.53	92. 22

V. CONCLUSION

In this paper, we designed the first deep-immune-memory network model that discovers the effect of the correlation between software metrics and vulnerable clones for open-source software projects. The analysis of software vulnerabilities was performed using deep learning-based classifiers on the basis of immune network model. A new framework, named the Deep-Immune-Memory Network (Deep-IMNs), was established as the generalized correlation for the vulnerable software metrics. The discussion demonstrated that clone detection by utilizing software metrics and the Immune-neural-network model was a good method of code-clone detection, analysis, and clone prediction.

The findings demonstrate that software metrics can discriminate between vulnerable and non-vulnerable functions. Furthermore, strong relationships between the mentioned metrics and the vulnerabilities that are present in the studied functions can be revealed. Ultimately, the findings show that there is a possibility that vulnerable functions will have other vulnerabilities in the future. The clone in sequence will be discovered by supplementary improvement approaches in the future. It is also crucial to assess the suggested methodology on other datasets.

ACKNOWLEDGMENT

This article does not contain any studies with human participants performed by any of the authors.

REFERENCES

[1] S. Kim, S. Woo, H Lee., and H. Oh, "VUDDY: A Scalable Approach for Vulnerable Code-Clone Discovery", IEEE Symposium on Security and Privacy, Doi: DOI 10.1109/SP.2017.62, 2017.

[2] S.M. Ghaffarian, and H.R. Shahriari , "Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey", ACM Compt. Surv., Vol:50 no:4, pp:1-36, Nov. 2017.

[3] Y. Fang, S. Han, C. Huang, R. Wu, "TAP: A static analysis model for PHP vulnerabilities based on token and deep learning technology", PLoS One. 2019; 14(11): doi: 10.1371/journal.pone.0225196, 2019.

[4] J. Chung, et al., "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.Cluster Comput 22, 9847–9863 (2019). https://doi.org/10.1007/s10586-018-1696-z.

[5] H.K. Dam , et al., "Automatic feature learning for predicting vulnerable software components", IEEE Transactions on Software Engineering, Doi: 10.1109/TSE.2018.2881961, pp: (99):1-1,2018.

[6] Z. Li , Y. Shao, "A Survey of Feature Selection for Vulnerability Prediction Using Feature-based Machine Learning", International Conference on Machine Learning and Computing, (ICMLC'19)Doi:10.1145/3318299.3318345, pp: 36–42, 2019.

[7] Z.Li , et al., "VulDeePecker: A Deep Learning-Based System for Vulnerability Detection", Cryptography and Security, Doi: 10.14722/ndss.2018.23158, 2019.

[8] M. Zagane, M.K. Abdi, M. Alenezi, "Deep Learning for Software Vulnerabilities Detection Using Code Metrics" , IEEE Access, Doi: 10.1109/ACCESS.2020.2988557, 2020.

[9] R. Russell , L. Kim, L. Hamilton, T. Lazovich , J. Harer, O. Ozdemir, P. Ellingwood, and M. McConley , "Automated vulnerability detection in source code using deep representation learning", in Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), pp. 757–762, 2018.

[10] M. White, M. Tufano, C. Vendome , D. Poshyvanyk, "Deep learning code fragments for codeclone detection", 31st IEEE/ACM International Conference, Doi: 10.1145/2970276.2970326, 2016.

[11] M. Zagane and M. K., Abdi ''Evaluating and comparing size, complexity and coupling metrics as Web applications vulnerabilities predictors,'' Int. J. Inf. Technol. Comput. Sci., vol. 11, no. 7, pp. 35–42, Jul. 2019.

[12] W. Wang, G. Li, B. Ma, X. Xia, Z. Jin, "Detecting Code Clones with Graph Neural Network and Flow-Augmented Abstract Syntax Tree", arXiv preprint arXiv:2002.08653, 2020 - arxiv.org, 2020.

[13] Md. Z. Alom, T. M. Taha, et al., (2019), "A state-of-the-art survey on deep learning theory and architectures". Electronics, 8, 292; doi:10.3390/electronics8030292.

[14] J. Zhang, X. Wang, H. Zhang, H. Sun, K. Wang, and X. Liu, "A novel neural source code representation based on abstract syntax tree," in Proceedings of the 41st International Conference on Software Engineering. IEEE Press, 2019, pp. 783–794.

[15] S.K. Singh , A. Chaturvedi, "Applying Deep Learning for Discovery and Analysis of Software Vulnerabilities: A Brief Survey ,Soft Computing: Theories and Applications". Advances in Intelligent Systems and Computing, vol 1154. Springer, Singapore. https://doi.org/10.1007/978-981-15-4032-5_59, 2020.

[16] B. Bowman, H.H. Huang, "VGRAPH: A Robust Vulnerable Code Clone Detection System Using Code Property Triplets , IEEE European Symposium on Security and Privacy (EuroS&P), 2020.

Prospectives of Big Data Analytics and Explainable Machine Learning in Identification of Probable Biomarkers of Alzheimer's disease

Afreen Khan
Department of Computer Science
Aligarh Muslim University
Aligarh, India
afreen.khan2k13@gmail.com

Swaleha Zubair
Department of Computer Science
Aligarh Muslim University
Aligarh, India
swalehazubair@yahoo.com

Samreen Khan
Mayo Institute of Medical Sciences
Lucknow, India
drsamreen2k4@gmail.com

Abstract— The recent advancement in healthcare domain results in the generation of a large amount of clinical, imaging, and medication data. The extensive analysis of such data targets employing big data analytics helps in timely identification of various diseases which thereby aids in building precautionary measures. Alzheimer's disease (AD) is the most common neurodegenerative disorder worldwide. To find an effective management strategy for AD; clinical, biological, and behavioral data of various cohorts are gathered, managed and broadcasted through various AD coordinating centers. In general, the collected data used to be imbalanced, incongruent, heterogeneous and sparse. In the present study, we employed the big data technology and machine learning correlate for modeling the bulky Alzheimer's disease Neuroimaging Initiative (ADNI) dataset to identify the potential biomarkers of AD. A total of 12741 data values and 1907 clinical variables for 1738 subjects were used. About 20 variables out of this AD big data, were identified as the most suitable biomarkers for prediction of AD respectively. Through ML explainability modeling, we identified the correlation and significance of various cognitive, MRI, PET and CSF metrics in contrast to the risk factors i.e., age and APOE4. The approach used in this study could be beneficial for AD-based research enrichment in pre-clinical tests, where enrolling patients at the jeopardy of cognitive degeneration is critical for verifying the efficiency of the study.

Keywords—Alzheimer's disease, big data, biomarker, explainability, machine learning

I. INTRODUCTION

Alzheimer's disease (AD), a neurodegenerative disorder, is the most common form of dementia. It is a convoluted disease, majorly distinguished by deposition of neurofibrillary tangles and β-amyloid (Aβ) plaques, comprised of tau amyloid fibrils [1]. This is allied with loss of synapse, extending to the neurodegeneration yielding in further memory damage and other severe related cognitive issues [2]. According to the 2019 World Alzheimer report, there will be an estimated 131.5 million people worldwide living with Alzheimer by the year 2050 [3]. Notably, AD cases have significantly increased in developed countries in recent years. However, there are no known methods or treatments which can either measure or slow down AD progression effectively and precisely in its initial stages. Thus, to predict clinical degeneration in the future, advance the development of novel drugs and using as outcome measures in clinical tests, there is a pressing requirement to find the related biomarkers [4].

AD data includes all the features of big data. These are characterized by 5 Vs viz. volume, variety, velocity, veracity, and value. The major objective of handling the AD big data which is collected from the public databanks is to discover potential biomarkers; thus finding a remedy for the ailment. Healthcare domains these days are considering the deployment of artificial intelligence, machine learning and deep learning systems on a large scale. When it comes to the operation of such systems in real-time, accountability and transparency becomes important in particular. Moreover, if such systems fail to deliver enhanced model interpretability, and eventually explainability, then the potential impact and hence the ability of artificial intelligence is thus limited.

Machine learning (ML) model explainability refers to the process of extracting insights from any ML model that is human-understandable. From the context of medical diagnosis, in particular to AD prediction and detection, ML models are employed to build imperative decisions [5]. To depend on the results of systems which are powered by the ML models, we ought to understand how these systems predict the AD. An explainable ML model does not specify its own explanation. Such models are complex to be comprehended by humans specifically, thereby they need superfluous methods to know how they make predictions.

The key task performed in this study was to identify important AD features among a bundle of AD big data. This is an analytical study which emphases on understanding the relationship among several AD features, to find the right combination of biomarkers to predict AD. Furthermore, significant features from the large pool of dataset were found and mapped in a simple interactive manner employing the ML model explainability.

II. BACKGROUND

A. Big Data: Challenges in Healthcare

In general, big data is referred to a collection of an enormous amount of data which comprises varied types of data. In the current digital era, big data has played a significant role due to the substantial development of healthcare technologies [6]. The healthcare domain and other sectors have been known for the sources of big data viz. volume and variety, therefore, the healthcare industry has acquired its effect due to the impact of big data [7]. There is no particular technique or a straightforward approach to implementing big data technology. The complexity is a result of the volume, variety, and veracity of the bulky datasets. Alzheimer's disease data consists of all the properties of big data which is high-dimensional [8]. Concerning AD data, these 5 Vs are illustrated in Fig. 1. Despite this, the challenging task is deriving inferences from the result of high-dimensional data. Specifically, the curse of dimensionality obstructs from understanding the end results of AD diagnosis.

VOLUME	VARIETY	VELOCITY	VERACITY	VALUE
• Several new attributes are being collected for Alzheimer's research. • There has been an increase in the contribution of different cohorts via numerous AD-related programmes. • This has made the AD data volumetric.	• Alzheimer's disease comprises a variety of data. • Viz. text, audio, images, and semi-structured data, gathered from different tracking devices.	• Velocity is described by the speed with which the data is generated, stored and managed. • From the context of AD research, real-time processing techniques facilitate real-time decision making.	• Veracity refers to the truthfulness of data. • It deals with the data quality concerns such as noise, biases and abnormality in data. • Alzheimer's data includes multi-source, heterogeneous, distinct incomplete and sparse values.	• Value signifies to the ability to turn data into meaningful information. • The value in the context of AD data refers to extracting value and converting it to significant knowledge.

Fig. 1. Alzheimer's Disease Big Data.

The 5 Vs complexity of big data of biomedical data have increased enormously. In AD research, high-throughput textual and image data has been used to come up with a probable solution to cure this disease. Many researchers have constructed predictive machine learning and deep learning models for diagnosis of AD and also, to find therapeutic attributes for early drug development [9]. At the same time, several varied computational algorithms are being programmed to combat AD. Along with this, many software tools to employ these algorithms on petabytes of AD data are being developed as well. With the upsurge of information, computational models and scientific practices employed in AD research, the tests and trials in its data analysis are also becoming exponential.

B. *Alzheimer's Disease: A Public Health Concern*

The old age is the main risk factor for the occurrence of cognitive impairment or AD in the later stage of life [10]. In general, AD is a syndrome, defined by numerous distortions in memory and is distinguished by the underperformance of cognitive abilities. Several pathological examinations have shown an accumulation of neurofibrillary tangles and amyloid plaques in the brain. This deposition causes neurodegeneration and ultimately leads to cognitive disorders [11]. However, the correct etiology of AD syndrome is still unidentified, but we can deduce that it is complex and multifactorial.

Double aging is a concept which means increment in life expectancy and a speedy rise in the percentage of oldest old in contrast with the youngest old. Due to this phenomenon perceived in high-income countries and the growing life expectancy and epidemiological change in lower and mid-income countries, the total number of people living with AD worldwide will only rise in the future and is expected to grow into 74.7 million cases till 2030 and 131.5 million cases till 2050 [12]. Thus, the World Health Organization has acknowledged AD as a public health main concern in 2012 [13].

C. *Machine Learning Model Explainability*

The fuel of technology is bulky data as per current times. The majority of data that prevails today are unreliable, noisy and are deficient of definite trends and behavior [14]. This is because of the existence of several errors making the data unstructured [15]. Machine Learning model can be thought of as a function, where the features are considered as the input while the predictions are regarded as output. Moreover, ML model is believed to be a black box, where they can make noble predictions but the logic behind those estimates is incomprehensible. Regarding this, an explainable ML model is such a function which is complex enough to be understood by a human. Thus, we require such a method which can dig into the black-box model and get familiar with its working.

This is what an explainable ML model does. It extracts those insights from the highly complicated ML models, which a normal exploratory data analysis fail to do at the time of data pre-processing. In this paper, we have framed an approach in identifying the right combination of biomarkers for the prediction of AD and what all bulky set of features have in common when it comes to AD diagnosis. This is pictorially represented in Fig. 2. Several methods have been applied while structuring this paper, to come up with a probable solution.

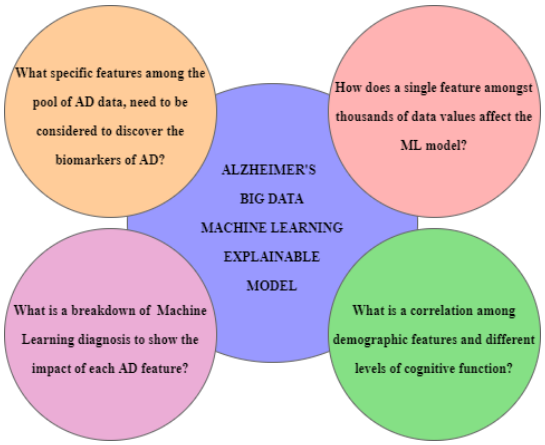


Fig. 2. ML Explainable Model.

III. MATERIALS AND METHODS

A. *Data Description*

Alzheimer's Disease Neuroimaging Initiative (ADNI) disseminates the Alzheimer's study dataset [16]. This database includes clinical, genetic, cognitive and imaging data collected at various participating sites of North America. Our study utilizes a comprehensive longitudinal dataset derived from the ADNI study.

B. *Study Participants*

The dataset obtained from ADNI for our study consists of 1738 subjects. The dataset encompasses baseline and time-series measurements for each feature. Our study data included either Alzheimer's Disease (AD), Cognitively Normal (CN), Early Mild Cognitive Impairment (EMCI), Late Mild

Cognitive Impairment (LMCI), or Significant Memory Complaints (SMC) participants.

The dataset consists of 12741 data values for 1907 number of features. Because of this amount of big data, this makes our approach applied to this study more legitimate. We divided it into eight different categories of data i.e. cognitive test, demographic information, diffusion tensor imaging, electroencephalography, genetic test, magnetic resonance imaging (MRI), positron emission tomography (PET), and cerebral spinal fluid (CSF) measurements. In general, data categorization aids in understanding the big picture i.e. what type of measurements or tests need to be performed to find out AD at the earliest.

C. Data Preparation

After data gathering, the next challenge was data wrangling (pre-processing). Each feature column was analyzed in detail to eliminate administrative and redundant data. In the medical domain, categorical data are ubiquitous. However, several robust algorithms are there that prove better for numerical data but fail to work with the categorical data. In our study, there were too many categorical data. This is a bottleneck while performing analytical study during data pre-processing. Consequently, this was handled well with many scikit-learn libraries.

D. Imputation of Missing Values

As ADNI data is maintained for the biomedical research of Alzheimer's disease, a huge number of demographic, genetic, MRI and other biomedical attributes are gathered. During the collection, several attributes cannot get recorded due to several reasons, which causes missing data in the dataset. These missing values are inevitable during the data collection process [17]. In this study, missing data were treated using imputation. The downstream data mining and statistical methods require full data matrix. For this reason, imputation is a practical solution.

IV. RESULTS

A. Demographic Characteristics

From this dataset of 1737 subjects, the gender distribution was found to be 780 females and 957 males. As a percentage of subjects by gender, there were 44.86 percent females while 55.14 percent males. The age range was [55-90] years for females with a mean age of 73 years and [54-91] years for males with a mean age of 75 years. The subjects' marital status was found to be distributed between four classes i.e. 1310 married, 206 widowed, 150 divorced and 64 unmarried while the rest of the values were unknown. Furthermore, the dataset also contained the ethnicity of the subjects. It was either Hispanic/Latin (58 subjects) or non-Hispanic/Latin (1668 subjects). These are the count of baseline values i.e. when the subject initially arrived. These 1737 subjects visited after a rollover of 6 months for either 2 or 3 years, making it a total of 12,738 data values. As mentioned in Section II(B) above, the baseline diagnosis is represented by the following: Alzheimer's Disease (AD), Cognitively Normal (CN), Early Mild Cognitive Impairment (EMCI), Late Mild Cognitive Impairment (LMCI), or Significant Memory Complaints (SMC).

B. Correlational Analysis: Risk Factors vs Brain Metrics

There are two major risk factors in the development of AD i.e. age and APOE4. Based on these risk factors, we performed correlational analysis employing 3 different tests, Pearson, Spearman and Kendall. These were performed with cognitive measures (CDRSB, MMSE, ADAS11, RAVLT 5 sum), MRI measures (Hippocampus, Whole Brain, Entorhinal, MidTemp), PET measures (FDG, AV45), and CSF measures (Amyloid, Tau, PTau proteins). Along with age and APOE4, we took gender attribute also, to discover any novel associations among the various metrics.

In our experimental analysis, we discovered that age has a significant relation with all four cognitive measures, APOE4 does not have any significance with that of Clinical Dementia Rating Sum of Boxes (CDRSB) but extremely significant with other three metrics. Next with MRI, PET and CSF measures, we found that both age and APOE4 possess a significant relationship. In some cases, there was an extremely significant association such as age and MRI measures. Moreover, we found no significance or weak significance amongst gender and the four set of brain metrics.

C. Permutation Testing

Permutation testing evaluates the statistical significance of the ML algorithm [18]. The next part of this study was to determine the features that have a huge impact on predictions while ML modeling. We applied a better method of determining the feature importance i.e. permutation importance. It is evaluated once the ML model is fitted. Thus, in this study, once the ML model was fitted using the Random Forest algorithm, the next challenge was to determine the features that have the largest influence on predictions. This helps further in determining the class of suitable biomarkers that may further aid in the diagnosis of AD.

The highest value was shown for CDRSB i.e. 0.2474 ± 0.0199. The next highest value was shown for Mini Mental State Examination (MMSE) as 0.2074 ± 0.0267. ADAS13 feature gave an importance value of 0.0056 ± 0.0070 while ADAS11 showed a value of 0.0037 ± 0.0037. Interpreting these values suggests us that the first value (before ± sign) describes by what value does the performance of the model reduced. While the value after the ± sign signifies by what amount does the performance varied. These values show that the most important feature in AD prognosis is CDRSB. MMSE also has a great impact on the AD diagnosis, being it another cognitive score.

D. Partial Dependence

The feature importance showed the set of important features that affect predictions mostly. On the other hand, how or in what way the feature affects the prediction is determined through the partial dependence. In this study, the next most important question was to uncover how the highly affecting features are related. To understand the suitability of these as a right biomarker for the AD research, we uncovered those features which showed higher importance. In this paper, we are illustrating CDRSB and MMSE values pictorially in Fig. 3 and Fig. 4. These plots are built on complex model i.e. on the fitted Random Forest. This holds the property of capturing even more severe complicated patterns in comparison to simpler models.

The y-axis in Fig. 3 shows the 'change in the prediction' and the blue region means the level of confidence. Through Fig. 3, we can comprehend that the CDRSB score increases the chances of correct AD diagnosis. But beyond a certain limit, it holds a reduced amount of influence on the

predictions. Fig. 4 indicates the plot for MMSE. There are eight grid points in this. Initially, it causes low predictions but after a definite period, a change in prediction can be observed, which is a good indication for it to be the right choice as a biomarker.

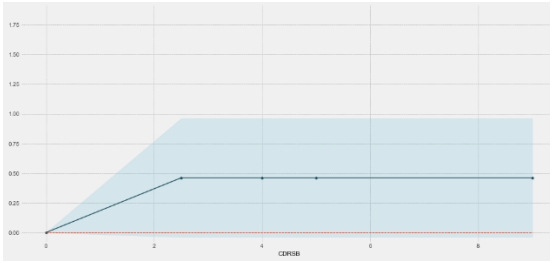


Fig. 3. Partial Dependence for CDRSB.

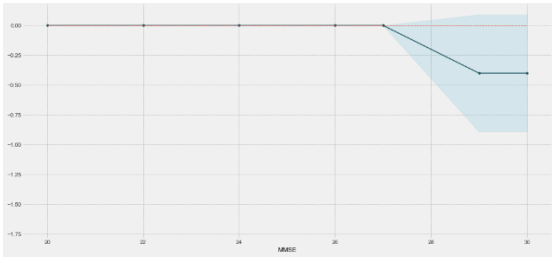


Fig. 4. Partial Dependence for MMSE.

Altogether, the smoothness of these plots appears to be much more plausible. This shows that whatever amount of big data is there i.e. huge number of features and there corresponding subject's values, the importance and hence their respective partial dependence plots can be evaluated and can be further examined whether or not these specific set of features hold a good choice as biomarkers for AD diagnosis.

E. Identification of Robust Biomarkers

Apart from the above findings, we move ahead to break down the working of a model for a specific prediction. For this, we have SHAP values (SHapley Additive exPlanations) [19]. SHAP has a property of splitting the evaluated diagnosis to demonstrate each feature's impact. We followed this approach to describe the prediction of a feature by calculating each feature's contribution to AD prediction. Later, it was aggregated to show the powerful ML model insights (Fig. 5). It posed to be a strong approach in AD ML explainability. This helped in determining the right set of biomarkers for AD diagnosis among the 1907 provided features in the dataset. By implementing this approach, it further aids in modeling a definitive ML-based system for the AD diagnosis and other related research based on this.

From Fig. 5, it can be seen the importance of features in decreasing order. Here, we have shown the plot only for top influential features. We can comprehend through this, that the features with high absolute SHAP values are significant. Also, because we are required to have only those features that have global importance, in real picture, we calculate the mean of the absolute SHAP values per feature. The values in Fig. 5 are measured as the average absolute SHAP values. The CDRSB was found to be the most important feature, changing the predicted absolute AD probability by a few decimal points.

Moreover, there is a considerable difference amongst both of the applied approaches (Section 4.3 and 4.5). The one described in Section 4.3 i.e. permutation importance is centred on the idea of a decrease in model performance. While the SHAP method is based on feature attributions magnitude.

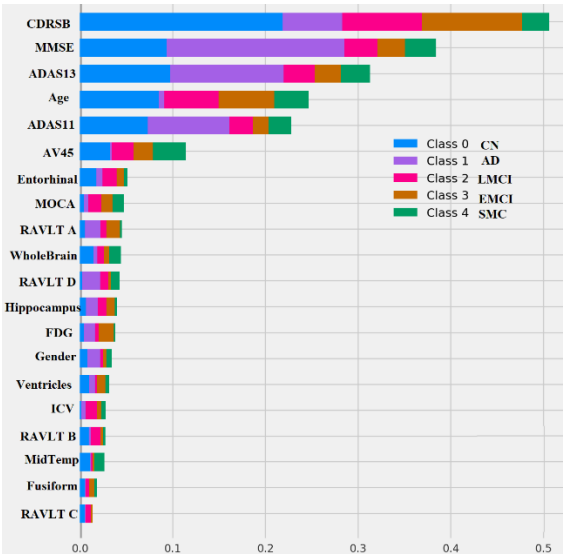


Fig. 5. ML Explainability of Potential Biomarkers.
(where A, B, C and D in RAVLT (Rey Auditory Verbal Learning Test) are the sum of 5 trials, trial 5–trial 1, trial 5–delayed and percent forgetting)

Based on the above findings, the resultant set of biomarkers was further used for the prediction of AD. Random Forest (RF) classifier outperformed other ML classifiers. After hyperparameter tuning, it gave an accuracy of 87.72 percent on the defined set of twenty biomarkers. While XGBoost classifier gave slightly lesser accuracy than RF. This is because while we performed tuning and Bayesian optimization, it reduced the accuracy of the predicted ML model.

V. DISCUSSION

As the clinical tests in general focus on presymptomatic subjects, the study populations which are most likely to get converted to MCI during the conventional trials, may perhaps lessen the needed sample sizes and also, the expenses required for assessing study participants [20]. ADNI data is used in several of the AD-based studies and numerous research has been performed in this direction using different datasets provided by it. The approach shown in this study might be beneficial for deciding on the candidate biomarkers from the ADNI sampling pool.

This study ascertains on establishing an understanding of a large set of brain metric features of AD in older adults. We looked over in elucidating the ML explainability model for AD big data. In our results, we observed that men were more likely to have AD when compared to women. This is more likely due to the prevalence of numerous major disparities that often appear amongst men and women in the presentation, existence, and development of psychiatric symptoms. Many past findings have reported that women are more susceptible to progress towards AD [21]. This is because they are at higher risk of depression in contrast to men [22]. Also, the APOE4 gene affects men and women differently [22]. One of the authors in their paper observed that age, APOE4, and gender are the most critical factors in AD development [23].

In our study, we found that the CDRSB scale has the highest feature importance. This is true in general as it is a coherent assessment tool which is being used in many dementia-related studies. The CDRSB scale is correlated with other performance cognitive metrics like the MMSE, Abbreviated Mental Test, and comprehensive psychometric tests [24]. The MMSE cognitive test is a measure of cognitive impairment, which has been employed in the detection of AD. Our results show MMSE as next best biomarker for the prognosis of AD. From above results, we can comprehend well that the cognitive measures i.e. CDRSB, MMSE, ADAS11, MOCA and RAVLT (5 sum), MRI measures i.e. Hippocampus, WholeBrain, Entorhinal and MidTemp, PET measures i.e. FDG and AV45 prove to be highly important biomarkers in the diagnosis of AD. In our study, the CSF measures i.e. Amyloid, Tau and PTau protein were found to have a significant correlation with age and APOE4 but in ML explainability model, it failed to compete with other features and was not marked as a top biomarker. This is because we had a large number of missing values for these three CSF measures. And missingness of values has a huge impact on the feature importance, thereby affecting the overall model performance.

There are several ML models used for classifying AD and CN subjects. And these models use cognitive-based assessments i.e. CDR and MMSE score or MRI-based imaging volume attributes [25]. In comparison to previous studies, our study investigated the relationship between a large pool of AD features individually to derive the correct set of biomarkers. When the class of correct features is chosen, it directly affects the performance of the ML model. This was the basis of this study where we provided the comprehensive Alzheimer's big data ML explainable model. This is motivated by the hypothesis that distinct regions are affected differently by AD and that changes in separate regions can be captured by different features.

VI. CONCLUSION

This paper proposed to investigate suitable biomarkers derived from brain regions to detect AD. Our results indicate that the cognitive, MRI, PET measures are major biomarkers in comparison to CSF and DTI (Diffusion Tensor Imaging) metrics. Using this set of biomarkers will show a great difference between AD and CN subjects. The study further confirmed upon the role of the cognitive assessment in the identification of AD using Random Forest classifier. Also, several important insights were extracted from the sophisticated classifiers employing the ML explainable model.

ACKNOWLEDGMENT

The data used in this study were acquired from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database

REFERENCES

[1] Hardy J. Alzheimer's disease: the amyloid cascade hypothesis: an update and reappraisal. J Alzheimers Dis. 2006;9(3 Suppl):151-3. doi: 10.3233/jad-2006-9s317. PMID: 16914853.

[2] Weiner MW, Veitch DP, Aisen PS, et al. "Alzheimer's Disease Neuroimaging Initiative. The Alzheimer's Disease Neuroimaging Initiative: a review of papers published since its inception", Alzheimers Dement. 2012 Feb;8(1 Suppl):S1-68. doi: 10.1016/j.jalz.2011.09.172. Epub 2011 Nov 2. PMID: 22047634; PMCID: PMC3329969.

[3] "World Alzheimer Report 2019 Attitudes to dementia," 2019.

[4] Mueller SG, Weiner MW, Thal LJ, et al. "The Alzheimer's disease neuroimaging initiative", Neuroimaging Clin N Am. 2005 Nov;15(4):869-77, xi-xii. doi: 10.1016/j.nic.2005.09.008. PMID: 16443497; PMCID: PMC2376747.

[5] A. Khan and S. Zubair, "A Machine Learning-based robust approach to identify Dementia progression employing Dimensionality Reduction in Cross-Sectional MRI data," *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, Riyadh, 2020, pp. 237-242.

[6] X. Wang, Y. Wang, C. Gao, et al. "Automatic diagnosis with efficient medical case searching based on evolving graphs," *IEEE Access*, vol. 6, pp. 53307–53318, 2018.

[7] R. Raja, I. Mukherjee, BK Sarkar, "A Systematic Review of Healthcare Big Data", Scientific Programming, vol. 2020, Article ID 5471849, 2020. https://doi.org/10.1155/2020/5471849.

[8] Saeid Amiri, Bertrand S. Clarke & Jennifer L. Clarke (2018) "Clustering Categorical Data via Ensembling Dissimilarity Matrices", Journal of Computational and Graphical Statistics, 27:1, 195-208.

[9] A. Khan and S. Zubair, "An Improved Multi-Modal based Machine Learning Approach for the Prognosis of Alzheimer's Disease," J. King Saud Univ. - Comput. Inf. Sci., 2020.

[10] Barnes DE, Yaffe K, "The projected effect of risk factor reduction on Alzheimer's disease prevalence", Lancet Neurol 2011; 10(9): 819-28.

[11] Anstey KJ, Cherbuin N, "Herath PM. Development of a New Method for Assessing Global Risk of Alzheimer's Disease for Use in Population Health Approaches to Prevention", Prevention Science 2013; 14(4): 411-21.

[12] Anstey KJ, Mack HA, Cherbuin N, "Alcohol consumption as a risk factor for dementia and cognitive decline: meta-analysis of prospective studies", Am J Geriatr Psychiatry 2009; 17(7): 542-55.

[13] Kalantarian S, Stern TA, Mansour M, Ruskin JN, "Cognitive impairment associated with atrial fibrillation: a meta-analysis", Ann Intern Med 2013; 158(5 Pt 1): 338-46.

[14] A. Khan and S. Zubair, "Expansion of Regularized Kmeans Discretization Machine Learning Approach in Prognosis of Dementia Progression," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020, pp. 1-6.

[15] Sivarajah, U., Kamal, M.M., Irani, Z., Weerakkody, V., "Critical analysis of Big Data challenges and analytical methods", J. Bus. Res. 2017 (70), 263–286.

[16] http://adni.loni.usc.edu/

[17] A. Khan and S. Zubair, "Usage Of Random Forest Ensemble Classifier Based Imputation And Its Potential In The Diagnosis Of Alzheimer's Disease," Int. J. Sci. Technol. Res., vol. 8, no. 12, pp. 271–275, 2019.

[18] Golland P., Fischl B, "Permutation Tests for Classification: Towards Statistical Significance in Image-Based Studies", in Information Processing in Medical Imaging: 18th International Conference, C. Taylor and J. A. Noble, Eds., IPMI, 2003.

[19] Lundberg, Scott M., and Su-In Lee, "A unified approach to interpreting model predictions", Advances in Neural Information Processing Systems. 2017.

[20] Breitner, J.C. (2016), "How can we really improve screening methods for AD prevention trials?", Alzheimer's & Dementia: Translational Research & Clinical Interventions, 2: 45-47.

[21] A. Khan and S. Zubair, "Longitudinal Magnetic Resonance Imaging as a Potential Correlate in the Diagnosis of Alzheimer Disease: Exploratory Data Analysis," *JMIR Biomed. Eng.*, vol. 5, no. 1, pp. 1–13, 2020.

[22] Hara Y. How does Alzheimer's affect women and men differently?. 2018 Jul 02. URL: https://www.alzdiscovery.org/cognitive-vitality/blog/how-does-alzheimers-affect-women-and-men-differently

[23] Riedel BC, Thompson PM, Brinton RD, "Age, APOE and sex: triad of risk of Alzheimer's disease", J Steroid Biochem Mol Biol 2016 Jun;160:134-147.

[24] W. S. Lim, J. J. Chin, et.al, "Clinical Dementia Rating Experience of a Multi-Racial Asian Population," *Alzheimer Dis. Assoc. Disord.*, vol. 19, no. 3, pp. 135–142, 2005.

[25] M.-J. Chiu et al., ``Plasma tau as a window to the brain-Negative associations with brain volume and memory function in mild cognitive impairment and early alzheimer's disease," Hum. Brain Mapping, vol. 35, no. 7, pp. 3132_3142, 2014.

Identification of Medical Forum Posts on Hypertension and Cholesterol Based on Machine Learning

Mansur Alp Toçoğlu
Department of Software Engineering
Manisa Celal Bayar University
Manisa, Turkey
mansur.tocoglu@cbu.edu.tr

Aytuğ ONAN
Department of Computer Engineering
Izmir Katip Celebi University
Izmir, Turkey
aytug.onan@ikcu.edu.tr

***Abstract*— With advancements in information technology, users are increasingly producing content, including health information and support, on microblogging sites and medical forums. In meeting the needs of patients for health information, the patient-generated content accessible on medical forums plays a crucial role. Given the immense number of questions and the low number of experts, a significant portion of the questions remain unanswered. To obtain adequate answers in a reasonable time, guiding questions and forum posts to relevant experts, according to subject preferences becomes an essential issue. In this paper, we intend to classify medical forum posts on hypertension and cholesterol based on conventional text representation models and classification algorithms. In this scheme, we used three models (i.e., unigrams, bigrams, and trigrams) and three types of word representations (i.e., term frequency, term presence, and TF-IDF). Furthermore, the ensemble combinations of various attribute sets and representations have been considered. In this way, nine representation models were obtained. Our results show that when combined with artificial neural networks, the TF-IDF-based representation scheme provide good results with an accuracy of 92.18%.**

Keywords—text classification, medical forums, hypertension, machine learning, topic classification

I. INTRODUCTION

With its progressively expanding and evolving attributes, the World Wide Web is one of the richest sources of information. With many types of multimedia, such as, text, video, etc., the Web allows access to many web sites. There has been a continuous growth in the number of blogs, microblogs, and online multimedia. Blogs, forums, wikis, social networks, and review websites are too the essential implies of replying users' questions and making choices in different zones within the setting of healthcare [1]. The patient-generated content available on medical forums plays a key role in fulfilling patients' needs for health information. However, many questions will not be answered as many questions and a limited number of specialists are provided. Driving questions and forum posts to specific specialists according to thematic preferences becomes an important factor in getting the right answers in a timely manner.

Medical forums are unique sources of user-generated medical, health and wellness information. This unique content provides feedback and encouragement through the personal experiences of the individual, not available anywhere else [2].

Hypertension, too, alluded to as high blood pressure, may be a long-term therapeutic condition in which the blood vessel blood pressure is determinedly raised [3]. Due to its high frequency and concomitant rise in infection hazard, hypertension is an imperative open wellbeing issue around the world [4]. For cardiovascular, cerebrovascular, and renal clutters, it is the foremost critical hazard calculate [5]. Critical numbers of individuals all-inclusive endure from hypertension. In Poland, the most elevated predominance was found, with 68.9 percent among men and 72.5 percent among ladies [6].

In a comparative way, the dangers of heart malady and stroke are expanded by raised cholesterol. Generally, over the top cholesterol is mindful for one third of ischemic heart infection [7]. Given the predominance of hypertension and raised cholesterol universally, there are numerous questions on the Internet significant to these infections.

In this paper, based on conventional text representation models and classification algorithms, we point to recognize medical forum posts on hypertension and cholesterol. We have analyzed 2,113 cholesterol-related and 3,437 hypertension-related forum posts from medhelp.org [8]. We utilized three sorts of attributes (i.e., unigrams, bigrams, and trigrams) and three sorts of word representations in this plot (i.e., term-frequency, term-presence and TF-IDF). In model, the varieties of distinctive attribute sets and representations of the text have been considered. Nine models of representation have been gotten in this way. For assessment task, five machine learning methods (specifically, support vector machines, logistic regression, Naïve Bayes, random forest, and artificial neural systems) have been utilized.

The rest of this paper is organized as takes after: In Section II, the related work on text classification and medical forum posts have been examined. In Section III, the methodology of the study has been presented. In Section IV, the exploratory strategy and experimental results have been examined. At last, Section V presents the concluding comments of the study.

II. RELATED WORK

Medical forums are unique sources of medical, health and wellness knowledge generated by users. For this reason, text mining on medical forums became very popular and has attracted many researchers' attention. This section briefly discusses the earlier research contributions in the field.

Ali et al. [9] focused on the polarity of emotions conveyed in textual content messages, shared on clinical forums. For assessment challenge on sentiment analysis, distinct attribute engineering schemes, such as, range of phrases observed as

subjective in present day sentence, range of adjectives, range of adverbs, range of pronouns, range of phrases having previous polarity as positive, range of phrases having previous polarity as negative, range of phrases having previous polarity as impartial and range of terms with pronouns had been taken into consideration together with Naïve Bayes, support vector machines and logistic regression.

Similarly, Sokolova and Bobicev [10] displayed a machine learning based approach for opinion examination of messages posted on a medical microblogging platform. In this conspire, microblogging posts have been relegated into one of the five classes, as support, appreciation, disarray, actualities, and facts/sentiments.

Carrillo-de-Albornoz et al. [11] presented an opinion analysis system for medical forum posts based on machine learning. In this plot, content-based attributes (such as, word embeddings, concept embeddings and bag-of-words), space particular attributes, arrange attributes, sentiment-based attributes and linguistic attributes have been assessed in conjunction with machine learning classifiers, such as, K-nearest neighbor, Naïve Bayes, random forests and voting ensemble method.

Using online drug surveys, Gopalakrishnan and Ramaswamy [12] examined persistent fulfilment with solutions. For that work, a machine learning-based approach to extremity discovery based on artificial neural systems was presented. Concurring to the discoveries of an experiment on a corpus of sedate conclusions collected from the askapatient.com site, outspread premise work systems outflank the other plans. In a comparative way, Salas-Zarate [1] presented a sentiment examination system for sedate supposition examination based on psycholinguistic feature sets.

Liu et al. [13] categorized therapeutic field issues agreeing to whether healthcare experts or clients utilizing measurable and category attributes to construct a learning model based on support vector machines. In another work, Roberts et al. [14] classified restorative questions, as understanding, common information, and investigate situated, based on lexical, semantic, and syntactic attribute sets.

In a similar way, Jalan et al. [14] presented a deep learning approach to question classification on medical forum posts. In this scheme, deep learning architectures, such as, bidirectional long-short term memory, bidirectional long-short term memory with attention mechanism, hierarchical networks and deep neural models based on medical attributes have been evaluated.

III. METHODOLOGY

In this part, the dataset collection process, attribute extraction plans, and supervised learning methods utilized within the observational examination have been displayed.

A. Dataset Collection and Preprocessing

In this research, we collected a dataset to identify medical forum posts on hypertension and cholesterol. We have analyzed 2,113 cholesterol-related and 3,436 hypertension-related forum posts from medhelp.org. In the medical forum, we have collected three types of posts, i.e., titles of queries, queries, and answers to queries. The distribution of the dataset based on different types of posts has been summarized in Table I.

TABLE I. THE DISTRIBUTION OF DATASET

Type of post	Categories		
	Hypertension	***Cholesterol***	***Total***
Title of query	942	453	1395
Query	942	453	1395
Answer	1553	1207	2760
Total	3437	2213	5550

In this analysis, we pre-processed the collected dataset to prepare for the operations. We removed all the punctuation marks, numeric characters, and extra spaces in the first stage. In the field of computational linguistics, stemming and lemmatization are text normalization schemes utilized to prepare text, phrases, and documents for further processing. In this study, we utilized stemming, lemmatization, and raw scheme from Python NLTK. By using the Turkish language stop-word list, we deleted stop-words offered in Python NLTK [16, 17].

B. Text Representation Schemes

For language modeling and computational linguistics tasks, N-gram modeling may be a common attribute representation conspire. An n-gram may be a touching arrangement of n things from a content report occurrence given. Phonemes, syllables, letters, words, or characters may be the objects in this conspire. Word-based n-grams and character n-grams have been commonly utilized in common dialect handling errands [17, 18]. N-gram measure 1 was alluded to as "unigram" and N-gram measure 2 was alluded to as "bigram," whereas N-gram measure 3 was alluded to as "trigram."

We have utilized word-based n-gram models, where unigrams, bigrams and trigrams have been considered, to demonstrate medical forum posts. We have considered three distinctive models within the vector space model (VSM) to classify microblogging posts, specifically, term presence-based representation, term frequency-based representation, and TF-IDF-based representation. The number of occurrences of words within the reports was numbered in terms of frequency-based representation, specifically that each record was spoken to by a rise to length vector with the comparing word checks.

The presence or absence of a word in each record has been utilized to portray content records in terms of presence-based representation. Term scoring plans can moreover be utilized to demonstrate content archives, in expansion to recurrence and presence-based representation plans, where the meaning of terms/words on a report or corpus has been illustrated. We have utilized TF-IDF method and gotten nine diverse arrangements of the empirical analysis dataset in this way.

C. Classification Algorithms

Five supervised learning algorithms (i.e., support vector machines, logistic regression, Naïve Bayes, random forest, and artificial neural networks) were evaluated in the classification process. The remainder of this section briefly explains the supervised learning algorithms.

Support vector machines (SVM) are machine learning techniques that can be used to tackle classification and regression tasks. SVM can viably arrange datasets with both linear and non-linear attributes [19]. SVM builds a hyperplane

in a higher-dimensional space to address order and relapse errands. The objective of the hyperplane is to accomplish a decent division by arriving at the greatest distance to the closest training objects of the classes, alluded as a useful edge.

Logistic regression (LR) is a machine learning method that utilizes a direct capacity of an assortment of indicator factors to demonstrate the probability of any occasion happening [20]. Linear regression can create great outcomes. Notwithstanding, the membership esteems delivered by linear regression cannot generally be inside the reach [0–1], which is anything but a satisfactory scope of probabilities. In LR, a direct model is based on the changed objective variable while eliminating the issues portrayed previously.

The Naïve Bayes method (NB) is a probabilistic classifier for supervised learning on the hypothesis of Bayes. Because of the supposition of conditional independence, it has a simple structure. It tends to be adequately utilized in a wide scope of uses, including text mining and web mining, notwithstanding its essential structure [21, 22].

Random Forest is a machine learning algorithm. It produces a forest, as the name suggests, and somehow spontaneously does it. The "forest" that it has developed is a series of decision trees that are often trained by the "bagging" process. The general principle of the bootstrap aggregating approach is that a mixture of learning models would improve the overall result [23, 24]

Artificial neural network (ANN) is a computational model consisting of several processing elements which obtain inputs and produce outputs based on the predefined activation functions [24]. Artificial neural networks; emerged as a result of an attempt to mathematically model the learning process, inspired by the human brain. For this reason, research on this topic began with the simulation of neurons, which are the biological units that make up the brain, and its application in computer systems, and were later used in many fields in tandem with the advancement of computing [25, 26].

IV. Experimental Procedure And Results

Evaluation measures, experimental procedures and experimental findings have been discussed in this section.

A. Experimental Procedure and Evaluation

In the empiric study, 10-fold cross-validation was utilized. In this plot, the introductory dataset is arbitrarily separated into 10 commonly exclusive datasets. The training and testing sets are reproduced 10 times and each portion is tried and prepared 10 times. The comes about recorded in this area are the 10-fold values. Three isolated strategies of extraction of attributes (specifically, term-presence based representation, term-frequency based representation and TF-IDF weighting conspire) have been considered within the empiric examination. In expansion, different n-gram sizes (from n = 1 to n = 3) have been tried. In this way, we got 9 diverse setups for the dataset. Five supervised learning algorithms have been utilized within the empiric consider classifying medical forum posts. To survey the effectiveness of the classification methods, classification accuracy has been considered.

B. Experimental Results

In Table II, the experimental results for unigram-based representation schemes with TF-IDF weighting have been presented with stemmer, lemmatization, and raw scheme on five supervised learning algorithms. As can be seen from the results appeared in Table II, unigram-based representation schemes with TF-IDF weighting with lemmatization, stemmer and raw plot yield comparative prescient execution in terms of classification accuracy, with the most noteworthy average performance on lemmatization.

TABLE II. Experimental Results For Representation schemes

Representation Scheme	**SVM**	**LR**	**NB**	**RF**	**ANN**
Unigram, TF-IDF, Stemmer	91,43	91,47	83,70	91,28	91,62
Unigram, TF-IDF, Lemmatization	91,47	91,56	83,90	91,31	91,60
Unigram, TF-IDF, Raw	91,45	91,53	84,16	91,33	91,39

Regarding the predictive performances recorded on Table II, the most noteworthy accuracy values are by and large gotten by artificial neural systems, the second most elevated accuracy values are by and large gotten by logistic regression, the third most noteworthy accuracy values are by and large gotten by random forests method. The lowest predictive exhibitions are for the most part gotten by Naïve Bayes learning method.

To evaluate the performance of different attribute sizes on text representation, we have considered four different attribute sizes (i.e., 250, 500, 1000, and 1500).

Table III presents the empirical results for different attribute sizes on classification algorithms with unigram and TF-IDF based representation (with lemmatization). As it can be observed from the results recorded in Table III, the classification algorithms abdicate the most noteworthy predictive performance when attribute size breaks even with to 1000.

TABLE III. Experimental Results For Different Attribute Sizes

Attribute Size	**SVM**	**LR**	**NB**	**RF**	**ANN**
250	89,15	89,34	84,07	89,23	89,32
500	90,64	90,70	83,24	90,89	90,84
1000	91,47	91,56	83,90	91,31	91,53
1500	91,77	91,56	81,15	91,39	92,18

In Table IV, the test comes about for compared representation plans on five supervised learning methods have been displayed. Because it can be observed from the empirical comes about recorded in Table IV, the lowest predictive performances have been for the most part gotten by trigram-based representation models (i.e., trigram term-frequency, trigram term-presence and trigram TF-IDF). The most noteworthy predictive performance among all the compared arrangements has been gotten by unigram TF-IDF based content representation scheme in conjunction with artificial neural systems, with a classification accuracy of 92.18%.

To summarize the most findings of the experimental investigation, we show fundamental impacts plot for representation plans and classifiers, distinctive include sizes and content representation models in Figure 1, 2 and 3, individually.

TABLE IV. EXPERIMENTAL RESULTS FOR COMPARED CONFIGURATIONS

Representation	SVM	LR	NB	RF	ANN
Unigram, Term-frequency	88,29	91,30	70,95	91,68	92,01
Unigram, Term-presence	89,70	91,03	70,42	91,32	91,58
Unigram, TF-IDF	91,77	91,56	81,15	91,39	92,18
Unigram-Bigram, Term-frequency	88,23	91,22	74,41	91,53	91,89
Unigram-Bigram, Term-presence	89,86	91,26	74,50	91,53	91,49
Unigram-Bigram, TF-IDF	91,89	91,77	85,24	91,77	91,95
Bigram, Term-frequency	81,79	84,51	66,51	84,11	84,84
Bigram, Term-presence	82,48	84,62	65,40	83,51	84,64
Bigram, TF-IDF	84,49	84,84	72,72	84,26	84,87
Unigram-Bigram-Trigram, Term-frequency	88,13	91,24	73,50	91,37	91,93
Unigram-Bigram-Trigram, Term-presence	89,84	91,12	73,97	91,56	91,58
Unigram-Bigram-Trigram, TF-IDF	91,89	91,79	84,82	91,62	91,89
Trigram, Term-frequency	69,98	71,63	59,51	71,09	72,01
Trigram, Term-presence	69,63	71,38	59,07	70,74	71,61
Trigram, TF-IDF	71,43	71,95	61,20	70,97	72,03

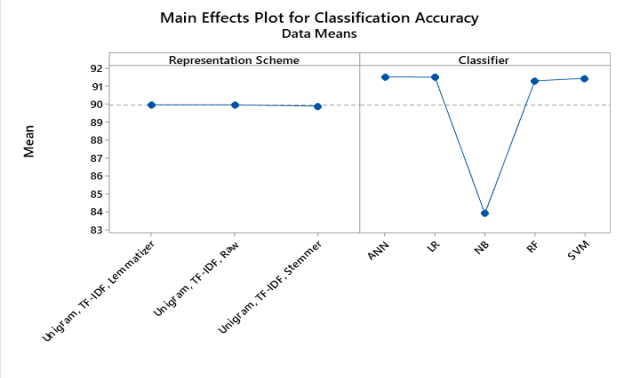


Fig. 1. Main effects plot for classification accuracy based on representation schemes and classifiers

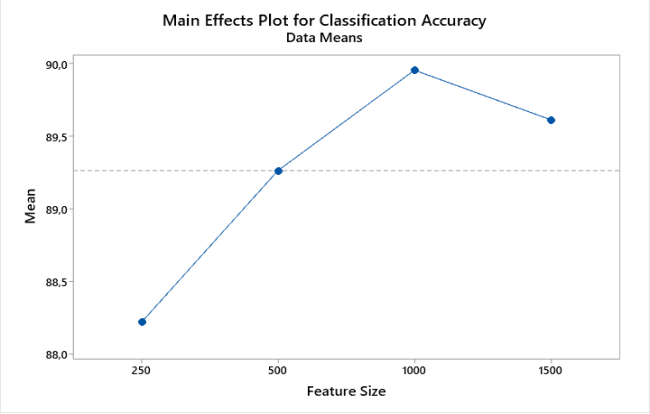


Fig. 2. Main effects plot for classification accuracy based on different attribute sizes

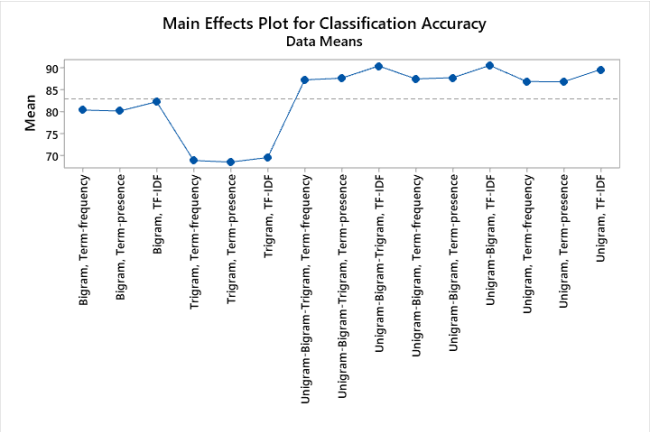


Fig. 3. Main effects plot for classification accuracy based on compared text representation models

V. CONCLUSION

With advances in information technology, users are increasingly generating content, including health information and help, on microblogging sites and on medical forums. A critical function in satisfying patients' needs for health information is the patient-generated content that can be accessed via medical forums. Considering the huge number of questions and the small number of experts, a substantial proportion of the questions remain unanswered. To obtain appropriate answers within a fair period, directing questions and forum posts to the relevant experts will become a key issue, depending on the subject preferences. Medical forums are unique sources of user-generated fitness, health, and wellness knowledge. This unique content offers input and motivation through the personal experience of the individual, which is not available anywhere else. Owing to the incidence of hypertension and elevated cholesterol worldwide, there are a range of web-based questions related to these diseases. In this paper, we intend to classify medical forum posts on hypertension and cholesterol based on conventional text representation models and classification algorithms. In this scheme, we used three types of attributes (i.e., unigrams, bigrams, and trigrams) and three types of word representations (i.e., term-frequency, term-presence and TF-IDF). Furthermore, the ensemble combinations of various attribute sets and representations have been considered. In this way, nine representation models were obtained. Our results indicate that, in combination with artificial neural networks, the unigram and TF-IDF based representation scheme yields promising results with a classification accuracy of 92.18%.

In the future, we plan to increase the size of the corpus. We aim to analyze the predictive performance of ensemble learning methods and deep learning architectures on the corupus.

REFERENCES

[1] M. del Pilar Salas-Zárate, G. Alor-Hernández, J. L. García-Alcaraz, L. O. Colombo-Mendoza, M. A. Paredes-Valverde, and J. L. Sánchez-Cervantes, "A Sentiment Analysis Method for Analyzing Users Opinions About Drugs for Chronic Diseases," in Data Analysis and Optimization for Engineering and Computing Problems, Springer International Publishing, 2020, pp. 217–228.

[2] C. C. Yang and X. Tang, "Estimating User Influence in the MedHelp Social Network," IEEE Intell. Syst., vol. 27, no. 5, pp. 44–50, Sep. 2012, doi: 10.1109/mis.2010.113.

[3] J. Stamler, “Blood pressure and high blood pressure. Aspects of risk.,” Hypertension, vol. 18, no. 3_Suppl, pp. I95–I95, Sep. 1991, doi: 10.1161/01.hyp.18.3_suppl.i95.

[4] P. K. Whelton, “Epidemiology of hypertension,” The Lancet, vol. 344, no. 8915, pp. 101–106, Jul. 1994, doi: 10.1016/s0140-6736(94)91285-8.

[5] J. He and P. K. Whelton, “Epidemiology and preventation of hypertension,” Medical Clinics of North America, vol. 81, no. 5, pp. 1077–1097, Sep. 1997, doi: 10.1016/s0025-7125(05)70568-x.

[6] P.M.Kearney, M. Whelton, K. Reynolds, P. K. Whelton and J. He, “Worldwide prevalence of hypertension: a systematic review,” Journal of hypertension, vol. 22, no. 1, pp. 11-19, 2004.

[7] P. Barter et al., “HDL Cholesterol, Very Low Levels of LDL Cholesterol, and Cardiovascular Events,” N Engl J Med, vol. 357, no. 13, pp. 1301–1310, Sep. 2007, doi: 10.1056/nejmoa064278.

[8] Q. Xu, S. C. Chia, J.-H. Lim, Y. Li, B. Mandal, and L. Li, “MedHelp: enhancing medication compliance for demented elderly people with wearable visual intelligence,” Sci Phone Appl Mob Devices, vol. 2, no. 1, Apr. 2016, doi: 10.1186/s41070-016-0006-5.

[9] T. Ali, D. Schramm, M. Sokolova, and D. Inkpen, “Can I hear you? Sentiment analysis on medical forums,” in *Proceedings of the sixth international joint conference on natural language processing* (pp. 667-673).

[10] M.Sokolova, and V. Bobicev, “What sentiments can be found in medical forums?,” in *Proceedings of the International Conference Recent Advances in natural language processing* (pp.633-639).

[11] J. Carrillo-de-Albornoz, J. Rodríguez Vidal, and L. Plaza, “Attribute engineering for sentiment analysis in e-health forums,” PLoS ONE, vol. 13, no. 11, p. e0207996, Nov. 2018, doi: 10.1371/journal.pone.0207996.

[12] V. Gopalakrishnan and C. Ramaswamy, “Patient opinion mining to analyze drugs satisfaction using supervised learning,” Journal of Applied Research and Technology, vol. 15, no. 4, pp. 311–319, Aug. 2017, doi: 10.1016/j.jart.2017.02.005.

[13] F. Liu, L.D. Antieau, and H. Yu, “Towards automated consumer question answering: automatically separating consumer questions from professional questions in the healthcare domain,” Journal of Biomedical Informatics, vol. 44, no. 6, pp.1032-1038, 2011.

[14] K.Roberts, L.Rodriguez, S. E. Shooshan, and D. Demmer-Fushman, “Resource classification for medical questions,” in *Proceedings of AMIA Annual Symposium* (pp.1040-1046).

[15] R. Jalan, M. Gupta, and V. Varma, “Medical Forum Question Classification Using Deep Learning,” in Lecture Notes in Computer Science, Springer International Publishing, 2018, pp. 45–58.

[16] E. Loper and S. Bird, “NLTK,” presented at the the ACL-02 Workshop, 2002, doi: 10.3115/1118108.1118117.

[17] M. A. Toçoğlu, “Sentiment Analysis for Software Engineering Domain in Turkish,” Sakarya University Journal of Computer and Information Sciences, Dec. 2020, doi: 10.35377/saucis.03.03.769969.

[18] M. A. Toçoğlu and A. Onan, “Satire Detection in Turkish News Articles: A Machine Learning Approach,” in Communications in Computer and Information Science, Springer International Publishing, 2019, pp. 107–117.

[19] C. Cortes and V. Vapnik, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1023/a:1022627411411.

[20] M. Kantardzic, “Data Mining.” Wiley, Oct. 17, 2019, doi: 10.1002/9781119516057

[21] A. Onan, “Classifier and attribute set ensembles for web page classification,” Journal of Information Science, vol. 42, no. 2, pp. 150–165, Jun. 2015, doi: 10.1177/0165551515591724.

[22] A. Onan, S. Korukoğlu, and H. Bulut, “Ensemble of keyword extraction methods and classifiers in text classification,” Expert Systems with Applications, vol. 57, pp. 232–247, Sep. 2016, doi: 10.1016/j.eswa.2016.03.045.

[23] L. Breiman, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/a:1010933404324.

[24] F. Eshragh, M. Pooyandeh, and D. J. Marceau, “Automated negotiation in environmental resource management: Review and assessment,” Journal of Environmental Management, vol. 162, pp. 148–157, Oct. 2015, doi: 10.1016/j.jenvman.2015.07.051.

[25] M. Negnevitsky, “Artificial intelligence: a guide to intelligent systems”, Addison-Wesley, 2005.

[26] A. Onan, “On the Performance of Ensemble Learning for Automated Diagnosis of Breast Cancer,” in Advances in Intelligent Systems and Computing, Springer International Publishing, 2015, pp. 119–129.

Towards a Privacy preserving Machine Learning based Access Control for the Internet of Things

Aissam Outchakoucht
LRI Laboratory, ENSEM School
Hassan II University, Morocco
IPI, Paris, France
aissam.outchakoucht@gmail.com

Hamza Es-Samaali
LRI Laboratory, ENSEM School
Hassan II University, Morocco
IPI, Paris, France
hamza.essamaali@gmail.com

Oussama Mounnan
LabSI Laboratory
Ibn Zohr University, FSA
Agadir, Morocco
oussama.mounnan@gmail.com

Anas Abou El Kalam
Cadi Ayyad University
ENSA School, Marrakech, Morocco
a.abouelkalam@uca.ac.com

Siham Benhadou
LRI Laboratory, ENSEM School
Hassan II University, Morocco
benhadou.siham@gmail.ma

***Abstract*— The Internet of Things (IoT) is pushing us towards adopting original security approaches to manage the huge amount of data they produce while respecting its computing and storage constraints. For that matter, this paper endorses getting the most out of machine learning (ML) algorithms as a means to build smart and dynamic Access Control (AC) solutions for the IoT. Unfortunately, ML requires abundant amount of data for training, whereas a large part of the data exchanged by IoT devices is private, hence this paper is offering a way out to reconcile these two concepts through inter-organizational differential privacy. The proposed solution is backed up by an implementation and shows promising results compared to popular AC models and standards.**

Keywords—internet of things, privacy, machine learning, access control, security

I. INTRODUCTION

The Internet of Things (IoT) offers unprecedented technological opportunities. However, controlling access to billions of devices remains a major obstacle to the large-scale adoption of this technology. In this regard, this paper will attempt to endorse exploiting machine learning (ML) techniques in Access Control (AC) given their effectiveness when it comes to abundant amounts of data, by proposing an enhancement to those kinds of solutions with privacy preserving modules.

In fact, by design, IoT is a network that could not be efficient under centralized architectures. One solution to this is to move towards models that support collaboration between their stakeholders. However, a large part of the data exchanged by IoT devices falls under the category of sensitive or personal data, in short, they must remain private. So how can we use this data for training and improving AC models while respecting privacy?

Moreover, there are many IoT intrinsic characteristics which require smart and dynamic management instead of traditional and impractical one; we believe that the IoT should benefit from those characteristics always considered as weaknesses. In order to do so, our research has confronted many questions that we have tried to answer in this paper, namely:

- The immense number of IoT devices, along with the huge amount of data generated by those devices.
- The majority of AC models offer static mechanisms that go against the dynamic nature of the IoT
- Often AC models are either too complex or centralized, in both cases their adoption in IoT environments is inappropriate.
- In models that propose collaboration between different entities, the issue of privacy is rarely discussed.
- Lots of models lack a proof of work implementation, therefore an empirical evaluation of their feasibility.
- The absence of holistic frameworks that go beyond simply defining board policies or enforcement mechanisms to understanding the context of each device and continuously improving those policies, without falling into the trap of static management or role explosion.

In the following section we will discover the importance of collaboration through some of the researches that have tried to address this issue. We will focus more on solutions offering a dynamic and smart AC policy management, particularly one of our recent works in this area. Then, in Section III we will address the issue of privacy protection when collaborating across organizations to build up broader and more accurate ML models. Next, section IV provides the -theoretical and technical- details of our implementation which are applied to a smart city case study, before moving to section V where we discuss and evaluate the results compared to the most popular AC models and standards. Finally, we'll present some open perspectives and conclusions in section VI.

II. COLLABORATION IN DISTRIBUTED ENVIRONMENTS

Several research studies [1][2] have confirmed that centralized approaches are far from practical when dealing with IoT systems. In this section, we explore the concept of collaboration and how to benefit from it to answer these problems.

A. Related works

In a previous work [3] we have thoroughly studied a large spectrum of AC models and standards as well as their usage in the IoT world; In this paper we will focus mainly on OrBAC-based models because of the advantages they present for the IoT. In fact, Organization Based Access Control (OrBAC) [4] provides the concept of organization by design,

which means that there is no need to add extra dimensions to refer to the owner or policy maker of the IoT devices. Moreover, OrBAC is distinguished by an advanced level of abstraction required to reduce the complexity produced by the colossal number of devices. Not to mention the key component for dynamic and smart decisions: the context; it is embodied in every rule of OrBAC, which will facilitate the collection of contextual information in real time from the end nodes, thus leading to better AC decisions.

Actually, OrBAC is one of the richest AC models in terms of internal modules and applicability to many realistic situations. It presents an original dimension, namely the concept of organization; moreover, it makes a clear distinction between the abstract level (role, view, activity) and the concrete one (subject, object, action).

However, one of the major drawbacks of this model, especially when it comes to IoT environments, is that it is based on a totally centralized architecture and does not provide or support distribution, collaboration and interoperability. Nevertheless, several studies have tried to extend this model to meet these needs.

a) Poly-OrBAC [5] : An extension of OrBAC in which the authors have introduced two new concepts to both remain faithful to the OrBAC notation, and at the same time integrate the collaboration aspect:

- Virtual User: a mold that the organization owning the object will use for external roles.
- Service image: used to represent external objects within the organization providing the roles.

Poly-OrBAC uses electronic contracts to establish collaboration terms, conditions, and penalties, and then uses web services to ensure the exchange of heterogeneous services and data.

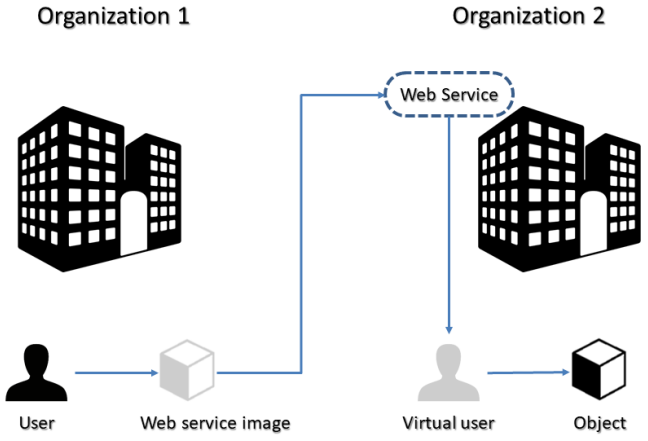


Fig. 1 Collaboration according to Poly-OrBAC

b) Organization-to-Organization: O2O [6] uses a totally different approach. The basic idea is the introduction of a Virtual Private Organization (VPO) that encompasses both organizations interested in collaboration. The resources engaged in a collaboration will therefore be considered as internal resources of this virtual organization.

Since the AC policy of the VPO is only a union of the policies of the two collaborating organizations, O2O presents many drawbacks regarding coherence (which will be amplified in the context of the IoT). Not to mention the lack of details concerning the negotiation stage, which remains an important step in the implementation of such a model [7].

c) Distributed Integrity-OrBAC: I-OrBAC [8] is an extension of OrBAC that focuses on the integrity aspects. DI-OrBAC reinforces this model by making it collaborative.

In this model, the authors have combined the strengths of the two aforementioned propositions. On the one hand, DI-OrBAC integrates the concept of virtual organization while at the same time trying to integrate conflict resolution steps. On the other hand, it gets rid of the centralized approach through electronic contracts for the negotiation about the terms of use.

d) Pervasive Based Access Control: PerBAC [1] is an AC model for IoT environments. It also offers a collaboration component based on three types of contracts: *a public contract* with a minimum number of conditions (if any), often used for public resources; *a private contract* with terms predefined by the object owner, and *a custom contract* that offers more flexibility in negotiating terms of use.

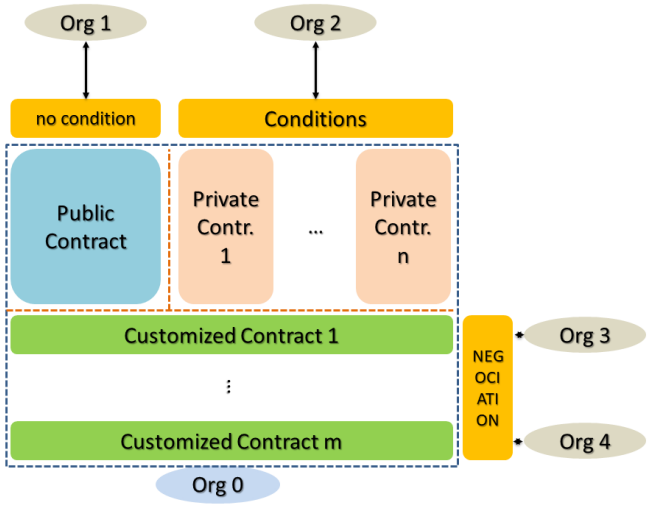


Fig. 2 Collaboration according to PerBAC

B. ML based solution

In a former paper [3], we tackled the issue regarding collaboration by means of attributes, learning and prediction. In fact, by using prediction the model was able to handle queries for which there was no corresponding rule in the AC policy, this has been achieved by extracting the logic behind that policy through machine learning techniques. So as not to go into details, the framework could be segmented into three main phases: *(i)* Pre-request tasks, which focus on defining AC policies; *(ii)* Request processing, which includes all actions triggered by an access request until the subject receives an authorization/rejection in return; *(iii)* Post-request actions, which are responsible for policy learning and improvement.

Basically, this framework uses learning to provide three functionalities that we consider of a great importance in IoT environments.

- *Prediction*: When the request contains some fields that the system has never processed, and therefore are not taken into consideration when defining the

AC policy, the Policy Decision Point (PDP[a]) uses machine leaning algorithms to train on the basis of existing data and then predicts the decision in question. Something that could not be done in a classical management of AC requests.

- *Dynamicity*: By adding the probability of confidence regarding every decision, this model offers the possibility to update them according to previous experiences.
- *Exploration*: This property consists in exploring additional new features to improve the accuracy of future predictions. It is mainly useful when we activate the collaboration aspect, making use of experiences from other organizations.

It's worth noting that the proposed framework is decentralized. In fact, the concept of organization is introduced to decompose complex IoT environments into reasonable and manageable groups, not to transform them into a giant centralized whole. For example, if we need to manage AC in a smart city situation, the framework will treat this multidimensional platform as a set of organizations interacting and collaborating with each other.

Nevertheless, the exchange of information between organizations during the collaboration phase has a major drawback that may slow down, or even block, the adoption of this feature, especially in IoT environments: *Privacy*. In the next section we explore this issue in more depth.

III. PRIVACY PRESERVING MACHINE LEARNING

We are supposedly living in the golden age of data. Still, protecting one's anonymity and privacy is one of the major constraints blocking the sharing of useful data for the common good of all stakeholders. Since we have seen the importance of collaboration in the IoT, we devote this section to studying how to preserve privacy while sharing data between different organizations for the sake of training.

In a study based on U.S. Census data [9], 87% of the U.S. population can be uniquely identified by three pieces of information: date of birth, gender and zip code. Basically, a rule of thumb that should be retained from this example, and many others [10] is: Anonymized data isn't. The problem of privacy cannot be solved by simply hiding some information from the database, let's see why and how to address it.

A. K-anonymity

Let's suppose that we have a database on some organizations that have agreed to share their experiences in order to improve their respective AC systems. Indeed, each of these organizations implements its version of the AC model and carries out its training phase based on the parameters it deems relevant.

Obviously, performing training on the basis of parameters that have already shown their effectiveness is a big step towards more efficient training and therefore more accurate predictions. However, sometimes organizations while willing to share this information may want to remain anonymous. Actually, sharing a feature implies that the organization in question has used it to train its system, this is something that not every organization would want to share.

The following example illustrates that even by deleting the names, anyone who knows that *Organization E* was founded in 1962 and the fact that it contributed to this dataset could infer which feature it employed for training, as the date of establishment (or a combination of the attributes for that matter) uniquely identifies it in this database.

TABLE I FICTITIOUS DATABASE ABOUT SOME COMPANIES

Name	**date of establish-ment**	**Nature of the company**	**Head Quarters ZIP Code**	**Works in ML domain**	**Shared feature**
Org_A	1965	Public	1729-1	Yes	Theft
Org_B	1964	Private	1729-2	No	Accident
Org_C	1967	Public	1729-3	Yes	Weather
Org_D	1969	Private	1729-4	No	Max speed
Org_E	1962	Public	1729-5	Yes	Theft
Org_F	1951	Private	1729-6	Yes	Delay
Org_G	1959	Public	1729-7	No	Auto pilot
Org_H	1953	Public	1729-8	No	Delay
Org_I	1959	Private	1729-9	Yes	Auto pilot

One solution to this problem is a technique, called k-anonymity, which involves the redaction of individual records so that no set of characteristics corresponds to a single data record. The purpose of k-anonymity is to make it difficult to link sensitive (shared features) attributes to insensitive (the rest) ones.

There are two main ways to eliminate information from a table to make it k-anonymous: we can remove the information entirely, or we can magnify it.

TABLE II FICTITIOUS DATABASE ABOUT SOME COMPANIES - ANONYMOUS VERSION

Name	**date of establish-ment**	**Nature of the company**	**Head Quarters ZIP Code**	**Works in ML domain**	**Shared feature**
*	1960-70	Public	1729**	Yes	Theft
*	1960-70	Private	1729**	No	Accident
*	1960-70	Public	1729**	Yes	Weather
*	1960-70	Private	1729**	No	Max speed
*	1960-70	Public	1729**	Yes	Theft
*	1950-60	Private	1729**	Yes	Delay
*	1950-60	Public	1729**	No	Auto pilot
*	1950-60	Public	1729**	No	Delay

[a] Key component of XACML, responsible for providing the access decision

*	1950-60	Private	1729**	Yes	Auto pilot

In Table II, names have been removed, dates of establishment and zip codes have been “blurred”. The result is that the table is now anonymous (formally, it’s called 2-anonymous) in the sense that there is no combination leading to exactly one organization. Indeed, "a 1966 ML based public company" now corresponds to three different entries. Thus, an organization's record cannot be uniquely re-identified from his insensitive information.

However, although k-anonymity may prevent the re-identification of records in the strict sense, it must be emphasized that re-identification is neither the only nor the main risk to privacy. Indeed, in previous table if we know that *Org_E* is a “1966 ML based public company”, of course we cannot identify the exact feature it had sent, yet we can know for sure that it is either “Theft” or “Weather”, a serious invasion of privacy. But the concept of k-anonymity also suffers from an even more serious and subtle problem, namely that its guarantees disappear entirely when several datasets are published, even if all were published with a k-anonymity assurance [10].

B. *Differential Privacy*

Differential Privacy (DP) [11] is built upon the idea that we had better be comparing what an attacker might learn from the data if any particular entry was included in this dataset with what he might learn if it was not.

Usually by adding some noise before or after data aggregation, DP protection ensures that adding or deleting data from a single individual does not change the likelihood of a result in any significant way. Therefore, DP is a mathematical assurance highly demanded for the collection and sharing of aggregated information about users while retaining the individual information of each user.

In fact, machine learning algorithms we use nowadays depend on the amount of data they are provided with during the training phase. Therefore, and to make the framework we summarized in section II-B effective in real world IoT scenarios we need to leverage data not only from one organization, but from many. But in order for those organizations to collaborate and share their data and results, which are often sensitive, they must demand a guarantee of an anonymous processing.

In this regard, we opted for techniques that ensure DP during the collaboration phase. In our implementation, we used randomized response algorithm to guarantee this property.

IV. IMPLEMENTATION

Although the phases and components of the model have been elucidated earlier, it is of paramount importance to have a real implementation of those theoretical notions. To achieve this, we have developed a Proof of Concept based on a case study of a typical IoT environment, namely a Smart City (SC).

In fact, the SC has always been the intuitive case to illustrate this vision where devices not only interact but depend on (or even control) each other. This section mainly focused on the collaboration phase in a SC situation to illustrate how our previously explained framework could be implemented in such a complex environment. Actually, we took the case of an autonomous car rental agency (CRA) as our primary organization. All the details and codes used during this implementation are available in open source on GitHub [12].

As we mentioned in section II-B as well as in the example provided in III-A, in the exploration phase, the organization in question introduces other features in the learning phase in order to explore and study their impact on the prediction.

In our implementation, and after training with a CatBoost model [13], we discovered other relevant features that we can use afterwards to improve the efficiency of our AC policy, including “previous accidents” (of the car brand, of a specific person or under certain conditions) and “weather”. We can even visualize the contribution of each of these features to the accuracy of the model, as shown in Figure 3.

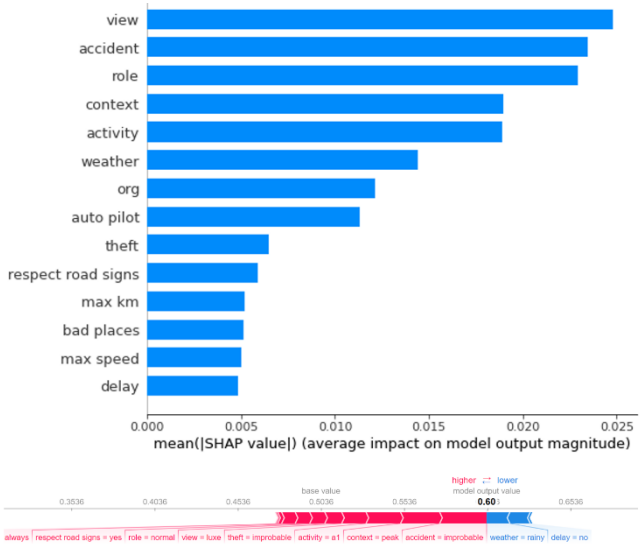


Fig. 3 Features importance and their contribution to the predictions

In order to ensure DP when exchanging this type of information between organizations, we developed an adapted randomized response (RR) implementation. Suppose you conduct a survey, to protect the anonymity of your participants, RR suggests that they never send correct information 100% of the time. When you ask a participant for an answer, she tosses a coin, for Tails she tells the truth, while if the coin settles on Heads, the candidate gives a random answer (tosses the coin a second time, tells the truth for Tails and lies for Heads).

Note that RR actually satisfies an even stronger constraint than DP since this latter only requires that no one is able to guess much better than randomly whether a given result was produced with or without a specific data entry. However, RR promises not only that the final average is differentially private, but also that the entire data set is. Furthermore, with RR we do not need to trust a central entity since each participant is noising her own answer before sending it in.

Here is the pseudo code that details how that algorithm is adapted to our case study:

```
Input     : truth
Output    : response
Algorithm :

1.  SET threshold AND columns
2.  IF (truth IN columns) THEN
3.     rand ← random number between 0 and 1
4.     IF (rand <= threshold) THEN
5.        response ← truth
6.     ELSE
7.        REMOVE truth FROM columns
8.        response ← random choice FROM columns
9.     ENDIF
10. ELSE
11.    DISPLAY("error: the input must be in columns")
12. ENDIF
```

Fig. 4 Pseudocode of randomized response algorithm

Let's say that during the exploration phase, our CRA finds a new relevant feature and it wants to share it with the other CRAs without being disclosed as the source of this information: We start by setting a threshold (say 0.8) and a set *S* of parameters (the names of the possible features in this type of experiment), then we take the real answer (the name of the feature that a given organization finds relevant). The algorithm keeps the true answer by the probability of 0.8 (threshold) and selects a random response from *S* otherwise.

Thanks to this protocol, organizations could share data for the benefit of everyone without worrying about disclosing their identities.

Finally, it is important to mention that the execution of these algorithms does not increase the response time during an access request. This exploration phase is executed after answering access requests.

V. Results & Discussion

In this section we discuss how our proposition has addressed some access control requirements in IoT environments as described in the introduction of this paper. Table III below summarizes these solutions.

TABLE III OUR MODEL'S SOLUTIONS TO AC PROBLEMS IN IOT

Problem	Proposed solutions
Huge amount of data	Intelligent processing (with ML)
Low computing and storage capacity of IoT objects	Multi-layer architecture Algorithms adapted to each layer Subsequent treatment Not using ML when it is not the best solution
Large number of IoT devices	OrBAC and abstraction layers Prediction in case of new requests
Centralization	Collaboration
Static models	Use of attributes and contexts Optimization of the AC policy The exploration phase
Privacy	Differential privacy protection and randomized response algorithms
Language's rigor	Based on OrBAC and XACML
Proof of work	Case study and implementation

The following table IV compares our proposition to the most popular AC models and standards according to different metrics.

TABLE IV COMPACTED COMPARISON OF OUR MODEL AND OTHER AC MODELS AND STANDARDS

Tests / Model	R BAC [14]	A BAC [15]	Or BAC [4]	XAC ML [16]	Our model
Simplicity	✓				
Multi-layer Architecture			✓		✓
Granularity		✓		✓	✓
Collaboration					✓
Adaptability to the IoT		✓	✓		✓
Privacy					✓
Machine Learning					✓
Proof of Work	✓	✓	✓	✓	✓
Large scale implementation	✓	✓		✓	

The most popular models and standards we studied, namely RBAC, ABAC, OrBAC and XACML, have all proven their feasibility and been implemented, some of them even on a large scale; yet, despite being set up, the implementation of our solution remains a Proof of Work. As for collaboration, machine learning or privacy, except for ours, none of the aforementioned models offers an integrated tool to provide these functionalities.

Clearly, in terms of complexity RBAC is by far the simplest, it provides optimal execution time thanks to its lighter conception compared to the other models. On the other hand, the raw use of attributes without any type of aggregation in ABAC makes this model one of the most complex. Right in the middle stands the proposition of this paper, a holistic framework - which we call DEPPAC, for Dynamic, Explorer, Predictor and Privacy preserver Access Control. In fact, DEPPAC is relatively complex since it is based partly on ABAC; however, the use of abstraction layers and prediction contribute largely to noticeable improvements in response time.

Regarding the adaptability to the IoT, RBAC does not offer a wide collection of abstract entities apart from roles, which leads to the problem of role explosion, not to mention the lack of contextualization. Actually, ABAC might appear better suited for IoT, nonetheless despite the importance of attributes in the IoT context, the complexity of ABAC and the lack of aggregation layers make its implementation very challenging. As for OrBAC, the abstraction layers and the dimension of context it introduces qualify it as a good candidate for the IoT, nevertheless it remains a centralized model by design, and it does not offer any learning mechanism. On the other hand, when we study XACML closely we can see that its popularity as an AC standard and the rigor with which its language is written could be useful in solving problems of heterogeneity and incompatibility in the IoT, however, as the other models it lacks abstraction and learning paradigms. On the contrary, DEPPAC is oriented -by design- towards IoT environments. It provides a multi-level architecture and integrates context and learning mechanisms. Moreover, the introduction of the paradigm of collaboration frees it from the centralized aspect of OrBAC.

That being said, we can affirm that we have addressed all the AC requirements that we presented at the beginning of our study, which makes our proposition an advantageous

candidate in IoT environments compared to other AC solutions.

In addition, we believe that there are some very active lines of research that can push further the adoption of our framework and increase its effectiveness, such as transfer learning [17], federated learning [18], Generative Adversarial Networks (GANs) [19] for generating training data, as well as advances in software and hardware [20].

In the light of the foregoing, we believe that endowing the proposition we exposed in II-B with the kind of private collaboration we outlined so far would result in a holistic framework -DEPPAC- for controlling accesses in IoT networks. This solution is the result of the interference between deterministic and ML-based systems, crowned by the privacy aspect of inter-organizational information sharing.

VI. Conclusion

In this paper we have explored a new proposal to provide intelligent and privacy-friendly management of access control in the context of the Internet of Things. This solution, called DEPPAC, provides smart and dynamic processing (using Machine Learning) to manage the huge amount of data generated by IoT devices. In addition, the limitations due to low computing and storage capacity were managed through the multi-level architecture and post-decision processing. Furthermore, the large number of IoT devices that was always considered a major obstacle to securing this type of environments has been alleviated by the introduction of abstract layers, collaboration and the concept of prediction in case of new queries. Last but not least, we thoroughly discussed in previous sections how to endow it with differential privacy protection through randomized response algorithm.

The proposal was supported by an implementation applied to a case study of an Internet of Things environment. Technical details and codes are deployed in the GitHub under an open source license. The implementation was consolidated by an evaluation of DEPPAC against reference AC models and standards.

Besides, we believe that there are some very promising research fields that could stimulate and speed up the wide spread adoption of frameworks like ours. Take Transfer Learning for example, it refers to when a Machine Learning model is developed for a task, but is then reused as a starting point for a new task, it serves to quickly and effectively address new but similar problems. It reduces the need for data related to the specific task as well as the time and resources needed to learn from scratch.

Another very active line of research these days is Federated Learning. It has been proposed by Google and its main idea is to create ML models based on datasets that are distributed across multiple devices (mainly smartphones) while protecting the privacy of the creators of the data. Not to mention the advances in hardware and how the giants of technology (Google, NVIDIA, Intel, ...) are now competing to produce devices that minimize the size/efficiency ratio. The competition is also fierce in the software sphere, whether in the academic or industrial fields, researchers are pushing algorithms to be increasingly optimized so that they could be embedded in constrained objects.

References

[1] S. El Bouanani, M. A. El Kiram, O. Achbarou and A. Outchakoucht, "Pervasive-Based Access Control Model for IoT Environments," in IEEE Access, vol. 7, pp. 54575-54585, 2019.

[2] I. Bouij-Pasquier, A. A. El Kalam, A. A. Ouahman, and M. De Montfort, “A Security Framework for Internet of Things,” in CANS, 2015.

[3] A. Outchakoucht, A. A. E. Kalam, H. Es-Samaali and S. Benhadou, “Machine Learning based Access Control Framework for the Internet of Things” International Journal of Advanced Computer Science and Applications (IJACSA), 11(2), 2020.

[4] A. A. E. Kalam et al., "Organization based access control," Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, Lake Como, Italy, 2003, pp. 120-131.

[5] A. Abou El Kalam, Y. Deswarte, A. Baina, and M. Kaaniche, “PolyOrBAC: A security framework for Critical Infrastructures,” Int. J. Crit. Infrastruct. Prot., vol. 2, no. 4, pp. 154–169, 2009, doi: 10.1016/j.ijcip.2009.08.005.

[6] F. Cuppens, N. Cuppens-Boulahia, and C. Coma, “O2O: Virtual private organizations to manage security policy interoperability,” Inf. Syst. Secur., pp. 101–115, 2006, doi: 10.1007/11961635_7.

[7] Abdeljebar AMEZIANE EL HASSANI, “Contrôle d’accès dans les systèmes distribués à grande échelle,”, PhD thesis, 2018.

[8] A. Ameziane El Hassani, A. Abou El Kalam, A. Bouhoula, R. Abassi, and A. Ait Ouahman, “Integrity-OrBAC: a new model to preserve Critical Infrastructures integrity,” Int. J. Inf. Secur., vol. 14, no. 4, pp. 367–385, 2014.

[9] L. Sweeney, “k-anonymity: A model for protecting privacy,” Int. J. Uncertainty, Fuzziness Knowlege-Based Syst., vol. 10, no. 5, 2002.

[10] A. Kearns, M. and Roth, “The ethical algorithm: the science of socially aware algorithm design”, Oxford University Press, 2019.

[11] Abadi, M. et al. “Deep Learning with Differential Privacy.” Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[12] A. OUTCHAKOUCHT, “Car Access: An implementation of the framework DEPPAC,” 2020. [Online]. Available: https://github.com/aissam-out/Car_Access.

[13] Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin, “CatBoost: gradient boosting with categorical features support”, Workshop on ML Systems at NeurIPS, 2017.

[14] R. S. Sandhu, “Role-based Access Control,” Adv. Comput., 1998, doi: 10.1016/S0065-2458(08)60206-5.

[15] V. C. Hu et al., “Guide to Attribute Based Access Control (ABAC) definition and considerations,” NIST Spec. Publ., 2014, doi: 10.6028/NIST.SP.800-162.

[16] OASIS Standard., “eXtensible Access Control Markup Language (XACML) Version 3.0,” 2013.

[17] K. Weiss, T. M. Khoshgoftaar, and D. A. Wang, “survey of transfer learning”. J Big Data 3, 9, 2016.

[18] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated Optimization: Distributed Machine Learning for On-Device Intelligence,” CoRR, vol. abs/1610.0, 2016.

[19] Goodfellow, I. J., “Generative Adversarial Networks”, arXiv e-prints, 2014.

[20] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” IEEE Commun. Surv. Tutorials, 2019, doi: 10.1109/COMST.2019.2904897.

Evaluation of Outlier Algorithms for Anomaly Detection

Pınar Ersoy
Technology
Commencis
Istanbul, Turkey
pinar.ersoy@commencis.com

Mustafa Erşahin
Software Development
Commencis
Istanbul, Turkey
mustafa.ersahin@commencis.com

Deniz Kılınç
Computer Engineering Department
Izmir Bakırçay University
İzmir, Turkey
deniz.kilinc@bakircay.edu.tr

***Abstract*— Anomaly detection is a critical problem that has been researched within diverse research areas and application disciplines. This article aims to construct a structured and comprehensive overview of the selected algorithms for anomaly detection by targeting data scientists, data analysts, and machine learning specialists as an audience. In total, nine outlier detection algorithms were introduced. Using a real-time streaming anomaly detection dataset called Numenta Anomaly Detection Benchmark (NAB), we selected two similar datasets to perform a more precise comparison. Furthermore, we trained and tested these nine algorithms with Python Outlier Detection (PyOD) package by means of performance and time complexity perspectives. Copula-Based Outlier Detection (COPOD) and Clustering-Based Local Outlier Factor (CBLOF) demonstrated persistently satisfactory performance in both datasets achieving the highest Receiver Operating Characteristic (ROC) scores. Similarly, COPOD and Principal Component Analysis (PCA) recorded the least execution time in seconds in both cases.**

Keywords*— *anomaly detection, outlier prediction, local outlier factor, angle-based outlier detection, clustering-based local outlier factor, copula-based outlier detection, k-nearest neighbor, one-class svm, isolation forest, machine learning

I. INTRODUCTION

An unexpected change that performs highly divergent attitudes from other observations in a time period can be represented as abnormal behavior. In other words, anomaly detection can be defined as the measure of specifying the outliers in the existing dataset which acts considerably different from the rest of the data points by profiling them as non-conforming normal points.

Anomaly detection can be accepted as the way toward discovering exceptions in a dataset. Outlier points are the piece of information which stand apart from other data points and do not adjust to the normal pattern in a dataset. Outlier detection calculations have wide applications in business, logical, and security spaces where secluding and following up on the consequences of outlier identification process is crucial [1].

In the case of having marked deviant points in the existing dataset, the supervised algorithms can be utilized for detecting abnormal points while lacking of labelled anomalous activity information in the dataset, unsupervised outlier detection algorithms may be applied to identify them with distance or density based approaches by considering the neighborhood points [2]. Prior to examining the algorithm approaches, the term outlier or anomaly shall be explained and the logic of that information appeared in a dataset should be perceived.

In this study, we addressed the evaluation of the anomaly detection algorithms from a supervised machine learning perspective. Using a real-time corpus of 58 time-series dataset with a novel scoring mechanism provided by A. Lavin and A. Subutai in their article [3], we identified optimum dataset features that improve prediction accuracy. Moreover, we trained and tested a lot of machine learning models such as Minimum Covariance Determinant (MCD), Principal Component Analysis (PCA), Local Outlier Factor (LOF), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Ordering Points To Identify the Clustering Structure (OPTICS), Isolation Forest iForest) and One-Class Support Vector Machine (SVM) on the datasets that we created from the original dataset via feature selection and transformation.

This paper is organized as follows: Section 2 presents some timely work on anomaly detection algorithms in the literature. Section 3 explains the details of the datasets and machine learning models that we experimented with. In Section 4, we present our empirical findings and discuss them. Finally, Section 5 concludes the study and provides some future directions for further research.

II. RELATED WORK

Anomalous points might be produced by errors in the data; however, it could point out to a historically or currently existing unidentified or hidden process or behavior.

As the publicly available data volume reaches in mass amounts, outlier detecting algorithms are modified to run on these data sets to predict the unusual patterns. For instance, a “suspiciously high” count of login trials might outline a possible cyber intrusion or a considerable increase in incoming network traffic can be pointed to malicious activity in network systems. Considering these activities, they hold a shared aspect that they are “appealing” and “unusual” to the data scientists and data analysts. The “curiosity” or real-life applicability of anomalies is an essential element of anomaly detection.

Comprehending the source of the anomalies is crucial since it will direct what activity shall be performed on the recognized exceptions. Nevertheless, identifying precisely what caused an exception is a repetitive errand and it could be difficult to track down the reasons for anomalies in the dataset. In this section, we present some notable and recent work on the topic. Most of these studies rely on data mining and machine learning techniques because enormous quantities of operational data regarding aviation operations are now possible to collect, store, and process. Besides, applications based on machine learning techniques have already achieved promising performances in remarkably diverse domains like computer vision, natural language processing, medical diagnosis, fraud detection, and so on. Therefore, it is perfectly normal to see many studies that try to utilize machine learning for anomaly detection.

A. Subsets of Anomaly Detection

The outlier detection work of Chandola et.al. [4] proposes a certain number of outlier detection methodologies based on specified groups as point, contextual, and collective anomalies.

1) Point Anomaly: The first one subset highlights a specific item in a dataset which stands largely dissimilar from the rest of the points corresponding to its attributes.

2) Contextual Anomaly: The second subset can be regarded as contextual anomaly. It points out anomalous pattern corresponding to points that belongs to context-based knowledge and it may be frequently examined in time-series data [5], [6]. This sub-group of of anomalous behaviour may not be identified in absence of contextual information.

3) Collective Anomaly: The final subset of anomaly detection can be presented as collective anomalies. In this type, outlying items can be composed of multiple instances of points that may not constitute anomalous behaviour individually.

III. ML-Based Anomaly Detection Approaches

Noticeably varied approaches attempt differing perspectives and frequently diverse assumptions of what explicitly established an outlier. Several types of approaches will be discussed in more detail in this section.

A. Distance-Based Approaches

Distance-based outlier detection is one of the identification approaches in the outlier detection field. The distance-based [14] methodology expects clients to choose appropriate distance, scale features with limited effectiveness on high-dimensional datasets.

1) Minimum Covariance Determinant (MCD): Distinguishing exceptions in multivariate information is not clear. At the point when the measurement surpasses two, visual examination of the information can be misleading since multivariate anomalies may not be obvious on all dimensions. An outlier can be represented as a point with an interval higher than different distance which can classically be identified as Mahalanobis distance [7]. Minimum Covariance Determinant acts as the covariance estimator that is to be applied to Gaussian-distributed data. It scans for the small groups of a elements with the smallest determinant of a covariance matrix. The MCD algorithm tends to learn a rotationally symmetrical shape and works best with elliptically symmetric unimodal distributions [8]. By considering this fact, the proper usage of this algorithm is to apply on a unimodal distributed data. The more the size of the data and unimodality gets lower, the more the performance of the algorithm diminishes.

2) Principal Component Analysis (PCA): The multivariate datasets with their high-dimensionality burden may be displayed as a complex issue to work on to extract their features in a leaner way. Plotting of information to have more insights may be limited to a few dimensional plots with some explanatory depictive approaches. Principal component analysis (PCA) contributes to reduction in the number of features of the data collection by finding a substitute arrangement of directions [9]. A couple of latest investigations showed the ideas of PCA to identify numerous exceptions in multi-dimensional datasets [10], [11]; which point out the algorithm has a quick and vigorous ability to distinguish anomalies.

B. Probabilistic Approaches

Exception recognition alludes to the recognizable proof of uncommon things that are degenerate from the overall data distribution. To cope with detection anomalies, probabilistic approaches consider deterministic models.

1) Angle-Based Outlier Detection (ABOD): Distinguishing exceptions in a mass group of data points is a significant mining technique in targeting various clusters of elements in a data collection. To be able to eliminate the negative impacts of multiple dimensionality effects of any data set, it is crucial to find a viable solution which is free from distance calculations which might be generated from the distance-based approaches in the previous section. Angle-Based Outlier Factor (ABOF) can portray the disparity in routes of elements comparatively to each other with an implication of the corresponding point is situated outside of certain groups of data that are gathered [12]. By considering the definition, anomaly score for this approach can be correlated to the angle between the corresponding data points.

2) Copula-Based Outlier Detection (COPOD): Existing methodologies experience the negative effects of high computational multifaceted nature, low prescient ability, and restricted interpretability. The Copula-Based Outlier Detection can be classified as a probabilistic model with a parameter-free structure which is relieved of training burden with its disparate structure of marginal distributions, remarkably explainable with the dimension-based generated graph for outliers and a reduced computational cost by scaling efficiently for spacious datasets with great number of dimensions [13]. COPOD approach is an emerging algorithm and it is motivated by deterministic functions for demonstrating multivariate variable distribution. Initially, it develops an exact copula, and afterward utilizes it to foresee tail probabilities of each given element to decide its degree of outlier potential.

C. Density-Based Approaches

The density-based anomaly identification techniques can viably tackle the issues of high-dimensionality by portraying the level of anomaly points that are evaluated by nearby frequencies.

1) Local Outlier Factor (LOF): The Local Outlier Factor computes the corresponding density anomaly factor of each element respective to its surrounding points, which is utilized to depict the level of exception in the data [14]. The degree of deviation of the observations computes a score that calibrates the local density of a elements by comparing them to their neighbor points. The rationale is to diagnose the samples that have a descend density than their enclosed points. Precisely, inlier points are allocated by the methodology of k-nearest neighbors with measuring the distance between items. The computation cost of the LOF approach can be high by calculating the LOF estimation of all values, which is burdensome to implement to the anomaly

recognition of the enormous scale of data set [15]. This complexity concern was evaluated with two steps by firstly, considering reducing the dimensions of a dataset to decrease the mathematical complexity and secondly applying an optimized k-nearest neighbors approach to scale down the distance calculation efforts [16] with an advanced version of kd-tree indexing [17].

D. Density-Based Clustering Approaches

Density-based clustering methods discover the clusters by separating them according to their volumes of densities.

1) Clustering-Based Local Outlier Factor (CBLOF): The Clustering-Based Local Outlier Factor (CBLOF) approach attempts to compute the anomaly score positioned on cluster-based local outlier factor. CBLOF accepts a set of elements and the grouped model which was created by one of the clustering approaches with the capability of organizing the groups into clusters with differing sizes by utilizing the parameters of the algorithm [18]. The outlier score can be determined dependent on the size of the cluster. The point assigns to the closest enormous group. An undesired issue might happen as irregular points around smaller-sized groups may not be discovered in case of not weighting for anomaly factor. In such situations, anomaly scores are merely calculated dependent on their distance to the nearest broader sized groups.

2) K-Nearest Neighbors (KNN): K-nearest neighbors (KNN) approach can be described as the kind of machine learning algorithm that computes similarity using the distance metric between two points on a feature space. The nearest neighbor principal rule assigns to an unmarked element to the cluster of the closest of a set of formerly grouped items. In the context of outlier detection, a data point in with its distance to its nearest neighbor can be positioned as the anomaly score which might be underlined as a method to calculate the density which can be applied for both classification as well as regression based analytical problems [19].

E. Isolation-Based Approaches

Isolation-based algorithms identifies anomalies by isolating elements with their nature of being rarity without a need of either distance or density metric calculations.

1) Isolation Forest (iForest): Isolation Forest algorithm is expressed on the idea that outliers represent data points that are few and different with a considerable distance from the cores of the non-outlier clusters [20]. Isolation Forest is connected to tree ensemble method which is generated from decision trees. By using decision trees, subdivisions are created with a incidental selection of a factor followed by a value in the extent of a detailed variable list.

F. Classification-Based Approaches

Classification-based methods can be applied on data which have skewed nature with producing binary groups.

1) One-Class SVM (OCSVM): Apart from the ordinary type of SVMs, one-class SVM endeavors to become familiar with a threshold which accomplishes the most extreme edge between the items and the center point in the high dimensional space [21]. The methodology of One-Class SVM can provide more accurate outputs in higher dimensional datasets. The purpose of this algorithm to diverse the items from the core to magnify the gap between the features to the center.

IV. Materials and Method

In this section, we give the details of the dataset used and the method applied in this study.

A. Dataset

Numenta Anomaly Benchmark (NAB) intends to detect outliers in a stream of data which do not harmonize to previous arrangement of behavior for the defined data stream [29]. NAB dataset endeavors to give a supervised setting to test and monitor the outlier recognition calculations on a group of streaming data. The ideal responsible algorithms synchronize with the streaming timeline-based data over a diversified set of datasets with conforming to evolving metrics.

As the dataset, we use NAB that includes several real datasets with their labels with various fields. In addition, their outlier detection tags can be applied on the streaming data. We use two NAB datasets, “Ambient Temperature System Failure” which consists of the ambient temperature in an office environment, and “Machine Temperature System Failure” which holds temperature sensor data of an internal component of a machine. These datasets were chosen since they both keep temperature values over a timeline that prepare equivalent settings for an algorithm-based comparison. Descriptions of the datasets are depicted in the Table I for Ambient Temperature System Failure with an abbreviation of DS1-ATSF and Table II for Machine Temperature System Failure with an abbreviation of DS2-MTSF.

TABLE I. Structure of the DS1-ATSF dataset

Column name	***Description***	***Number of distinct values***
timestamp	Ordered unique timestamp record	7.267
value	Ambient temperature value with a float64 format for each timestamped record	7.267

TABLE II. Structure of the DS2-MTSF dataset

Column name	***Description***	***Number of distinct values***
timestamp	Ordered unique timestamp record	22.683
value	Machine temperature value with a float64 format for each timestamped record	22.695

The selected datasets hold features of a time-series formatted data with a timestamp and an individual real number as a corresponding value.

B. Data Preprocessing

In this study, we examined the outlier detecting issue from both supervised and unsupervised approaches varies to the presented algorithms. Even though the data sets process the target variable Y, the unsupervised methodologies only use

the X variable. The role for Y variable stands for validation procedure.

As deep dived into data tables, both consist of the columns timestamp and value which hold categorical and numerical values respectively with several distinct values as seen in Table 1, and Table 2. To be able to utilize timestamp feature, we formatted it and saved it as a new variable.

Since the number of variables are too few, we derived some additional variables as hour and day of week by using timestamp feature. The correlation between these variables was tested before the prediction phase. Also, for the contamination parameter, outlier points are added to the datasets with a column called an anomaly.

C. Experimental Setup

There exist several approaches while detecting anomalous behavior in a dataset. Being able to find the correct metrics to compare them in appropriate way is a tough task to accomplish. In this paper, as a first step, we choose datasets that shares similar theme as temperature values associated with a point in time.

The application of the proposed system was developed based on Jupyter Notebook [22] using Anaconda 3 with Conda version 4.9.0. The entire system was executed on a local personal computer, Dell Precision 5530 with 64-bit operating system, x64-based processor. The system was run on a local server with a 256 GB hard disk, Intel(R) Core(TM) i7-8850H CPU, 2.60GHz, 2592 MHz processor having 6 cores, 12 logical processors, and 64 GB (32Gx2) of RAM on Windows 10 Pro with the version of 10.0.19042 as the operating system.

As the major programming language, we used Python 3.7.6 for the entire data preprocessing, model training, and model evaluation phases since it has a broad range of scientific packages that empowers data manipulation with Pandas (version 1.0.1) [23], Numpy (version 1.19.1) [24], and various modeling techniques with Scikit-Learn (version 1.0.3) [25].

The main objective of this paper is to introduce the existing anomaly detection approaches with a wider perspective to highlight their capabilities with respect to their categories. Therefore, the chosen outlier detection methodologies were tested on an already developed anomaly detection Python toolkit which is called as PyOD [26] to be able to evaluate all the algorithms in an officially acknowledged library.

In the context of our experiments, we chose the following algorithms of Angle-Based Outlier Detection (ABOD), Cluster-Based Local Outlier Factor (CBLOF), Copula-Based Outlier Detection (COPOD), Isolation Forest (iForest), K-Nearest Neighbors (KNN), Local Outlier Factor (LOF), Minimum Covariance Determinant (MCD), One-Class Support Vector Machine (OCSVM) and Principal Component Analysis (PCA) for comparison by their existing detector categories. By using those algorithms, we executed the corresponding classifiers with their default parameters in their PyOD implementations. The default parameters of these algorithms are introduced in Table III. For the parameter "contamination", the windowed anomalous labels of the Numenta Dataset were used to calculate the proportion of outliers for each dataset. Besides, we standardized model variables to get more insightful model performance; since the variables in datasets were left-skewed.

TABLE III. DEFAULT PARAMETER VALUES OF PYOD ALGORITHM CLASSIFIERS

Algorithm	***Default parameter values***
ABOD	n_neighbors:10, contamination:0.1, method:'fast'
CBLOF	n_clusters=8, contamination=0.1, clustering_estimator=None, alpha=0.9, beta=5, use_weights=False, check_estimator=False, random_state=None, n_jobs=1
COPOD	contamination=0.1
iForest	n_estimators=100, max_samples='auto', contamination=0.1, max_features=1.0, bootstrap=False, n_jobs=1, behaviour='old', random_state=None, verbose=0
KNN	contamination=0.1, n_neighbors=5, method='largest', radius=1.0, algorithm='auto', leaf_size=30, metric='minkowski', p=2, metric_params=None, n_jobs=1
LOF	n_neighbors=20, algorithm='auto', leaf_size=30, metric='minkowski', p=2, metric_params=None, contamination=0.1, n_jobs=1
MCD	contamination=0.1, store_precision=True, assume_centered=False, support_fraction=None, random_state=None
OCSVM	kernel='rbf', degree=3, gamma='auto', coef0=0.0, tol=0.001, nu=0.5, shrinking=True, cache_size=200, verbose=False, max_iter=- 1, contamination=0.1
PCA	max_depth:3, learning_rate:0.1, n_estimators:100

D. Model Evaluation Metrics

To be able to accurately measure the anomalous point detection both in performance and time of the selected algorithms, we applied the ROC Area metric. ROC Area can be calculated by using the given below formulas with labelled with the abbreviations of True Positive as TP, False Negative as FN, True Negative as TN, and finally False Positive as FP.

$$ROC = 1/2\ (TP / (TP + FP) + TN / (TN + FP)) \quad (1)$$

In addition, the execution times of each algorithm were calculated to be able to use it while evaluating each algorithm in an additional metric. To achieve this, timers were added at the beginning and ending of each algorithm as a starting and ending time values in seconds. After the execution of each algorithm, the time between the end time and the start time was subtracted to create the duration value in seconds.

V. FINDINGS & RESULTS

We trained nine machine learning models and tested them on two experiment datasets by using Python toolkit for outlier detection (PyOD). We present model scores and time metrics for each dataset in Tables 4 and Table 5, respectively. In the tables, highest values for each score and time are emphasized with bold font.

TABLE IV. EXPERIMENT RESULTS FOR DS1-ATSF

Algorithm	***Anomaly Percentage from NAB Labels***	***ROC Score***	***Execution Time (seconds)***
ABOD	0.028	0.4652	0.892
CBLOF	0.028	**0.6531**	0.192
COPOD	0.028	**0.7505**	**0.032**
iForest	0.028	0.5423	0.482
KNN	0.028	0.2195	0.217

Algorithm	*Anomaly Percentage from NAB Labels*	*ROC Score*	*Execution Time (seconds)*
LOF	0.028	0.5957	0.056
MCD	0.028	0.2357	0.836
OCSVM	0.028	0.4886	0.520
PCA	0.028	0.4635	**0.003**

TABLE V. EXPERIMENT RESULTS FOR DS2-MTSF

Algorithm	*Anomaly Percentage from NAB Labels*	*ROC Score*	*Execution Time (seconds)*
ABOD	0.018	**0.8895**	2.820
CBLOF	0.018	**0.9262**	0.272
COPOD	0.018	**0.8022**	**0.072**
iForest	0.018	**0.9011**	1.006
KNN	0.018	**0.8419**	0.699
LOF	0.018	**0.8997**	0.196
MCD	0.018	0.6658	1.630
OCSVM	0.018	**0.8132**	5.762
PCA	0.018	0.4685	**0.007**

ROC can be accepted as well performing indicator of a valid prediction model. When we examine the results in the tables, we observe that model scores for Experiment Dataset DS1-ATSF are consistently exceptionally low when compared to the scores of the DS2-MTSF dataset.

Since both tables consist of timeseries based temperature values, and similar outlier percentages scattered around the datasets, the major factor for this difference can be recognized as the smaller number of samples of the DS1-ATSF dataset compared to the DS2-MTSF one. Therefore, we can infer that the number of samples acts as a crucial parameter anomaly detection algorithms performance.

A. Evaluation of Algorithms

A critical observation that PCA algorithm performed very poorly for all experiment datasets which resulted in extremely low ROC scores.

On DS2-MTSF dataset, apart from PCA and MCD algorithms, all other classifiers accomplished considerably high ROC values. On the other hand, on DS1-ATSF dataset, except for CBLOF and COPOD algorithms, the rest of the classifiers recorded exceptionally low ROC scores.

Besides the PCA algorithm, CBLOF also showed consistently satisfactory performance on both datasets. According to the results for both datasets, the Cluster-Based Local Outlier Factor (CBLOF) algorithm achieved the highest ROC value, especially for DS2-MTSF dataset, reaching as high as 92% ROC score.

B. Evaluation of Execution Time

Execution time of an algorithm indicates an insight about its time complexity. Although the vital effect of generating a high score for ROC in the model evaluation stage, the runtime of an algorithm might affect computational systems in a negative way. To reduce its resource consuming side effects, it is critical to monitor the execution period of each algorithm with model performance

In contrast with the low-scored ROC performance of the PCA approach, its runtime duration practices as the lowest among the algorithms. As the second least time-complexity holding algorithm, we can highlight CBLOF and COPOD which have promising ROC scores on both datasets. metrics.

On the other hand, Angle-Based Outlier Detection (ABOD) algorithm recorded one of the longest runtime durations on both datasets. For DS2-MTSF dataset, the highest time complexity belongs to One-Class SVM (OCSVM) algorithm which is pursued with Angle-Based Outlier Detection (ABOD) that was observed as the maximum runtime duration in DS1-ATSF dataset.

VI. DISCUSSION

Detecting anomalous points which invoke to the identification of uncommon items which differ from the overall conveyance of a population, is an essential problem to be solved since it might point to non-performing mechanisms. With vast amount of existence in various fields, multiple types and groups of approaches shall be applied and tested to find the most suitable one.

Performances of the machine learning models indicated that they are dependent on the size of the datasets on which they were trained since the description of the datasets share common basis as containing time-based temperature values. The dataset that the machine learning algorithms presented satisfying performance was the one that held the higher number of samples.

Execution time performances of each algorithm follow synchronized pattern by proving their time complexities over different sized datasets. As the highest passed-time recorded algorithm in dataset DS1-ATSF can be observed as second highest in the DS2-MTSF. The similar condition also valid for the lowest time-complexity possessing algorithms are the same for both datasets.

The main objective of this paper was to introduce the different types of outlier detecting approaches fostering with various statistical backgrounds. With this aim, the dataset was chosen from the same field as Numenta Anomaly Detection Benchmark and for the algorithm part, Python Outlier Detection (PyOD) package was selected which is composed of several anomaly detection algorithms to be able to evaluate all the detection classifiers by treating them equivalently by using an officially approved scientific software package.

The dataset resource and package decision did contribute to our processes while implementing and comparing them. However, the Numenta dataset only holds timeseries datasets with a few numbers of features for each table which might not satisfy the sufficient number of variables while testing a good performing high dimensional classifier.

Furthermore, we introduced a higher number of anomaly detection approaches than the experimented ones in this paper, since the PyOD algorithm set consists of limited number of outlier identification approaches which restricted our number of tested algorithms.

VII. ACKNOWLEDGEMENT

Funding for this work was partially supported by the Research and Development Center of Commencis Technology accredited on Turkey - Ministry of Science.

VIII. CONCLUSION

For further steps, a multivariate dataset can be preferred to be able to compare the algorithms by means of space complexity in a more solid way. Since we executed all the algorithms with their default parameter configurations except for contamination, hyper-parameter tuning can be performed for each algorithm to obtain more optimized results.

REFERENCES

[1] A. Ramchandran and A. K. Sangaiah, "Chapter 11 - Unsupervised anomaly detection for high dimensional data - An exploratory analysis, " in Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications, 2018, pp. 233-251.

[2] V. Kotu and B. Deshpande, Chapter 13 - Anomaly Detection, Data Science 2nd ed., 2019, pp. 447-465.

[3] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark," in 14th International Conference on Machine Learning and Applications (IEEE ICMLA'15), 2015.

[4] V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection: a survey," in ACM Computing Surveys, vol. 41, 2009.

[5] A. S. Weigend, M. Mangeas, and A. N. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting," in International Journal of Neural Systems, vol. 6, pp. 373–99, 1995.

[6] S. Salvador, P. Chan, and J. Brodie, "Learning states and rules for time series anomaly detection," in FLAIRS conference, 2004, p. 306–311.

[7] C. Fauconnier, and G. Haesbroeck, "Outliers detection with the minimum covariance determinant estimator in practice," in Statistical Methodology 6, 2009, pp. 363-379.

[8] M. Hubert, and M. Debruyne, "Minimum covariance determinant" in John Wiley & Son s, Inc, 2009.

[9] K.H. Esbensen, Multivariate Data Analysis - In Practice, 5th ed, CAMO Process AS, Esbjerg, Denmark, 2005.

[10] G. Stefatos and H. A. Ben, "Cluster PCA for outliers detection in high-dimensional data," in IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 3961-3966.

[11] B. N. Saha, N. Ray, and H. Zhang, "Snake validation: A PCA-based outlier detection method," in IEEE Signal Processing Letters vol. 16, 2009, pp. 549-552.

[12] H.P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'08), Las Vegas, NV, 2008, pp. 444-452.

[13] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. "COPOD: copula-based outlier detection," in IEEE International Conference on Data Mining (ICDM). IEEE, 2020.

[14] M. M. Breunig, H. P. Kriegel, R.T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in ACM sigmod record, vol. 29, 2000, pp. 93–104.

[15] C. Zhangyu, Z. Chengming, and D. Jianwei, "Outlier detection using isolation forest and local outlier factor," in Proceedings of International Conference on Research in Adaptive and Convergent Systems, Chongqing, China, September 24–27, 2019.

[16] K. Seung, N. W. Cho, B. Kang and S. H. Kang, "Fast outlier detection for very large log data," in Expert Systems with Applications, 2011, pp. 9587–9596.

[17] J. L. Bentley, N. W. Cho and B. Kang and S. H. Kang, "K-d trees for semidynamic point sets," in Proceedings of 6th Annual ACM Symposium Computational Geometry, 1990, pp. 187-197.

[18] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," in Pattern Recognition Letters, vol.24, 2003, pp.1641–1650.

[19] T. Cover, and P. Hart, "Nearest neighbor pattern classification," in IEEE Transactions on Information Theory, vol.13, 1967, pp.21-27.

[20] F.T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forests," in Proceedings of International Conference on Data Mining, 2008, pp. 413–422.

[21] B. Scholkopf, J.C. Platt, J.C. Shawe-Taylor, A. C. Smola and R.C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation 13(7), 2001, pp.1443–1471.

[22] B. M. Randles, I. V. Pasquetto, M. S. Golshan, and C. L. Borgman "Using the jupyter notebook as a tool for open science: An empirical study," in ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2017.

[23] W. McKinney, "Pandas: A foundational python library for data analysis and statistics," in Python for High Performance and Scientific Computing, vol. 14, 2011.

[24] Greenfield, P., Miller, J. T., Hsu, J. & White, R. L. "Numarray: a new scientific array package for python.," In PyCon DC, 2003.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," in Journal of Machine Learning Research, vol. 12, 2011, pp. 2825-2830.

[26] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A python toolbox for scalable outlier detection," Journal of Machine Learning Research, vol. 20 num. 96 , 2019, pp. 1-7.

Link Analysis and Web Search: A review

1st Badaruddin
Department of Computer Science
Shah Abdul Latif University
Ghotki Campus
Ghotki, Pakistan
badaruddin.chachar@salu.edu.pk

2nd Ali Raza
Department of Computer Science
Muhammad Alii Jinnah university
line 4: Karachi, Pakistan
ali.raza@jinnah.edu

3rd Abdul Aziz
School of Software
Dalian University of Technology
Dalian, China
abdulaziz@mail.dlut.edu.cn

***Abstract*— In this technological era each and every day significant advancements are taking place in the field of information technology especially in information retrieval and big data mining. Links are the best sources for carrying out the data with processing capabilities and because of that one of the leading change in our daily life is availability of efficient and accurate web search via various search engines like Google. In fact, Google was not the first search engine but it helped a lot to defeat the spammers who were the major cause of useless web search. Graphs have gained a very good popularity in every technological and traditional fields because of their excessive use and of course increased in capacity and performance. There is much information to be gained by analyzing the links derived from the different networks. By analyzing the links data access is very easy and fast that's why their efficiency has improved and the need of that type of particularly widespread system has grown. In this short paper we have reviewed a bunch of papers to deeply understand the overall idea of link analysis and web search.**

Keywords — Directed Graphs, Social Networks, Media Networks, Information network Systems.

I. INTRODUCTION

The inefficient queries that produces results that are less than satisfying, generating endless links to irrelevant pages even though those pages may contain the right query keywords is very often in search engines during web search. As we are growing in terms of information technology we are facing great challenges to reduce the inefficiency, spreading awareness and stopping the spammers that are leading us to the irrelevancy and not getting the desired results. So it has been observed that link analysis can have significant contributions to web page retrieval from search engines, to web community discovery, and to the measurement of web page influence. It can help to rank results and find high-quality index/hub/link pages that contain links to the best sites on the topic of interest as well as links tells the what is more important on the web [1].

Graphs are everywhere and we are all surrounded by data, that data can be represented by graphs especially data gathered from the different networks like social networks including Facebook, Twitter. WeChat and Weibo and many more. Graph can easily represent the information as graph in a sense of set of nodes and edges or connection between them or intersection between them. Another data that also can naturally be represented as networks are the networks of the information or information nets like in figure 1, in this case what we are seeing is the map of science, So, every node is a different journal or a different conference and edges between the journals for publication venues, one journal is citing another journal. So based on this citation network between journals, we can basically visualize how different disciplines sciences and sub fields of science are relating to each other. Another example of big part of kind of networks is the web, Web itself can be considered and represented as Graph, and we will focus on these kind of structures of the web. So the first question is how we represent a web as a graph. We will represent web as a directed graph: Nodes will correspond to the web pages and directed links between those web pages that corresponds to be as Hyperlinks. If web is a big and giant network, then the question arises that how is the web organized. The way people try to approach to organize the web is to naturally curated by human. So for example Yahoo back in 1996, their originally idea was to take all the webpages on the web and manually categorize them into the set of different categories and sub categories. The basic theme of Yahoo was to take every page and categorize them into this giant hierarchy. Of course with recent emergence, time shows that the web has been famous for too quickly [2].

So the next way how to organize the web and how to find things on the web is Web Search. The interesting thing in terms of the web search is, there is rich literature in the particular field of information retrieval that covers the problem of how do we find a document in a large set of documents. So, in our case in web search we can think of every webpage as a document and the whole web is giant corpus of documents and our goal is to find a relevant document to a given query in a huge set. However, the traditional information retrieval was interested in finding these documents in relatively small collection of trusted documents likewise, in newspapers collection or patent collections. The web is very different, the difference is first the web is huge and the second thing is web is full of untrusted documents, random things, spam, unrelated things and so on. So the big question is which web pages should trust and which webpages are fake and irrelevant. Another question that may rise here is how do we identify the set of relevant or trustworthy webpages in this huge web network [3].

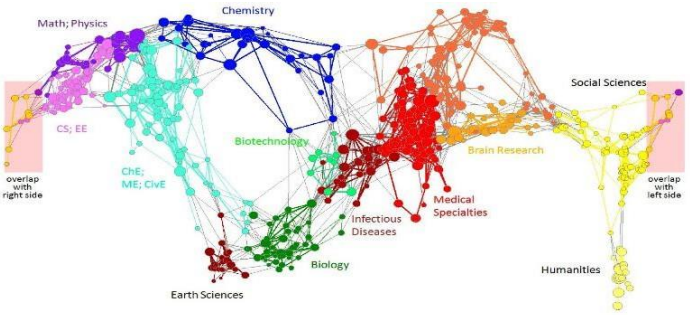


Fig.01 Citation Network and Map of Science

At early stages of search engines, the basic thing was to crawl the web and index pages by words they contained because search queries would respond according to the list of words within the pages that contains those words and attempt to order the pages matching a search query by importance.

II. Aim of Study

The use of link analysis techniques to estimate the importance of a page made web search engines useful so in this modern and progressive technological era traditional link analysis is very necessary but very accurate and efficient because Google presently claims that more than one hundred factors are required for calculating the result of a single query. As a result, web search is facing two big challenges and they are like firstly web contains many sources of information then who to "trust" and secondly what would be the best answer to a query. As the first challenge is very clear like who to trust on to the web because many pages are legitimate and which pages are spam or somehow fabricated. The idea here is to use the structure of the link web graph to understand these things. The idea is simple like trustworthy webpages will link to each other. So, here is the idea of page rank algorithm and then the other problem is sometimes queries can be rather ambiguous, for example anyone can ask what is the best answer to query newspaper and there is no any good answer to this query and the goal is if we identify all good newspapers on the web is to again look at the web structure in order to identify set of pages or newspapers linking to each other and get the results out of the structure [4-6].

The way of addressing both of these challenges is to basically realizing that the web as a graph is a very rich structure. So, one thing that is very mandatory to understand that we can try to think this problem as rank nodes of a big graph. So basically we would like to compute a score of every node in this web graph. The idea is some nodes will collect lots of links so they will have high importance and some other nodes will have a small number of links from untrusted sources so they will have low importance. In recent days' usage of graphs are playing a key role in various sectors and is helping intensifying security, results and portability of information. Simply in order to understand and compute the importance of the nodes in a graph, there are several approaches. Broadly thinking of these approaches are called Link analysis approaches are link analysis techniques because analyzing the links of the web graph to compute the importance of nodes in a graph is the main task of these techniques. The following section will present the details of these approaches [5].

A. *Page Rank*

Suppose we have links in a graph as votes and consider a page or a node in a graph is as important as the number of link it has. But it is notable thing here that what would be these links like either we are talking about in-links or out-links. For-example take a look at in-links. Because in-links are kind of harder as it's very easy to have a page with lots of out-links but it's harder to have a page that lots of other pages on the web point to. The second thing is it's not enough to just consider in-links but we also have to consider about from where this link is coming from. Likewise, a link from a given webpage maybe from dlut.edu.cn has more importance than some other webpage that receives only few in-links. So the idea is not all in-links are equal and links coming from important pages have worth more. Because importance of a page is depending on the importance of the other pages that are pointing to it. So this is kind of recursive definition where importance of your given page depends on the importance of the other pages as shown in figure 2. This kind of importance can measure the importance of the graph. In figure 2, the size of nodes is proportional to its page rank scores and we have normalized the page rank scores so that the sum is to be one hundred [7-9].

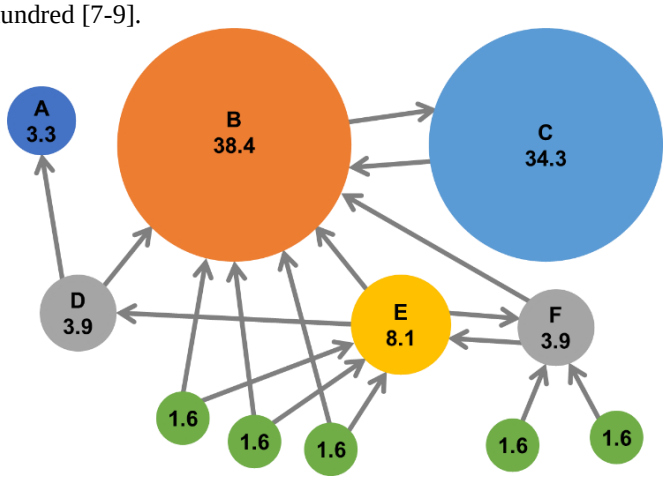


Fig.02. Page Rank Scores of different nodes via a graph

Additionally, page rank score is computed for importance of nodes in a node graph network and using recursive formulation where we take each link as a vote and we think importance of a node is proportional to the importance of the source page. For clearly understanding the recursive formulation let's take a page j with its importance r_j and page j has n outgoing links and the importance of page j is r_j is split in its out-going links evenly and each link gets r_j / n votes. Similarly, we can find the importance of J as the sum of votes that it receives as in-links [10]. Like in figure 3, the importance r of page j is simply importance of page i divided by three because page i has three out-going links plus importance of page k divided by four because page k has four out links. So this is how we compute the page rank score of page j, and now we have the page rank score of page j and the score further get spread outside of page j along the three out-going links and the equation goes like;

$$rj = ri/3 + rk/4$$

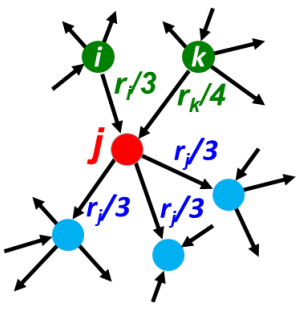


Fig.03. Computing the importance

i. The Flow Model

A page is important if it is pointed by other important pages so simply a vote from an important page is worth more.

Hence it is called flow model through the network and sometimes it is also called as flow formulation of the flow model. To make sure how it works let's have a deeper look onto it. As shown in figure 4, a very small web graph contains three websites namely a, m and y. It is clearly shown in figure 4 that the structure like y has the self-link and a link points to a, and a point to y and m and m point to backward and so on. Initial idea is simple as mentioned earlier that the vote from an important page is worth more. So we can say a page is more important if its pointed by other important pages. Like already defined in previous slide, we will assign the importance r to the page j and we will call this important score as rank. So this is where from the page rank terminology comes from [10-12].

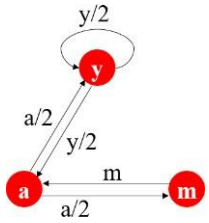


Fig.04 Flow Model

Importance of page j is simply the sum of all the other pages that point to it, importance of that page i divided by the out degree of the page. So now what we can do is basically this means for every node in the network we obtain a separate equation. So for example the importance of node y in network is simply the importance of y divided by 2 plus importance of a divided by 2 because y has 2 outgoing links (as one link points to itself) and similarly node a has 2 out-going links so we take RA divided by 2. Going further like we say the importance of the node m is just simply the importance of node a divide by 2 (again why divided by 2 because a has 2 out-going links) so half of its importance goes to m and half of its importance goes to y.
"Flow" equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

Let's define a rank r_j for page j as the following

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

ii. Matrix Formulation

Linear algebra is always considered as the best option when dealing with matrices. The flow equations in terms of graph can be represented as a big matrix of values.

$$\text{If } i \to j \text{ then } M_{ji} = 1/d_i \quad \text{else} \quad M_{ji} = 0$$

Here M is column stochastic matrix and the idea is like if a page i, points to page j, then we will have a non-zero entries in the cell j i, and if the page j does not point to page i, then we will have zero entry there. So now the question here is what is the value of non-zero entries in the cell of matrix. Hence if i, point to j, then the corresponding entry j,i in the matrix M will be 1 over the out degree of node i, So this means that our matrix M is called the column stochastic matrix which means that every column in out matrix sums to one [13-15].

So for now we have presented our whole graph as a matrix, and we can also take all the page rank scores of nodes represent it as a vector. So the way we do this is that we think we have one entry per page so we can think of our pages as number 1,2,3,4,5 up to n and we have a vector of length n and every entry in this vector basically corresponds to the page rank score of a given page. So here sum of all the entries of our vector equals to one. So now what is interesting is we can take our flow equation kind of our previous basic equation and write it in terms of the matrix M and vector R. So we can say that rank vector R equals to the matrix M times to the vector R again. So now we can say we have a big equation as M is fixed and we can find what are the values of R [14].

B. Hubs Authorities (HITS)

Hyper Link Induced Topic Search algorithm is a classical link analysis algorithm for analyzing the web structure. The algorithm takes into the consideration of the structural information of links but ignores the correlation between pages and topics. This strategy determines the importance of a page by analyzing the reference relationship between pages, so as to determine the crawling sequence of the focused crawler on the page. The algorithm is to find out the authority pages and hub pages from a set of web pages, according to user query and through analyzing the forward and backward linkages. An authority page is an authoritative page most relative to the query topic (authority is of influence and is accepted by most of people); while a hub page is a web page that points to the link set of authoritative pages [15]. Each page requires two measurement values: authority weigh and hub weight, according to which the importance of a page to a specific topic can be judged.

Many algorithms, including the HITS, are based on hypothesis. The HITS algorithm uses two basic hypotheses:
Hypothesis 1: a good "authority page" is linked by many good "hub pages", and;
Hypothesis 2: a good "hub pages" points to many good "authority pages".
Hypothesis 1 describes what a "good" authority page is: a page linked by many good "hub pages". Hypothesis 2 describes what a "good" hub page is: a page points to many good "authority pages". Form above basic hypotheses we cans see a mutually reinforcing relationship between the hub and authority page, i.e. the higher quality of a hub page, the better the authority page will be, and vice versa, if an authority page has higher quality, then the hub page that points to the authority page has a higher quality. Based on such a mutually reinforcing relationship, those hub and authority pages that have highest quality can be figured out through iterative computations.

Furthermore, HITS is an important algorithm for web structure mining. In response to problems of this algorithm, many scholars have proposed various improved algorithms which are still in evolution.

C. Knowledge Graph Embedding:

Knowledge graphs are considered as the organized representation about the real work data where nodes are used to represent entities whereas edges are used for connecting two entities with a relation. For example: people and places.

Link prediction is the best example of graph embedding as it is used to find the existence of a link between two entities and then predict that link for the said task. In other words, the prediction of the links is necessary to check whether the two links between two nodes are based on the attribute information and the actual relationship that has been observed [15-19].

Conclusion

Link analysis techniques are the mostly used techniques to assess the relationship between the nodes and for bringing betterment in web search algorithms. In this epidemics and pandemics situations of COVID-19, link analysis techniques have played a vital role in medical research. Other areas where link analysis can be used are; search engine optimization for any search engine, security risk analysis, fraud detection and stock market research etc. So to conclude this paper, there are various purposes of link analysis but mostly three purposes are considered for link analysis. Firstly, for the data which could be of the interest, secondly to find the anomalies where rules and regulation have been desecrated and mitigate these violations. Lastly, to determine the new and innovative techniques for web search, graph modeling and graph learning with real time analysis of social life.

REFERENCES

[1] Dong Wang, Zheng Chen, Li Tao, Wei-Ying Ma, Liu Wenyin "Ranking User's Relevance to a Topic through Link Analysis on Web Logs" in WIDM

[2] Pawel Jurczyk Eugene Agichtein "Discovering Authorities in Question Answer Communities by Using Link Analysis" in CIKM

[3] Deng Cai Xiaofei He Ji-Rong Wen Wei-Ying Ma "Block-level Link Analysis" in SIGIR 2019

[4] Luca Becchetti Stefano Leonardi "Link Analysis for Web Spam Detection" in ACM Transactions on the Web, Vol. 2, No. 1, Article 2

[5] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger "LinkSUM: Using Link Analysis to Summarize Entity Data"

[6] Weiming Yang "An Improved HITS Algorithm Based on Analysis of Web Page Links and Web Content Similarity" International Conference on Cyberworlds.

[7] Puspita Datta Prof Vaidhehi V "Influencing the PageRank using Link Analysis in SEO" International Journal of Applied Engineering Research"

[8] Samir Kumar Jalal "Exploring Web Link Analysis of Websites of Indian Institute of Technology" Journal of Library & Information Technology

[9] K. Kumar "Web Impact Factor and Link Analysis of Indian Council of Agricultural Research (ICAR) Organizations" International Journal of Knowledge Content Development & Technology

[10] Brian D. Davison, Apostolos Gerasoulis, Konstantinos Kleisouris, Yingfang Lu, Hyun-ju Seo, Wei Wang, and Baohua Wu "DiscoWeb: Applying Link Analysis to Web Search" ACM Conference on Hypertext and Hypermedia

[11] Gui-Rong Xue Hua-Jun Zeng Zheng Chen Wei-Ying Ma Hong-Jiang Zhang ChaoJun Lu "Implicit Link Analysis for Small Web Search" in SIGIR.

[12] Lan Nie Brian D. Davison Xiaoguang Qi "Topical Link Analysis for Web Search" in SIGIR

[13] ROSSI et al "Knowledge Graph Embedding for Link Prediction: A Comparative Analysis"

[14] Tsioutsiouliklis et al "Fairness-Aware Link Analysis"

[15] Mariana Cláudia master's dissertation "Analysis of web information-seeking behavior of users with different levels of health literacy" @Faculty of Engineering - University of Porto

[16] Armin et al "What are Links in Linked Open Data? A Characterization and Evaluation of Links between Knowledge Graphs on the Web" in Journal of Data and Information Quality (ISSN 19361956)

[17] Aboubakr Aqle, Dena Al-Thani, and Ali Jaoua "Can search result summaries enhance the web search efficiency and experiences of the visually impaired users?"

[18] Yin Myo Kay Khine Thaw, Myo Ma Ma and Khin Myat New Win "Capabilities of e-Commerce Technology By Using Web Mining"

[19] Marie-Theres Nagel et al "How Do University Students' Web Search Behavior, Website Characteristics, and the Interaction of Both Influence Students' Critical Online Reasoning?"

[20] Falah Al-akashi "SAMA: a real-time Web search architecture" in International Journal of Computers and Applications

[21] Youngho Jo et al "Web behavior analysis in social life logging" in The Journal of Supercomputing (2021) 77:1301–1320

.

A Real-Life Predictive Maintenance: A Case Study from Industry

İlknur Kurban
Software Development
Eldor Corporation
İzmir/TURKEY
ilknur.kurban@eldor.com.tr

Mehmet Tekeli
Technology
Eldor Corporation
İzmir/TURKEY
mehmet.tekeli@eldor.com.tr

Özgür Selmanoğlu
Technology
Eldor Corporation
İzmir/TURKEY
ozgur.selmanoglu@eldor.com.tr

Onur Tekir
Software Development
Eldor Corporation
İzmir/TURKEY
onur.tekir@eldor.com.tr

Murat Şahin
Software Development
Bakırçay University
İzmir/TURKEY
murat.sahin@bakircay.edu.tr

Deniz Kılınç
Computer Engineering Department of Faculty of Engineering and Architecture
Bakırçay University
İzmir/TURKEY
deniz.kilinc@bakircay.edu.tr

***Abstract*—Predictive maintenance is a set of techniques that uses data analysis methods to detect anomalies in operations and processes, as well as potential equipment defects, so failures can be detected before they occur. Predictive maintenance reduces the lost production hours due to maintenance and the cost of spare parts and consumables. In this work, predictive maintenance solution of Eldor Corporation in İzmir is presented as a real-life case study. By collecting various data measurements such as vibration, acceleration, and temperature from smart sensors, a parametric trend analysis algorithm and a software system for anomaly detection were developed. During the design of the system and the algorithm, a data set consisting of 10 variables and 1.071.572 observations, including old pump failures was created and used. The final system detected six fault events before they occurred between 30.07.2020 and 30.11.2020.**

Keywords—predictive maintenance, anomaly detection, trend analysis

I. Introduction

Growing competition, globalization of enterprises, advances towards total quality management and continuous technological changes are some of the key factors that cause major changes in the structure of enterprises. These modifications have also been moved to the production field as they are most directly related to the effectiveness and sustainability of manufacturing and maintenance processes [1]. Consequences in production and maintenance reveal the need to change the focus of maintenance strategies. The core principle of maintenance is that continuous monitoring of process, system, material or commodity conditions guarantees the maximum time between repairs and thereby minimizes the amount and expense of unplanned outages that impact the output, consistency of the service and overall performance of the production process [2]. Predictive maintenance is a maintenance type that tracks the efficiency and state of equipment during a regular service to decrease the risk of failures. Predictive maintenance is also identified as condition-based maintenance and has been applied in the manufacturing field since the 1990s [3].

In this study, predictive maintenance experimental works in Eldor Corporation are utilized using sensor data installed on the pumps in a part of the production line. Various data measurements such as vibration, acceleration and temperature are collected from the smart sensors. Controlled tests are also carried out in a way that may cause bearing failure. Finally, a parametric trend analysis algorithm and software for anomaly detection have been developed that detected six failure events before they occurred between 30.07.2020 and 30.11.2020.

The rest of the paper is organized as follows. Section 2 presents predictive maintenance works in the literature. Section 3 explains background and methods are introduced. Section 4 explains the experimental study including dataset, developed solution and events detected. Finally, the last section concludes this paper and discusses future work.

II. Related Work

When the literature is searched, it is seen that many studies have been conducted in the field of predictive maintenance.

Martha et al. [4] conducted a research using data from a metallurgical company. In the study, they used rule-based methods for predictive maintenance and obtained positive results. In addition, feature selection methods were used to determine which data will be used before analyzing.

Radyha Saha et al. [5] researched new technologies coming with industry 4.0 and big data. Their studies were conducted on big data platforms by using predictive maintenance. In the study, they used new technologies and tools such as Kafka, Kinesis and RabbitMQ. The scope of the study is on rail transportation and wing energy.

Chia Yan Li et al. [6] proposed a study on detecting bearing failure on electric motors. They tried to find the faults based on the vibration data. They developed and trained a model using various regression methods such as weighted least squares regression (WLS) and feasible generalized least squares regression (FGLS), and Support vector regression (SVR).

Wo Jae Lee et al. [7] have researched tool wear and bearing failure for predictive maintenance. They worked on data driven models. During the operation, six different signals, the DC spindle motor current, the AC spindle motor current, the table vibration, the spindle vibration, the acoustic emission at the table and the acoustic emission at the spindle

were collected. Two powerful classification techniques, support vector machines (SVM) and artificial neural networks (recurrent neural network and convolutional neural network) were trained and tested.

III. Background and Methods

This section presents the problem definition, maintenance types, trend analysis and outlier detection methods utilized in the predictive maintenance task.

A. Problem Definition

Eldor Corporation was established by Pasquale Forte in 1972 and is an international group leader in the automotive business and supplier to the main automotive manufacturers across the world. Eldor operates having 3.500 employees in 20 locations with presence in Italy, Germany, USA, China, Turkey, Brazil, Japan, and South Korea. Figure 1 shows a graph of 4 days of data collected from one sensor (Radial Axis Peak Acceleration) in a production line belonging to Eldor Turkey. In the figure, the window size for the moving average is determined as 30 minutes.

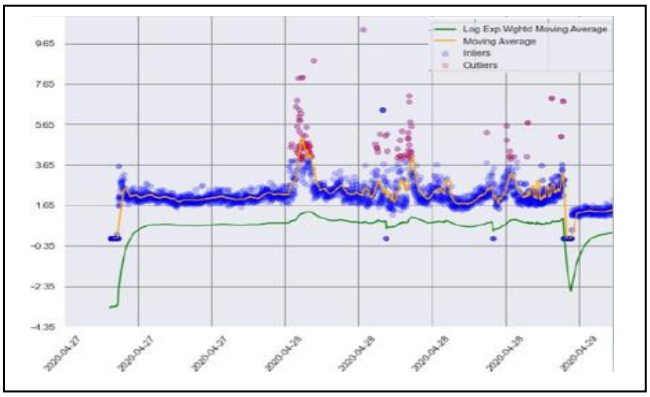


Fig. 1. Four days view of outliers.

The orange line is drawn by taking the average value every 30 minutes and sliding the window in one-minute steps. Green lines are used for the exponential weighted moving average and a logarithmic transformation has been performed on the smart sensor data to calculate this value. Exponential weighted moving averages over the stationary series are defined by determining 30-minutes windows. As it is seen from the graph, some outlier values may seem visible. However, it is not possible to visually understand whether these outliers are signs to failure events or not.

B. Maintenance Types

- *Corrective Maintenance:* Corrective maintenance is performed when a problem is detected while working on another job. The problem is solved at that time or at another scheduled time. Problems can be caught instantly with corrective maintenance [9, 10].
- *Preventive Maintenance:* This type of maintenance can also be called periodic maintenance. Maintenance is carried out regularly in a certain period of time or in a planned manner according to a certain criterion. The aim is to replace the equipment of the system regularly before the failure occurs to prevent unexpected failures of the operation of a system. The disadvantage of preventive maintenance is the equipment that is replaced despite having a longer usable life [11, 12].
- *Predictive Maintenance:* Predictive maintenance is a type of maintenance performed using a developed complex software application that evaluates the collected sensor data of the equipment to prevent unplanned malfunctions and maximize service life. Predictive maintenance is categorized into two groups; i) statistically based and ii) condition based [13]. In statistics-based predictive maintenance, statistical methods are used to determine when maintenance will be performed in the future. Condition-based predictive maintenance predicts the occurrence of a fault by monitoring the equipment instantaneously and continuously and analyzing it with data of normal operating conditions obtained in the past.

C. Trend Analysis and Outlier Detection in Predictive Maintenance

The regular movement that is generally observed in time series in the long term is called “trend” [14]. The trend showing the average level of the time series is the growth measure of the relevant series. Upward direction of the trend indicates growth, while downward direction indicates shrinkage in the series. Outlier detection is defined as the process of descriptive analytics performed to find unusual or extreme samples in a large data set, without prior knowledge of which samples to search.

Conducting an outlier analysis by ignoring the trends in a time series problem will result in obtaining results that are both incomplete and impossible to interpret. Especially in predictive maintenance problems where time-dependent conditions are constantly monitored, it is critical to detect anomalies by following trends on time series data [15].

IV. Experimental Study

In this section, we present detailed information about the dataset and the experimental procedures applied.

A. Dataset

The dataset covers the period for a production line between 04.03.2020 and 25.05.2020. It consists of 10 features and 1.071.572 observations. It contains data of seven pumps including four vacuum and three blowers. The information about collected measurement variables are shown in Table 1.

Table 1. Measurement variables (features) collected from sensors.

Measurement Variable	Description
Radial Axis RMS Vibration	The average of vibration squares on the radial axis.
Axial Axis RMS Vibration	The average of vibration squares on the axial axis.
Radial Axis RMS Acceleration	The average of acceleration squares on the radial axis.
Radial Axis Peak Acceleration	The peak of acceleration on the radial axis.

Radial Axis Curtosis	The curtosis of radial axis RMS vibration.
Radial Axis Crest Factor	A parameter of a waveform, such as alternating current or vibration, showing the ratio of peak values to the effective value. In other words, the crest factor indicates how extreme the peaks are in a waveform.
Temperature	The temperature value in Celsius.

B. Developed Trend Analysis Algorithm

In the study we developed a parametric trend analysis algorithm for anomaly detection that can be easily adapted for similar tasks in production. The dataset that contains the past pump failures and maintenance have played a critical role in the development of the algorithm. Detailed exploratory data analysis was performed on the smart sensor data before and after pump failures. Time series visualizations show that an upward trend and outliers were observed before the pump failure occurred. The algorithm tries to detect the uptrend and anomaly value by shifting every hour at the window size determined for all smart sensor features.

In this context, the window size, the degree of trend to be detected and the outlier threshold value should be entered parametrically for each smart sensor feature. Before explaining how the trend algorithm works, pre-processing is performed on the raw smart sensor data. In these pre-processing stages, processes such as deleting missing data, making the frequency of the data a regular one minute, getting a 60-minutes moving average and finding an hourly average are applied.

After pre-processing steps, a line that will represent all hourly averages is fitted with the help of a least square's polynomial fit. The optimal line made up has an equation and hence a slope that represents the trend's intensity and direction. The degree of the trend is determined by taking the arctangent of this slope coefficient. If the trend level is higher than the trend level specified as a parameter in the configuration file, a trend alarm is issued.

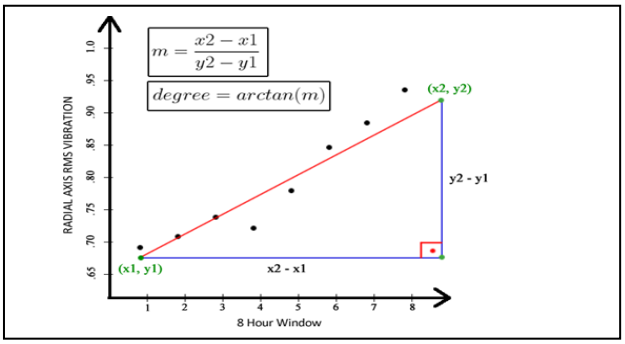


Fig. 2. The logic of the trend detection algorithm.

Figure 2 can help to understand the working logic of the trend detection algorithm. The example displays the 8-hour preprocessed values of the RMS variable representing the vibration data. On this graph, where a serious increase was observed from the first hours to the last hours, the red line was drawn with the least squares method. The slope coefficient and slope degree are calculated to determine the trend through the equation of the line drawn.

C. Events Detected Before Failures

In this study six successful events between 30.07.2020 and 30.11.2020 were detected before failures. The developed system determines a trend considering at the last 8 hours. A sample range is determined and the slope grade in this range is considered to find a true slope. The degree of the slope is the angle of the linear line passing through the set of points calculated for 8 hours. The system allows the results to be fine-tuned by changing the slope grade thresholds. Besides, users are warned by the system when there is a real failure in the pumps.

Table 2. Successful events detected by the system.

Event	**Slope Degree**	**Feature**	**Trend Start Date**	**Pump Change Date**
Ev1. Pump X	7.84	Radial Axis RMS Vibration	27.07.20	30.07.20
Ev2. Pump X	9.22	Radial Axis RMS Vibration	19.08.20	21.08.20
Ev3. Pump Y	13.71	Radial Axis PEAK Acceleration	23.09.20	25.09.20
Ev4. Pump X	18.29	Radial Axis RMS Vibration	28.09.20	28.09.20
Ev5. Pump Y	6.83	Radial Axis PEAK Acceleration	15.10.20	16.10.20
Ev6. Pump X	9.86	Radial Axis PEAK Acceleration	04.11.20	05.11.20

Table 2 shows information about the events. Slope degree value is the related feature maximum slope degree in trend list. In the paper, event 6 and event 8 are explained with figures.

Event #1: It was the first successful event detected by the developed system. There was a mechanical problem, and a pump was changed with another. Event was detected because of the increase in vibration and acceleration values. The trend's slope was measured at 7.84.

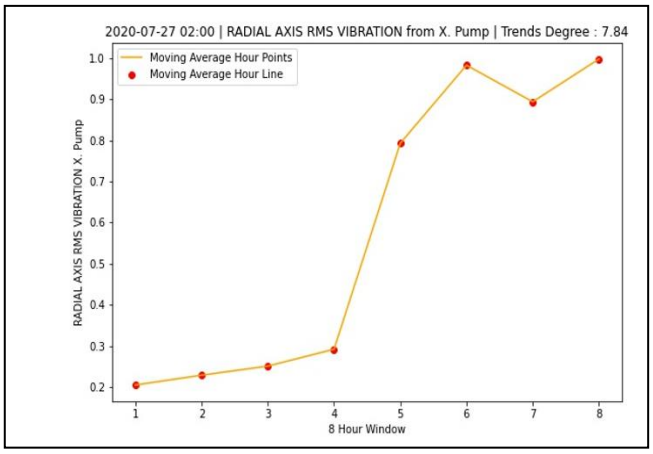


Fig. 3. Event#1 and its eight ours window.

Event #6: This event was the last detected trend. The pump was changed considering the radial axis RMS acceleration trend.

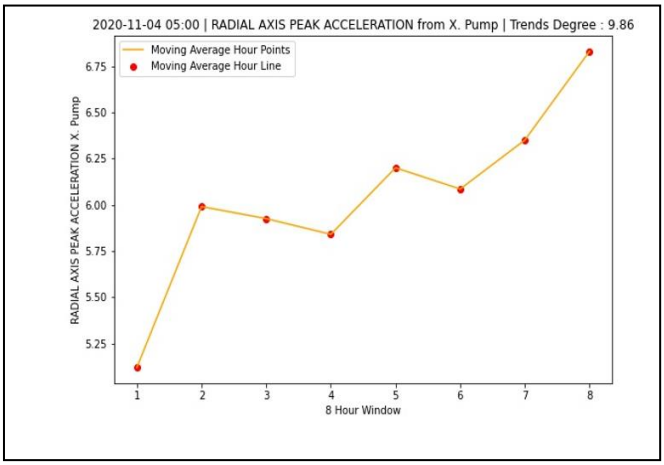


Fig. 4. Event#6 and its eight hours window.

CONCLUSION AND FUTURE DIRECTIONS

Maintenance is a set of activities aimed at always keeping a machine's condition and performance like new. In literature different types of maintenance exist such as corrective, preventive, and predictive. In this study, a trend analysis algorithm and software for anomaly detection have been developed and possible events of failures are prevented in six real-life scenarios between 30.07.2020 and 30.11.2020 in Eldor Turkey. The results also show that the developed system is capable to find trends and possible malfunctions approximately 2 days before they occurred. As a future work, if enough labeled data (healthy and unhealthy states) is collected, it is planned to use machine learning classification algorithms such as decision tree, SVM, and ensembles.

REFERENCES

[1] Jonsson, P. (1999). Company-wide integration of strategic maintenance: an empirical analysis. International journal of production economics, 60, 155-164.

[2] [2] Mobley, R. K. (2011). Maintenance fundamentals. Elsevier.

[3] [3] March, S. T., & Scudder, G. D. (2019). Predictive maintenance: strategic use of IT in manufacturing organizations. Information Systems Frontiers, 1-15.

[4] [4] Fernandes, M., Canito, A., Bolón-Canedo, V., Conceição, L., Praça, I., & Marreiros, G. (2019). Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry. International journal of information management, 46, 252-262.

[5] [5] Sahal, R., Breslin, J. G., & Ali, M. I. (2020). Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. Journal of Manufacturing Systems, 54, 138-151

[6] [6] Lee, C. Y., Huang, T. S., Liu, M. K., & Lan, C. Y. (2019). Data science for vibration heteroscedasticity and predictive maintenance of rotary bearings. Energies, 12(5), 801.

[7] [7] Lee, C. Y., Huang, T. S., Liu, M. K., & Lan, C. Y. (2019). Data science for vibration heteroscedasticity and predictive maintenance of rotary bearings. Energies, 12(5), 801.

[8] [8] Ben-Daya, M., Duffuaa, S. O., Raouf, A., Knezevic, J., & Ait-Kadi, D. (Eds.). (2009). Handbook of maintenance management and engineering (Vol. 7). London: Springer.

[9] [9] Adolfsson, E., & Tuvstarr, D. (2011). Efficiency in corrective maintenance - a case study at SKF Gothenburg.

[10] [10] Wang, Y., Deng, C., Wu, J., Wang, Y., & Xiong, Y. (2014). A corrective maintenance scheme for engineering equipment. Engineering Failure Analysis, 36, 269-283.

[11] [11] De Almeida, A. T., Cavalcante, C. A. V., Alencar, M. H., Ferreira, R. J. P., de Almeida-Filho, A. T., & Garcez, T. V. (2015). Preventive maintenance decisions. In Multicriteria and Multiobjective Models for Risk, Reliability and Maintenance Decision Analysis (pp. 215-232). Springer, Cham.

[12] [12] Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., & Vasilakos, A. V. (2017). A manufacturing big data solution for active preventive maintenance. IEEE Transactions on Industrial Informatics, 13(4), 2039-2047.

[13] [13] Hashemian, H. M. (2010). State-of-the-art predictive maintenance techniques. IEEE Transactions on Instrumentation and measurement, 60(1), 226-236.

[14] [14] Maurya, M. R., Rengaswamy, R., & Venkatasubramanian, V. (2007). Fault diagnosis using dynamic trend analysis: A review and recent developments. Engineering Applications of artificial intelligence, 20(2), 133-146.

[15] Susto, G. A., & Beghi, A. (2016, September). Dealing with time-series data in predictive maintenance problems. In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-4). IEEE.

Assessment of Current Computational Intelligence Methods on Benchmark Function

Soner Kızıloluk
Computer Engineering
Malatya Turgut Özal University
Malatya, Turkey
soner.kiziloluk@ozal.edu.tr

Umit Can
Computer Engineering
Munzur University
Tunceli, Turkey
ucan@munzur.edu.tr

Bilal Alatas
Software Engineering
Firat University
Elazığ, Turkey
balatas@firat.edu.tr

***Abstract*— Metaheuristic solution methods proposed for solving complex, multidimensional and nonlinear optimization problems that do not guarantee the exact solution for every problem. New solution methods are suggested by researchers to find better solutions. In this study, among the current metaheuristic optimization algorithms, Jellyfish Search Optimizer (JS), Carnivorous Plant Algorithm (CPA), and Giza Pyramids Construction (GPC) Optimization algorithms were compared using 4 benchmark tests. The dimensions of the benchmark tests were selected as 10, 30 and 50, respectively. Jellyfish Search Optimizer algorithm showed the best performance and Carnivorous Plant Algorithm was the slowest running algorithms.**

Keywords—intelligent computation, metaheuristic optimization, benchmark function, performance

I. INTRODUCTION

Many real world applications in various fields of science and engineering can be turned into optimization problems. These problems tried to be solved are nonlinear, multimodal and highly complex problems. Metaheuristic optimization algorithms are used to find the closest solution to the optimum solution in a reasonable time frame in cases where the solution space of a problem cannot be evaluated entirely and is infinitely large. Metaheuristic optimization algorithms are very popular methods and there are four important reasons for this; (a) They are based on simple concepts and are easy to implement; (b) they do not need the gradient information of the objective function; (c) they can overcome the local minimum point; and (d) they can be used to solve different problems in various fields [1-3]. Metaheuristic optimization algorithms can be classified according to the areas they are inspired by. Metaheuristic optimization algorithms are divided into six classes according to their inspirations: evolutionary-based, physics-based, chemistry-based, swarm-based, human-based and plant-based.

The most popular evolutionary methods are the Genetic Algorithm [4] and the Differential Development Algorithm [5]. Electromagnetism-Like Algorithm [6] and Gravitational Search Algorithm [7] can be given as an example for physic based methods. Artificial Chemical Reaction Optimization [8] algorithm can be given as an example to chemistry-based methods. Methods developed in nature by imitating the social behavior of animals and using their swarm intelligence are called swarm-based algorithms. These are algorithms inspired by the behavior of animals such as moths, bees, birds, cats, wolves and whales. Algorithms such as Particle Swarm Optimization [9], Artificial Bee Colony [10] and Bat Algorithm [11] can be given as examples of swarm-based algorithms. There are many algorithms developed inspired by various human-based sources such as sports, music, education and management style. League Championship Algorithm [12] and Golden Ball Optimization [13] are examples of sports-inspired algorithms. Harmony Search [14] and Melody Search [15] algorithms are methods inspired by music. Inspired by the parliamentary system, the Parliamentary Optimization Algorithm [16] simulated the functioning of the parliamentary system. The Teaching-learning-based optimization algorithm [17] was developed, inspired by a teacher's effect on students' learning in the classroom. Apart from all these categories, plant-based algorithms inspired by plant behaviors have been proposed by researchers to solve complex problems. Root mass optimization algorithm [18-19] and Flower Pollination Algorithm [20] are examples of such techniques.

Although many new optimization algorithms have been developed by researchers, they are still not sufficient to produce satisfactory results in solving these challenging problems [21. New heuristic methods are suggested by researchers in order to provide more successful solutions to optimization problems. JS [3], CPA [22], and GPC optimization [23] are the current metaheuristic methods proposed in 2020. In this study, three up to date metaheuristic algorithms proposed in the last year were examined and these algorithms were compared using four frequently used benchmark functions for the first time. The results obtained are given in detail.

II. COMPUTATIONAL INTELLIGENCE METHODS

As scientific research continues, people's discoveries increase, which leads to the emergence of different sources of inspiration for the development of new metaheuristic algorithms. In this section, inspiration sources and the pseudo codes of three different current metaheuristic optimization algorithms, which have been compared in the experimental study, are given under subheadings.

A. Jellyfish Search Optimizer

Jellyfish are creatures that exist in various sizes and colors and can live at different depths in waters of different temperatures. Feeding methods of jellyfish are different from each other. Some jellyfish use their tentacles to bring food to their mouths, others use the filtering method to feed on what the currents bring. Some jellyfish actively hunt their prey and immobilize them by stinging them with their tentacles. The JS algorithm was inspired by the oceans foraging behavior of jellyfish and mathematically modeled a jellyfish population. The algorithm based on this model is based on three rules [3].

1) A jellyfish either follows the ocean current or moves within the swarm, and a "time control mechanism" determines the transition between these movements.
2) Jellyfish moves in the ocean for search of food. A jellyfish is attracted to places where there is more food available.

3) The location and the fitness function of the location decides the amount of food found.

The basic steps of the JS algorithm are given in the pseudo code in Fig. 1. The details of the equations mentioned in the pseudo code can be found in the relevant article [3].

Begin
Define objective function $f(X)$, $X = (x_1, \dots, x_d)^T$
Set the search space, population size (n_{Pop}), and maximum iteration (Max_{int})
Initialize population of jellyfish $X_i (i = 1,2, \dots, n_{Pop})$ using logistic chaotic map
Calculate quantity of food each X_i, ($f(X_i)$)
Find the jellyfish at location currently with most food (X^*)
Initialize time: t=1
Repeat
For i=1;n_{Pop} **do**
Calculate the time control $c(t)$ using Eq. (17)
If $c(t) \geq 0.5$: Jellyfish follows ocean current
(1) Determine ocean current using Eq. (9)
(2) New location of jellyfish is defined by Eq. (11)
Else: Jellyfish moves inside a swarm
If $rand(0,1) > (1 - c(t))$: jellyfish exhibits type A motion (Passive motions)
(1) New location of jellyfish is defined by Eq. (12)
Else: Jellyfish exhibits type B motion (Active motions)
(2) Determine direction of jellyfish using Eq. (15)
(3) New location of jellyfish is defined by Eq. (16)
End if
End if
Check boundary conditions and calculate quantity of food at new location
Update the location of jellyfish (X_i) and location of jellyfish currently with the most food (X^*)
End for i
Update the time: t=t+1
Until stop critierion is met
Output the best results and visualization (Jellyfish bloom)
End

Fig. 1. Pseudo code of JS algorithm [3].

B. Carnivorous Plant Algorithm

While most plants are food sources for animals, this is the opposite for carnivorous plants. Carnivorous plants survive in harsh conditions and hunt animals such as flies, butterflies, lizards and mice [22]. At this point, the Carnivorous Plant Algorithm (CPA) was proposed, inspired by the hunting behavior of carnivorous plants to survive in harsh conditions and their pollination adaptation for reproduction. The attractiveness, entrapment, digestion and reproduction strategies of carnivorous plants were modeled mathematically for optimization. Fig. 2. shows the pseudo code of the CPA algorithm. The details of the equations in the pseudo code can be found in the relevant article [22].

C. Giza Pyramids Construction (GPC)Optimizaion

The Pyramids of Giza are a building block consisting of three great pyramids built in ancient Egypt. According to archaeologists, the construction methods of these pyramids of different sizes are different from each other as the construction is spread over time. One of the most important issues in the construction of these pyramids was the issue of how to manage porters, slaves, masons, metalworkers and carpenters, each with different responsibilities in the field of construction. They were guided by the pharaoh's agent. In addition, the construction of the pyramids has been optimized due to the limitation of the construction materials used, the construction time problem and the stone blocks used. The GPC algorithm has emerged as an algorithm that was inspired by the methods, technologies and strategies of the period in which these pyramids were built [23]. The pseudo code of the GPC algorithm is given in Fig. 3. The details of the mentioned equations in the pseudo code can be found in the relevant article [23].

Define objective function $f(\vec{x})$, $\vec{x} = (x_1, x_2, \dots, x_d)$
Define iteration number within a group, *group_iter*,
attraction rate, *attraction_rate*, growth rate, *growth_rate*,
reproduction rate, *reproduction_rate,*
number of carnivorous plants, *nCPlant* and number of prey, *nPrey*
Initialize a population of n individuals with d dimensions randomly
Evaluate the fitness of each individual
Sort the individuals based on the fitness value
Identify the best individual, g^*as the first rank carnivorous

Repeat until stopping condition is met
Classify the top *nCPlant* individuals as carnivorous plants
Classify the remaining *nPrey* individuals as prey
Group the carnivorous plants and prey

//Growth process of the both carnivorous plants and prey
for i=0: *nCPlant*
for *Group_cycle*=1: *group_iter*
if *attraction_rate*> a generated random number
// The prey trapped and digested
Generate new carnivorous plant using Eq. (7)
else
// The prey escapes from the trap
Generate new prey using Eq. (9)
end for
end for

// Reproduction process of the first rank carnivorous plant
for *i*=1: *nCPlant*
Generate new carnivorous plant based on the first rank carnivorous plant using Eq. (11)
end for
Evaluate the fitness of each new carnivorous plant and new prey
Combine the previous and newly generated carnivorous plants and preys
Sort the individuals and select top n-ranked individuals to next generations
Identify the current best individual, g^*, as the
end while
Display the current best solution, g^*,

Fig. 2. Pseudo code of the CPA algorithm [22].

STEP 1:
Generate initial population array of stone blocks or workers (Population size)
Generate position and cost of stone block or worker
Determine best worker as Pharaoh's agent
STEP 2: for FirstIteration to MaxIteration **do**
STEP 3: for i=1 to n **do** (all n stone blocks or workers)
Calculate amount of stone block displacement using Eq. (4)
Calculate amount of worker movement using Eq. (7)
Estimate new position Eq. (8)
Investigate possibility of substituting workers Eq. (9)
Determine new position and new cost
if new_cost< Pharaoh's agent cost **then**
Set new_cost as Pharaoh's agent cost
end if
END STEP 3
Sort solutions for next iteration
END STEP 2
END STEP 1

Fig. 3. Pseudo code of the GPC algorithm [23].

III. EXPERIMENTAL STUDY AND RESULTS

In this section, benchmark functions used to compare current metaheuristic optimization algorithms are explained and the results obtained by using these functions are given in detail.

TABLE I. BENCHMARK FUNCTIONS

Function	**Equation**	**Feature**	**Value Range**	**Optimum**
Sphere (F1)	$\sum_{i=1}^{d} x_i^2$	Unimodal	±100	0
Rosenbrock (F2)	$\sum_{i=1}^{d-1}[100(x_i^2 - x_{i+1})^2 + (1 - x_i^2)]$	Unimodal	±30	0
Ackley (F3)	$20 + e - 20\,exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)$	Multimodal	±32	0
Levy (F4)	$sin^2(\pi y_1) + \sum_{i=1}^{d-1}(y_i - 1)^2[1 + 10\,sin^2(\pi y_{i+1})] + (y_n - 1)^2, y_i = 1 + \frac{x_i - 1}{4}$	Multimodal	±10	0

A. Benchmark Functions

There are many benchmark functions in the literature in order to evaluate the performance of metaheuristic optimization algorithms. These functions have the difficulty and complexity of real life engineering problems. These functions are often used to evaluate and compare optimization algorithms for convergence, precision, robustness, and overall performance. The natural functions, complexities and other properties of these comparison functions can be obtained from their definitions and the difficulty levels of these functions can be adjusted by changing the size and interval parameters [24-25]. In this study, four benchmark functions, Sphere (F1), Rosenbrock (F2), Ackley (F3) and Levy (F4) were used. These functions and their properties are shown in Table I.

B. Experimental Results

All experimental studies were performed on the MATLAB 2020b platform. For all metaheuristic algorithms compared in the study, the initial population numbers were taken as 30 and the maximum number of iterations as 1000. Algorithms are tested in 3 different difficulties by taking the dimension values of the functions as 10, 30 and 50. The performances of the algorithms for each test function were evaluated according to the results obtained in 30 independent run. Comparative test results obtained are given in Table II for 10 dimensional functions, Table III for 30 dimensional functions and Table IV for 50 dimensional functions. In the tables, the mean, best and standard deviation values obtained in 30 independent runs are given. Mean and best values represent the performance of algorithms at convergence to global optimum. Standard deviation indicates whether the algorithm works stable or not. Average run times (sec.) obtained at 1000 iterations in 50 dimensional test functions are given in Table V.

TABLE II. RESULTS OBTAINED IN 10 DIMENSIONAL TEST FUNCTION

		JS	CPA	GPC
F1	Mean	**2.63E-84**	3.50E-76	8.02E-25
	Best	**4.43E-94**	6.80E-84	1.31E-29
	Std.	**9.86E-84**	1.68E-75	3.12E-24
F2	Mean	**0.06227**	2.73034	7.19726
	Best	**6.14E-08**	0.03693	6.54867
	Std.	**0.17952**	3.29056	0.26851
F3	Mean	**1.48E-15**	0.17039	1.64E-13
	Best	**8.88E-16**	4.44E-15	**8.88E-16**
	Std.	**1.32E-15**	0.44129	2.30E-13
F4	Mean	**5.71E-27**	0.11120	0.60254
	Best	2.10E-30	**1.50E-32**	0.45402
	Std.	**1.95E-26**	0.34592	0.07917

TABLE III. RESULTS OBTAINED IN 30 DIMENSIONAL TEST FUNCTION

		JS	CPA	GPC
F1	Mean	**1.18E-46**	1.66E-15	4.60E-22
	Best	**2.40E-75**	1.35E-17	3.74E-26
	Std.	**6.34E-46**	5.94E-15	1.10E-21
F2	Mean	**0.10381**	59.63437	27.43257
	Best	**4.58E-05**	4.01466	26.85205
	Std.	0.43619	59.80774	**0.20578**
F3	Mean	**3.49E-15**	0.86503	3.27E-12
	Best	**8.88E-16**	1.45E-09	1.87E-14
	Std.	**1.57E-15**	1.05580	3.58E-12
F4	Mean	**1.24E-09**	4.17400	2.30824
	Best	**2.73E-12**	0.63338	2.10798
	Std.	**1.74E-09**	2.01820	0.08005

TABLE IV. RESULTS OBTAINED IN 50 DIMENSIONAL TEST FUNCTION

		JS	CPA	GPC
F1	Mean	**1.21E-50**	0.00193	2.27E-21
	Best	**5.55E-66**	9.26E-07	5.10E-26
	Std.	**6.54E-50**	0.00915	5.22E-21
F2	Mean	**0.04310**	123.45188	47.58939
	Best	**0.00076**	27.19353	47.03578
	Std.	**0.08878**	63.91115	0.40007
F3	Mean	**3.97E-15**	2.67599	6.41E-12
	Best	**8.88E-16**	0.01144	3.95E-13
	Std.	**1.21E-15**	1.04502	5.96E-12
F4	Mean	**3.28E-07**	8.87983	4.11957
	Best	**3.19E-09**	1.99635	3.99599
	Std.	**6.14E-07**	4.30005	0.07264

TABLE V. AVERAGE RUN TIMES (SEC.) OBTAINED AT 1000 ITERATIONS IN 50 DIMENSIONAL TEST FUNCTIONS

Function	JS	CPA	GPC
F1	**0.6006**	10.4534	3.6321
F2	**0.6591**	9.8305	3.6458
F3	**0.5997**	10.4969	3.3960
F4	**1.9460**	11.0680	4.6248

IV. CONCLUSION

Metaheuristic algorithms are algorithms that provide near-optimum solutions in acceptable times for many optimization problems, can be easily adapted to those problems without making changes on the problems of interest like classical algorithms, and can offer solution strategies even in the presence of different types of decision variables and constraints. Because of these advantages, researchers have proposed many metaheuristic algorithms in the last few decades and are still being added to the literature in new algorithms. In this study, the performances of JS, CPA, and GPC, which are among the newest metaheuristic algorithms,

were tested in detail using 4 benchmark test functions of 10, 30 and 50 dimensions. According to the test results, it was seen that the JS, which gave the best solutions in all benchmark functions, was the most successful algorithm. JS algorithm gave the best results in terms of run time. The worst performance was shown by CPA. This study is important for researchers to easily access some of the most up-to-date metaheuristic optimization algorithms, as well as to help researchers choose the appropriate algorithm by giving them a preliminary idea about the performance of metaheuristic algorithms that they can use in their studies.

REFERENCES

[1] S. Mirjalili and A. Lewis**,** 2016. The whale optimization algorithm. *Advances in engineering software*, *95*:51-67.

[2] S. Gao and C.W. de Silva, 2018. Estimation distribution algorithms on constrained optimization problems. *Applied Mathematics and Computation*, *339*:323-345.

[3] J. S. Chou and D. N. Truong, 2021. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, *389*:125535.

[4] J. H. Holland, 1992. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

[5] R. Storn and K. Price, 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4):341-359.

[6] S.I. Birbil and S.C. Fang, 2003. An Electromagnetism-like Mechanism for Global Optimization. *Journal of Global Optimization,* 25:263-282.

[7] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, 2009. GSA: a gravitational search algorithm. *Information sciences*, 179(13):2232-2248.

[8] B. Alatas, 2012. A novel chemistry based metaheuristic optimization method for mining of classification rules. *Expert Systems with Applications*, *39*(12):11080-11088.

[9] J. Kennedy and R, Eberhar, 1995. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE.

[10] D. Karaboga and B. Akay, 2009. A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, *214*(1):108-132.

[11] X. S. Yang and A. H. Gandomi**,** 2012. Bat algorithm: a novel approach for global engineering optimization. *Engineering computations*, 29(5):464-483.

[12] **A**. H. Kashan, 2014. League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. *Applied Soft Computing*, 16:171-200.

[13] E. Osaba, F. Diaz and E. Onieva, 2014. Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. *Applied Intelligence*, 41(1):145-166.

[14] K. S. Lee and Z. W. Geem, 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer methods in applied mechanics and engineering*, 194:3902-3933.

[15] S. M. Ashrafi and A. B. Dariane, 2011**.** A novel and effective algorithm for numerical optimization: melody search (MS). *11th International Conference on Hybrid Intelligent Systems* (HIS) (pp. 109-114). IEEE.

[16] A. Borji and M. Hamidi, 2009. A new approach to global optimization motivated by parliamentary political competitions. *International Journal of Innovative Computing, Information and Control*, 5(6):1643-1653.

[17] R. V. Rao, V. J. Savsani and D. P. Vakharia, 2012. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences*, *183*(1):1-15.

[18] X. Qi, Y. Zhu, H. Chen, D. Zhang and B., Niu, 2013. An idea based on plant root growth for numerical optimization. In *International Conference on Intelligent Computing* (pp. 571-578), Springer, Berlin, Heidelberg.

[19] Ü. Can and B. Alataş, 2015. Bitki zekâsında yeni bir alan: kök kütlesi optimizasyonu. *Türk Doğa Ve Fen Dergisi*, *8*.

[20] X. S. Yang, 2012. Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation* (pp. 240-249). Springer, Berlin, Heidelberg.

[21] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, 2020. Gradient-based optimizer: A new Metaheuristic optimization algorithm. *Information Sciences*, *540*:131-159.

[22] K. M. Ong, P. Ong and C. K. Sia, 2021. A carnivorous plant algorithm for solving global optimization problems. *Applied Soft Computing*, 98:106833.

[23] S. Harifi, J. Mohammadzadeh, M. Khalilian and S. Ebrahimnejad, 2020. Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization. *Evolutionary Intelligence*, 1-19.

[24] B. Alatas, E. Akin, A. B. Ozer, 2009. Chaos embedded particle swarm optimization algorithms, *Chaos, Solitons and Fractals*, 40(4):1715–1734.

[25] S. Kızıloluk and A. B. Özer, 2016. Melez elektromanyetizma benzeri-parçacık sürü optimizasyon algoritması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 7(3):515-526.

Comparative Analysis Of Machine Learning Algorithms For Mapping Of Debris Coverd Glacier Through Remote Sensing Data:A CaseStudy Of Hunza Basin

Rahila Parveen
Department of Computer Sciences
Karakrum Inernational University
Gilgit Baltistan, Pakistan

Aftab Ahmad Khan*
Department of Computer Sciences
Karakrum Inernational University
Gilgit Baltistan, Pakistan
E.mail. aftab.ahmed@kiu.edu.pk

Dostdar Hussain
Department of Computer Sciences
Karakrum Inernational University
Gilgit Baltistan, Pakistan

***Abstract*— The Himalayas are one of the main clusters of glaciers outside the polar regions, but the fate of the Hindu Kush Himalayan glaciers has been the subject of heated debate due to its quick melting and retreat. As glaciers are the best signs of climate change and water resources. Despite the significance of glaciers, there is still a lack of detailed and consistent glacier inventory data in most parts of the world. Mapping, monitoring and inventory of glaciers at a universal scale is therefore very important. In view of the enormity and inaccessibility of mountain glaciers, remote sensing is probably the only way of re-mapping glaciers in an organized and cost-effective manner. In this research work, three supervised machine learning algorithms are used for automatic mapping of glacier, debris filled glacier and non-glaciated areas using multi-temporal sentinel-2 data. The machine learning algorithms used in this study are support vector machine (SVM), artificial neural network (ANN) and random forest (RF). In order to classify, the high altitude shisper glacier of the Hunza Basin was chosen as an area of interest, and three main classes were deliberated, i.e. Glacier, glacier-covered debris and glacier-free area. The data was divided into the training set (70 percent) and the test set (30 percent). Finally, the accuracy of each classifier is compared to the referenced data for assessment. The result suggested that each classifier classifies the class in a very accurate manner. Comparatively, random forest (RF) performed the best in all three classes relative to ANN and SVM. This precise classification of the debris covered-glacier will help in assessment of water sources, climate change and for hazard management.**

Keywords*— *Remote Sensing Sentinel-2, Support Vector Machine(SVM), Artificial Neural Network (ANN), Random Forest(RF)

I. INTRODUCTION

The Hindu-Kush-Karakoram-Himalaya area (HKKH), also referred to as "Asian Water Towers," [1] is among the largest glacier cover outside the northern hemisphere. It contains several highest peaks and largest glaciers [2]. It is the greatest freshwater resource since it includes the largest collection of snow and glaciers.

Glaciers are the best indicators of climate change, and their meltwater benefits irrigation system and fills other water requirements [3]. As Pakistan is an agricultural country and its economy is heavily dependent on irrigation systems, climate change affects water resources and disrupts the agricultural system and power capacity. Water flow variations also create problems among provinces in terms of reducing water flows during dry seasons and higher flows resulting in flooding during wet seasons [4]. Thus, any minor change in glaciers will affect the population and may disorder their living standards in several ways. Debris filled glaciers often play a significant role in the balance of the glacier mass and the melting rate of ice .Therefore, it is important to observe, monitor and record glaciers to obtain imparts on climate change , water resource management and understand the dynamics and reactions of glacier [5]. Traditional glacier mapping approaches based on field surveys and analyzing topographical maps, which are very time consuming and not feasible in remote areas particularly in mountain areas. Thus, there is still a lack of accurate data of glaciers in most parts of the world [6].

In view of the vastness and unreachable nature of mountain glaciers, remote sensing is the most efficient method for accurately mapping glaciers [7]. Several remote sensing techniques have been used to map and monitor glaciers. These methods include splitting between visual interpretation and the band ratio of remote sensed images. In addition, machine learning algorithms that include supervised algorithms, unsupervised algorithms and decision tree methods are also proposed for the detection of specific information [8][5]. Similarly, glaciers filled with debris are identified using band ratio and manual on-screen digitization. Many researchers have introduced new semi-automatic methods for high-precision mapping of glaciers, but these techniques have not been applied to glaciers covered by debris due to the spectral properties of supraglacial debris and periglacial debris [9]. Few researchers focused on debris covered glacier through onscreen manual digitization. However, the accuracy of manual digitization of debris-covered glaciers varies on the basis of image resolution and personal expertise resulting in different mappings for the same glacier. To address this challenge, we use machine learning algorithms to classify glaciers, glacier-covered debris and non-glaciated areas from remote sensed data. The aim of this study is to estimate the performance and importance of machine learning algorithms in remote sensing for the classification of debris-covered glacier.

II. Study Area

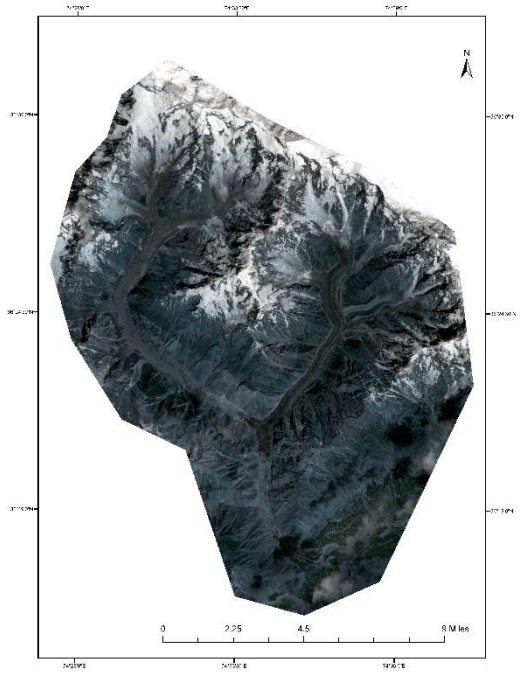


Figure 1. Shisper glacier as selected study area.

The study area is shisper glacier, located in the Hunza Valley, northern Pakistan (Fig. 1). Shisper is a 16.5 km long surge-type glacier located in the Karakorum region (Lat: 36.35-36.48° N; 74.57-74.61° E) in Hunza Valley, Pakistan, spanning 26 km.

III.Material

Various types of data have been obtained, processed, and analyzed using various methods and techniques. These different types of data include sentinel 2 Imagery, Landsat 8 imagery and Shape file of boundary of study area (Table 2).

Table 2. Description of data used.

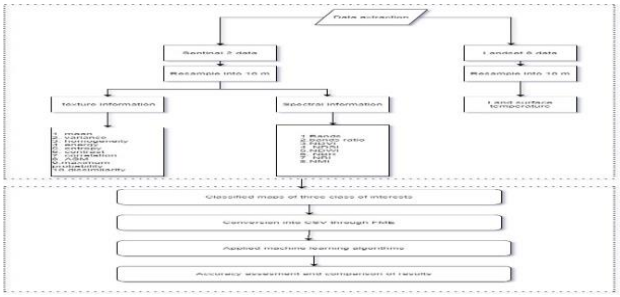
Data Type	**Source**	**Spatial Resolution**	**Revisit Time**	**Recording Time Period**	**Spectral Bands**
Sentinel	Google Earth Engine Website (Supported by USGS)	10, 20, 60 (m)	5 Days	(20-7-2019)	13 Bands
Landsat 8	Google Earth Engine Website (Supported by USGS)	30 (m)	16 Days	(20-7-2019)	9 Bands

A. Sentinel and Landsat data product

Sentinel-2 offers high-resolution multi-spectral imagery with limited time to review and produce vast amounts of information. In addition, Sentinel-2 tracks thirteen separate electromagnetic spectrum frequencies, from a minimum of about 440 NM to a maximum of about 2190 NM, with a spatial resolution of 10, 20 and 60 m[10]. The sentinel data is in the form of a grid tile with a spatial resolution of 10 meters and 20 meters, while the Landsat imagery has a spatial resolution of 30 meters, which has been re-sampled to 10 meters, making it easier to use thermal data with the sentinel dataset. In addition, sentinel images were also re-sampled to 10 meters with a spatial resolution of 20 meters and processed in ArcGIS. The images we used have already been rectified and incorporated from the raw digital numbers (DNs) to the top of the atmosphere (TOA) using GEE radiometric parameters.

.

Figure 3. 1: Methodology Flowchart



IV. Methodology

The complete workflow of this method is shown in the figure 2. This process is made up of three steps. A collection of features was produced at the initial stage. These characteristics include spectral characteristics and texture characteristics extracted using ArcGIS. The spectral characteristics include reflectance indexes of all required bands, normalized indexes of vegetation variations, water indexes and many more. In addition, mean, variance, homogeneity, contrast, dissimilarity, entropy, energy, correlation, and angular momentum are included in the texture function. Along with these two characteristics, the surface temperature of the Earth from Landsat-8 imagery band-10 was included. The FME workbench of Safe Software was used in the second phase to convert all extracted features to CSV files and to merge all files for a single dataset to be generated.

In the last phase, the training set was applied to three commonly used machine learning algorithms to classify each class, i.e. land, ice and glacier, and the accuracy assessment was carried out using the reference data to determine the performance of each classifier.

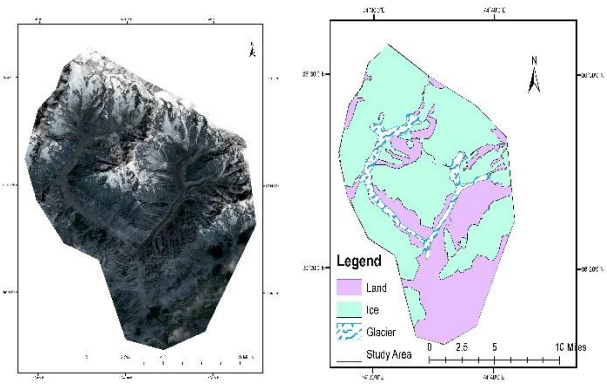


Figure 3. (a) The area of interest, from where the training and test dataset were extracted. (b). The reference image for the area of interest showing three classes: glaciers, debris-covered glacier and non-glacier areas.

A. FEATURE EXTRACTION

We extracted two types of features which are texture characteristics and spectral characteristics. The details of each feature is given below.

1) TEXTURE FEATURES

Texture provides data on the spatial arrangement of the color or intensity of an image. Texture is categorized by the spatial level distribution that often co-occurs in a picture with different combinations of gray levels. Gray Degree Co-occurrence Matrix1 (GLCM) is a technique for image processing. It is an estimate of how often in a picture different combinations of gray levels co-occur. Textural characteristics were extracted from the co-occurrence matrix of the gray level given by the moving 3 x 3 window, including mean, variance, homogeneity, contrast, dissimilarity, entropy, energy, correlation, and second angular momentum. By using SNAP software, we collect these textures [11].

2) SPECTRAL FEATURES

The spectral features involve reflectance indexes of all necessary bands, normalized vegetation difference indexes, water indexes and barren index. The spectral features were extract directly from the required ten bans which include Blue, Green, Red, Vegetation Red Edge 1, Vegetation Red Edge 2, Vegetation Red Edge 3, NIR, Vegetation Red Edge 4, Shortwave Infrared 1, Shortwave Infrared 2. Sseveral water indices were calculated by using these bands. the importance indexes are NDVI, NDSI, NDWI, NMI and NBR. We calculated all these indexes using the formula of each index. The formula of each index is given below.

$$NDWI = (Green - NIR)/ (Green + NIR)$$

$$NDVI = (NIR - Red)/ (NIR + Red)$$

$$NDSI = (Green - SWIR)/ (Green + SWIR)$$

$$NMI = (NIR - SWIR1)/ (NIR + SWIR1)$$

$$NBR = (NIR - SWIR)/ (NIR – SWIR)$$

A. FEATURE ANALYSIS USING MACHINE LEARNING

Total 80 features were extracted which include spectral features (10 bands, land surface temperature, NDVI, NDSI, NDVI, NMI, NDWI, NBR) and texture information which include six features of each bands and Sar information. We used python to apply machine learning algorithms. Since it offers concise, readable code, and consists of a lot of code libraries for ease of use. Its syntax is simpler and easier to understand. Here three machine learning algorithms have been trained and tested on processed data, i.e. SVM, ANN and RF. Each classifier was independently applied to the data that generated a classification map for each class, i.e. Glacier debris-covered glacier and non-glacier area. A sample of dataset is given below

CLASSIFICATION

1). SUPPORT VECTOR MACHINE (SVM) CLASSIFICATION

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression problems [12]. However, it is mainly used for classification problems. In the SVM algorithm, each data object is plotted as a point in n-dimensional space (where n is the number of features, we have), with the value of each feature being the value of a specific coordinate. Then we perform classification by finding the hyper-plane which differentiates the classes very well. SVM's primary objective is to divide the datasets into classes in order to find a maximum marginal hyperplane (MMH) and it can be achieved in the following two steps. First, SVM can iteratively produce hyperplanes that best segregate the classes. Then the hyperplane that correctly divides the classes will be chosen. In practice, the SVM algorithm is implemented with a kernel that converts the input data space into the appropriate form. SVM uses a kernel trick technique in which the kernel takes a small dimensional input space and converts it into a larger dimensional space. In simple terms, by introducing more dimensions, the kernel transforms non-separable problems into separable problems. It makes it more effective, scalable and precise for SVM.

2). RANDOM FOREST (RF) CLASSIFICATION

Random Forest is a popular machine learning algorithm which is part of a supervised learning technique [13]. It can be used for both classification and regression problems in ML. It is based on the principle of Ensemble Learning, a method of combining multiple classifiers to solve a complex problem and improve the output of the model. Random Forest is a classifier that contains a number of decision trees on various dataset subsets and uses the average to improve the predictive accuracy of the dataset. Instead of relying on a single decision tree, the random forest makes predictions from each tree based on the majority of predictions and predicts the final results. The greater number of trees in the forest leads to higher precision and prevents over-fitting problems. Hyper parameters in the random forest are either used to improve the predictive capacity of the model or to speed up the model. We used some essential hyper parameters to improve predictive capacity. Firstly, there is the hyper parameter of the n estimators, which is just the number of trees the algorithm produces before the maximum vote is taken or the average forecast is taken. Another essential hyper parameter is max features, which is the maximum number of random forest features considered to be a node split. The last major hyper

parameter is min sample leaf. This specifies the minimum number of leaves required to divide the internal node.

3). ARTIFICIAL NEURAL NETWORK (ANN) CLASSIFICATION

An artificial neural network is a network of basic components called neurons that receive input, alter their input, and modify their components, internal state (activation) by that input, and generate output by input and activation [14]. By connecting the output of some neurons to the input of other neurons, the network is guided and directed Graph. ANNs are non-linear statistical models that demonstrate a complex relation between inputs and outputs in order to explore a new pattern. The computational complexity of ANN training is comparatively higher, but it can learn to classify complex nonlinearly separable data with high accuracy. Some parameters, such as the number of hidden layers and the number of neurons in each layer, must be defined during the ANN design and training. These features can change after the individual application. There is no general and transparent method for selecting these parameters. It uses almost a trial and error approach, but takes more computing time and is not an efficient measure. It is therefore necessary to suggest an approach in order to find an optimal combination of parameters that will influence the efficiency of the ANN

V. EXPERIMENTAL RESULTS

The classification of glacier, debris covered glaciers and non-glaciated areas through machine learning algorithms was carried out. Moreover, the performance of machine learning algorithms was examined on the same dataset and investigated the accuracy of each classifier. Supervised learning models need well labeled input data to be trained on it and give expected output. Training is the backbone of supervised learning. During its training process, the system is fed with vast quantities of data, which informs the system what output should be obtained from each particular input value. To check the outcome of the training and calculate the accuracy, the trained model is then presented with test data. Moreover, supervised learning depends on many parameters and the parameter selection is the most crucial stage. Therefore, we manually select training samples for each class of interest and employed grid search technique for finding optimal parameters to each class of interest which were used in data training and testing for each class. The following section will describe the implementation details.

B. TRAINING SAMPLE SELECTION

While supervised learning required labeled training sample for each class of interest. Therefore, we have collected training samples for each class i.e. glacier, debris covered glacier and non-glaciated area. In total we collected 22,75450 pixel points, of which 273052 points were labeled as debris covered glacier, 953823 as glacier and 1048575 as non-glaciated areas. Furthermore, the collected dataset was split into training set (70%) and testing set (30%). As a whole 1592815 pixels were used for training set and 682635 pixels were used as testing set respectively. Training samples were collected using stratified sampling, whereby the samples were widely distributed across the entire field of study and reflected, without prejudice, each class of interest.

C. MODEL OPTIMIZATION AND TUNING

Optimization or tuning of the model in machine learning is the most critical step in improving accuracy. For the best tuning parameter values, the analysis can be carried out in many ways, but most fall into two key categories: those

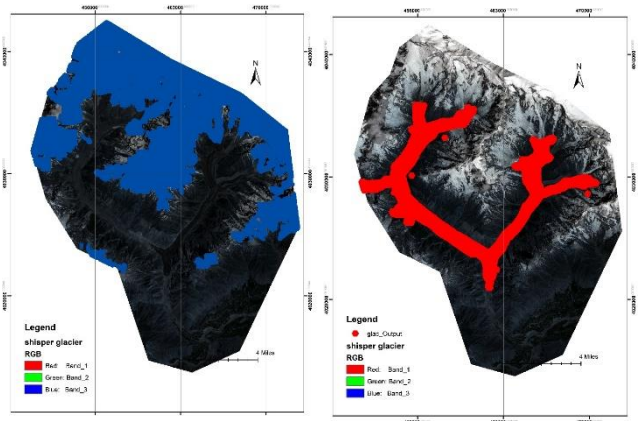


Figure 4. Labelling of three classes.

that predefine which values to measure and those that incrementally determine the values. We used identify the best combination of parameter values for the model. The sklearn library offers a method that allows us to define a set of possible values that we want to try on the data for the given model and trains on the data and identifies from a combination of parameter values the best estimator [15]. For the development and implementation of models for all three ML algorithms, we used k-fold cross-validation (SVM, ANN, RF). Optimal values for parameters were obtained by cross validation in the original training data collection. Finally, the set of parameters for which the model produced optimal results was selected for model evaluation. Basic Parameter Information for each classifier tuning using grid-search is described underneath.

1). SVM MODEL PARAMETER SELECTION

To apply SVM meritoriously and accurately we used RBF (radial basis function) kernel which is most popular and commonly used in SVM. The RBF kernel is a function whose value depends on the distance from the source or the point from which it originates.it used the following format for two data points Xi and Xj: K (xi, xj) = exp (-γ || xi - xj||2) where γ is gamma, which represents the width of the kernel, as this value of the gamma increases, the model results in overfitting and if the value of the gamma decreases, the model gets under fits. Another vital parameter of the RBF kernel is cost C, which is used to standardize error of misclassification and γ. The C parameter trades against optimizing the margin of decision-making from the correct classification of training examples. A smaller margin for higher C values would be accepted if the decision making feature were easier to correctly identify all training points. At the expense of training precision, a lower C will encourage a greater margin, hence a simpler decision making feature. In other words, "C" is the regularization parameter in the SVM. Using k-folds to choose the best values for C and γ Cross-validation (CV) data is divided into k subsets (k=10). One subset is used to evaluate data while remaining k-1 training subsets are used for training purposes.

2). ANN Model PARAMETER SELECTION

Although the ANN classifier has a relatively higher number of classifiers, we have chosen to fine-tune only the

momentum and learning rate. Other parameters have been set to the immovable value. As the number of iterations=1000, as activation, sigmoid as activation function, and scientifically calculated the number of hidden iterations. The neurons must be 200 in the secret layer. The rate of learning was Checked between 0.001 and 0.5 while the 25 momentum is between 0 and 0.9. The values that provided maximum accuracy were 0.1 and 0.8 correspondingly for learning rate and momentum.

3). RF MODEL PARAMETER SELECTION

There is the hyper parameter of the n estimators in RF, which is just the amount of trees generated by the algorithm before taking the maximum vote or taking the average forecast. Max features, which is the maximum number of random forest features deemed to be a node split, is another critical hyper parameter. Min sample leaf is the last major hyper parameter. This specifies the minimum number of leaves required for the internal node to be broken. To boost the speed of the model, we have the following parameters. The hyper parameter for N jobs informs the engine how many processors it can use. When it has a value of one, only one processor can be used. To say that there is no boundary, we used the value of '-1'. The hyper parameter of the random state renders the model output repeatable.

D. CLASSIFICATION RESULTS AND POST PROCESSING

The same collection of features was extracted from the test data after selecting the best parameters and training the models and fed into the classifiers being trained. The classifier created a classification map for one of the three classes by assigning the classifier, with pixels defining a class. In the background class, the unclassified pixels were allocated. For better interpretation, the findings obtained were labeled with distinct color as shown in Figure.4.

E. ACCURACY ASSESSMENT

In this section the efficiency of the proposed method has been evaluated in form of Total Accuracy (OA), Precision (P), Recall (R) and f1-score for delineation of glacier, debris covered glacier and non-glaciated area. A comparison between the results obtained and the reference data was conducted to determine the classification accuracy. 2275450 were the total pixels chosen for the accuracy evaluation. By delineating the study area of the three groups, the reference data for the selected area was prepared manually. The classification accuracy obtained from the three selected supervised machine learning algorithm is summarized in Table 4 and fig 5.

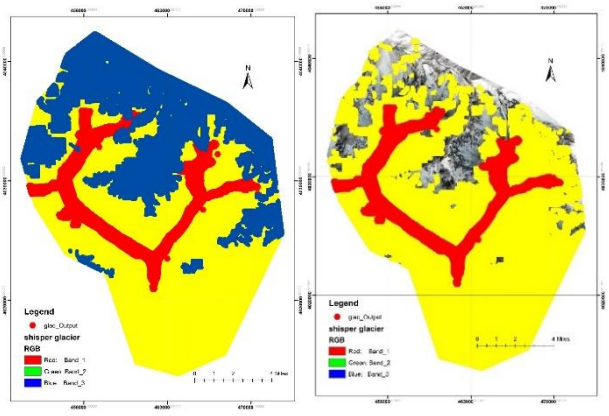


Fig 5. Classification results.

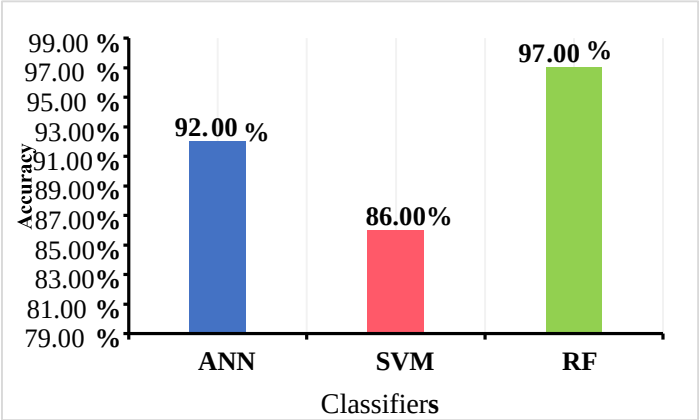


Fig 6. Accuracy of classifiers.

Table 4. Accuracy assessment of three algorithms for classification of glacier, debris-covered glacier and land

Cover-type	Classifier	Precision	Recall	f1_score	Accuracy
Debris-Covered glacier	RF	97.00	99.00		97.00
Glacier	RF	97.00	99.00		97.00
Land	RF	96.00	97.00		97.00
Debris Covered glacier	ANN	94.00	93.00		92.00
Glacier	ANN	90.00	94.00		92.00
Land	ANN	93.00	89.00		92.00
Debris Covered glacier	SVM	91.00	90.00		88.00
Glacier	SVM	89.00	91.00		88.00
Land	SVM	87.00	88.00		88.00

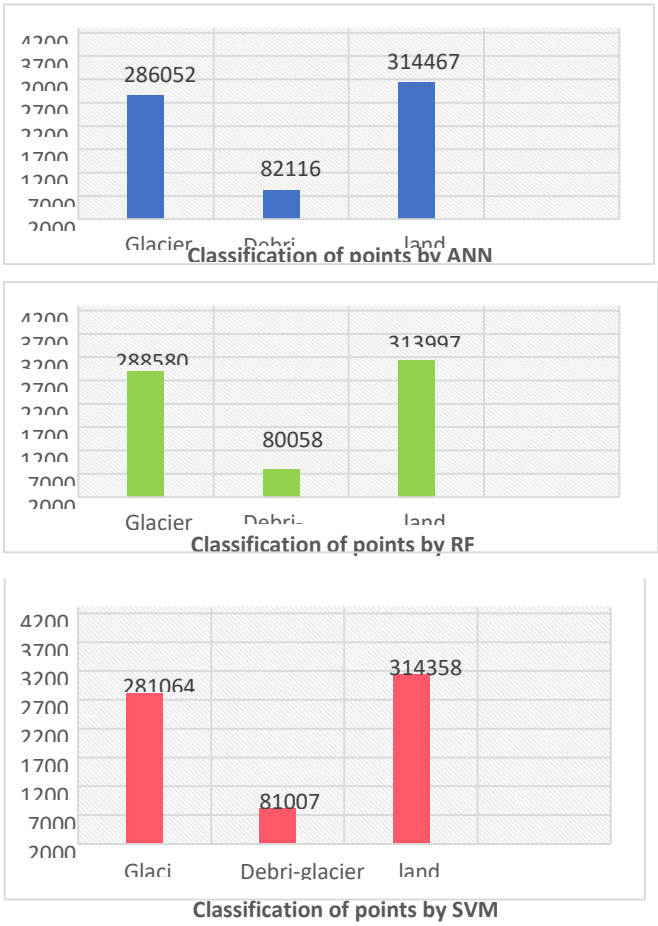


Fig 6. Classification of each classifier.

For each classifier, the table and graphs illustrate good results. For RF, ANN and SVM classifiers, the f1-score for the glacier class is 97, 92% and 90%. Likewise, 98%, 94% and 91% were created for the debris-covered glacier class by the same classifiers. Additionally, the classification accuracy for the non-glacier region class was 96%, 91% and 86% respectively. In fact, all three classifiers created good classification accuracy, but for all three classes, RF provided the highest classification accuracy compared to SVM and ANN. While the least classification accuracy was provided by SVM. Due to the fact that they have high spectral similarities between them, there were more miscellaneous classifications for SVM and ANN classifiers, especially for the class of non-glaciated and debris covered glacier. Figures also show that the findings of the classification for all three classifications showed good consistency across the entire region of the analysis. although, there were some confusion in some areas between non-glaciers and glaciers with debris. Due to spectral similarity, the distinction between the two groups can be very difficult to find. Also, due to clouds in the photos in some areas, there were some factual inaccuracies. Although we chose the image with minimal cloud effect, several misclassifications resulted.

VI. Conclusions

The primary objective of this research was to analyze the significance of machine learning algorithms using remote sensing data to detect glacier and glacier-covered debris and non-glaciated areas. We have used publicly available multi-temporal sentinel 2 data for this purpose. Shisper (Hunza Basin) was chosen as a research area located in northern Pakistan. This research was done using three of the most commonly used machine learning algorithms to achieve our goal, i.e. SVM, ANN, RF. Furthermore, three interest categories were defined: glacier, debris covered by glaciers and non-glaciated areas. For training purposes, the whole study area was used to collect samples and then a collection of spectral and texture characteristics was obtained. All the features were grouped together and individually fed into each classifier in order to classify. For the three classes of interest, classifiers then generate classification outcomes. Moreover, for validation purposes, the results of the classification have been compared with the referenced data. The results show, after the validation process, that RF is relatively more precise than SVM and ANN. Compared to the reference data, the experimental results were more accurate. When manually classifying debris filled glacier and ground, some problems were seen because they looked identical with bare eyes. Finally, we conclude that the three SVM, ANN, and RF ML models are effective in mapping glacier-covered debris and glacier on remote sensing data that other methods have not accurately mapped. However, as they enhance the performance analysis of geospatial data in a more precise way, ML algorithms are regarded as core modules in the GIS and RS disciplines. In the identification of water sources and glacier inventory details, our highly detailed classification method will help. We would like to integrate this methodology into the region's seasonal imagery in the future and recognize improvements that have taken place year after year. So, we can foresee the glacier's future and take precautions for the time being.

REFERENCES

[1] G. Tolt, M. Shimoni, and J. Ahlberg, “A shadow detection method for remote sensing images using VHR hyperspectral and LIDAR data,” in *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2011, pp. 4423–4426.

[2] T. D. Acharya, A. Subedi, and D. H. Lee, “Evaluation of machine learning algorithms for surface water extraction in a landsat 8 scene of nepal,” *Sensors (Switzerland)*, vol. 19, no. 12, 2019.

[3] S. R. Bajracharya and P. Mool, “Glaciers, glacial lakes and glacial lake outburst floods in the Mount Everest region, Nepal,” 2009.

[4] S. Surveying, “an Index-Based Shadow Extraction Approach on High-Resolution,” 2011.

[5] A. A. Khan, A. Jamil, D. Hussain, M. Taj, G. Jabeen, and M. K. Malik, “Machine-Learning Algorithms for Mapping Debris-Covered Glaciers: The Hunza Basin Case Study,” *IEEE Access*, vol. 8, pp. 12725–12734, 2020.

[6] K. Hewitt, “The Karakoram Anomaly? Glacier Expansion and the ‘Elevation Effect,’ Karakoram Himalaya,” *Mt. Res. Dev.*, vol. 25, no. 4, pp. 332–340, 2005.

[7] "Gomathi, M., Geetha Priya, M., & Krishnaveni," in *Proceedings - 39th Asian Conference on Remote Sensing: Remote Sensing Enabling Prosperity, ACRS 2018*, 2018, pp. 3277–3284.

[8] D. Hussain and A. A. Khan, "Machine learning techniques for monthly river flow forecasting of Hunza River, Pakistan," *Earth Sci. Informatics*, 2020.

[9] W. E. Grabs and J. Hanisch, "Objectives and prevention methods for glacier lake outburst floods (GLOFs)," IAHS Publ, 1993.

[10] H. Yang, Z. Wang, H. Zhao, and Y. Guo, "Water body extraction methods study based on RS and GIS," *Procedia Environ. Sci.*, vol. 10, no. PART C, pp. 2619–2624, 2011.

[11] M. Zühlke *et al.*, "SNAP(Sentinel appliation platform) and the ESA SENTINEL 3 Toolbox," *Sentin. Sci. Work.*, vol. 734, p. 21, 2015.

[12] W. S. Noble, "What is a support vector machine?," *Nature Biotechnology*, vol. 24, no. 12. Nature Publishing Group, pp. 1565–1567, Dec-2006.

[13] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, Jan. 2005.

[14] J. J. Hopfield, "Artificial Neural Networks," *IEEE Circuits Devices Mag.*, vol. 4, no. 5, pp. 3–10, 1988.

[15] A. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. 2016.

Landslide Hazard Assessment For Karakorum Highway Using Machine Learning and Deep Learning Approaches

Wajid Hussain
Department of Computer Sciences
Karakrum Inernational University
Gilgit Baltistan, Pakistan

Aftab Ahmad Khan*
Department of Computer Sciences
Karakrum Inernational University
Gilgit Baltistan, Pakistan

Israr Hussain
Department of Computer Sciences
Karakrum Inernational University
Gilgit Baltistan, Pakistan

Abstract*— Mass movements such as landslides are a crucial natural hazard in mountainous regions all over the world. One of the most vulnerable areas of Pakistan to landslide accidents is Gilgit-Baltistan. The great variability of domestic geological factors combined with the difficulties of precipitation forecasting and the level of events associated with the earthquake, the need for organized procedures and techniques for measuring and forecasting the danger and the harmful effect of slop failures. In this paper we use numerous machine and deep learning-based approaches for the classification of land sliding hazard assessment on KKH (Karakorum Highway). Four machines learning and deep learning algorithms, including SVM (support vector machine), (RF) random forest, Decision Tree and CNN (convolutional neural network), are utilized and evaluated on each database. By using machine learning and deep learning approaches, the proposed landslide identification method shows outstanding robustness and great potential in addressing the landslide classification problem. Our experimental results demonstrated that RF might perform better significantly in this case.

Keywords—Landside classification, Machine learning, Deep learning, CNN, SVM, Decision Tree and RF

I. INTRODUCTION

Landslides identification plays a significant role in landslide risk assessment and management. Landslides are one of the hilly and mountainous regions with the most widespread and common natural hazards. Where loss of life, collateral destruction, and significant damage to the community and road and highway foundations can be caused. In Karakoram Mountains eight numerous kinds of mass movements have been recognized, avalanche, rock falls, debris flow, rockslides, rotational slip, flow slides, slumps and creep [1].

Some of the prior studies [2], [3], [4], [5] have been carried out and tried to addressed the problem of landslide classifications in different regions of Gilgit Baltistan by using traditional approaches. After numerous successful and approved studies there are still many of practical applications and opportunities ahead that need to study further for real world applications using deep learning-based techniques. Moreover, no such study has been carried out in the region using such advance machine learning and deep learning-based approaches.

Machine learning and deep learning approaches have been proved to be a powerful and promising tool in many geotechnical applications [10], [11] as well as in landslide classification. In 2012 Moosavi et al. [6] compared ANN, SVM with object-oriented techniques in creating landslide inventories. Later on, Van Den Eeckhaut et al. [7] utilized data segmentation and SVM to classify forest landslides with DMT derivatives. Li et al. [8] used SVM and random forests (RF) to identify forested landslides in the TG (Three Gorges) area of China including DTM derivatives based on object-oriented methods. In 2016, Ding at al. developed an automatic landslides recognition technique using CNN and texture change detection using pre- and post-landslides optical images. Recently in 2019 Ghorbanzadeh et al [9]. analyzed the performance of ANN, SVM and CNN in detection landslide areas in with optical images and DTM derivatives. After all this many challenges still not address very well concerning the application of machine learning and deep learning towards landslide identification and classification. This study is step forward in this direction and in this study, we adopted machine learning and deep learning approaches for land sliding classification.

II. STUDY AREA

The Gilgit District is located in Pakistan between 660 45 E to 660 12 E and 380 45 N to 380 60 N. I was planning to carried out this research for the KKH (Karakorum Highway) now CPEC (China Pakistan Economic Corridor) district with an area of around 60 km^2 in the Nager division of the Gilgit province. It consists of large alpine glaciers and fresh water, fed by high mountains glaciers. Landslides, earthquakes and floods are not uncommon in the region, frequently damaging parts of the highway.

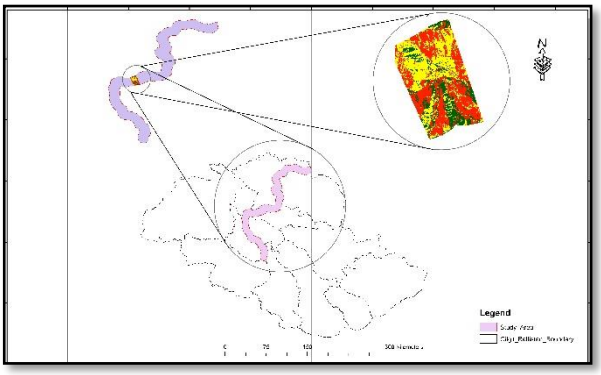


Fig. 1 Location map of the study area

III. METHODOLOGY

Google Earth Engine, ArcGIS 10.3.1, FME Workbench 2020.1, Anaconda Navigator, Jupiter Notebook, Spider, Google Co laboratories, Microsoft Excel 2016 and Mendeley 1.19.4 are the applications used for this analysis. GEE is used for data collection for a desired clip. In ArcGIS 10.3.1, to identify the study area, the downloaded DEM was processed.

Slop, Aspect, Geology, Road, Streams, Land Cover, Faults and Precipitation are the meanings of the study area. This instrument is also used to construct maps of the research area and CSV archives. In addition, Python is used for inserting, combining and tagging. Finally, in the whole work, referencing was provided using Mendeley. Figure 1 shows the overall Methodology flowchart.

A. Clipping Data

The primary move in the analysis was to clip the desired data that lies geographically between 35o76'977"N latitude and 74o57'402" E longitudes and runs through Jutal, Rahimbad, finishing at the top of Khunjarab. In Google Earth Engine (GEE), this move was easily carried out by writing simple python script and vector data. E.g., Var clip = function (img_1) {return img. Clip (export_geometry);};

In analysis work, it is important to clip the field of concern without missing the key data because it needs less data processing time. In other words, the retrieval of Unclipped information requires a great deal of time and is done at an impossible Pace.

B. Topographic factor

Topological factors include slope and aspect. The slope angle is considered the main parameter of the slope stability. Aspect related parameters such as exposure to sunlight, winds (dry or wet), rainfall (degree of saturation), soil moisture and discontinuities may control the occurrence of landslides.

C. Anthropological factor

Anthropological factors considered in this study area are land use and distance to roads. A land cover map was developed from the Sentinel 2 image through supervised classification.

D. Geological factor

Considered for this study include geology and proximity to faults. Geology plays an important role in landslide susceptibility studies because different geological units have different susceptibilities to activate geomorphologic processes e.g., Alluvium, Paleozoic Rock, Early Mesozoic & Late Paleozoic rocks, Permian Rocks.

E. Machine learning classifiers

The next move after data preprocessing was to implement classifiers for machine learning on processed data. Python was used for this point, which can be easily trained and produced using processed data to create the ML model. The algorithms tested were Random Forest (RF), Decision Tree, Convolutional Neural Network, and Algorithm SVM. Along with their experimental findings, Section IV has explained certain algorithms.

F. Designing graph

Chart architecture has been carried out in Python, ArcGIS and Excel which is one of the best for graph analysis. In a more effective, readable and engaging manner, graphs display the effect of land sliding and non-land sliding behavior.

III. EXPERIMENTS AND ANALYSIS

We have evaluated the machine learning and deep learning algorithms performance on dataset. First, we will discuss the experimental setup and describe the datasets used in our experiments. Then we report the findings of our study.

A. Experimental Setup

We have conducted extensive experiments using Keras, TensorFlow backend in python with necessary modifications. We have conducted all our experiments on CPU platform. The algorithms were developed on Python 3, OpenCV to implement our solution. First, we imported all necessary Keras libraries and open sources such as broken.

B. Datasets

The inventory of landslides is created by visual analysis of Sentinel 2 images, which have been counter-verified in Google earth maps and calibrated accordingly for field data and boundaries.

TABLE.1 DATA SOURCE AND DESCRIPTION

Spatial Data		Scale	Resolution	Source
Factor		Point Data		Satellite Image, ArcGIS 10.2
Topographic Factor	Aspect Slope Streams		30 m	SRTM Shuttle Radar Topography Mission (USGS) United States Geological Survey
Anthropological	Road land Cover		30 m	Google Image
Geological Factor	Geology Precipitation Faults		30 m	Survey of Pakistan

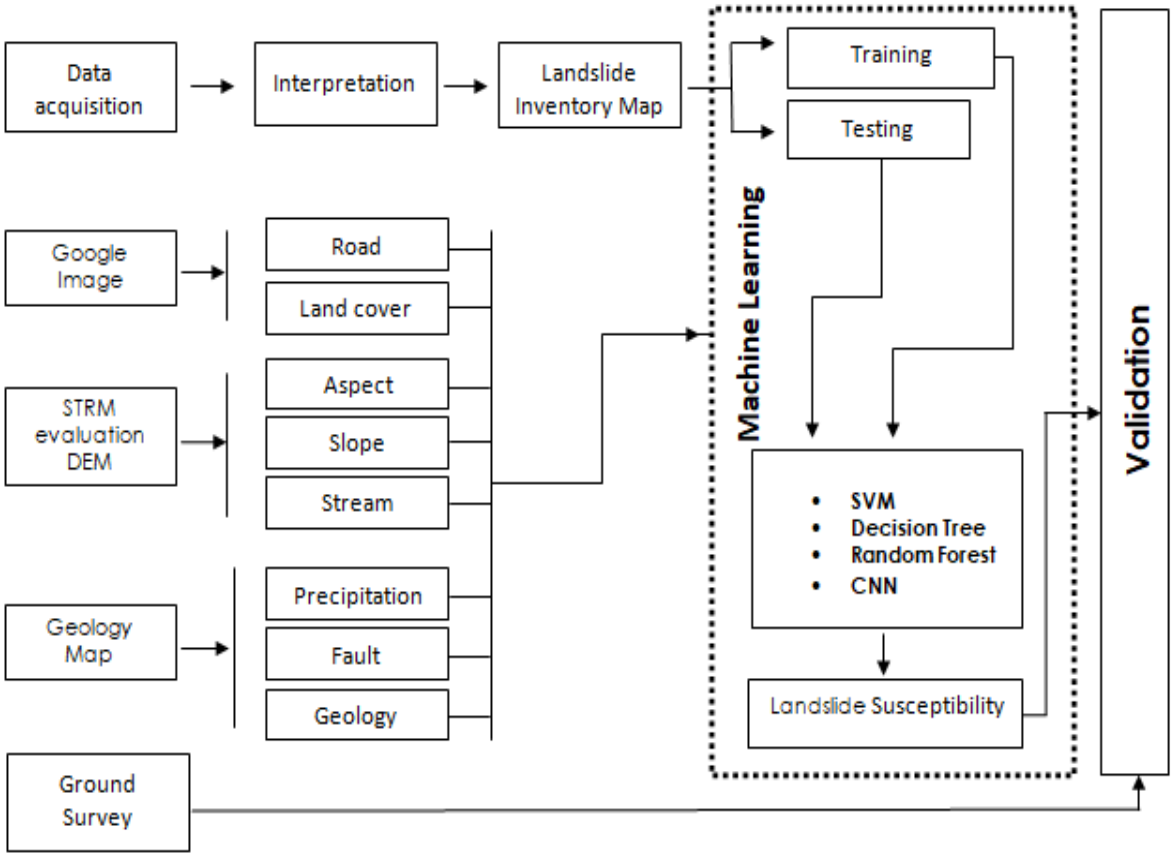


Fig.2 Flowchart of the methodology adopted for the study

A. Random Forest

It is a supervised algorithm for machine learning that can be used for regression and classification tasks. In this analysis, 80% of training samples and 30 percent for testing were randomly chosen by the algorithm to train the RF model. As preparation, these two classes are considered. The system used the number of decision trees for this form of classification with regard to random bootstrap datasets to construct the RF.

Each tree creates predications, but with the aid of majority voting, the classifier. extracts the best predictions from them and generates final strong results. According to this RF classifier operating mechanism, our device results in a processing time of 2200 seconds. Finally, for two separate classes of the region chosen, 96 percent classification accuracy was achieved

B.SVM (Support Vector Machine)

The machine learning algorithm used for the problem of classification or regression is supervised. The optimum boundary between potential outputs is found. It is possible to extend them to linear and non-linear problems. They were the best General algorithm for machine learning until 2006.

C.Decision Tree Classifier

One of the simplest and most common classification algorithms to recognize and analyze is the Decision Tree. The Decision Tree belongs to the group of algorithms for supervised learning. The decision tree algorithm can also be used for solving regression and classification problems, as opposed to other supervised learning algorithms. The purpose of using a decision tree is to build a training model that can be used by studying basic decision rules obtained from previous data to predict the class or meaning of the goal variable. Decision Trees, beginning from the root of the tree to predict a class mark for a record. We compare the values of a root attribute with the attributes of the record. We obey the branch corresponding to that value on the basis of similarity and leap to the next node.

D. CNN (Convolutional Neural Network)

One of the most convenient layers is specifically for image recognition, labeling, and segmentation. For either 2D or 3D data collection, CNN may be added. CNN has three types of layers. Pooling Layer, Convolutional Layer and totally attached layer.

TABLE 2: CLASSIFICATION RESULTS OF CLASSIFIERS

S.NO	Classifier	Accuracy (%)	Completion time (seconds)
1.	Random Forest	96%	2200 sec
2.	SVM	95%	2000 sec
3.	Decision Tree	68%	2100 sec
4.	CNN	77%	1900 sec

a. Simulation Results and analysis

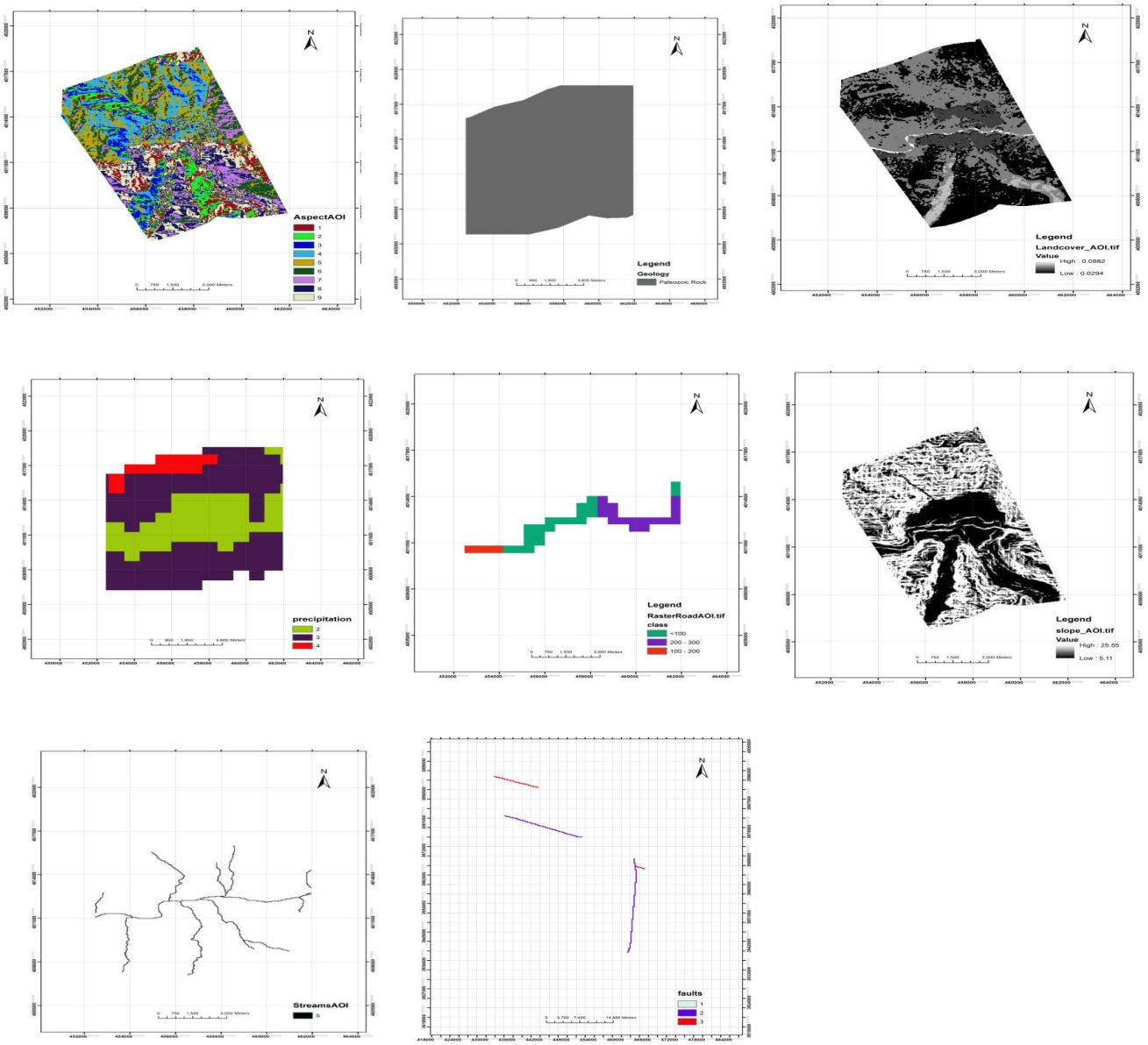


TABLE 3: RESULTS OF CLASSFIERS

S.NO	Classifier	Accuracy (%)
1.	Random Forest	96%
2.	SVM	95%
3.	Decision Tree	68%
4.	CNN	77%

IV. CONCLUSION

In this research landslide susceptibility mapping was regarded as a classification task in the input space of DEM, geological, anthological and topological and hydrological derived attributes. Pixels (30 m resolution) were classified into two categories: landsliding area and non landsliding area. Four different Machine learning techniques which learn from expertly labeled data were compared: support vector machine, Random Forest, Decision Tree, Convolutional neural Network. For training (70%) and testing (30%). The result of given classifier in 95%,96%,68% and 77%. Random forest classifier performs well as compare to all other classifier in this dataset.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGMENT

This work was supported by Karakorum University. The authors are expressing their sincere thanks to the Karakorum University and Computer Science department for their constant encouragement.

REFERENCES

[1] Owen, L.A. (1996) Quaternary Lacustrine Deposits in a High Energy Semi-Arid Mountain Environment, Karakoram Mountains, Northern Pakistan. Journal of Quaternary Science, 11, 461-483.

[2] Karim, E. (2006) Hazard and Vulnerability Assessment of Sherqila Village District Ghizer NAs Pakistan. Dissertation, University of Geneva, Geneva.

[3] Kanwal, S., Atif, S. and Shafiq, M. (2016) GIS Based Landslide Susceptibility Mapping of Northern Areas of Pakistan, a Case Study of Shigar and Shyok Basins. Geomatics, Natural Hazards and Risk, 8, 348-366.

[4] Rao, A.L. (2014) History Profiles of Major Natural Disaster Events in Gilgit Baltistan. Pakistan GLOF Project Climate Change Division.

[5] Ahmed, M.F. and Rogers, J.D. (2014) Creating Reliable, First-Approximation Landslide Inventory Maps Using ASTER DEM Data and Geomorphic Indicators, an Example from the Upper Indus River in Northern Pakistan. Journal of Environmental & Engineering Geoscience, 20, 67-83..

[6] Moosavi, V., Talebi, A. and Shirmohammadi, B., 2014. Producing a landslide inventory map using pixel-based and object-oriented approaches optimized by Taguchi method. *Geomorphology*, *204*, pp.646-656.

[7] Ding, Anzi, Qingyong Zhang, Xinmin Zhou, and Bicheng Dai. "Automatic recognition of landslide based on CNN and texture change detection." In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 444-448. IEEE, 2016.

[8] Li, X., Cheng, X., Chen, W., Chen, G. and Liu, S., 2015. Identification of forested landslides using LiDar data, object-based image analysis, and machine learning algorithms. Remote sensing, 7(8), pp.9705-9726.

[9] Ghorbanzadeh, Omid, Thomas Blaschke, Khalil Gholamnia, Sansar Raj Meena, Dirk Tiede, and Jagannath Aryal. "Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection." Remote Sensing 11, no. 2 (2019): 196.

[10] A. A. Khan, A. Jamil, D. Hussain, M. Taj, G. Jabeen, and M. K. Malik, "Machine-Learning Algorithms for Mapping Debris-Covered Glaciers: The Hunza Basin Case Study," *IEEE Access*, vol. 8, pp. 12725–12734, 2020.

[11] D. Hussain and A. A. Khan, "Machine learning techniques for monthly river flow forecasting of Hunza River, Pakistan," *Earth Sci. Informatics*, 2020.

Music Makam Recognition by Using Convolutional Neural Network

Yucel Cimtay
Image and Signal Processing Group
HAVELSAN A.S.
Ankara, Turkey
ycimtay@havelsan.com.tr

***Abstract*—Turkish music pieces are represented by different kinds of makams. These makams have a strong history since they exist for millenniums. Music pieces are mostly used for the education of musicians. There are more than 500 makams in Turkish music. Although there are several studies which implement music makam recognition, they use the traditional machine learning classification methods and mostly the note decomposition prior to classification. In this study, we propose the first work which uses the music sound directly with a CNN network. Therefore, this study can be implemented in real-time and can recognize the makam only by listening to the raw music. In this work, 89.7% overall classification accuracy is achieved for 8 different music makams.**

Keywords—Music note, artificial intelligence, music makam, musicians

I. INTRODUCTION

The developments in communication technologies has enabled people to record and share musical data in digital environments. In Western music, there are many tools for converting musical data to digital forms [1, 2]. However, the digitization studies for Turkish music is mostly limited to scanning the note papers and saving them as .pdf or image formats. Today, for representation of musical melodies, the mostly used file format is MusicXML. This format is used for transferring musical data into digital form and transmitting it over the world [3, 4]. The study in [5] introduces a system which queries information on a MusicXML database. The study in [6] introduces a corpus for Turkish music as a part of CompMusic project. This study is very important in terms of creating a reachable digital platform which includes 6000 audio recordings and important information about them such as information about recordings, makams, scores, artists, compositions and the interrelations between them.

II. RELATED WORK

Each makam has its own characteristics like the tonics, sequence of notes, and decision tone. Therefore, classification of melodies with respect to their makams is a very challenging issue for machine learning studies. [7] is the first study which classifies Turkish music melodies according to their makams. The study in [8] uses frequency histograms of notes in order to detect the music tone automatically. The study in [9] estimates the makams by detecting the tonics. It uses about 260 recordings and reports 94.9% accuracy for tonic classification. The study in [10] creates an open-source toolbox called Morty for mode recognition and tonic identification. The scientists report 95.8%, 71.8% and 63.6% accuracy in tonic identification, mode recognition and joint mode and tonic estimation tasks, respectively.

The study in [11] is a base study which recognize the makam by applying N-gram based statistical analysis to 847 melodies composed of 13 different makams. The classification accuracy on makam recognition is reported as 86% - 88%. Similar to Turkish makam recognition, [12] employs Dastgah recognition and [13] implements Dastgah classification. The study in [14] reaches 89.7% classification accuracy for 10 different Turkish makams: Hicaz, Nihavend, Hüzzam, Rast, Kürdilihicazkar, Uşşak, Hüseyni, Mahur, Muhayyerkürdi and Hicazkar by using MusicXML format and Random Forest classification algorithm.

The study in [15] increases the number of makams to 12 and applies 10 different machine learning algorithms in order to classify the makams. It achieves different accuracies with respect to the used machine learning algorithm. In average, the success rate changes from 82.18 to 88.12. In terms of makams, according to F-measure data, Hicaz is the one with the maximum success with 99.35%. The other most successful makams are: Rast with 97.37%, Nihavend with 98.40%, Kürdilihicazkar with 98.60% and Saba with 97.89%. The authors also report that they have problem with classification the makams: Beyati, Hüseyni, Muhayyer and Uşşak makams since they have the same emphasized notes (A) and very similar makam scales.

The study in [16] combines neural networks and achieves 95.83% classification accuracy on 9 makams. In [17], authors use probabilistic neural network and handles 89.40 accuracy. The study in [18] performs a learning approach by using DBN (Deep Belief Networks) and achieves 93.1% accuracy for 7 different makams. They use the first 20 seconds of each melody and divide each sample into 20ms pieces and extracts MFCC (Mel Frequency Cepstral Coefficients) to use in classification. Different from the general approach of using the symbolic music data, the authors use the commercial music sounds directly.

Using the symbolic data requires a pre-estimation of symbolic sequence of music melodies prior to the classification. Similarly, detecting the makams by using the tonics or the decision note is not enough for classification of music pieces, since different makams can have the same tonics and/or the decision note. By considering these points and getting use of recommendations of professional musicians, in this work, like [18] we use the music data directly which is downloaded from the website [19] and we use the state-of-the-art classification method: CNN (Convolutional Neural Networks) in order to recognize the makams of musical pieces.

III. DATASET

A. Creation of the Dataset

There is much music source today on internet. Thus, one of the most open data sources could be music data. In this study, we have downloaded music data from the website [19]. Since it groups the music pieces by their makams, it is very useful for these kinds of studies. This study focuses on 10

different makams: ussak, hicaz, rast, nihavent, segah, saba, huzzam, kurdi, mahur and hicazkar. In order to create a dataset, 100 music pieces from each makam is downloaded.

B. *Preprocessing of Data*

Each musical piece was split into 15 sec. portions. The reason of choosing the segmentation window length as 15s is that with the recommendation of professional musicians, we consider that the makam characteristic and note sequence can be understood in 15s. Normally, sometimes, for the exact understanding of the makam of any music piece, 15s may not be enough also. Because any piece may start on some makams but at the last part it may evolve to any other makam. Therefore, in these kinds of cases, musicians decide that this piece belongs to the last makam instead of its starting makam. Of course, this is a very detail information however what should be known is that music makam detection is a very complex issue. The number of segments belong to each makam is given in Table 1.

TABLE I. NUMBER OF 15S SEGMENTS WITH RESPECT TO MAKAMS

Makam	Number of Segments
Hicaz	2195
Nihavent	1575
Ussak	1838
Saba	765
Rast	1934
Segah	1610
Huzzam	2128
Kurdilihicazkar	1950
Mahur	2900
Hicazkar	2756

IV. PROPOSED METHOD

In order to remove the effect of different levels of sounds, each 15s data is first normalized. Each normalized 15s piece are windowed by a window size 25ms with 10ms overlapping. For each window, MFCC (Mel-Frequency Cepstral Coefficients) [20] are extracted. As a result, the dimension of the data, also the input data to the network, for each makam is prepared as $(n, 1499,13)$ where 13 shows the number of MFCC coefficients, 1499 is the number of windows and n is the number of 15s. samples which can be different for different makams.

In this study, we propose a 1D CNN structure. Since we extract the MFCC coefficients of each music segment, we prefer not to use very deep network or adopt any pretrained models to this problem. We use serial VGG-blocks [21], batch normalization, dropout and dense layers in our network. The structure of a VGG-block is given in Fig. 1.

In this study 3 convolutional layers with (64, 128 and 32) number of filters and max pooling layer with pool size 2 are used to form one VGG block. Following each VGG block a batch normalization layer is used. Totally 7 VGG blocks are used with batch normalization layers. Following 7 VGG blocks, dropout layer is used to sort the overfitting problem and Dense layers are used to increase the adoptability. The big picture of network structure is given in Table 2.

Number of convolutional layers is very important in terms of increasing the accuracy, in addition we discovered that the Dropout Layer is very effective on producing high classification accuracy.

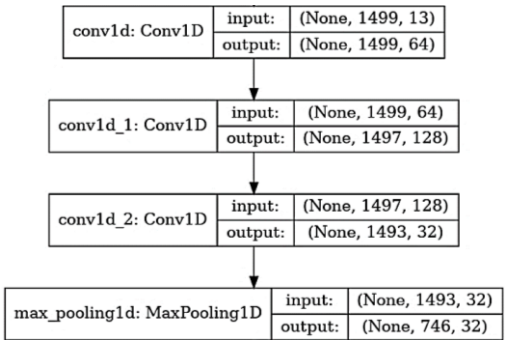


Fig. 1. VGG block structure

The CNN network is designed by using Python-TensorFlow framework with GPU. To train the network, 70% of music segments are used following a shuffling operation. To test the trained network, we used the left 30% of music segments. Training process is shown in Fig. 2. Since the accuracy improvement stops at epoch 246, we show the training graph up to 246.epoch.

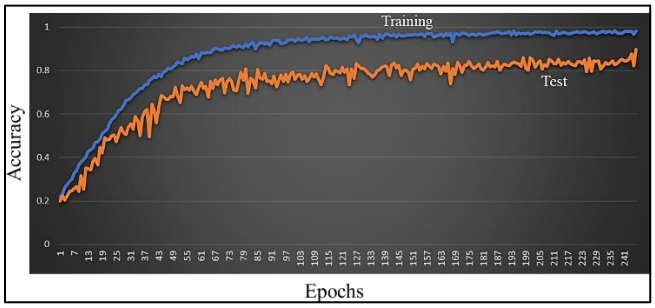


Fig. 2. Training Process of CNN

TABLE II. LAYER DESCRIPTION OF PROPOSED NETWORK

Layer (type)	Output Shape
VGG-block-1	(None, 746, 32)
batch_normalization	(None, 746, 32)
VGG-block-2	(None, 370, 32)
batch_normalization	(None, 370, 32)
VGG-block-3	(None, 182, 32)
batch_normalization	(None, 182, 32)
VGG-block-4	(None, 88, 32)
batch_normalization	(None, 88, 32)
VGG-block-5	(None, 41, 32)
batch_normalization	(None, 41, 32)
VGG-block-6	(None, 17, 32)
batch_normalization	(None, 17, 32)
VGG-block-7	(None, 5, 32)
batch_normalization	(None, 5, 32)
Dropout	(None, 5, 32)
Flatten	(None, 160)
Dense	(None, 128)
Dense	(None, 128)
Dense	(None, 128)
Dense	(None, 8)

The training parameters of proposed CNN is given in Table 3.

TABLE III. TRAINING PARAMETERS OF CNN

Parameter	Value
Optimizer	adam
Loss	categorical_crossentropy
Shuffle	True
Number of epochs	300
batch_size	64

V. RESULTS

The classification results are given as confusion matrix in Table 4. Overall accuracy is calculated as 89.7% for 8 makams. The grey column on the most right-hand side and the one on the bottom are recall and precision scores for each makam. The highest recall score 97% is achieved for Ussak, which is also the worst one with 77% of precision score. One reason for the low score of precision for Ussak is that other makams generally include Ussak segments compare to other makams. Therefore, the network may classify some segments as Ussak and since they are labelled as the other makams when preparing the data, the precision decreases. However, these kinds of details should be considered in future studies and the segments should be investigated separately.

Both recall and precision are successful for Rast makam and Saba is also classified successfully.

TABLE IV. CONFUSION MATRIX OF MAKAM CLASSIFICATION 8 MAKAMS

Target Makam										
	Hicaz	823	40	27	7	16	6	7	15	0.87
	Nihavent	0	625	31	5	5	4	2	3	0.93
	Ussak	3	10	767	1	3	2	1	2	0.97
	Saba	0	5	5	301	7	3	4	3	0.92
	Rast	2	28	35	6	735	2	6	15	0.89
	Segah	7	34	22	7	20	580	14	6	0.84
	Huzzam	12	16	60	23	21	6	752	22	0.82
	Kurdili hicazkar	18	18	46	3	6	2	6	737	0.88
		Hicaz	Nihavent	Ussak	Saba	Rast	Segah	Huzzam	Kurdili Hicazkar	0.89
		0.95	0.81	0.77	0.85	0.90	0.96	0.95	0.92	

Predicted Makam

In this study, we also trained the network with 7 different makams which are handled in Kizrak et al. [18] in order to conduct a benchmark. These makams are: Hicaz, hicazkar, kurdilihicazkar, mahur, nihavent, rast and ussak. We have achieved an average accuracy of 86.1%. On the other hand, we apply the DBN (Deep Belief Network) which was found as the most successful method in [18] on our dataset. Results are shown in Table 5.

TABLE V. BENCHMARK RESULT

Study	Accuracy (%)
Kizrak et al. [18]	21.7
Proposed	86.1

The accuracy for proposed study is lower for 7 makams compared to 8 makams. The reason is that the selected makams for 7 makams case are similar to each other in terms of musical characteristics. Therefore, classification is harder, and accuracy is a bit lower.

VI. CONCLUSION

In this study, we have implemented a deep learning CNN model for makam estimation of music pieces. Eight makams have been chosen for training of CNN model. The findings handled in this work is very valuable in terms of automatic makam detection. The importance of this study is that it shows that machines also can detect the makams like human. This is a success of artificial intelligence.

As it is indicated in the text, music makam detection is always a complex issue. Because, for a music piece which belongs to one specific makam, it is very possible to find different makam segments which is embedded in this music piece. Therefore, instead of detecting the makam for whole music piece, we have estimated the makam for each 15s music data.

Results shows that artificial intelligence is very successful in makam detection. The accuracy can be increased by examining each specific duration music data according to the makams, increasing the number of samples and using deeper network models. In the future these points will be covered by the author and results will be presented to the literature.

ACKNOWLEDGMENT

The author would like to thank the Conductor of Elazig Classical Turkish Music Choir “Kenan CIMTAY” for his valuable contributions in studying the theory of music makam detection.

REFERENCES

[1] O'Brien C. ve Plumbley M., Automatic Music Transcription Using Low Rank Non-Negative Matrix Decomposition, 25th European Signal Processing Conference (EUSIPCO2017), Kos Adası-Yunanistan, 28 Ağustos-2 Eylül, 2017.

[2] Sigtia S., Benetos E., Dixon S., An end-to-end neural network for polyphonic piano music transcription, IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 24 (5), 927-939, 2016.

[3] Good M., MusicXML for Notation and Anaylsis, Computing in Musicology, 12, 113-124, 2001.

[4] MusicXML 3.0 Tutorial, Avaiable on: https://wpmedia.musicxml.com/wp-content/uploads/2012/12/musicxml-tutorial.pdf. Accessed on: 23/12/2020.

[5] Ganseman J., Scheunders P., D'haes W., Using XQuery on MusicXML Databases for Musicological Analysis, 9. International Society for Music Information Retrieval Conference (ISMIR 2008), Philadelphia-USA, 433-438, 14-18 Eylül, 2008.

[6] Uyar B., Atlı H.S., Şentürk S., Bozkurt B., Serra X., A Corpus for Computational Research of Turkish Makam Music, 1. International Workshop on Digital Libraries for Musicology (DLfM), London, England, 1-7, 12 Eylül 2014.

[7] Gedik, A.C., Bozkurt, B., 2008, "Automatic Classification of Turkish traditional Art Music Recordings by Arel Theory", Conference on Interdisciplinary Musicology, Thessaloniki, Greece.

[8] Gedik, A.C., Bozkurt, B., 2010, "Pitch-Frequency Histogram-Based Music Information Retrieval for Turkish Music", Signal Processing, Vol. 90, pp. 1049-1063, Elsevier.

[9] Şentürk, S., Gulati, S., Serra, X., 2013, "Score Informed Tonic Identification for Makam Music of Turkey", 14. International Society for Music Information Retrieval Conference (ISMIR 2013), Curitiba, Brasil.

[10] Karakurt, Altuğ & Şentürk, Sertan & Serra, Xavier. (2016). MORTY: A Toolbox for Mode Recognition and Tonic Identification. 9-16. 10.1145/2970044.2970054.

[11] Ünal, E., Bozkurt, B., Karaosmanoğlu, M.K., 2012, "N-Gram Based Statistical Makam Detection on Makam Music in Turkey Using Symbolic Data", 13. International Society for Music Information Retrieval Conference (ISMIR 2012), Porto, Portugal.

[12] Darabi, N., Azimi, N., Nojumi, H., 2006, "Recognition of Dastgah and Makam for Persian Music with Detecting Skeletal Melodic Models", 2. IEEE BENELUX/DSP Valley Signal Processing Symposium,

[13] Abdoli, S., 2011, "Iranian Traditional Music Dastgah Classification", 12. International Society for Music Information Retrieval Conference (ISMIR 2011), Florida, USA, 275-280, 2011.

[14] Abidin, D., Öztürk, Ö., Özacar Öztürk, T., 2017, "Klasik Türk Müziğinde Makam Tanıma Için Veri Madenciliği Kullanımı", Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, Vol. 32(4) , pp. 1221-1232.

[15] Abidin, Didem & Özacar, Tuğba & Ozturk, Ovunc. (2018). USING CLASSIFICATION ALGORITHMS FOR TURKISH MUSIC MAKAM RECOGNITION. Selcuk University Journal of Engineering ,Science and Technology. 6. 377-393. 10.15317/Scitech.2018.139.

[16] Kalender, N., Ceylan, M., Karakaya, O., 2012, "Türk Müziği Makamlarının Sınıflandırılması için Yeni Bir Yaklaşım: Kombine YSA", ASYU 2012 Akıllı Sistemler Yenilikler ve Uygulamaları Symposium, Trabzon, Türkiye.

[17] Kizrak, M.A., Bayram, K.S., Bolat, B., 2014, "Classification of Classic Turkish Music Makams", Innovations in Intelligent Systems and Applications (INISTA 2014), Alberobello, Italy, 394-397, 23-25 June 2014.

[18] Kizrak, M.A., Bolat, B., 2015,"Classification of Turkish Music Makams bu Using Deep Belief Networks", IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Malatya, Türkiye, 2015.

[19] Available on: https://www.notaarsivleri.com/. (Accessed date:27.12.2020)

[20] Available on: http://www.practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs (Accessed date:12.01.2020)

[21] Available on: https://machinelearningmastery.com/how-to-implement-major-architecture-innovations-for-convolutional-neural-networks/ (Accessed date:12.01.2020)

A Comparison of Obstacle Dependant Gaussian and Hybrid Potential Field Methods for Collision Avoidance in Multi-Agent Systems

Fethi Candan, Yusheng Peng, Lyudmila Mihaylova
Department of Automatic Control and Systems Engineering
University of Sheffield, Sheffield, S1 3JD, United Kingdom
{fcandan1, ypeng39 and l.s.mihaylova}@sheffield.ac.uk

***Abstract*—In this paper collision avoidance methods - the velocity obstacle, obstacle dependant Gaussian potential field and hybrid potential field methods, are compared and tested on multi-agent systems. Extensive evaluation is presented with a number of case studies with a different number of agents, with static and dynamic and obstacles. The advantages and disadvantages of each method are discussed. The optimisation of the static and dynamic coefficients of the hybrid potential field method is performed via a genetic algorithm. The results from the tests are from 1000 independent runs and show that the hybrid potential field method can avoid reliably collisions in multi-agent systems.**

***Index Terms*—multi-agent systems, unmanned air vehicles, collision avoidance, potential field**

I. INTRODUCTION

In recent years, unmanned systems such as unmanned aerial or ground vehicles (UAV or UGV), have been subject to intensive research. Especially, UAVs have gained popularity and have been applied to civilian and military areas: aerial photography, surveillance, agriculture, search and rescue operations. For these reasons, some problems such as UAV path planning and control are of significant importance. Numerous linear and nonlinear control methods have been designed and deployed into UAVs and collision avoidance is an inherent component of them.

Both linear and nonlinear models of UAV dynamics have been used and embedded in collision avoidance and control methods. For example, such linear and nonlinear proportional integral derivative (PID), fuzzy PID controllers are developed in [8], an H_∞ control approach is presented in [1], a linear quadratic regulator (LQR) in [5] and a Model Predictive Control (MPC) approach in [14].

Modelling swarms of agents is another important task, especially linked with collision avoidance. In swarms the components can be seen at an aggregated level rather than at individual level of each agent/UAV [17]. There are also challenges related with the coordination, communication in real time and achieving these task for each single agent [4]. Multi-agent systems (MAS) have gained significant interest and recent results are described in the survey paper [10].

This paper focuses on methods for collision avoidance based on efficient path planning. Each agent/ UAV is considered as an agent. Path planning has been one of the important UAV and tasks aiming to general a trajectory which is optimal according to a certain criterion. In order to accomplish the given task, the robot/UAV needs to reach a desired target point, after starting with an initial point. The agent trajectory generation needs to be performed to avoid both static and dynamic obstacles [4].

There are different methods able to solve efficiently path planning problems [2], [3], [9], [12], [20]–[22], [26]. The potential field (PF) [22] and velocity obstacle (VO) methods [16] are two of the most efficient solutions

The PF methods have been widely used to solve path planning problems since they have several advantages: they are easy to implement and provide smooth trajectories [9], [22], [26].

In [22], PF methods have been studied in depth for path planning. Then, artificial potential field, based on traditional PF method, has been validated with performance criteria in terms of a solving local minima problem and obstacle avoidance. Collision avoidance can also be solved with geometric types of methods [20] and have been compared with an A* graph search method [13] especially for UAVs [3]. The A* method has been used to find the shortest path in real-time applications to find a new path when facing with dynamic obstacles.

From the VO methods, the reciprocal velocity obstacle method [24], outperforms the generic VO method and has been tested in real-time MAS. In [11] and [12], an interval VO approach, with collision avoidance cone has been proposed and compared with traditional VO, reciprocal, hybrid reciprocal VO and optimal reciprocal collision avoidance methods. Also, each algorithm has been assessed in terms of their advantages and disadvantages.

Traditional PF, VO, obstacle dependant Gaussian Potential field (ODG-PF) and hybrid potential field methods (HPFM) have been applied in several fields [12], [17]. As the PF and VO were designed to deal separately with static and dynamic obstacles, they require improvements in complex situations where both static and dynamic obstacles exist.

The HPFM combines the strategy of PF and VO by using the form of Vector Field Histogram (VFH) [6], [7]. In the HPFM, features of two types of obstacles are considered and the weights of each type of obstacles could be easily adjusted regarding different environments.

This work compares VO, ODG-PF and HPF methods in terms of solving collision avoidance problems, with respect to the length of the generated trajectory and task time costs, respectively. Moreover, to find dynamic and static coefficients of HPFM, a genetic algorithm (GA) [25] has been used. Then, an optimal trajectory providing collision avoidance has been determined with these coefficients.

This paper is organized as follows. Section II describes the methods of the potential field, velocity obstacle and hybrid potential field. Simulation results are given in Section III. A thorough evaluation and validation of these approaches is given. Section IV presents the conclusions and future work.

II. ADVANCED OBSTACLE AVOIDANCE METHODS

This section presents the velocity obstacle (VO) [16] and Potential Field (PF) [22] methods for obstacle avoidance in multi-agent systems (MAS). These methods are briefly described in the next section II and their performance is demonstrated over an example [19].

A. Potential Field Method

The potential field method is a well-known global path planning method able to determine easily distances to multiple agents. Assuming that the agent has a point mass and moves in a two dimensional (2-D) work space, the following notation is adopted: $\mathbf{q} = [x\ , y]^T$, where x and y are the agent coordinates and ^{T} is the transpose operation.

Different cost functions have been suggested in the literature [18], [23]. Considering a 2-D Cartesian system, the agent and obstacle can be visualised positioned at a certain distance from each other, as shown in Fig. 1. The attractive potential function $\mathbf{U}_{att}(\mathbf{q})$ that is most widely used takes the form of:

$$\mathbf{U}_{att}(\mathbf{q}) = \frac{1}{2}\xi\rho^m(\mathbf{q}, \mathbf{q}_{goal}), \tag{1}$$

where a positive scaling factor ξ is defined and the distance $\rho(\mathbf{q}, \mathbf{q}_{goal})$ between the current agent's location $\rho(\mathbf{q}$ and its goal location $\mathbf{q}_{goal})$ is shown. Here the m coefficient can have values 1 or 2. When $m = 1$, the attractive potential function is conic in shape, otherwise it is parabolic.

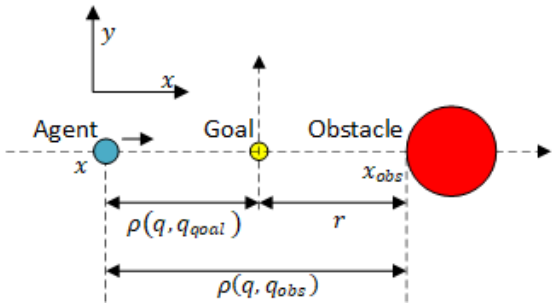


Fig. 1. Positions of the agent, goal and obstacle in a 2-D space

The associated attractive force $\mathbf{F}_{att}(\mathbf{q})$ is then given by the attractive potential's negative gradient $-\nabla U_{att}(\mathbf{q})$ as follows

$$\mathbf{F}_{att}(\mathbf{q}) = -\nabla U_{att}(\mathbf{q}) = \xi(\mathbf{q}_{goal} - \mathbf{q}). \tag{2}$$

One of the widely used potential driving functions takes the form of:

$$U_{rep}(\mathbf{q}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{q},\mathbf{q}_{obs})} - \frac{1}{\rho_0}\right)^2 & , \text{if } \rho(\mathbf{q}, \mathbf{q}_{obs}) \le \rho_0 \\ 0, & , \text{if } \rho(\mathbf{q}, \mathbf{q}_{obs}) > \rho_0 \end{cases} \tag{3}$$

where ρ_0 is the threshold distance.

When η is a positive scaling factor, $\rho(\mathbf{q}, \mathbf{q}_{obs})$ represents the minimum distance from agent $\mathbf{q}$ to the obstacle and $\mathbf{q}_{obs}$ represents the point on it.

$$\mathbf{F}_{rep}(\mathbf{q}) = -\nabla U_{rep}(\mathbf{q}). \tag{4}$$

The sum of the attractive and repulsive forces forms the combined force which is applied to the agent.

$$\mathbf{F}_{total} = \mathbf{F}_{att} + \mathbf{F}_{rep}. \tag{5}$$

B. Obstacle Dependant Gaussian Potential Field Method

This subsection presents the Obstacle Dependant Gaussian method (ODG-PF), based on Vector Field Histogram. In real-time path planning, this method is relying on processing the sensor data, because generally other methods uses global environment data. Considered with respect to the Potential Field method, it is a notable global path planning algorithm in terms of easy calculation and accurate results.

For generating the potential field, the width ω_k, the distance d_k to the obstacle and the calculated angle ω_k are given by equation (6) for this method.

$$\Phi_k = 2arctan\left(\frac{\omega_k}{2d_k}\right) \tag{6}$$

The distance d_k between obstacles and agents is linked with the function A_k as follows:

$$A_k = \frac{1}{2}e^{d_k} \tag{7}$$

Equation (8) shows the repulsive function f_k and direction ϑ to avoid each obstacle. Generally, the variable ϑ is between 0 and 359 degrees and it changes one degree for each step:

$$f_k(\vartheta) = A_k e^{-\frac{2(\theta_k - \vartheta)^2}{\Phi_k^2}} \tag{8}$$

The repulsive function $f_{rep}(\vartheta)$ is determined when several obstacles are identified in a scan by basically summarising the repulsion region of each obstacle

$$f_{rep}(\vartheta) = \sum_{k-1}^{n} f_k(\vartheta) = \sum_{k-1}^{n} A_k e^{-\frac{2(\theta_k - \vartheta)^2}{\Phi_k^2}}. \tag{9}$$

The attractive field $f_{att}(\vartheta)$ is based on the θ_{target} angle, referring to the orientation of the target and γ is a coefficient of the attractive function:

$$f_{att}(\vartheta) = \gamma\left|\theta_{target} - \vartheta\right|. \tag{10}$$

Lastly, the total potential field function is summed with the repulsive and attractive functions.

$$f_{total}(\vartheta) = f_{rep}(\vartheta) + f_{att}(\vartheta). \tag{11}$$

The ODG-PF in [9] shows an insensitive feature for the γ coefficient in an attractive field compared to the k_{att} coefficient in general PF, which provides a greater flexibility of the method to the environment. Other benefit of ODG-PF is that it requires a reasonable amount of data, which makes it easy to deploy in real time.

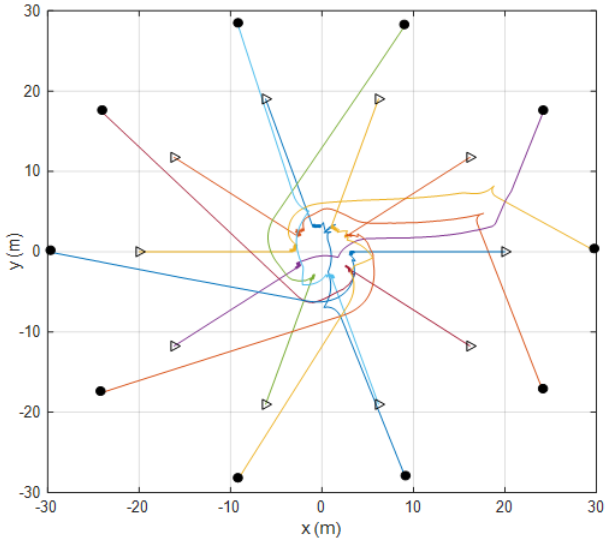


Fig. 2. ODG-PF trajectory training result

Fig. 2 shows agents' trajectories with the optimal coefficient value of ODG-PF (γ). This coefficient value is determined by using empirical testing.

C. Velocity Obstacle Method

The velocity obstacle (VO) method [16] considers a set of all velocities of an agent which at some point in time could result in a collision with another agent, if the other agent retains its current velocity. If the agent chooses a velocity inside the velocity obstacle region, then the two agents will inevitably collide. If it chooses a velocity outside the velocity obstacle region, it is assured that such a collision will not happen. Compared to other local collision-free trajectory planning methods, the VO method could help avoid moving obstacles, since the possible future obstacle trajectory has been considered during the collision avoidance process.

The VO method [16] has a critical concept which is the Collision Cone (CC), conical area in the velocity space. In Fig. 3, it has been agents A, obstacles B and centered position of objects are p_A and p_B, respectively.

The variables r_A and r_B represent the radii of the two objects A and B, respectively. The cut line from the central point p_A to the circle based on p_B and r_c = r_A + r_B radius can be described as two boundaries λ_{AB}. The VO_{AB} area is a collision cone. So, it is necessary to apply only a small translation through Minkowski sum($\oplus$) to produce VO_{AB}

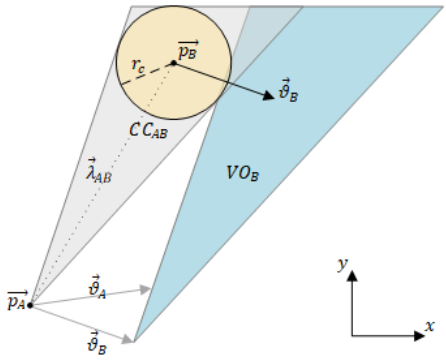


Fig. 3. Velocity obstacle methods

$$VO_{AB} = CC_{AB} \oplus v_B. \tag{12}$$

The velocity group VO_{AB} is described with (12). It is possible to select the collision-free velocity in time k from the velocities that satisfy the condition: $v \notin VO_{AB}$

$$V_i^{optimal} = argmin_{v \notin VO_{AB} | v - v_A^{pref}|}. \tag{13}$$

For the VO technique it is also important to find an optimal velocity beyond the VO area. Currently, only the derivation to the preferred velocity is included in optimum velocity selection. It can be expressed as follows

$$VO_{AB} = CC_{AB} \oplus \frac{(v_A + v_B)}{2}. \tag{14}$$

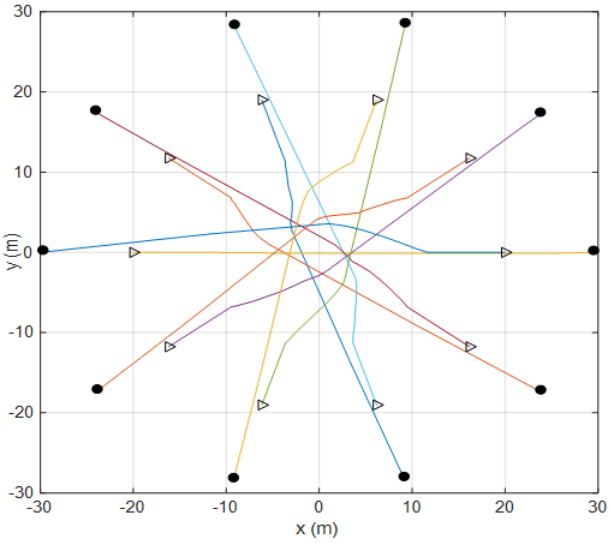


Fig. 4. VO trajectory training result

Next, the velocity obstacle coefficient values have been determined based on training and these results are shown in Fig. 4.

D. Hybrid Potential Field Method based on Velocity Obstacle and Potential Field

A Hybrid Velocity Field method (HPFM), a novel collision avoidance method is proposed. This method combines ideas from the "Potential Field and "Velocity Obstacle" methods.

The method considers a potential collision event which is based on a geometric space, while VO_{AB} is a term based on the velocity space. Equation (15) is used for the transformation of space velocity information into the geometric space. For each agent the speed is translated to relative displacements

$$x_v = (v_{agt} - v_{obs}) \times t_s(v_{agt} \in VO_{AB}) \tag{15}$$

in VO_{AB} that are in the CC_{AB}. Then, the velocity danger level l_v is represented as:

$$l_v = \frac{d_v}{|x_v|} \tag{16}$$

The repulsive function which is related with dynamic obstacles is described as a function. This function

$$f_{rep}^{VO}(\vartheta, v) = e^{-l_v} \tag{17}$$

contains velocity(v) and angle (ϑ) values.

The total potential field function can be determined as below:

$$f_{total}(\vartheta, v) = f_{att}(\vartheta) + \sum_{s=1}^{S} f_{rep}(\vartheta) + \sum_{d=1}^{D} f_{rep}^{VO}(\vartheta, v). \tag{18}$$

Due to the effects of dynamics constraints, the total potential field function and velocity filtering vectors are determined with the equation

$$f_{rech}(\vartheta, v) = \begin{cases} v < v_{max} \\ \sqrt{\left(\frac{dv}{dt} - v\frac{d\vartheta}{dt}\right)^2 + \left(2v\vartheta - v\frac{d\vartheta}{dt}\right)^2} < a_{max}. \end{cases} \tag{19}$$

When adding this inequality to the total function, five different coefficient values can be obtained. These coefficients are for attractive functions (γ, β), statics (ζ, δ) on repulsive functions and dynamics (α) on repulsive functions

$$\begin{aligned} f_{total}(\vartheta, v) &= (\gamma\,|\theta_{target} - \vartheta| + \beta\,|v - v_{pref}|) \\ &+ \left(\zeta \textstyle\sum_{s=1}^{S} e^{-\frac{2(\theta_s - \vartheta)}{\delta \Phi_k^2}}\right) + \left(a \textstyle\sum_{d=1}^{D} e^{-l_v^d}\right). \end{aligned} \tag{20}$$

The five coefficient values of HPFM have been optimised to find the optimal trajectory and velocity for each agent. A genetic algorithm (GA) has been used in the evaluation structure for optimisation of HPFM [15]. The following equation

$$f_{fit}(\gamma, \zeta, \delta, \alpha, \beta) = L_{traj} + 10E_{coli} + \frac{\Delta\theta}{20} + L_{sep} \tag{21}$$

defines the fitness function in a way to reduce the collision possibilities and provides an optimised trajectory.

In the HPFM, a two-dimensional potential field where the magnitude and direction of the velocity can be determined separately has been created. When the force HPFM is applied, the total cost is given as follows

$$T_{cost} = \frac{v_{max}}{v_{interval}} \times \frac{360^\circ}{\Theta_{interval}}. \tag{22}$$

and it includes a product of the considered velocity and angles.

When the speed or angle are small, the HPF optimisation method performs well. However, the computational cost increases when the resolution of the speed and angle increases. Since significant computational costs are not suitable for real-time applications, the smart optimisation algorithm can be a good solution to lower the computation cost T_{cost}.

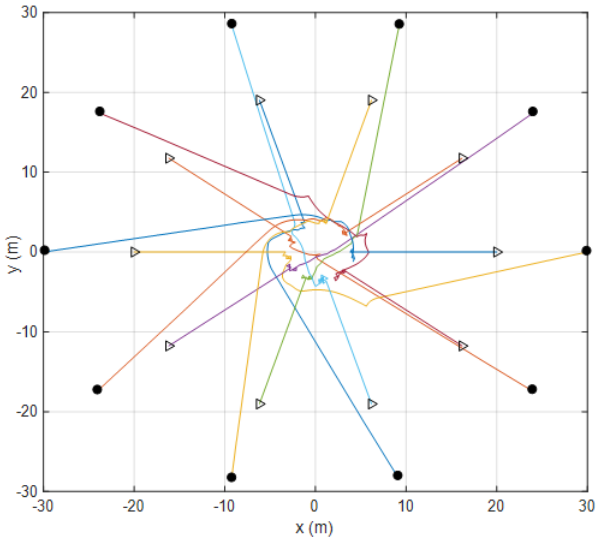


Fig. 5. HPFM trajectory training result

Empirical methods have some difficulties to determine all coefficients. In order to use fitness function via the GA, optimal coefficients have been determined. Fig. 5 shows the obtained result after the training phase. The next section presents testing results from the VO, ODG-PF and HPFM using the same coefficients.

III. SIMULATION RESULTS

In this paper, the experiment has been carried out via MATLAB / Simulink. A point based mathematical model of the agent has been used in the testing stage. Moreover, all determined obstacle avoidance methods have been tested with static and dynamic obstacles. The testing parameters are given in Table I.

After 1000 independent Monte Carlo runs, several performance criteria have been evaluated to determine the accuracy of the algorithms.

Fig. 6 shows trajectory results of VO methods. Multiple collisions have been observed in the conventional VO methods. However, the VO method has some advantages in terms of choosing the optimal trajectory with the minimum task time.

For testing, all coefficients have been fitted with the GA for HPFM and found with empirical methods for ODG-PF and VO.

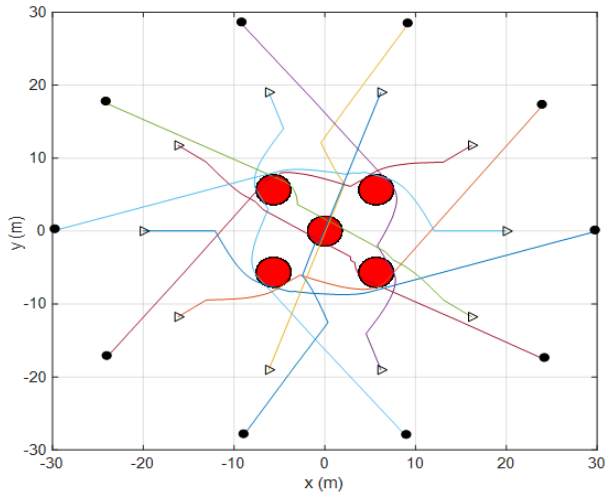


Fig. 6. VO trajectory test result

Fig. 7 shows results with the ODG-PF which has the best results when compared to VO methods with respect to the minimum collision and computational time. Fig. 7 shows that all agents achieve the goal although they have chosen long paths to move.

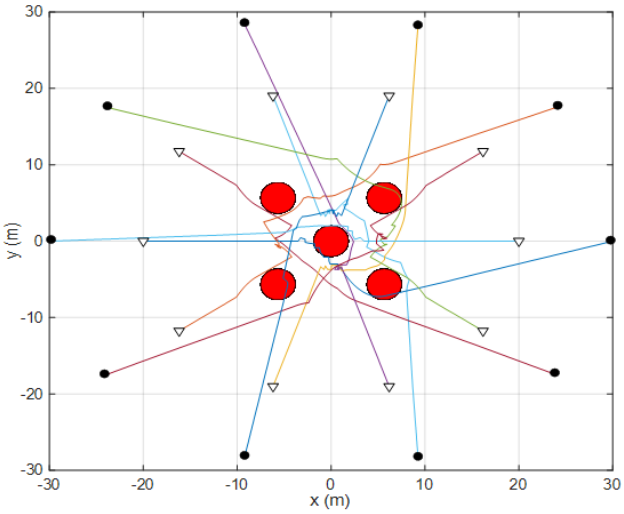


Fig. 7. ODG-PF trajectory test result

Fig. 8 shows the optimal HPFM trajectory, found based on the optimal coefficients calculated using a GA. The HPFM is a combination of the VO and PF, with an improved cost function. Thanks to this combination, it has better collision avoidance results than the VO and better length trajectory than the ODG-PF.

Results for trajectories obtained with all methods are given in Table I. Each value is determined from the mean value of the 1000 runs. Considering the computational time, the ODG-PF

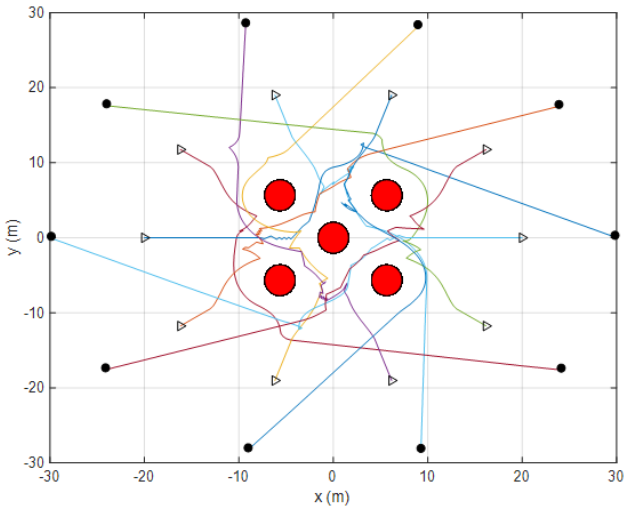


Fig. 8. HPFM trajectory test result

TABLE I
TEST RESULTS OF VO, ODG-PF AND HPFM

Parameters	VO	ODG-PF	HPFM
Computation Time	0.017 sec	**0.008 sec**	0.04 sec
Collision	23.46 times	10.61 times	**5.75 times**
Length of Trajectory	54.58 m	**47.83 m**	62.38 m
Task Time Cost	34.33 sec	**26.40 sec**	36.22 sec

alorithm is faster than the others. Also, the trajectory length and task time cost of ODG-PF are better than those of the other algorithms. However, the collision avoidance results of HPFM are the best, after that we can rank ODG-PF and VO respectively as good results. In this context, HPFM is more reliable than the others. This shows the potential of the HPFM algorithm to be part of MAS or swarm systems.

Fig. 9 gives a performance map of the compared algorithms with respect to easiness of use, task performance, trajectory smoothness and reliability, respectively. This performance map shows the importance of the considered criteria for MAS path planning. Different methods can be used for different tasks for MAS depending on the mission requirements.

The ODG-PF method which was designed for a static scenario is obviously the best one with respect to simplicity and task performance aspects. However, in MAS, the level of reliability is one of the most critical aspects. With respect to reliability, the HPFM gives the best result in the comparison of the other methods. Meanwhile, both of HPFM and ODG-PF produce the trajectories that have almost the same level of smoothness.

IV. CONCLUSION AND FUTURE WORKS

This paper compares collision avoidance methods that could be used in centralised and decentralised UAV collision avoidance systems. The trajectory smoothness and complexity of the the velocity obstacle and potential field methods were analysed. The evaluation of the methods includes UAVs in unknown environments, with static and dynamic obstacles.

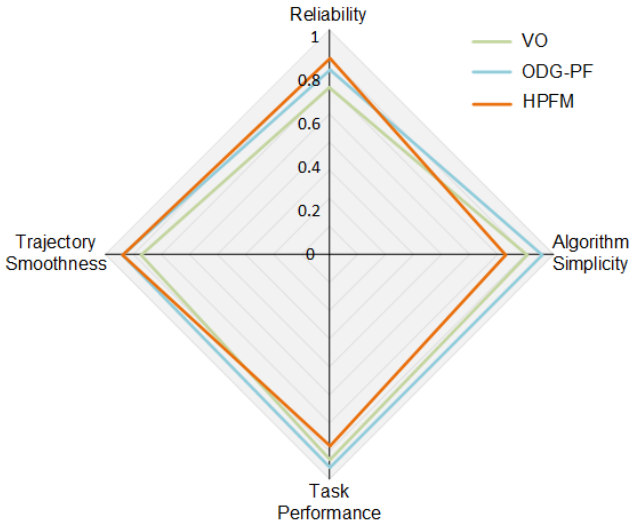


Fig. 9. Performance Map

The Hybrid Potential Field method (HPFM) is compared with the potential field and velocity obstacle methods. The method combines the PF and VO algorithm so that it can achieve the task of collision avoidance in complex environments. The HPFM used a form like the Vector Field Histogram to solve the collision avoidance task. The HPFM has been optimised using a Genetic Algorithm and its performance improved by nearly fifty percent compared with the non-optimised HPFM. The ODG-PF, VO and HPFM are compared over three different testing scenarios.

In a complex situation such as the scenario with both dynamic and static obstacles and multiple agents, the HPFM behavior is more reliable and stable. However, this comes with an increased computational cost. The average computational cost of HPFM is generally several times higher than VO and ODG-PF. The optimal Velocity for each agent must be evaluated via fitness function for the HPFM.

Acknowledgements. We acknowledge the Turkish government for funding the doctoral scholarship of Fethi Candan and we thank the financial support of the UK's Engineering and Physical Sciences Research Council (EPSRC) Programme Grant EP/S016813/1.

References

[1] Y. M. Al-Younes, M. A. Al-Jarrah, and A. A. Jhemi, "Linear vs. nonlinear control techniques for a quadrotor vehicle," in *Proc. of the 7th International Symposium on Mechatronics and its Applications*. IEEE, 2010, pp. 1–10.

[2] B. Albaker and N. Rahim, "A survey of collision avoidance approaches for unmanned aerial vehicles," in *Proc. of the International Conf. for Technical Postgraduates (TECHPOS)*. IEEE, 2009, pp. 1–7.

[3] A. Alexopoulos, A. Kandil, P. Orzechowski, and E. Badreddin, "A comparative study of collision avoidance techniques for unmanned aerial vehicles," in *Proc. of the IEEE international C Conf. on Systems, Man, and Cybernetics*. IEEE, 2013, pp. 1969–1974.

[4] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, "Collision avoidance for aerial vehicles in multi-agent scenarios," *Autonomous Robots*, vol. 39, no. 1, pp. 101–121, 2015.

[5] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," in *Proc. of the International Conf. on Informatics, Electronics and Vision (ICIEV)*. IEEE, 2013, pp. 1–6.

[6] S. Bolbhat, A. Bhosale, G. Sakthivel, D. Saravanakumar, R. Sivakumar, and J. Lakshmipathi, "Intelligent obstacle avoiding AGV using vector field histogram and supervisory control," in *Journal of Physics: Conference Series*, vol. 1716, no. 1. IOP Publishing, 2020, p. 012030.

[7] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robotics Autom.*, vol. 7, pp. 278–288, 1991.

[8] F. Candan, A. Beke, and T. Kumbasar, "Design and deployment of fuzzy PID controllers to the nano quadcopter Crazyflie 2.0," in *Proc. of the Innovations in Intelligent Systems and Applications (INISTA) Conf.* IEEE, 2018, pp. 1–6.

[9] J.-H. Cho, D.-S. Pae, M.-T. Lim, and T.-K. Kang, "A real-time obstacle avoidance method for autonomous vehicles using an obstacle-dependent gaussian potential field," *Journal of Advanced Transportation*, vol. 2018, 2018.

[10] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[11] J. A. Douthwaite, A. De Freitas, and L. S. Mihaylova, "An interval approach to multiple unmanned aerial vehicle collision avoidance," in *Proc. of the Sensor Data Fusion: Trends, Solutions, Applications (SDF) Workshop*. IEEE, 2017, pp. 1–8.

[12] J. A. Douthwaite, S. Zhao, and L. S. Mihaylova, "Velocity obstacle approaches for multi-agent collision avoidance," *Unmanned Systems*, vol. 7, no. 01, pp. 55–64, 2019.

[13] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[14] G. Ganga and M. M. Dharmana, "MPC controller for trajectory tracking control of quadcopter," in *Proc. of the International Conf. on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 2017, pp. 1–6.

[15] Y. Hu and S. X. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4350–4355.

[16] Y. I. Jenie, E. Kampen, C. D. Visser, J. Ellerbroek, and J. Hoekstra, "Selective velocity obstacle method for deconflicting maneuvers applied to unmanned aerial vehicles," *Journal of Guidance Control and Dynamics*, vol. 38, no. 6, pp. 1140–1146, 2015.

[17] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[18] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proc. of the IEEE International Symposium on Industrial Electronics*, vol. 3. IEEE, 2001, pp. 1530–1535.

[19] Y. Peng, "The collision avoidance method for multi-agents system," Master's thesis, University of Sheffield, 2020.

[20] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, "Overview of path-planning and obstacle avoidance algorithms for uavs: A comparative study," *Unmanned systems*, vol. 6, no. 02, pp. 95–118, 2018.

[21] F. Rossi, S. Bandyopadhyay, M. Wolf, and M. Pavone, "Review of multi-agent algorithms for collective behavior: a structural taxonomy," *IFAC-PapersOnLine*, vol. 51, no. 12, pp. 112–117, 2018.

[22] E. Sabudin, R. Omar, and C. Che Ku Melor, "Potential field methods and their inherent approaches for path planning," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 18, pp. 10 801–10 805, 2016.

[23] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proc. of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1. IEEE, 2000, pp. 256–263.

[24] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. of the IEEE International Conf. on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[25] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[26] Z.-z. Yu, J.-h. Yan, J. Zhao, Z.-F. Chen, and Y.-h. Zhu, "Mobile robot path planning based on improved artificial potential field method," *Harbin Gongye Daxue Xuebao(Journal of Harbin Institute of Technology)*, vol. 43, no. 1, pp. 50–55, 2011.

Nonholonomic path planning for a mobile robot based on Voronoi and Q-Learning algorithm

Mustafa Mohammed Alhassow
Deptepartment of E. & Comp Engineering
Altinbas University
mustafaalshakhe@gmail.com

Oguz Ata
Deptepartment of E. & Comp Engineering
Altinbas University
oguz.ata@altinbas.edu.tr

Dogu Cagdas Atilla
Deptepartment of E. & Comp Engineering
Altinbas University
cagdas.atilla@altinbas.edu.tr

***Abstract*— This paper discusses a differential route planning problem for the mobile robot based on Voronoi and Q-learning algorithms (QL). The issues with planning in static settings with barriers are treated as a problem of seeking a feasible route between the original and target stage. Since the non-holonomic car robot is differentially constrained and geometrical environment with the concave shape of obstacle some modifications will be inserted. Any type of car can be as a robot model of our work. This is a special case of a single vehicle, which only goes forward at constant speed and can only turn left and right. Voronoi introduces the world surrounding smart agents for robots, video games, and military simulations, improving the reliability of route planning and reducing Q learning algorithms by upgrading the Q table, thereby reducing both space and time complexity, depends on prior knowledge of intelligent agents. The simulation setup was conducted using a 2D environment. The simulation of the proposed work can use a different style of maps. Finally, a comparison with other related works is performed and the result of these comparisons showed that our work provides a good trajectory with performance.**

***Keywords*— *Car-like robot, nonholonomic system, path planning, Q-Learning, Voronoi diagram*.**

I. INTRODUCTION

In an environment, many robot paths are possible to accomplish a defined target, but according to certain requirements the best path is chosen. The shortest distance and the shortest time are these conditions. The shortest distance is the most common criterion. Path planing is an acritically important issue, since it aims at searching for a path that is as short as possible, subject to certain restrictions, including the provided environment with free collision, as mentioned in [1]. Some of the well-known non-heuristics are cell decomposition, Voronoi diagrams, and B-Spline curve. In the cell decomposition technique, two methods are used. They are the exact and the approximate cell decomposition methods. The exact method is used to divide the search space into simple cells and builds the adjacency relationships among the cells. It explicitly determines the obstacles and build the cells as mention in [2], [3]. The combination of all the generated cells will produce the exact free space. However, determining the exact free space in a high dimensional environment is not an easy task, hence the approximate method was introduced. B-splines is a mathematical function that is used to form a curve using a few control points in a segment rather than the entire points in the segmented section. [4], applied the B-spline function with a modification by introducing additional control points in the neighborhood of each obstacle, they also develop methods to shift these new control points away from obstacles and into clear areas [5], introduced user-specified threshold flying altitude in the enemy terrain and used these new thresholds to generate a path for the Unmanned Aerial Vehicles. The Voronoi D* Algorithm is proposed to achieve high route planning performance and satisfy distance specifications for mobile robot applications [6], but it has a problem with local minima. Also considering complex environments structured with obstacles was introduced as mention in [7], but also it has a problem with that the trajectory is not good enough. Machine Learning (ML) algorithms are used to support individual mobile robots in unfamiliar environments as a preferable alternative tool. In that case ML emerges as the remedy by Reinforcement Learning (RL), which enables agents to adjust/change their behavior for maximum outcomes in a given environment[8]. The game theory, information theory, control theory, and situational analysis were originally included in RL [9], and was adopted as an autonomous handheld robotic navigator over time[9], [10]. Many RL algorithms have appeared in recent times thus concentrating on route planning and avoiding barriers. The priority is less to refine the route and much less to develop the course and reduce the time needed to achieve convergence. The Q-Learning(QL) method [11], is commonly used for RL implementation and has recently been used in Worldwide diverse areas of robotics. Due to its simplicity and convergence proof, QL is the most frequently used RL algorithm [12]. On the other hand RRT (Rapidly exploring random tree) algorithm proposed a solotion for a nonholononomic path planning but its has a bad trajectory [17].

The robot route preparation of the mobile robot through its surroundings [13], [14] is a fundamental problem in mobile robotics. In order to prevent Confrontations with known static and moving collisions objects, it is necessary to find a viable path between the initials and goals in the configuration field. This process can be done either by using local sensor information or by using global, previously established robot environment information. This dilemma can therefore be separated into two categories [15],

1) *Local (On-line)* - Generates direction from original to destination with local sensor information incrementally during travel.
2) *Global (off-line)* - Compute the entire route to the pre-motion target using global map information. Researches have

advanced over time toward improving learning algorithms in order to address shortcomings of classic approaches of route planning. Our paper focuses on the construction of an off-line solution based on algorithms of Voronoi and QL.

II. VORONOI DIAGRAM (VD)

Voronoi charts are most utilized as a part of computational geometry, Voronoi outlines have discovered their way in numerous application territories, for example, PC designs (impact identification, movement arranging). Voronoi diagram in view of Euclidean separation are the best known; such charts parcel the 2D plane in areas. We start by re-call the voronoi properties with descriptions of the element properties of 2D Voronoi graphs. Given an arrangement of (V) focusesP_1, P_2, ..., P_n in the plane, a Voronoi graph isolates the plane into n Voronoi districts with the accompanying properties:

- Each point pi lies in precisely one area.
- If a point q ∈ V lies in an indistinguishable locale from pi, at that point the separation from pi to any point [16].

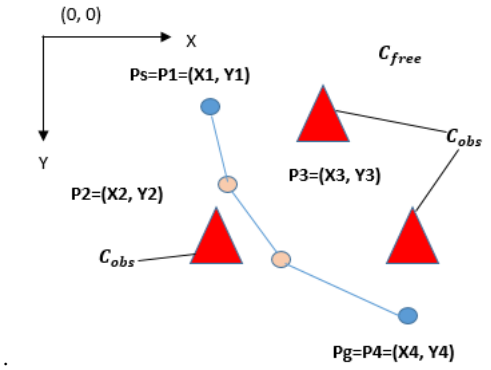


Fig. 1. Description of Robot Environment.

From Fig.1 p ∈ C ⊆V^2 its x and y coordinates can be interpreted, and we write p = (x, y). Because in this paper we are interested in line roads, the robot path P is defined by a sequence of (n) points p_1 , p_2, . . . , $p_n$in C. Hence, we write P = (p_1, p_2, . . . , p_n), whereby p_i∈ C for i = 1, . . . , n. Defining a mobile robot's start and goal as p_s, p_g, respectively, it must hold that p_1= $p_s$and p_n= p_g every robot route appropriate. The exact direction of the robot is then determined by following connections p_i and $p_{i+1}$of a path P by straight lines $L_{pi,pi+1}$for each i = 1, . . . , n − 1. Then the collection of points from the robot p_s to p_g is given by P_p⊆ C, whereby $P_p$consist of all points inside the segments of the line$L_{p1,p2}$, . . . $L_{pn-1,pn}$. We then refer to the P direction as collision-free if P_p ∩C_{obs} = ∅, That is, the crossroads enclosed by the road and by the obstacle area do not exist. We also implement the function for later use of various algorithms:

$$\text{Collision Free (p, p}\hat{}\text{)} = \begin{cases} true\ if\ \ L_{p,p\hat{}} \cap C_{obs} = \emptyset \\ false\ otherwise \end{cases}$$

This decides whether the direct relation $L_{p,p\hat{}}$ the field of obstacles is crossed by two p-pointsC_{obs} we write.

$$A = \{p/P_p \subseteq C_{free}\} \quad (1)$$

A selection of roads without obstacles. The gap between two points is further defined.

P_i= (x_i+y_i), P_j= (x_i+y_i) ∈ C as

$$d(p_i, p_j) = \sqrt{(x_i - x_{j)}{}^2 + (y_i - y_j)^2} \quad (2)$$

The key goal of this paper is to calculate appropriate obstacle-free robot paths between a given starting point using the notation introduced above $p_s \in C_{free}$and goal point $p_g \in C_{free}$.

III. Q-LEARNING (QL)

QL is based on the idea that if you produce a good result you will get a reward and if it produces poor results, you will receive a punishment, the parameter (r) as in (3) α and μ are learning parameters of the same equation, representing the condition as well as the action,$Qt(s_t, a_t)$is the importance of the state-action. The algorithm saves the approximate values in the simplest way$Qt(s_t, a_t)$, an operation for all possible pairs in a given (st) state [17].

$$Q_t + 1(s_t, a_t) = Q_t(s_t, a_t) +$$

$$\alpha_t \left[r(s_t) + \gamma \max_a Q_t(s_t, a_t) - Qt(s_t, a_t) \right] \quad (3)$$

Updating Q values as following:

$Q_t + 1\ (s_t, a_t)$ is the worth (quality) of an action in the next state to be transferred by the agent r (s_t). R is the father collection of the immediate strengthening received in State(s) r, r (s_t) belong to R (s_t), α is the pace of preparation (normally set between 0 and 1), μ is the discount rate time, t is a discrete time phase, t = 0,1,2. $\max_a Q_t(s_t, a_t)$is the Q value of the operation with a higher utility in the future state as mention in [16]. Choose action (a) to maximize the Q(s, a) and Q table updates to show how the function is done in any step(s). First the algorithm operates as follows, set the table passage Q(s),(a) to zero for each state operation match, search for the current state before the end of the time: pick an activity and perform it quickly award (r) observe the new state, and then change the table passage to Q(s) as follows:

$$Q_{t+1}(s, a) \leftarrow (1 - \rho) Q_t(s, a) +$$

$$\rho[R(s, a) + \gamma \max_a Q(s', a')] \quad (4)$$

IV. EXPERIMENT PROPERTIES and RESULT

In this work we have used for non-homonymic path planning both of Voronoi and Q-learning to build a path can pass through the obstacle. Our work aims to get the best path for the mobile robot by using both benefits of the VD. It also speed up the search by making the robot Moving from point of beginning to end by select the optimal path. This experiment starts with detecting the environment and specifies the obstacle position by using the Voronoi prosperities after that the Q-learning will

guide the robot to its final goal. In the simulation implemented we can use various map types that reflect the beginning, destination, and the obstacle block as the green point is the start state, the blue point is the destination point, and the red tringle is the concave shape of the obstacle as shown in Fig. 2.

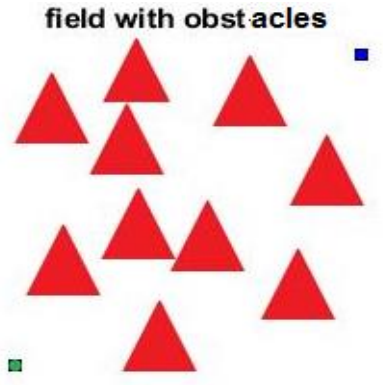


Fig. 2. The Start, Goal, and Obstacle in the Environment.

Voronoi will initialize and collect the space and generate the sequence of routs that will be away as far as from the obstacle.

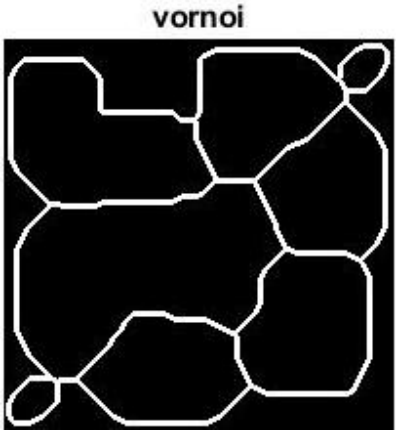


Fig. 3. Generated Voronoi Paths.

After all possible paths generated the Q-learning will be initialized and start taking the action per each state and get the reward and passing to the next state till reach the goal. The final step is to build the path and move from the start to the goal state.

Fig. 4. Building The Final Path That Make The Robot Reach The Goal State.

All this work is done in a static environment with a random obstacle and random start, end position. This research offers-ways to improve the working approach of stationary settings for learning skills. Be mindful of this end goal of the proposed work and to make sure that our algorithm has a decent execution that is better we compared our work with some related and the result as follow:

TABLE I. ILLUSTRATE THE COMPARISON OF OUR WORK WITH SOME RELATED.

Type of environment	**Comparison**		
	Algorithm	***Evaluation Time seconds***	***Evaluation Time seconds for proposed work***
Static	D-Star [5]	45sec	24sec
Static	RRT [17]	-	24sec

The simulation results will be affected the obstacle avoidance capability with better performance with respect to time and without wasting the space of the environment because of that we can get better trajectory when we used VD. RRT[17] has a bad trajectory because of that we have used VD to avoied this problem so our work is better than them as shown in fig 5, and 6.

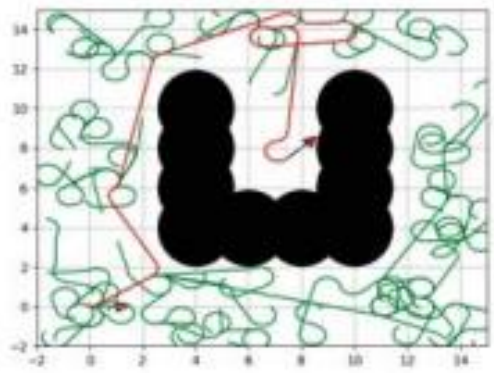


Fig. 5. Search Tree[17]

Also the simulation result that are considered using ROS programming environment, Stage simulator which provides a full view of the map using modified RRT algorithm is shown in fig6.

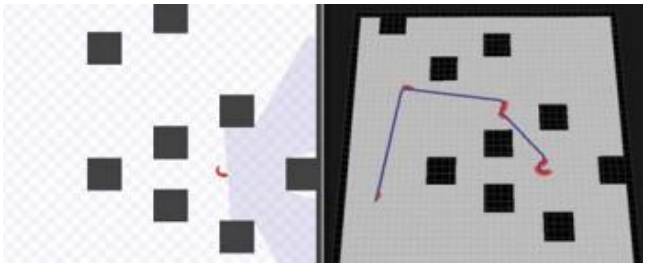


Fig.6. Robot's Trajectory[17]

From both these figures 5, and 6, we can see the bad trajectory and the wasting of search space, also one of the main

objectives of path planning is the shortest path problem which is not considered.

On the other hand D-star, it's admissible to fined the goal but it's stuck in local minima and we may losing time. In our work, we assumed that if there is no possibility to reach our goal there is no need to lose the time from the beginning the path is not possible and this is the power of VD. As a result, the VD supports QLwith prior knowledge because of that we get rewarding more than punishment as a status of the action.

V. CONCLUSIONS

Recently, technology for learning agents has been growing and reaches high standards. It is found in diverse implementations on the other hand. Time estimation and the construction of such modern learning algorithms that follow the precision of learning and time metrics has been very significant. This learning algorithm improves the learning technique, such that this method enhances accuracy as the time required for this process is integrated when the time is low as soon as the process has been compared more effectively than time parameters. Voronoi diagram introduced to produce a several routs that will reduce the time spend to collect the aria also to enhance the Q-table and increment the reward of each step. In this study we proposed nonholonomic path planning based on VD and QL to build a path avoided the obstacle. VD will support QL with a full view of paths. QL will generate Q-table. Q-value shall be specified as the estimated future discounted payment in the event that the agent takes the optimum direction (a) in the state (s). The Q-learning algorithm operates by estimating the values of the pair of states as well as the algorithm's leading method. The advantage of the proposed algorithm illustrate as: reduce the search spaces by applying voronoi algorithm. Reduce the number of states, and save the memory usage. Increasing the rewarding per each state of QL. Avoiding local minima problem, and solve the concave shapes of an obstacle.

Since we have got a good result of our work we are planning to apply it for a dynamic situation with random position of the obstacle.

REFERENCES

[1] M. Agarwal and P. Goel, “Path Planning of Mobile Robots using Bee Colony Algorithm,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 2, pp. 86–89, 2013.

[2] S. K. Debnath *et al.*, “Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles-A Review,” in *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019*, 2020, pp. 99–111.

[3] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, “A comprehensive review of swarm optimization algorithms,” *PLoS One*, vol. 10, no. 5, pp. 1–36, 2015.

[4] E.-M. Kan, M.-H. Lim, S.-P. Yeo, J.-S. Ho, and Z. Shao, “Contour based path planning with B-spline trajectory generation for unmanned aerial vehicles (UAVs) over hostile terrain,” *J. Intell. Learn. Syst. Appl.*, vol. 3, no. 03, p. 122, 2011.

[5] L. Jiang, S. Wang, J. Meng, X. Zhang, G. Li, and Y. Xie, “A Fast Path Planning Method for Mobile Robot Based on Voronoi Diagram and Improved D* Algorithm,” in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2019, pp. 784–789.

[6] M. Dirik and F. KOCAMAZ, “RRT-Dijkstra: An Improved Path Planning Algorithm for Mobile Robots,” *J. Soft Comput. Artif. Intell.*, vol. 1, no. 2, pp. 11–19, 2020.

[7] J. Liu *et al.*, “QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks,” *Comput. Commun.*, vol. 150, pp. 304–316, 2020.

[8] E. S. Low, P. Ong, and K. C. Cheah, “Solving the optimal path planning of a mobile robot using improved Q-learning,” *Rob. Auton. Syst.*, vol. 115, pp. 143–161, 2019.

[9] D. Luviano and W. Yu, “Continuous-time path planning for multi-agents with fuzzy reinforcement learning,” J. Intell. Fuzzy Syst., vol. 33, no. 1, pp. 491–501, 2017.

[10] C. Qu, W. Gai, M. Zhong, and J. Zhang, “A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning,” *Appl. Soft Comput.*, vol. 89, p. 106099, 2020.

[11] J. S. Park and J. H. Park, “Enhanced Machine Learning Algorithms: Deep Learning, Reinforcement Learning, and Q-Learning,” *J. Inf. Process. Syst.*, vol. 16, no. 5, pp. 1001–1007, 2020.

[12] C. Goerzen, Z. Kong, and B. Mettler, “A survey of motion planning algorithms from the perspective of autonomous UAV guidance,” *J. Intell. Robot. Syst.*, vol. 57, no. 1–4, p. 65, 2010.

[13] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[14] Z. Shiller, “Off-line and on-line trajectory planning,” in *Motion and Operation Planning of Robotic Systems*, Springer, 2015, pp. 29–62.

[15] N. Habib, D. Purwanto, and A. Soeprijanto, “Mobile robot motion planning by point to point based on modified ant colony optimization and Voronoi diagram,” in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2016, pp. 613–618.

[16] S. Li, X. Xu, and L. Zuo, “Dynamic path planning of a mobile robot with improved Q-learning algorithm,” in *2015 IEEE international conference on information and automation*, 2015, pp. 409–414.

[17] Živojević, Dino, and Jasmin Velagić. "Path Planning for Mobile Robot using Dubins-curve based RRT Algorithm with Differential Constraints." 2019 International Symposium ELMAR. IEEE, 2019.

Classification of Power Quality Events Using Deep Learning

Hammad Khalid
Department of Computer Science,
Istanbul Sabahattin Zaim Universitesi,
Istanbul, Turkey
https://orcid.org/0000-0001-9597-8455

Abdulfetah Shobole
Department of Electrical Engineering,
Istanbul Sabahattin Zaim Universitesi,
Istanbul, Turkey
https://orcid.org/0000-0002-3180-6504

***Abstract*— Power Quality (PQ) can be defined as a clean supply voltage that stays within the prescribed range in a smooth curve waveform. A power quality problem is defined as any problem that causes voltage or frequency deviations in a power supply, and it may result in failure or maloperation of a network. Therefore, continuous monitoring is also required in case of malfunction in these cases. In this paper, we have presented a deep learning-based power quality event classification method. We have used the proprietary electric relay wave-form data from The Turkish Electricity Transmission Corporation (TEIAS), as well as generated wave-form from MATLAB-Simulink, to train our model, using Convolutional Neural Networks (CNNs). The results proved to be effective, and can open the path to further research in this direction.**

Keywords*— *Power Quality, Deep Learning, Neural Networks, Convolutional Neural Networks, Electrical Distribution System.

I. INTRODUCTION

A power quality problem is defined as any problem that causes voltage or frequency deviations in a power supply, and it may result in failure or maloperation of a network. Power quality may also be described as the Load's ability to function properly. Without good power quality in an electrical network, Load's electrical and mechanical equipment may overheat, malfunction, fail prematurely, require high maintenance, and in many cases they may not operate at all. Power quality has various effects, and it is important to understand what equipment causes poor quality issues in electrical networks. Most of the poor quality issues are generated by devices in buildings. Non-linear loads are the major cause of poor quality issues. There are 10 power quality problems that need to be concentrated upon: Over-voltage, surges, spikes, transients, frequency variation, under-voltage, sags, blackouts, noises, and harmonics. An extensive literature survey suggests that there is no generally accepted method for characterization of these disturbances and suitable limits are not yet found in any international standard. One of the reasons for the lack of characterization methods is the difficulty of defining suitable site indices for each discrete disturbance type [1]. A diagram of power quality problems is shown in Fig. 1.

Power quality can be defined as a clean supply voltage that stays within the prescribed range in a smooth curve waveform. Over-voltage conditions can be defined over the range of a normal voltage or whatever normal voltage defines. Nominal voltage is a reference voltage, used to describe the batteries, cells, or electronic systems such as a 12 or 24 volt batteries. Under-voltage is under the lower boundary. The peak is the extreme condition of over-voltage, and blackout is the extreme condition of under-voltage. We can describe the clean power or high power quality voltage as a steady smooth sine wave. It is what we want for any computers, servers, or other Computing devices in a data centre. The first issue of power

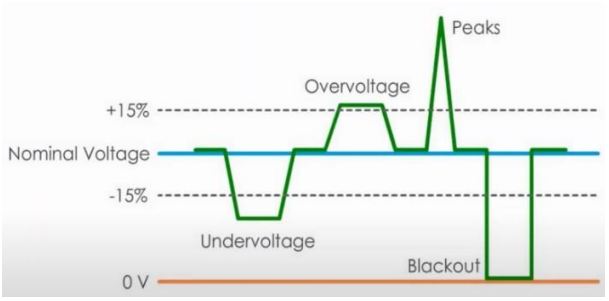


Fig. 1. Effects of PQ events on waveform.

quality that we discussed is over-voltage. Over-voltage refers to the power supply above the normal voltage range. Over-voltage can last for minutes to several days. It includes several different conditions such as surges, spikes, and transients. Surges occur over the short-term and they may be caused by lightning strikes or switching off heavier loads. In most situations, power surges are not a strong enough to interrupt or damage devices, but they can be very harmful sometimes. In a small office or home office, surge protectors are good enough. Spikes are and very short duration of voltage increases. They may be caused by lightning strikes or power outages. Transients are very fast voltage increase in terms nanoseconds. Frequency variations are a change in frequency. Voltage involves wavelengths. The longer the wavelength, the lower the frequency, and vice versa. The US electrical grid operates at 60 Hz, or 60 cycles per second. Anything but 60 Hz is considered frequency variation. Now, under-voltage is the opposite of the over-voltage. It is defined as a condition where the applied voltage drops to about 90\% of the normal range or nominal voltage value. They are also known as brownouts. This condition can last for a few minutes to days. Sags, or commonly referred to as dips, is the under-voltage condition over the short-term. It may be caused by switching on heavy loads. Sags also rarely damage electronic devices. A common cause of sags for industrial customers is turning on large loads such as large motors [2]. Blackouts are the total loss of power, which brings down the whole network, and interrupts normal business operations. Many times, blackouts become a a matter of life and death to a business. Noise is a term well known by most people. It is a high-frequency distortion of the voltage waveform caused by disturbances related to the Electromagnetic Interference (EMI) or Radio Frequency Interference (RFI). Finally, harmonics are unwanted frequencies, which superimposed on the fundamental waveform creates a distorted wave pattern. They are a recurring distortion of a normal wavelength.

II. Motivation and Previous Works

These power quality problems are not equally dangerous. Some of them may just affect the best performance of electronic devices, while some would damage devices, data, or even kill a business overnight. For a large datacentre, problem quality problems means potential disasters. The steady, clean and uninterruptible power supply, is the minimum requirement for business continuity. Power quality has a direct impact on electrical consumption and electrical demand. Bad power quality increases the current/amperes required in electrical networks. Good power quality decreases the current required in electrical networks. Therefore it is important to understand that not only has a damaging impact on all the electrical and mechanical equipment, it also has an impact on your electricity bills. This is why it is important that poor quality issues are addressed on a timely manner. A lot of actions have been taken to ensure efficient power quality. Many solutions in the form of devices have been presented. The approach that we have taken is with the perspective of Computer Vision.

Previously many works have been done with respect to power quality solutions. Regarding the deep learning solutions, Balouji and Salor [3] presented an image recognition method on the PQ event waveform images using a Multi-layer Convolution Neural Network (MLCNN). Their approach seemed to be very effective as their test data of power quality events classified with 100% accuracy. They used input images of dimensions (220x220x3). Their approach is quite similar to us, however, the specific amount of data they have used is not specified. This raises a doubt that their model has enough experience to tackle the future power quality events that may rise in the power system. Ahajjam et al. [4] have also done PW event classification using deep learning, however the difference that we have their work is that they have used temporal-spectral images to train teir CNN model. Their simulation results show that they are able to detect power quality disturbances with good accuracy, and are able to detect and distinguish between several number of power quality events in an image. Working on temporal-spectral images proved to give good results in power quality event classification.

Apart from deep learning based approaches, Saini and Kapoor [5] have presented a review of power quality events analysis. It mostly includes signal processing, and optimization techniques for power quality analysis. Mahela et al. [6] have also done a comprehensive review on power quality event analysis. This review includes Neural network based classification and also Support Vector Machine (SVM) based classification. The review of Saxena et al. [7] is somewhat dated, however they have performed good analysis on the key issues of power quality event analysis and presented some of the still well-known classification methods. An innovative approach to classify power quality events was presented by Chintakindi et al. [8], where they have proposed to use an improved version of the S-transform for the classification. They have simulated their proposed method on MATLAB software in accordance with IEEE Standard 1159. The effectiveness of the algorithm proved to identify multiple events at once. Similarly, Dash and Subudhi [9] have used S-transform for classification, however, they have combined a technique called Whale Optimization Algorithm (WOA), and SVM with it as well. They validated the proposed technique on real time signals obtained from the circuits.

III. Deep Learning Overview

The vision behind Deep Learning, and hence more accurately, Convolutional Neural Networks (CNN). CNNs are mostly used for problems like images recognition or speech recognition. The reason for this is that are CNNs outperform normal artificial neural networks in these types of problems. In fact, the accuracy of some CNN models at image recognition is even better than humans. The limitation in a normal artificial neural network is such that, as we understand an artificial neural network gets the data of each pixel as input, into the first layer of neurons. So if we have a 16x16 pixel image and we want to find what is in that image, all 256 pixels would be fed into the first layer. The problem in this case is that we would see the pixels randomly with no order and won't be able to identify the image. This is because we are not considering the effect of neighbouring pixels. If we consider the order of the pixels, only then we are able to identify what is the object in the image identifying the object. As the human brain works, we do not look at individual points or pixels, we recognize pattern in group of points or pixels. In fact, cells in our visual cortex respond to different patterns. Some respond to horizontal lines, some respond to vertical lines, and some respond to other complex patterns. The output of the lower-level neurons is then processed by higher-level neurons, to identify object in our visual field. CNNs are inspired from this concept. In, CNNs, instead of looking at each individual pixel, we look at a group of pixel. If you look at a group of pixels, we are more likely to pick up different features of the object in the image. So once we know the features, it is more likely that we can predict the object in the image.

As shown in Fig. 2, this concept is implemented in the way that the image is at the bottom, which is the input image to our Network. As seen on top of it, we will have a convolutional layer. This is the most important concept in CNNs. We have a convolutional layer which comprises of neurons, which take in information from a group of pixels in the previous layer. So in the first layer above the image, a neuron gets information stored in the pixels within the corresponding rectangular box as shown in the figure. This rectangular window is also referred to as the receptive field of the specific neuron. Similarly, the same process happens in the next layers with respect to the previous layers. This architecture allows the network to concentrate on lower-level features in the first layer, and then assemble these features into larger higher-level features in the next hidden layer, and so on.

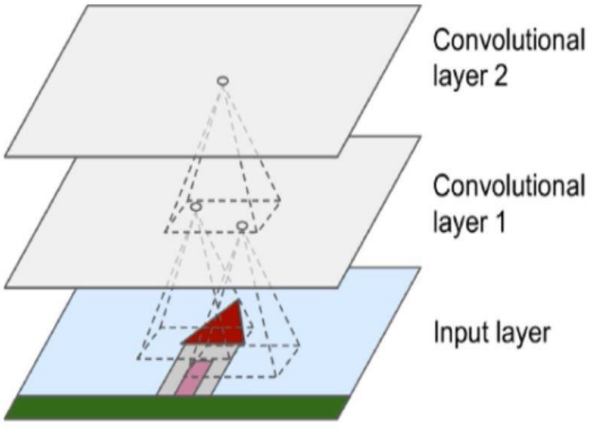


Fig. 2. Windows/Receptive fields in convolutional layers.

The concept of stride is such that the first neuron in the convolutional layer would be looking at a specific set of pixels. The next neuron in the convolution layer will have a slightly shifted receptive field/ rectangular window. This shift in the view from one neuron to another is referred to as stride. There is overlap between receptive fields if the stride is too small. Hence, there is small overlap if the stride is large, and large overlap if the strike is small. One thing to note from this is that if the stride is small, and overlap will be less, then fewer neurons will be required in the upper layer. Hence, the stride will determine the size of the upper layer as well.

As mentioned before, a cell in the convolutional layer gets information from a set of pixels in the previous layer. In case we have 25 pixels in a receptive field, we convert the 25 pixel values to one value using a filter. A filter is a matrix of the same dimensions as our receptive field. Therefore in case of 25 pixels, the 5x5 receptive field will have a 5x5 filter. We multiply each window pixel value with the corresponding filter value and add all of these products up. This product will represent information in the 25 pixels now. What we get after applying a filter is called a feature map, as shown in Fig. 3. Each feature map has some particular feature highlighted. We always use many types of filters, so that each filter creates different feature maps containing different features. This means that the convolutional layer is going to be a bundle of feature maps, and each feature map has some particular highlighted feature. As seen in Fig. 3, each cell on convolutional layer 2 will be getting information of all the feature maps in the previous layer, because only then can the cells in the convolutional layer 1 combine the different features to find more high-level features.

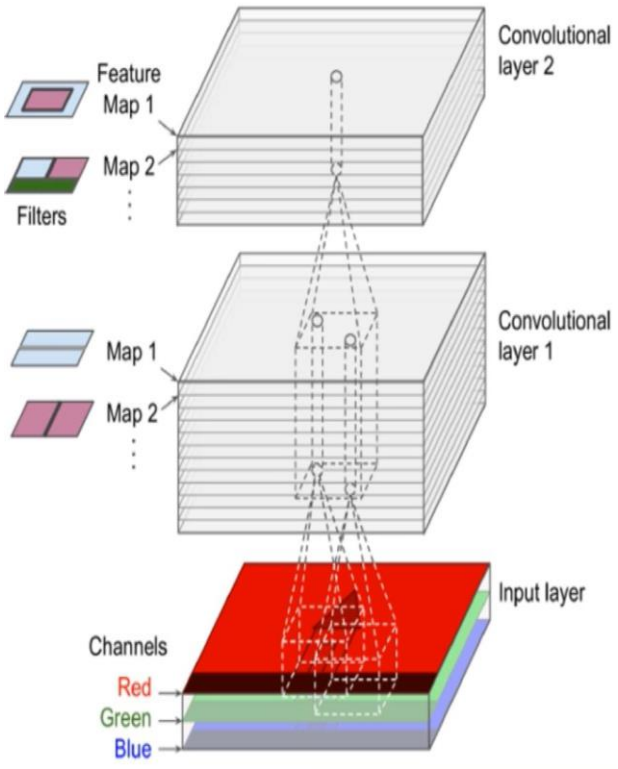


Fig. 3. Feature maps in convolutional layers.

A pooling layer is another concept used in our convolutional network to reduce the computational load, memory usage, and the number of parameters to be estimated. It basically subsamples i.e. shrinks the input image. We have the option of applying two kinds of pooling: Max pooling and average pooling. Commonly, max pooling works better because it highlights the main features, instead of averaging them out. It is trade off basically. We give away some extra information in the previous layer, to reduce the computational load on our system.

IV. Data Preparation

The data that we have used in this research work are the current waveform images of the practical electric relays at the Turkish Electricity Transmission Corporation (TEIAS). The data that we have is 3 years' worth of real-time data, which includes hundreds of power quality event occurrences. In this paper, we have used a few of the images, as well as generated wave-form data from MATLAB-Simulink, in order to present a general idea of the training and accuracy of the proposed overall work. After this, we will use all of the proprietary data obtained from TEIAS to train our deep learning model, so that it can detect and rectify all kinds of power quality events in the future using past 3 years' worth of experience from an actual power transmission company.

We have all of our data in the form of .JPEG images. In terms of data preparation, we have three folders: Training, validation, and testing. Each of these folders contain two sub-folders: Fault and no fault. In these folders, we have added images of our wave-forms that have noise and cause fault in the power equipment, and the ones that have no fault. Fig. 4 shows some samples of the input images that have noise in the wave-form, whereas Fig. 5 shows the ones that don't have any noise. This is the data we used for a the deep learning model. We have used 15 images in each of these folders. The dimensions of these images are not standardized, therefore we re-scaled the dimensions of all the images data into 255x155 pixel value. Fig. 4 shows some samples of the input images we have fed into our model.

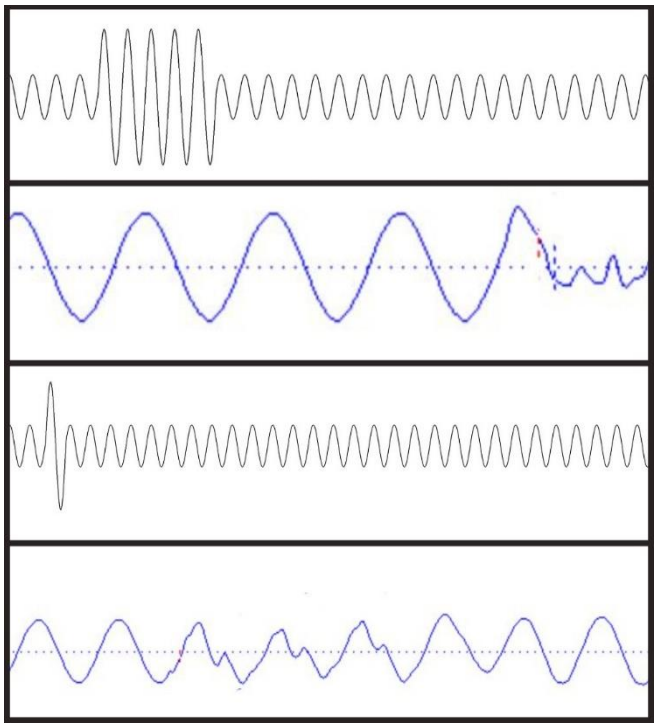


Fig. 4. Input image data of wave-forms with noise.

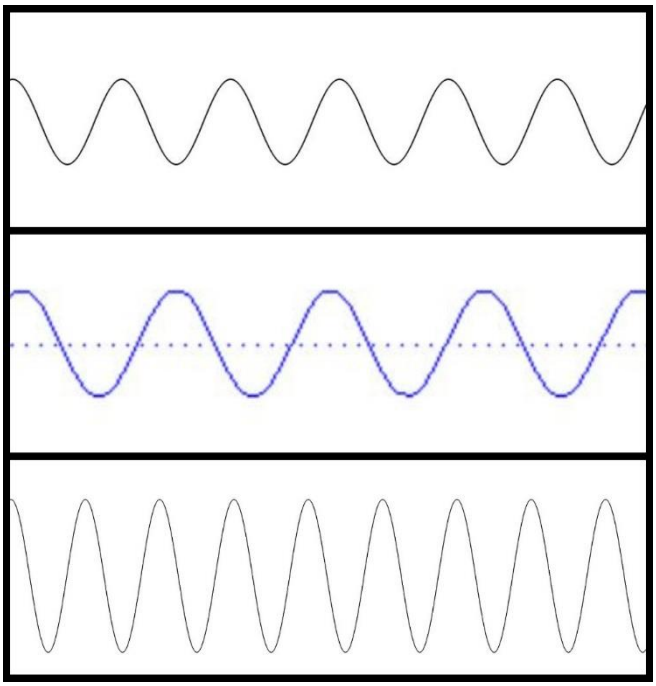


Fig. 5. Input image data of wave-forms without noise.

V. Proposed Method

In this section, we have described the approach that we have used in building and using our Python-based deep neural network model, for the recognition of power quality events, and then classifying them. As discussed before, power quality events consist of several different cases. The event that we focused upon in this paper is Noise.

Python has been used to design and train our CNN [10]. As mentioned before, the dimensions of the input images are not standardized, therefore we re-scaled the dimensions of all the images data into 255x155 pixel value. Before inputting the data into our model, we converted the images to the form of values. We read the picture files, decoded the .JPEG content to RGB grids of pixels, getting the three channels red blue green in our input. Then we converted these RGB floating-point values into values between 0 and 1.

Data pre-processing has been performed using the Keras image generator module. We have fed the waveform images directly from our folders to this module, and re-scaled the images inside this module as well. Using this module, we have the set the batches to be fed to the training in each epoch, to be 15. After the pre-processing of the images, we created the structure of our CNN model. We used 4 different convolutional layers with Max pooling. After that we applied a dense layer and finally the output neuron. For our first layer, we have a convolutional layer with 32 filters and 3x3 window. As we mentioned our input images dimensions before, our input size is 255x155x3. After this we have a max pooling layer with a window of 2x2. Another convolutional layer is added after this with 64 layers and the same 3x3 window. Following this layer, we add 2 more convolutional layers with 128 filters, 3x3 window, and a max pooling layer with a window of 2x2, each. We used Flatten after these convolutional layers, followed by a single dense layer with 512 neurons, and finally, the output layer with neuron, since we want to predict between 2 options.

VI. Results and Discussion

Initially, we compiled our model with the loss of binary cross-entropy [11][12], and used the RMSprop optimizer [13]. RMSprop has an advantage when dealing with image processing. We used the learning rate of 0.001, and trained our model for 40 epochs. As shown in Fig. 6, we have achieved a good accuracy score and low loss.

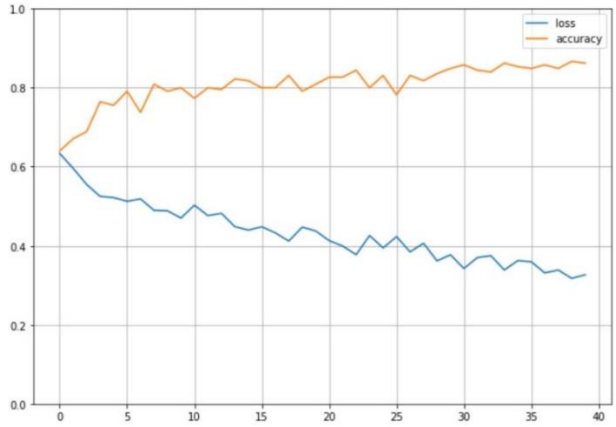


Fig. 6. Model training results (40 epochs).

The validation accuracy and as well as the loss has been plotted in Fig. 7. We are achieving good accuracy here as well, but according to the behaviour of the validation accuracy graph, it shows signs of over-fitting in our model. In order to fix this overfitting, we created dummy data. We did this by modifying our existing data into different forms by applying transformations like zoom, shear, rotation, height shift, width shift, and horizontal flip. Shearing means that we pull an edge of our picture, which makes the shape of the picture like a rhombus. Width shift means that we are shifting our whole image left or right, and height shift means shifting it upwards or downwards. One more change we made was adding a dropout layer to our model architecture. The dropout layer randomly deactivates 50\% of the neurons during each epoch. The model was then trained with the remaining different 50\% neurons at each epoch. Adding the dropout layer is very effective in removing overfitting in models. We trained our model again after making these changes.

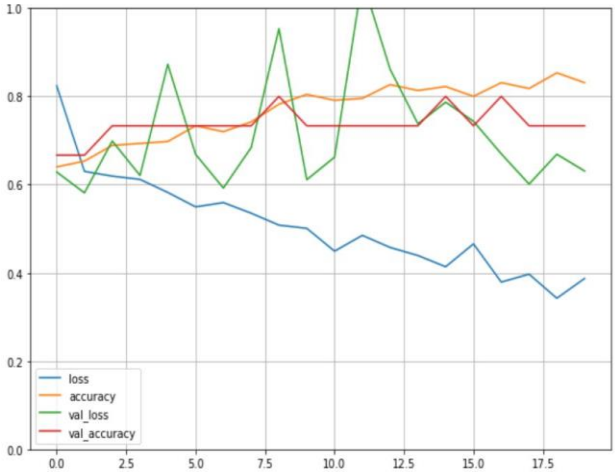


Fig. 7. Model training results with Overfitting (20 epochs).

The model was trained many times by changing several parameters, and Fig.8 shows how the results varied with each training. As seen in the figure, beyond 55 epochs, there is not room for much change or more accuracy. The accuracy that we achieved in detecting noise and disturbances in the wave-forms is very high. The results did not vary significantly after the training was performed with 55 epochs. The maximum possible accuracy was achieved at 60 epochs, and the results are shown in Table I.

TABLE I. BEST ACHIEVED RESULTS

Epochs	Results		
	Train. Accuracy	*Loss*	*Vald. Accuracy*
60	95.2%	5.6%	86.7%

This deep learning-based power quality event detection approach is to be used with an electric relay or other power equipment wave-forms would greatly prevent even minuscule power quality events to be overlooked. Human error is significant in observation and detection, which is why this approach can be further built upon in order to replace human effort completely in this respect.

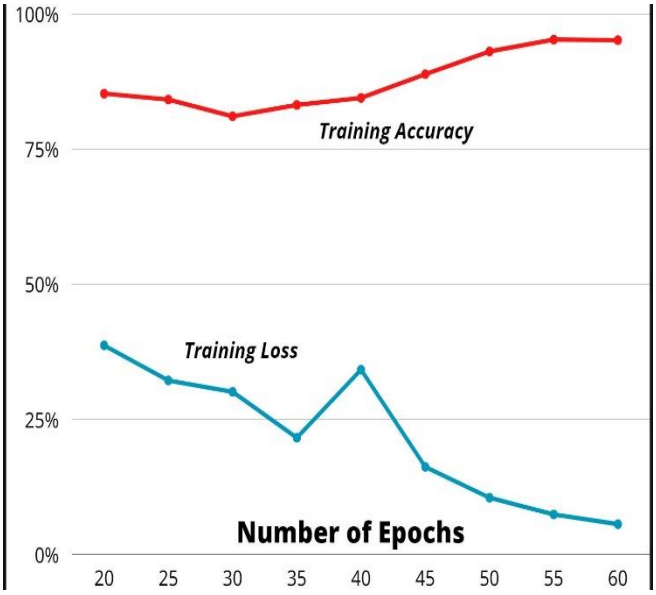


Fig. 8. Results achieved with each training performed.

VII. Conclusion

This paper presented an issue that is faced in power transmission and power quality field. Without good power quality in an electrical network, Load's electrical and mechanical equipment may overheat, malfunction, fail prematurely, require high maintenance, and in many cases they may not operate at all. In order to avoid this, many solutions are being provided in different perspective. We provided a deep learning image recognition based solution. The efficiency of the proposed approach proved to be good, which is further work will definitely be done in this direction, in order to not just minimize, but completely finish the issue at hand.

References

[1] H. M.S.C. Herath, Victor J. Gosbell, and Sarath Perera. Power quality (PQ) survey reporting: Discrete disturbance limits. IEEE Transactions on Power Delivery, 20(2 I):851–858, 2005.

[2] Kelvin Olikara. Power Quality Issues , Impacts , and Mitigation for Industrial Customers. Power-Wp002a-En-P, page 8, 2015.

[3] Ebrahim Balouji and Ozgul Salor. Classification of power quality events using deep learning on event images. 3rd International Conference on Pattern Analysis and Image Analysis, IPRIA 2017, (Ipria):216–221, 2017.

[4] Mohamed Aymane Ahajjam, Daniel Bonilla Licea, Mounir Ghogho, and Abdellatif Kobbane. Electric Power Quality Disturbances Classification based on Temporal-Spectral Images and Deep Convolutional Neural Networks. 2020 International Wireless Communications and Mobile Computing, IWCMC 2020, pages 1701–1706, 2020.

[5] Manish Kumar Saini and Rajiv Kapoor. Classification of power quality events - A review. International Journal of Electrical Power and Energy Systems, 43(1):11–19, 2012.

[6] Om Prakash Mahela, Abdul Gafoor Shaik, and Neeraj Gupta. A critical review of detection and classification of power quality events. Renewable and Sustainable Energy Reviews, 41:495–505, 2015.

[7] D Saxena, K Verma, and S Singh. Power quality event classification: an overview and key issues. International Journal of Engineering, Science and Technology, 2(3):186–199, 2010.

[8] Sruthi Reddy Chintakindi and D. V.S.S.Siva Sarma. Classification of Power Quality events using improved S-transform. 12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015, (5):1–5, 2016.

[9] Sambit Dash and Umamani Subudhi. Multiple power quality event detection and classification using modified S transform and WOA tuned SVM classifier.

[10] Bendong Zhao, Shanzhu Xiao, Huanzhang Lu, and Junliang Liu. Waveforms classification based on convolutional neural networks. Proceedings of 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2017, pages 162–165, 2017

[11] A. Usha Ruby, Prasannavenkatesan Theerthagiri, I. Jeena Jacob, and Y. Vamsidhar. Binary cross entropy with deep learning technique for image classification. International Journal of Advanced Trends in Computer Science and Engineering, 9(4):5393–5397, 2020

[12] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. arXiv, (NeurIPS), 2018.

[13] Raniah Zaheer and Humera Shaziya. A Study of the Optimization Algorithms in Deep Learning. Proceedings of the 3rd International Conference on Inventive Systems and Control, ICISC 2019, (August):536–539, 2019.

Improving Sentiment Analysis Based on Gated Recurrent Unit Model by Using Feature Selection

Mohammed Hussein Abdalla
Department of Computer Science
University of Raparin,
Rania, Iraq
muhamad_it@uor.edu.krd

Fatih Özyurt
Department of Software Engineering,
Firat University,
Elazig, Turkey
ozyurtfatih@gmail.com

***Abstract*— in recent years, sentiment analysis has received a great deal of research due to the dramatic growth in online social media use. For learning-based methods, many technically advanced strategies have been used to improve the performance of the forms. Feature selection is one of these strategies and has been studied by many researchers. However, one unresolved problem is choosing the right number of features to get the best sentiment analysis performance from learning-based methods. Therefore, we examine the relationship between the number of features selected and the sentiment analysis performance of learning-based methods, aiming to reduce the complexity time and increase accuracy in the experimental data. In this paper, we worked on a dataset consisting of tweets for six major US Airlines and analyses the impact in sentiment analysis of combined methods of Feature Selection based on chi-square correlation with the GRU network. Finally, GRU with feature selection method scored the highest performance in all metrics and achieved an accuracy of 96.38%.**

Keywords—deep learning, feature selection, chi-square, sentiment analysis, Gated Recurrent Unit Model

I. INTRODUCTION

Unstructured data forms a big part of the created data in today's world. From social media comments to customer feedback and browsing trails, and for processing and analyzing such texts, demand for appropriate text classification schemes is essential. Being a subfield of machine learning and involving multiple algorithms to function, deep learning allows for the self-training of software programs to carry out predefined tasks to expose neural networks to large datasets [1]. Social media has provided a new device for making and sharing ideas with others. Receiving public opinion about social events, marketing activities, and product priorities attracted the scientific community and marketing companies [2]. Today, if one wants to buy a product, he is no longer limited to family and friend opinions, because users' comments and discussions about different products are available on websites. However, finding and evaluating opinions is not an easy task due to the expansion and various opinions. Each site usually has many text comments that are not easy to decrypt and summarize such data, so automated sentiment analysis systems are required [2]. The acceptance of the generated content development by users on websites and social networks such as Twitter has led to an increase in social media's power to express opinions about services, products and events. Today, the number of people using these networks to make decisions is increasing [3].

Using sentiment analysis systems, we can determine what people think about different topics and their opinions. We can evaluate the reasons for the failure or success of various social issues from the users' perspective by analyzing these ideas. In this paper, the Sentiment Analysis System uses NLP techniques and a sentimental vocabulary network for the top 10 US-based airline carriers. In this system, in the pre-processing data phase, the required information is extracted from the comments by separating the words and sentences, labelling the sentence components and rooting the phrase. After completing the learning process, the algorithm can use the model learned from the training set to classify (sentiment analysis) the rest of the data (which is not labelled) and label each text to the correct class [2]. This study introduces deep learning methods with a feature selection algorithm in Sentiment Analysis. Figure 1 shows the General Sentiment Analysis System.

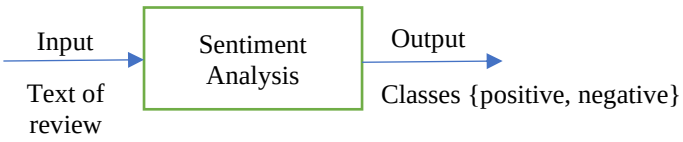


Figure.1. General Sentiment Analysis System

II. RELATED WORK

Twitter, is one of the foremost prevalent online social networks and a platform that can be used for microblogging, sharing views and opinions using various forms of expression like acronyms, abbreviations, emoticons. Grammar and spelling are not restricting users to communicate on the platform as such these features assist in categorizing sentiments in tweet bodies. In a study [4], Sharma S., Jain A. analyzed Twitter data through real-time data extraction. Preprocessing and feature extraction was applied to the text data, Chi-square test and principal component analysis (PCA), and different machine learning classifications (SVM, Naïve Bayes, Random Forest) Logistic Regression) were conducted against performance indicators. The study concludes that feature selection led to enhancing the accuracy of the backup vector classifier. Another study [5] works to lessen the problems related to the accuracy of sentiment analysis of aspect nature. Here, a feature selection method is proposed as principal component analysis (PCA) through which the best feature set of such research is specified. To enhance the accuracy of classification, irrelevant features are removed, and the redundant ones are lessened. Using Twitter's PCA of aspect-based, a new method of selecting features for the analysis in question is produced. "PCA is combined with the Sentiwordnet Glossary-based method, which is combined with the Support Vector Learning (SVM) framework for classification. Data set tests (HCTS), and benchmark (STS)

have an accuracy of 94.53% and 97.93%, respectively." When compared with other methods of this form, promising results are expected from the proposed approach to enhance this type of sentiment analysis. The study [6] attempted to classify text and sentiments in Twitter data by examining mean k-clustering and TF-IDF to weigh features depending on the simple Bayesian classifier and multiple Chi-squared feature selection algorithms. To show strength and weak points, this approach was compared to other variants of the Bayesian. After executing the experiments, the study concluded that the proposed method specifies more fantastic performance than too many others. To enhance the analysis of sentiments in Twitter data, a way of classification was proposed in [7] based on maximum entropy classification. They were using features such as n grammar, word clusters, and comment dictionaries. The result showed that applying various dictionaries of comment as attributes performance was increased compared to using only one even when extra cluster information and feature space were added.

III. Problem Statement

Sentiment analysis is meant to establishing specific strategies for language processing and text mining to differentiate textual data. The abundance of online datasets and sophistication of social media platforms led researchers to focus more on analyzing sentiment and its associated applications [8]. In recent years, several approaches to sentiment analysis have been designed and developed. Most of these approaches have been based on machine learning and deep learning algorithms. Although good results have been obtained with these two techniques, many literature references show that deep learning has better outcomes [1]. However, the deep learning algorithms' complex structure has increased computational time for model training in data. Examining previous research, we conclude that these models are usually very accurate in training data. In the test data, the accuracy of the model has decreased. These models face over fitting in sentiment analysis and text processing due to the high number of features. Also, there is an unbalanced data set problem in sentiment analysis. Unbalanced data is generally referred to as a dataset in which the number of samples representing a class is less than other samples in different classes. This situation becomes problematic in classification when a class, which is usually an absolute or minority class, is not shown in the dataset. In other words, the number of incorrect observations exceeds the correct observations in a class. This problem has been repeatedly observed in the analysis of sentiment analysis. Low data quality may affect classification performance due to class imbalances or incorrectly labelled items[9].

IV. METHODOLOGY

A. Dataset

The US Airline dataset was extracted from multiple sources for the top 10 US airline carriers, such as Twitter tweets and online reviews of Skytrax. It analyzed a total of 14640 tweets from 7700 users. Twitter data was scrapped as of February 2015 and contributors were asked to identify positive, negative and neutral tweets first, followed by a categorization of negative reasons (such as "late flight" or "rude service") [6]. The dataset performed by us on the 80-20 rule uses a train-test split where %80 of the data is used for training and is used for testing the remaining %20.

B. Text Pre-processing

Text pre-processing methods are essential because they provide the tools needed to convert text from natural language to machine-readable format. In this step, all letters in the text data are converted to lowercase or uppercase letters and numbers are converted to words. Numbers that are not related to text analysis and lead to meaningful data products can be removed from the text data. Also, punctuation marks are removed from accent marks, and the space between textual data, stop-words, scattered terms and specific words is removed. The steps and procedures for pre-processing text are examined [1]. After preparing text data, the next step is to normalize collected text data. The essential methods of normalization for text pre-processing are:

- Convert all letters in text data to lowercase letters or uppercase Letters.
- Convert numbers to words or remove numbers from text data.
- Delete punctuation, accent marks and diacritics.
- Clear Whitespaces of text data.
- Expanding abbreviations.
- Delete stop-words, sparse terms and particular words.
- Canonicalization of text data.

C. Feature Selection Based On Chi-square Correlation

Chi-square correlation-based feature selection is one of the filter-based feature selection methods. In this method, features that have a high correlation with each other are removed. This feature selection method measures the significant difference between the observed frequency and the two features' expected frequency. According to the two-variable data, the chi-square statistic is predicted based on the difference between what is seen in the data and what is expected in the absence of a relationship between variables. To fully appreciate the concept, two feature categories are studied [9] namely: independent (or predictor) and dependent (or response). The features that are heavily dependent on the reaction are being selected. However, when both are independent properties expected, and observed values are similar, they have a lesser Chi-Square value. Producing a significant value of the Chi-square is means having an error in the independence hypothesis. To make it more concise, any feature with high value can be used to train the model.

$$chi-2(t,c)=\sum_{t\in\{1\}}\ \sum_{C\in\{1\}}\ \frac{(N_t,C-E_t,C)^2}{E_t,C} \quad (1)$$

In this formula, we show the observed frequency with N, the expected frequency with E, period t and class C. The high value of chi-square indicates that the hypothesis of independence of two features is not correct. As the correlation between the independent and dependent variables increases, the chi-square index shows a weaker fitting. The chi-square value's sensitivity is to the sample size (this index is usually significant in the high number of samples). If there is a correlation between the independent category feature (predictor) and the dependent category feature (response), the probability of correct prediction of the class occurrence increases. As a result, the feature with the high value of chi-square is selected as the associated feature.

V. CLASSIFICATION TECHNIQUES

Classification in sentiment analysis means categorizing knowledge into various groups based on the equation to evaluate the text's sentiment. Our research used two techniques for binary classification (positive and negative) of the US Airline dataset, namely, the GRU network and the proposed model (GRU with chi-square).

D. GRU Network

The Gated Recurrent Unit network architecture was proposed by Chou et al. in 2014. This architecture is designed to address traditional recurrent neural networks' shortcomings, such as gradient vanishing problem and reduce overload in LSTM architecture. GRU is generally considered as a modified version of LSTM because both of these architectures have the same design and, in some cases, give excellent results in the same way. GRU uses concepts called update gateway and resets gateway. These gates are two vectors that are used to decide whether data is transmitted to the output or not. The unique point about these gates is that they can be trained to retain data about long time steps without changing over time (during different time steps). In Fig (6) the structure of the GRU network is described [10].

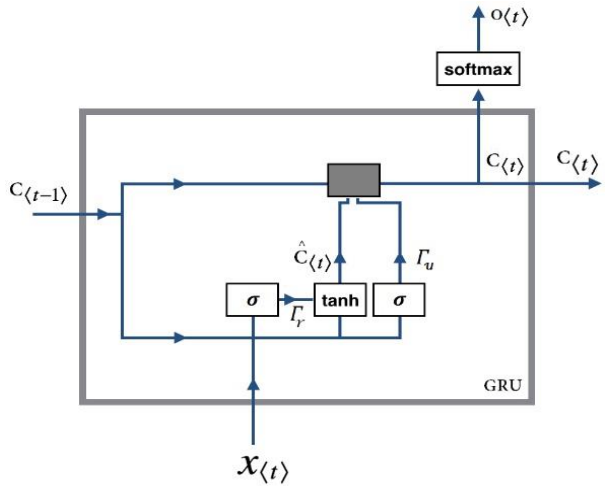


Figure. 2. The structure of the GRU network

step. With this new feature, the network can easily affect a state from several previous time steps in the next few time steps in long sequences. In other words, the network will keep elements from the distant past in its memory and exploit it.

A. Proposed Model (GRU with chi-square)

Gradients are used in neural networks to update parameters. Each parameter is changed according to the amount of effect it has on the network's final result. The vanishing problem indicates that the gradients' values gradually shrink as they move toward the beginning of the network, causing weight changes to be negligible. The training process slows down dramatically. This can cause the training process to be stopped in more severe cases, usually due to the network's considerable depth. Recurrent neural networks have a great depth because each time step is a layer in the feed-forward networks in this type of networks. The possibility of a gradient explosion and the gradient values gradually become so large that eventually the model fails and an overflow occurs in the calculations. In the LSTM network, we provide the network with the ability to retain multiple features by imposing restrictions on the freedom of parameters (by placing new gates) in the optimization process. But the ability to retain various features from several time steps back in the LSTM network is problematic. Therefore, I used the GRU network in this article. In this network, the forget gate and the input gate are combined and turned into a single update gate. Single and hidden states are also integrated, so the GRU model is more straightforward than the standard LSTM model. This added feature in the GRU has an advantage over the LSTM:

- First, each unit can remember a specific feature in the next long time steps' input flow. Any feature that is recognized important by the GRU update gate can be retained without being overwritten and lost.
- The second and perhaps most crucial point is that this new feature practically creates shortcuts that ignore several time steps and allow the production error to transfer in the post-propagation phase without disappearing very quickly. This reduces the problems associated with vanishing gradients.

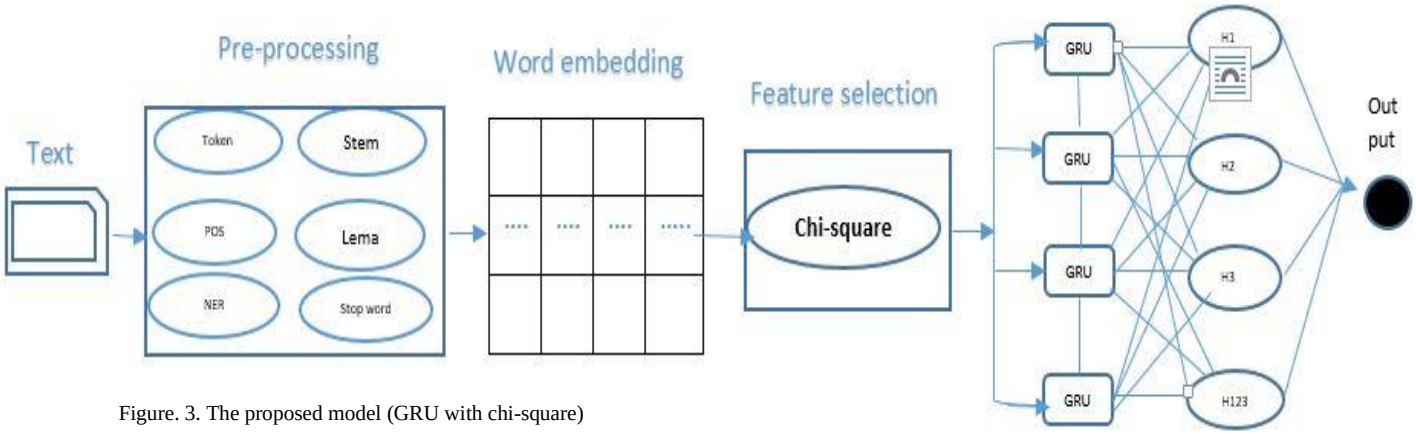


Figure. 3. The proposed model (GRU with chi-square)

Instead of LSTM model, we used the GRU model, because one of the main problems in the LSTM network is maintaining multiple features from several steps back in time is associated with problems. In the GRU recurrent neural network, the concept of update gates has been used to solve this problem. An update gate is a switch that specifies whether to use the previous state or input (or a combination of both) in a time

Despite the advantages of GRU network, GRU network training in training data is associated with high computational time. Also, reviewing previous research, we conclude that this model usually has good accuracy in training data, but the accuracy of the model decreases in experimental data [11]. To solve this problem, feature dimensions reduction method has been used. Feature dimensions reduction can be defined as a

combination of a search technique to suggest a subset of new features, along with an evaluation that rates a subset of different features. The simplest algorithm is to select a set of possible features that minimizes the error rate. This is a comprehensive search and is computationally difficult for every feature set except the smallest feature set. But the selection of the correlation-based feature of the independent category (predictor) and the dependent category (response) has a lower computational load. The chi-square correlation feature selection method is one of the correlation-based feature selection methods used in the proposed method. Fig 3 Shows the flowchart of the proposed method, which is a combination of chi-square correlation-based feature selection and GRU network and in the following, we will describe its steps [12, 13, 14, 15].

VI. RESULTS AND DISCUSSION

The results obtained through GRU and GRU with chi-square, both trained on the US Airline dataset, are discussed in this section, and the results are compared based on metrics such as precision, accuracy, recall, and f1 score. One of the key tasks in constructing any deep learning model is to evaluate its performance. By computing different metrics, the performance of each technique used in this work is calculated, and the ultimate aim behind working with other metrics is to understand how well a deep learning model can do on unseen data. The following metrics are used in this work: Accuracy is the proportion of the samples correctly analyzed to the total number of samples.

- Precision Aimed at measuring model exactness. It is calculated as the ratio of several correct reviews as predicted to the number of predicted reviews (TP+FP) in a way that a containing class (c) may be either negative or positive.

$$Precision = T_P/(T_P + F_P) \quad (2)$$

- Recall: : It is measured as the ratio of TP to the number of actual reviews (TP+FN) and having a negative or positive class. Through that, the model completeness is measured.

$$Recall = T_P/(T_P + F_N) \quad (3)$$

- Accuracy: This is aimed at the measurement of model correctness. Calculating accuracy is conducted by taking the ratio of correctly predicted reviews to total review under investigation.

$$Accuracy = (T_P + T_N)/(T_P + T_N + F_P + F_N) \quad (4)$$

- F1 score: Being primarily utilized for optimizing the model, it is an indication of the average recall and precision. The score can be used for optimizing recall or precision.

$$F_1 score = 2 * (Precision * Recall)/(Precision + Recall) \quad (5)$$

In the above equations, TP is the true positive and predicted correctly. FP is the false positive and predicted incorrectly, TN is the true-negative and predicted correctly. FN is the false-negative and predicted incorrectly. The experimented dataset split into training, validation, and testing set. The training set was given 80% of the dataset, while the testing was given 20% of the dataset. The following arrangements were derived from the outcomes of the proposed models. At 256 nodes, the number of nodes in the GRU was implemented. The networks began training from 8 to 1024, which increased by 2n in each experiment, where n is the number of nodes. To prevent the networks from remembering the training set that is useful for controlling the over fitting problem, each of the models is configured with a dropout layer. An Adam optimizer was used to compile the models, also trained by the MSE loss with the batch size of 100. The values of the hyper-parameters are chosen after performing many tests.---- Based on the presented results (Table 1), the Accuracy, Precision, Recall, and F1 score of the prediction of the proposed method (GRU+ chi-square) in the US Airline data set was 96.21%, 96.38%, 95.15% and 96.12%, respectively. Finally, the comparison of the results with the GRU network also shows that feature selection has a positive effect on increasing this network's accuracy. The GRU network's prediction accuracy based on the results obtained in Table (1) was76.54%. Which by chi-square feature selection approach reached 96.21%. We also had over-fitting due to a large number of features in the US Airline dataset. The chi-square method has solved this problem. Normally there are 23 features, but I chose 20 features, and the accuracy of the model has been dramatically increased.

Techniques	Accuracy	Precision	Recall	F1 score
GRU	76.54	76.55	76.12	76.30
GRU with chi-square2	**96.21**	**96.38**	**95.15**	**96.12**

Table 1: Results and performance comparisons of the models

Also, by examining the accuracy of the model in each epoch in Figure (4), we find that the proposed model's accuracy does not improve after the epoch 9 in the US Airline dataset. Specifically, the optimal values of the epochs for the US Airline database are 10, too.

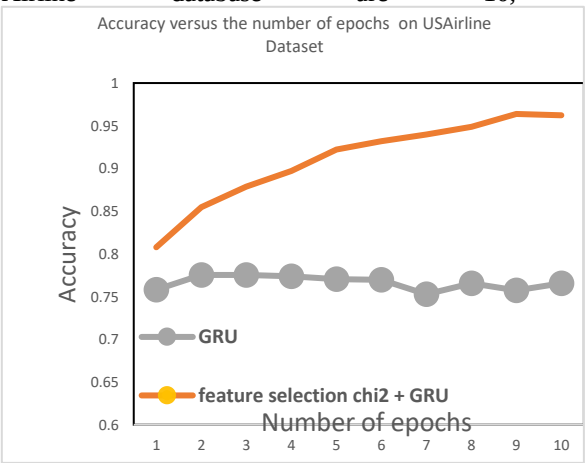


Figure 4: Best model statistics of correct and incorrect predictions

VII. CONCLUSION

In this paper, we present a combination of feature selection methods based on chi-square correlation with the GRU network, which increases the GRU network's computational accuracy. In explaining this result, it can be said that the GRU network usually has good accuracy in training data, but the accuracy of the model decreases in testing data. This problem has been solved with selecting the chi-square feature by choosing the features that are highly dependent on the response, and so increases the accuracy of the GRU network.

This method increases the probability of correctly predicting the class's occurrence by selecting the independent category (predictor) feature that has a high correlation with the dependent category (response) feature. However, the method presented in this framework has unavoidable problems such as generalizing the results based on other datasets and searching across the feature subsets required in the subset construction step, needs its computational complexity. Therefore, the chi-square feature selection method requires further studies, and its comparison with other feature selection methods is necessary. Also, we used the connection of the forget gate and the input gate to prevent the production of additional information.

REFERENCES

[1] Abdalla, G., & Özyurt, F. (2020). Sentiment Analysis of Fast Food Companies With Deep Learning Models. The Computer Journal.

[2] [S.Akyol, B.Alatas, Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization, Physica A: Statistical Mechanics and its Applications,Volume 540, (2020), 123094.

[3] M. Yasen, S. Tedmori, Movies Reviews Sentiment Analysis and Classification, 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT).

[4] Sharma, S., & Jain, A. (2020). An Empirical Evaluation of Correlation Based Feature Selection for Tweet Sentiment Classification. In Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies (pp. 199-208). Springer, Singapore.

[5] Zainuddin, N., Selamat, A., & Ibrahim, R. (2016, August). Twitter feature selection and classification using support vector machine for aspect-based sentiment analysis. In International conference on industrial, engineering and other applications of applied intelligent systems (pp. 269-279). Springer, Cham.

[6] Feature Selection Approach for Twitter Sentiment Analysis and Text Classification Based on Chi-Square and Naïve Bayes

[7] Griol, D., Kanagal-Balakrishna, C., & Callejas, Z. (2020). Feature Set Ensembles for Sentiment Analysis of Tweets. In Advances in Data Science: Methodologies and Applications (pp. 189-208). Springer, Cham.

[8] Özyurt, F. (2019). Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. The Journal of Supercomputing, 1-19.

[9] Özyurt, F. (2020). A fused CNN model for WBC detection with MRMR feature selection and extreme learning machine. Soft Computing, 24(11), 8163-8172.

[10] Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2020). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems, 115, 279-294.

[11] Cheng, Y., Yao, L., Xiang, G., Zhang, G., Tang, T., & Zhong, L. (2020). Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism. IEEE Access, 8, 134964-134975.

[12] Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. (2020). Fake news stance detection using deep learning architecture (cnn-lstm). IEEE Access, 8, 156695-156706.

[13] Rane, A., & Kumar, A. (2018, July). Sentiment classification system of Twitter data for US airline service analysis. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 769-773). IEEE.

[14] Wazery, Y. M., Mohammed, H. S., & Houssein, E. H. (2018, December). Twitter sentiment analysis using deep neural network. In 2018 14th International Computer Engineering Conference (ICENCO) (pp. 177-182). IEEE.

[15] Abdulkhaliq, S. S., & Darwesh, A. M. (2020). Sentiment Analysis Using Hybrid Feature Selection Techniques. UHD Journal of Science and Technology, 4(1), 29-40.

Real Time Emotion Recognition Using Convolutional Neural Network

Bah Abdoulaye

MSc in Computer Engineering,
Istanbul Sabahattin Zaim University

***Abstract*—Due to the evolution of technology, Human Computer Interaction has become a crucial field of interest. Over the past decade various research has been accomplished in this field and still are ongoing. Facial expressions play an important role in non-verbal communication and also in Human Computer Interaction. Emotions on a human face say a lot about our thought process and give a hint on what is going on inside our brain. In this Paper a method of Facial Emotion Recognition is implemented (FER) using Convolutional Neural Networks (CNN) which can be used for the classification of facial emotion expressions in real time. This work and experiment can be used for emotion analysis while people watch movie trailers or video lectures or other purposes.**

***Index Terms*—Convolutional Neural Network, Transfer Learning, Facial Expression Recognition, Neural Network**

I. Introduction

Human Computer Interaction has gained consideration due to the high usage rate of technology. Therefore, FER using machines has become a trend topic of interest for experts and researchers over the last decade.

Furthermore, an application or device able to detect and classify human expressions in real time is required. This emotion classification can be used later in psychology to understand the human mind or also help devices to have a better idea about the user needs or requirements.

FER is not limited to that, it can be used in association with other systems in order to furnish safety. For instance, ATMs could be set up such that they won't dispense money when the user is scared. In the gaming industry, emotion-aware games can be developed which could vary the difficulty of a level depending on the player's emotions. By judging their expressions during different points of the game, a general understanding of the game's Software for cameras can use emotion recognition to take photos whenever a user smiles[11].

This paper aims to discuss about a method developed to implement a FER system using CNN. The system will be able to do the classification of seven human face expressions such as happiness, neutral, fear, disgust, anger, surprise and sadness.

Furthermore, with the help of a webcam, the so-called model will be used for the categorization or classification of human faces in real time. This experiment can then be used later to analyze user expressions in order to make easier the understanding of user requirements and needs.

The rest of this paper is as follows: in Section 2, there is an introduction about the Related Works. Section 3 about the Methodology. In Section 4 we will show our experiment result, and the experiment settings. Then we finish with our conclusion in Section 5 and Future Works in Section 6.

II. Related Works

In the last previous years FER has been and still a trend topic for researchers because of its various usage in robotic, Computer Vision, and especially Human Computer Interaction[1][2][3]. Paul Ekman, 1994 has presented six universal expressions. He has described the positioning of faces, and the muscular movements required to create these expressions in his study (Ekman, 1997). This study has proved to be very useful in the research of FER.

The Facial Action Coding System (FACS), developed by Swedish anatomist Carl-Herman Hjortsjö, is a coding system used to taxonomize human facial movements based on their appearance on the face. This system, which was later adopted by Ekman & Friesen (2003), is also a useful method of classifying human expressions. FER systems were mostly implemented using the FACS in the past.

However, recently there has been a trend to implement FER using classification algorithms such as SVM, neural networks, and the Fisherface algorithm (Alshamsi, Kepuska & Meng, 2017; Fathallah, Abdi & Douik, 2017; Lyons, Budynek & Akamatsu, 1999). In order to promote researchers to improve the FER2013 which was designed by Goodfellow et al. Kaggle organized a competition. Using CNN with some image transformation was the technique which led the top three teams to success [7]. The winner, Yichuan Tang, achieved a 71.2% accuracy by using the primal objective of an SVM as the loss function for training and additionally used the L2-SVM loss function [8].

A paper presented by Pramerdorfer and Kampel [9] describes the approaches taken by six current state-of-the art papers and ensembles their network to achieve 75.2% test accuracy on FER2013, which is, to our knowledge, the highest reported in any published journal paper.

Among the six papers, Zhang et al. achieved the highest accuracy of 75.1% by employing auxiliary data and additional features: a vector of HoG features was computed from face patches and processed by the first FC layer of the CNN (early fusion).

They also employed facial landmark registration, suggesting its benefits even in challenging conditions (facial landmark extraction is inaccurate for about 15% of images in the FER dataset) [4]. The paper with the second highest accuracy by Kim et al. utilized face registration, data augmentation, additional features, and ensembling [5].

There are several challenges with implementing the FER system. Most datasets consist of images of posed people with a certain expression. This is the first challenge, as real time applications require a model with expressions which are not posed or directed. The second challenge is that the labels in the datasets are broadly classified, which means that in real time there might be some expressions which the system might be able to classify correctly.

There are many FER systems, such as Affectiva, and Microsoft's Emotion API (McDuff et al., 2016; Linn, 2015). These systems have become very popular in applications where FER is required.

III. Methodology

A. The Dataset

We used FER2013 dataset from the Kaggle competition (Goodfellow et al., 2013)to implement the FER system. The images in this dataset are black and white images, having 48x48 size grayscale images. The dataset contains 38,887 images varying in lighting, scale, and viewpoint.

The dataset file is in a csv format containing columns and such as "Label", "Number of images" and "Emotion"[10].Moreover the so-called dataset has 28,710 images as training set, as for the public and final test, 3587, 3590 respectively[10]. We can have a look at some examples of the dataset in Fig. 1, and the explanation of the dataset in Table 1.

An image of each class is shown.

Fig. 1 FER2013 dataset sample images

Table 1. FER2013 dataset description [6]

Label	Number of images	Emotion
0	4593	Angry
1	547	Disgust
2	5121	Fear
3	8989	Happy
4	6077	Sad
5	4002	Surprise
6	6198	Neutral

The Convolutional Neural Network was implemented with the help of Keras and Tensorflow. This Network can be improved with the help of a stronger CPU [10].

B. Convolutional Neural Networks

A Convolutional neural network is a neural network comprised of convolution layers which does computational heavy lifting by performing convolution. Convolution is a mathematical operation on two functions to produce a third function. It is to be noted that the image is not represented as pixels, but as numbers representing the pixel value. In terms of what the computer sees, there will simply just be a matrix of numbers. The convolution operation takes place on these numbers. We utilize both fully-connected layers as well as convolutional layers. In a fully-connected layer, every node is connected to every other neuron. They are the layers used in standard feedforward neural networks.

Unlike the fullyconnected layers, convolutional layers are not connected to every neuron. Connections are made across localized regions. A sliding "window" is moved across the image. The size of this window is known as the kernel or the filter. They help recognise patterns in the data. For each filter, there are two main properties to consider - padding and stride. Stride represents the step of the convolution operation, that is, the number of pixels the window moves across. Padding is the addition of null pixels to increase the size of an image. Null pixels here refers to pixels with value of 0. If we have a 5x5 image and a window with a 3x3 filter, a stride of 1 and no padding, the output of the convolutional layer will be a 3x3 image.

This condensation of a feature map is known as pooling. In this case, "max pooling" is utilized. Here, the maximum value is taken from each sliding window and is placed in the output matrix. Convolution is very effective in image recognition and classification compared to a feed-forward neural network. This is because convolution allows to reduce the number of parameters in a network and take advantage of spatial locality. Further, convolutional neural networks introduce the concept of pooling to reduce the number of parameters by downsampling. Applications of Convolutional

neural networks include image recognition, self-driving cars and robotics. CNN is popularly used with videos, 2D images, spectrograms, Synthetic Aperture Radars.[11]

C. Process of FER

The implementation of FER is divided into three steps. In the first step which is the preprocessing, we are preparing the dataset such that it works on a generalized algorithm and also works efficiently. In the second step, which is the face detection, we are detecting face from the images being captured in real time.
And our last step is the emotion classification, we are classifying the input image into our seven classes by the implementation of CNN. A description of the steps is shown in fig. 2.

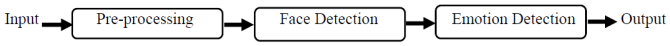


Fig.2 Implementation phases

Pre-processing:
To get rid of the noise, the variation in illumination, the color and size, that the input image may contain in order to get faster and accurate results on the algorithm, some preprocessing implementations were done on the images. The so-called used techniques were first of all Normalization, then Grayscaling, and lastly image resizing.

Normalization was done to remove the illumination variations in order to get better face image. Grayscaling was done to convert the colored image input into an image whose pixel value depends on the intensity of light on the image. Grayscaling is done as colored images are difficult to process by an algorithm[6]. Image resizing was done to remove the useless or unnecessary image parts. As a result it increases the computational speed and also diminish the memory.

Face detection:
Lets start with to the primary step of FER which is the face detection. Haar cascade are kind of classifiers able to detect an object in a video or an image for which they have been trained. They are trained over a set of positive and negative facial images[6].

The reason behind choosing Haar cascade is because of its fame in object detection and also its high accuracy results.

Haar features detect three dark regions on the face [6], the eyebrows for example.

The computer is trained to detect two dark regions on the face and using fast pixel calculation their location is decided. The unnecessary background data is successfully removed by Haar cascade from the image and detect the facial region from the image.

OpenCV was used for the implementation of Haar cascade classifiers for the face detection step. This method was originally proposed by Papageorgiou et al, using rectangular features which are shown in figure 3 (Mohan, Papageorgiou & Poggio, 2001; Papageorgiou, Oren & Poggio, 1998).

Fig.3. Haar features (Shan, Guo, You, Lu, & Bie, 2017)

Emotion Classification:
Here our model will be classifying the image into one of the seven categories – Sadness, Happiness, Anger, Disgust, Surprise, Fear, and Neutral. A category of neural network known as CNN was used for the training because of its high accuracy in image processing. First we split the dataset into training an testing datasets, and then trained on the training set. Feature extraction was done after feeding it into CNN.
The following steps were used for the emotion classification phase:

- *Data Splitting:* According to the usage label in the FER2013 dataset, the data was split into 3 categories such as Training, PublicTest and PrivateTest. For the generation of the model Training and PublicTest were used, and for the model evaluation PrivateTest.

- *Training and Model Generation:* The architecture of our neural network was as follows.

  - *Convolutional layer:* here a learnable filter which is randomly instantiated is convolved or slid over the input. The operation implements the dot product between the filter and each local region of the input. As output we have a 3D volume of multiple filters, which we can call feature map also.

  - *Max Pooling:* To diminish the spatial size of the input layer we use the pooling layer, to reduce the size of the computational cost and input.

  - *Fully Connected Layer:* In the fully connected layer, there is connection between each neuron from the previous layer and the output neurons. The size of the final output layer is equal to the number of classes in which the input image is to be classified.

  - *Activation function:* The activation function is used for only one reason which is to reduce the overfitting. The ReLu activation function was used in the CNN, because its gradient is equal to 1 always. That is to say that most of the error is passed back while the implementation of back propagation.

- *Softmax:* Softmax is a function which takes a vector of N real numbers and normalizes it between (0,1) values.

- *Batch Normalization:* To speed the training process we use batch normalization; another benefit of batch normalization is that it applies a transformation that conserve the mean activation close to 0 and the activation standard deviation close to 1.

D. Model evaluation

The obtained model during the Training phase was later evaluated on the validation test, that contains 3589 images.

E. Model usage to classify real time images

By using the concept of transfer learning, we can detect emotion in images captured in real time. The generated model from the training phase is composed of pretrained values and weights, that can be used to implement a new facial expression detection problem. Since the so-called model already contains weights, FER becomes faster for real time images. In fig. 4 the CNN architecture is shown.

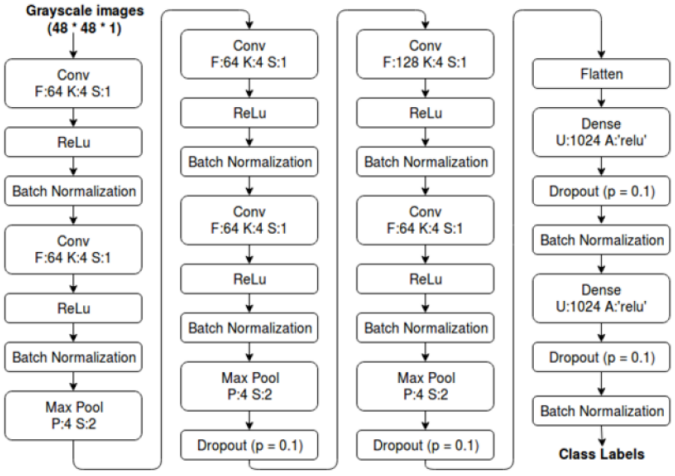


Fig. 4 CNN Architecture [6]

IV. Experimental Results

The results were obtained by the implementation of CNN algorithm. It was remarked that the training and testing set was decreasing after each epoch. The batch size was 256, which did not change during all the experiment.
To get better results the following changes were done in the neural network:

- *Epoch number:* We observed that the accuracy of the model increased when we increased the number of epochs. Furthermore, a high number of epochs causes overfitting. In the end we concluded that eight epochs resulted in minimum overfitting and high accuracy.
- *Layer numbers:* The neural network is composed of three hidden layer and one fully connected layer. Totally six convolutional layers were built using ReLu activation function.
- *Filters:* We observed some changes in the accuracy while we changed the number of filters. For the first two layers the number of filters applied on the network was 64, and for the third layer it was left as 128 constantly.

A. Accuracy:

The final state-of-the-art model gave a test accuracy of 60.12% and a training accuracy of 79.89% as we can see in the table. The used architecture was able to classify correctly 22936 out of 28709 images from the train set and 2158 out of 3589 images from the test set. In table 2 you can see the results of some experiments done on CNN.

Table 2. Accuracy from three experiments[6]

Experiment	Training Accuracy	Test Accuracy	Validation Accuracy
Experiment 1	63.22	56.56	89.01
Experiment 2	68.37	58.03	89.61
Experiment 3	79.89	60.12	89.78

Table 3. Shows a brief comparison of the proposed system with other related works[6].

Table 3. Comparison with related works [6]

Related work	Algorithm	Dataset	Results
Kumar, Kumar, & Sanyal, 2016	CNN	FERC-2013	Around 90%
Amin, Chase & Sinha, 2017	CNN	FER-2013	60.37
Shan, Guo, You, Lu & Bie, 2017	KNN	JAFFE, CK+	65.11, 77.27
Kulkrni, Bagal, 2015	Gabor, Log Gabor	FACES	82%, 87%
Minaee, & Abdolrashidi, 2019	Attentional CNN	FER2013	70.02%
Proposed	CNN	FER2013	89.78%

B. Loss and accuracy over time:

We remark that the loss and the accuracy decrease after each epoch. The training versus testing cure for the accuracy remains the same over the first five epochs. The training and test accuracy along with the training and validation loss obtained from our dataset FER2013 using CNN is given in Table.3.

Table 3. Accuracy per epoch [6]

Epoch	Training Accuracy	Validation Accuracy
1	29.10	43.33
2	47.81	50.65
3	55.60	56.90
4	60.13	57.65
5	64.07	57.95
6	67.00	59.63
7	69.95	59.01
8	72.88	60.13

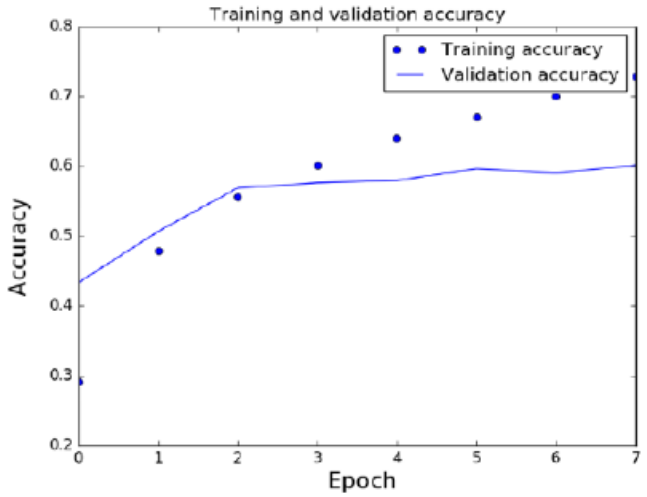


Fig.5. Graph of training and validation accuracy per epoch[6]

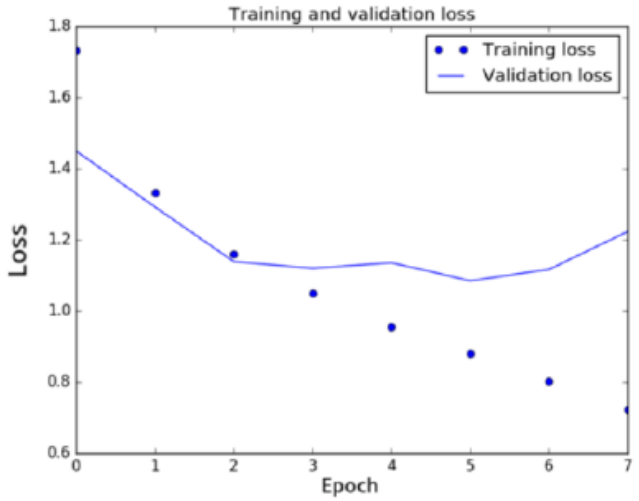


Fig.6. Graph of training and validation loss per epoch[6]

C. Confusion Matrix

The confusion matrix that was created for the test data is discussed in Figure 7. The dark squares around the diagonal indicates that the test data is conducting the classification correctly. Whereas the number of right classifications for disgust and fear is poor, as we observed.

The numbers on each side of the diagonal indicate the number of images that have been correctly identified. We can assume that the algorithm has been efficient and obtained state-of-the-art results as these numbers are lower relative to the numbers on the diagonal.

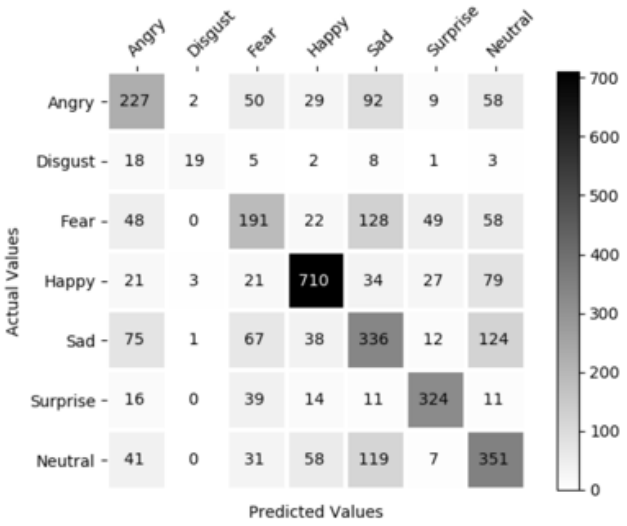


Fig. 7. Confusion Matrix represented as a heatmap [6].

V. Conclusion

In this paper we discussed about an approach for FER using CNN. We created a CNN model on the FER2013 dataset and experiments with the architecture were done to achieve a test accuracy of 0.6012 and a validation accuracy of 0.8978. This state-of-the-art model were used to classify emotions of users in real time with a webcam. The webcam captures images and uses them to classify the emotions.

VI. Future Works

Our remark is that to improve this model, we need more better images with more quality and more specific, we do not forget to mention that also the webcam and the background has a huge impact on the real time classification. Another opinon is that OpenCV should be improved for best image capturation. Also there were less numbers of images compared to the overall such as disgust, so this can affect the accuracy also. So more pictures should be used approximately for all emotions.

REFERENCES

[1] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), 2001.

[2] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In Proceedings of the IEEE Conference on Automatic Facial and Gesture Recognition, 2006.

[3] M. Pantic and J.M. Rothkrantz. Facial action recognition for facial expression analysis from static face images. IEEE Transactions on Systems, Man and Cybernetics, 34(3), 2004

[4] Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, "Learning Social Relation Traits from Face Images," in Proc. IEEE Int. Conference on Computer Vision (ICCV), 2015, pp. 3631–3639.

[5] B.-K. Kim, S.-Y. Dong, J. Roh, G. Kim, and S.-Y. Lee, "Fusing Aligned and Non-Aligned Face Information for Automatic Affect Recognition in the Wild: A Deep Learning Approach," in IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 48–57.

[6] Real Time Facial Expression Recognition Using Deep Learning, Isha Talegaonkar, Kalyani Joshi, Shreya Valunj, Rucha Kohok, Anagha Kulkarni 2019.

[7] Facial Expression Recognition, Amil Khanzada, Charles Bai, Ferhat Turker Celepcikay.

[8] Y. Tang, "Deep Learning using Support Vector Machines," in International Conference on Machine Learning (ICML) Workshops, 2013.

[9] Pramerdorfer, C., Kampel, M.: Facial expression recognition using convolutional neural networks: state of the art. Preprint arXiv:1612.02903v1, 2016.

[10] Human Emotion Recognition using Convolutional Neural Network in Real Time by Rohit Pathar,Abhishek Adivarekar,Arti Mishra,Anushree Deshmukh.

[11] Facial Emotion Recognition using Convolutional Neural Networks by Akash Saravanan, Gurudutt Perichetla, and Dr. K.S.Gayathri

DeCoV-CNN: A Simple CNN Model for Detection of COVID-19 Using Chest X-rays

Suba S
Centre for Computational Natural Sciences and Bioinformatics
International Institute of Information Technology
Hyderabad, India
suba.s@research.iiit.ac.in

Nita Parekh
Centre for Computational Natural Sciences and Bioinformatics
International Institute of Information Technology
Hyderabad, India
nita@iiit.ac.in

***Abstract*— Machine learning and artificial intelligence methods are well established in image analysis, making them suitable for the analysis of chest X-RAY (CXR) images. Earlier work on Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS) have confirmed their applicability in the diagnosis of pulmonary diseases. This has led to much recent attention in the analysis of CXRs using deep learning architectures for the detection of COVID-19 in a clinical setting. Here a simple Convolutional Neural Network (CNN) model is proposed for classifying CXR images into one of the three classes, *viz.*, Normal, Covid Pneumonia (CP) and Non-Covid Pneumonia (NCP). Here we show that performance of a simple CNN architecture is comparable to that of deep architecture like VGG-16 in classifying a CXR image as CP class. Dataset used for training is collated from multiple open-source repositories and, to the best of our knowledge, is the largest among all publicly available COVID-19 datasets. In this study impact of various factors like input image dimensions and batch size on model's performance is discussed. With portable chest radiography now being commonly used for early disease detection and follow up of lung abnormalities, there is a clear scope of the simple CNN model for assisting health experts in triaging patients in the current pandemic situation.**

Keywords*— *Convolutional Neural Network, consolidation, ground glass opacities, Residual connections

I. INTRODUCTION

The Corona Virus disease (Covid-19), first observed in Wuhan, China, in December 2019, is an infection of the lungs caused by a newly discovered virus named Severe Acute Respiratory Syndrome Corona Virus 2 (SARS-CoV-2). In this one year, more than 91.7 million cases have been reported globally with over 1.96 million deaths. The virus travels into the lungs through bronchial tubes causing inflammation of mucous membranes in lungs resulting in shortness of breath in patients. Main symptoms of the patients infected with the virus are fever and cough. Though in majority of individuals it manifests as mild to moderate respiratory illness, people with medical history such as cardiovascular diseases, diabetes, other respiratory ailments, cancer, etc. are susceptible to serious illness and death. Vaccinating the whole population is a humongous task, and early detection and isolation of the infected people are still the best ways in mitigating the situation.

Reverse Transcriptase-Polymerase Chain Reaction (RT-PCR) is the most sensitive test available to detect the virus. However, false negative rate of RT-PCR in Covid-19 detection is quite high; highest within first five days of exposure (~ 67%) and lowest on day 8 after exposure (21%). That is, in the best-case scenario, one out of five infected individual will have a negative test result. Further, viral load pkinetics of SARS-CoV2 differs from patient-to-patient [1]. It is also time consuming and laborious method and a faster preliminary screening method is desirable. Chest radiography (X-rays) or Computer Tomography (CT) imaging have been proposed for identifying COVID-like infections in the lungs. These alternative methods can be used as a preliminary tool before RT-PCR or along with RT-PCR for prioritizing admission of patients in hospitals/ICUs. Chest radiography image analysis requires the expertise of radiologists which may create a bottleneck in decision-making process. Computer aided systems using Machine Learning (ML) methods can assist the radiologists in interpreting CXR images and aid in faster triaging of patients. Convolutional Neural Network (CNN) is one of the popular architectures of Deep Learning models applied in image analysis. Various CNN architectures, *viz.*, Resnet, DenseNet, Inception, etc. have been proposed for diagnostic and prognostic analysis of CXR and CT scans of COVID-19 patients and have been shown to exhibit performance comparable to human experts [15]. In this paper we propose a multi-class classifier based on a simple convolution neural network (CNN) to classify new Covid-19 images from other pneumonia diseases and normal individuals with high accuracy and recall. Its performance with deep learning method, e.g., VGG-16, is discussed.

A. Abnormalities in the lungs

Chest X-Rays and CT scans are seen to exhibit characteristic abnormalities in the lungs of COVID-19 patients such as multiple regions of Ground Glass Opacities (GGO), which may sometimes be associated with Consolidations mostly in the middle and lower lobes of lungs. The abnormalities could be bilateral and are mostly seen in the peripheral areas of lungs. Like with any pneumonia, COVID-19 also causes the lung density to increase, seen as whiteness in the lungs in chest radiography images. Based on disease severity, the lung markings in the images exhibit different patterns; when partially obscured it is called a ground glass pattern and when completely obscured by the whiteness, it is called consolidation [2]. Progression of the disease is evident from the images in Fig.1; (a) normal, (b) showing GGO on first day and consolidations in (c) on tenth day for the same patient. Need for expert radiologists can become a bottle neck in triaging patients in pandemic and computer-assisted image reading can accelerate the decision-making process. Even though CT scans give more detailed features, most radiology societies around the world do not recommend CT scans for diagnosis of COVID-19 unless a patient is critically ill. Moreover, X-Ray machines are portable, cheaper, and faster compared to CT scans, and thus

more practical in handling the overload in pandemic situations.

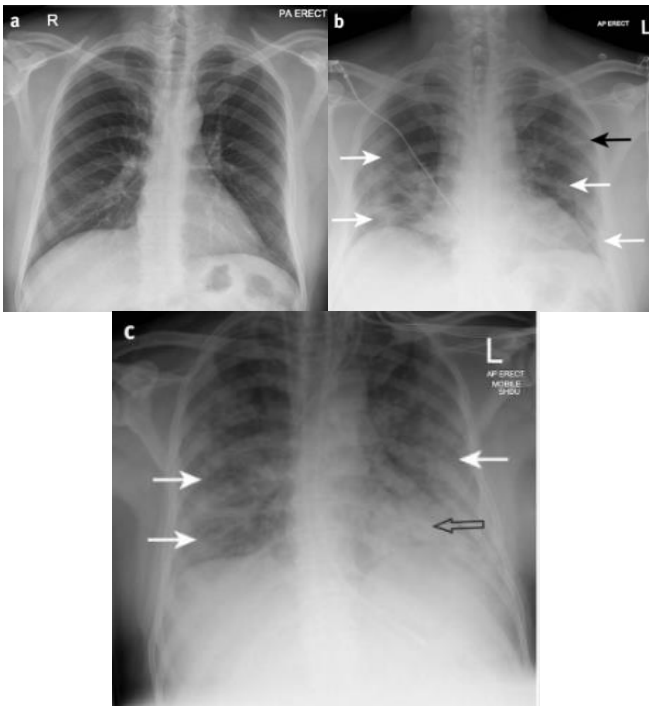


Fig. 1. (a) Normal Posterior-Anterior (PA) chest radiograph of a patient taken 12 months before COVID-19 infection (b) AP radiograph of the same patient taken when infected with COVID-19 taken on first day showing Ground-glass opacities (GGO) in the periphery of both lungs in the mid and lower zones, (c) Dense Consolidations in the image, taken on 10th day shows progression to severe COVID-19 Pneumonia. (Reproduced from [2])

II. Related Work

Various deep learning architectures have been proposed for classifying chest X-ray images into Normal, COVID-19 and Pneumonia due to various other causes such as bacterial, fungal, etc. These studies have either used pre-defined architectures such as ResNet, VGG, Exception, UNet, etc. pre-trained on ImageNet database [3] or have synthesized very complex models on these deep layered architectures. For example, CovidNet [4] uses a tailor-made design pattern based on ResNet architecture that comprises projection-expansion-projection-extension (PEPX) pattern of multiple Conv layers generated using a Generator-Inquisitor pair to obtain an optimal model. CoroNet based on Xception architecture consists of 71 layers, using depth wise separable convolution layers along with residual connections for classifying X-ray images into four classes, Normal, Pneumonia bacterial, Pneumonia Viral and COVID-19 [5]. Another DL model uses ConvLSTM and CNN for binary classification of COVID-19 and non-COVID-19 cases [6]. It has the capacity to encode spatiotemporal information in its memory cell and replaces fully connected layers in LSTMs to overcome redundancy in spatial data. VGG-16 architecture has been used in the model by [7]. In the first step it identifies if a CXR indicates presence of pneumonia and in the next step classifies Pneumonia samples into COVID or non-COVID pneumonia, a binary classification. CoroNet model [8], comprises two modules for three-class classification: a Task Based Feature Extraction Network module (TFEN) and COVID-19 Identification Network module (CIN). TFEN is a Feature Pyramid based AutoEncoder (FPAE) network with seven layers of convolutional encoder and decoder blocks, while the CIN is a pre-trained ResNet-18 network. Multiple pretrained models such as VGG-16 [9], VGG-19 [9], Inception V3 [10], etc. have been used and their performances compared with a customized CNN in [11]. Here the models are iteratively pruned by removing 2% of neurons from the convolutional layers with zero neuron activations. The best set of pruned models are considered for constructing an ensemble and various strategies used to combine their predictions for three-class classification. In [15], a review of deep learning approaches for classification of CXR and CT images to detect COVID-19 is discussed in detail.

These studies suffer from one common problem of data imbalance due to limited publicly available COVID-19 chest X-ray images. Various approaches have been used to address this problem. For example, using data augmentation methods which include operations such as rotation, horizontal flips, intensity shifts, zooming, etc. are applied to COVID-19 X-rays to handle class imbalance problem in some studies [4], [7]. In [6], image augmentation techniques using Generative Adversarial Network (GAN) was proposed to increase the number of COVID-19 images along with random under-sampling done on majority classes in [5]. Further, majority of the proposed models were pre-trained on ImageNet database for better initialization of weights [4], [5], [7], [8], [11].

III. Data

In this study, we have constructed Chest X-Ray dataset by collating six different publicly available data repositories., the links are summarized in Table I. Link to first five datasets is obtained from https://github.com/lindawangg/COVID-Net/blob/master/docs/COVIDx.md and the sixth dataset is provided by a cardiothoracic radiologist from Spain in his twitter handle and contains 132 images from 50 COVID-19 patients. COVID-19 images are collected from all the links except the NIHCC resource from which only normal and pneumonia images are obtained. The constructed dataset consists of a total of 14,308 images across 14,092 cases comprising 8066 normal (8066 cases), 5575 Pneumonia (5549 cases) and 749 COVID-19 images (477 cases). All the images were resized to 224×224 pixels. To the best of our knowledge, this dataset has the largest number of 749 publicly available Covid-19 images.

TABLE I. PUBLICLY AVAILABLE REPOSITORIES USED FOR CONSTRUCTING THE DATASET USED IN THIS STUDY.

	Links for Datasets	Reference
1.	https://github.com/ieee8023/covid-chestxray-dataset	[12]
2.	https://github.com/agchung/Figure1-COVID-chestxray-dataset	[4]
3.	https://github.com/agchung/Actualmed-COVID-chestxray-dataset	[4]
4.	https://www.kaggle.com/tawsifurrahman/covid19-radiography-database	
5.	https://www.kaggle.com/c/rsna-pneumonia-detection-challenge (which came from https://nihcc.app.box.com/v/ChestXray-NIHCC)	[13]
6.	https://threadreaderapp.com/thread/1243928581983670272.html	

To address the problem of data imbalance, 649 COVID-19 images used for training were subjected to image augmentation techniques and a total of 5663 training images were generated. The final dataset used for training and testing is given in Table II. The training set was further split into training and validation sets in the ratio 80:20.

TABLE II. BALANCED DATASET FOR MODEL TRAINING AND TESTING

Data	Normal	Pneumonia	COVID-19
Train	5966	5475	5663
Test	100	100	100

IV. EXPERIMENTS

Motivated by the success of CNNs in image analysis we propose a model for Detecting COVID-19 using CNN, called 'DeCoV-CNN' and its architecture is given in Fig. 2. The model consists of five layers of convolutional filters: the first layer has 16 filters followed by 32, 64, 128 and 256 filters in successive layers. All kernels are of size 3×3 and a zero padding was used to make the input and output width and height dimensions the same. As shown in Fig. 2, a 'maxpool' layer was added after first convolution layer and a 'batch normalization' followed by 'max pool' layer added for the remaining four convolutional layers. A dropout layer was added after the fourth and fifth convolutional layers to avoid overfitting. The convolutional layers were followed by dense layers with 512, 128, 64 and 3 nodes in each layer. Dropout layers were also used after each dense layer. The output layer had a 'softmax' activation function and previous layers of convolution and dense layers used 'Relu' and loss function used was 'categorical cross entropy'.

Two models were built on the above framework. In the first model called, CNN1, only one convolution layer was used in each convolution block, while in the second model, called CNN2, two Conv layers were used in each convolution block, shown by dotted lines in Fig. 2. Since convolutional layers are expected to capture more subtle patterns from CXR, performance of CNN1 was compared with CNN2 by adding an extra Conv layer in each block. The number of parameters in the two CNNs range from 2-6 million depending upon the input dimensions used. Further, a VGG-16 model, pre-trained on ImageNet database, was downloaded from Keras for evaluating the performance of the two models, CNN1 and CNN2. In VGG-16 model, after pre-training only the convolution layers were used for fine tuning and classification of CXR images. The top most fully connected layers in this model were replaced with four customized fully connected layers of 512, 256, 128 and 3 nodes in each layer respectively.

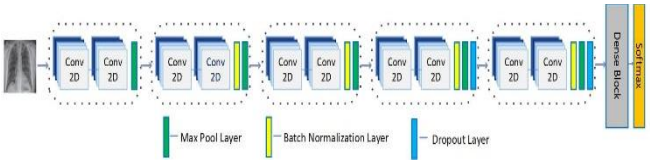


Fig. 2. Architecture of DeCoV-CNN with two convolution layers in each convolution block, model CNN2, is shown. In CNN1, the second Conv2D layer is removed from each convolutional block (shown as dotted rectangles).

V. IMPLEMENTATION DETAILS

The models were implemented using Tensorflow with Keras backend. Optimizer used was 'adam' and default learning rate was set to 0.001. Learning rate was scheduled to reduce by monitoring the validation loss for a period of 2 epochs, i.e., if the loss did not improve for two epochs, the learning rate was reduced. All the models were trained for 10 epochs since beyond 10 epochs the accuracy was found to be stagnate. The experiments were performed using Google Colabarotory with GPU. The training data was generated using 'ImageDataGenerator' class and 'flow_from_directory' function of Tensorflow with validation split of 0.2. The function takes path of a directory as input and generates batches of data for training after shuffling the data. Different batch sizes were used for analysis as described in the experiments section. Testing data was also generated using the same function with input directory set as 'test'.

VI. RESULTS

Performance analysis of the two CNN models for 3-class classification of chest X-Rays is carried out and compared with the pre-trained VGG-16 network. Here we show that the performance of CNN model, assessed in terms of sensitivity and PPV is comparable to the deep network, VGG-16 model, in classifying CXR into 3 classes, Normal, Covid Pneumonia (CP) and Non-Covid Pneumonia (NCP). An important feature of any DL model is its training cost which is typically measured in terms of the number of parameters required for achieving the classification. Most DL models proposed for identifying COVID-19 using CXR images require over 10 million parameters with initialization of weights by pretraining on ImageNet database [3]. Training such big networks is highly time consuming and difficult to deploy on smaller devices and in a clinical setting. The DeCoV-CNN model has much fewer parameters, ~ 2-6 million, thus making it easily portable to smart devices for clinical applications.

Accuracy and loss curves are two important plots that help us understand how the network is training. Loss curves give a snapshot of training process and the direction in which the network learns. We expect the loss to decrease as the network learns over the epochs. If training and validation loss plots start departing consistently, it is indication that training should be stopped at previous epoch. Accuracy plot gives information about over-training, and larger the gap between training and validation accuracy, larger is the problem of overfitting.

Three metrics used for the performance evaluation of models are: Accuracy, Sensitivity (recall) and Positive Predictive Value (PPV) or precision. Accuracy gives the proportion of true results to the total number of cases: $(TP+TN)$ $/(TP\ FP+TN+FN)$, where *TP*, *TN*, *FP* and *FN* correspond to the number of true positives, true negatives, false positives and false negatives, respectively. It measures fraction of correct classifications. Sensitivity is defined as the ratio of true positives to all positive cases, $TP/(TP+TN)$ and gives the proportion of correct positive results. Positive Predictive Value (PPV) gives the proportion of positively classified cases that are truly positive: $TP/(TP+FP)$, and is a measure of how relevant a positive result is. Accuracy is a less biased metric as it includes all four parameters *TP*, *FP*, *TN* and *FN* in its calculation.

Below we discuss performance evaluation of the proposed model DeCoV-CNN in classifying CXR images. In the first analysis we present the comparison of two CNN models, CNN1 and CNN2 with the deep network, VGG-16, pre-trained on ImageNet database. The dimension of the input CXR images considered is 224×224×3 and all the three networks are trained for 10 epochs with batch size of 64. Accuracy and loss curves on training and validation sets over training epochs are shown in Figure 3 for the three models. Training and validation accuracy is seen to improve over the epochs for two CNN models (3(a) and (b)), however, validation accuracy departs from the training accuracy in case of VGG-16 model (3(c)).

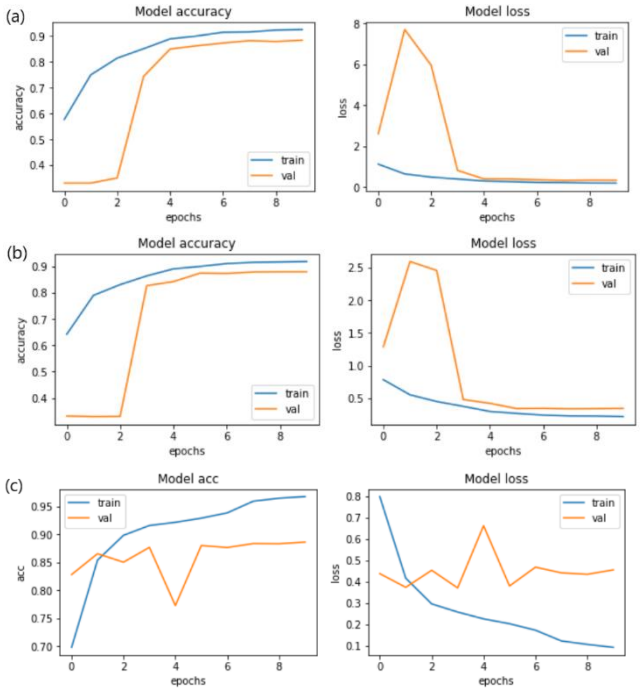


Fig. 3. Accuracy and loss curves for training and validation steps. (a) CNN1 model, (b) CNN2 and (c)VGG-16 model

Training and validation loss curves decrease and approach values 0.4 - 0.5 for CNN1 and CNN2, similar to validation loss for VGG-16 though its training loss was ~0.1. This indicates that a simple CNN architecture can learn better compared to the deep VGG architecture. Accuracy values are given in Table III and Sensitivity and Positive Predictive Value (PPV) metrics are summarized in Table IV. Though training accuracy is 96% for VGG-16 model, validation and test accuracies are no better than CNN models. Test accuracy achieved by CNN1 is 0.88, comparable to VGG-16, and better than CNN2, ~ 0.80. Among the two CNN models, CNN1 is observed to achieve better training and testing accuracies even with fewer number of convolutional layers. That is, the performance actually improved on decreasing the number of convolutional layers in this case. This shows that simple CNNs are as good as deep neural network like VGG for this problem. Looking at other metrics, for the COVID-19 class PPV of 0.99 is obtained for CNN1 and 0.96 for CNN2 and VGG-16. This indicates the ability of CNN1 model in labelling COVID-19 images correctly with higher precision, i.e., with fewer false positives. Sensitivity is slightly lower for CNN1 (0.67) and CNN2 (0.64) compared to VGG-16 (0.79) for COVID-19 class while it is comparable for other classes. Higher sensitivity of the model indicates fewer false negatives in classifying CXR into COVID-19 or other classes. With availability of more COVID-19 CXR images, we expect the sensitivity of CNNs to further improve.

Next, we assessed the effect of input image dimensions on the performance of two CNN models with the objective of finding any information loss on reducing the size of images. The networks were trained on two image dimensions, 150×150×3 and 224×224×3 for 10 epochs with batch size 128. No significant difference in the accuracy and loss curves for training and validation was observed for the two image dimensions considered for both the models (results not shown). This has a significant impact in storage as well as computational efficiency in training the model. Sensitivity and PPV values are summarized in Table V for the two image dimensions. From Table V no difference in PPV values is observed for the two image dimensions (0.96) and sensitivity values are also comparable, 0.79 (224×224×3) and 0.77 (150×150×3) for CNN1, indicating no information loss on reducing the image size. The test accuracy is also comparable, in fact marginally better in classifying COVID-19 cases using smaller image dimension for both the models (Table VI).

TABLE III. ACCURACIES OF CNN1, CNN2 AND VGG-16 MODELS USING IMAGE SIZE 224×224×3 AND BATCH SIZE 64

Model	Accuracy		
	Training	Validation	Testing
CNN1	0.925	0.875	**0.883**
CNN2	0.914	**0.892**	0.803
VGG-16	**0.968**	0.886	0.877

TABLE IV. SENSITIVITY AND PPV OF CNN1, CNN2 AND VGG-16 MODELS USING IMAGE SIZE 224×224×3 AND BATCH SIZE 64

Model	Metric	COVID-19	Normal	Pneumonia
CNN1	Sensitivity	0.67	**0.97**	0.89
	PPV	**0.985**	0.815	0.788
CNN2	Sensitivity	0.64	0.87	**0.91**
	PPV	0.955	0.821	0.717
VGG-16	Sensitivity	**0.79**	0.96	0.88
	PPV	0.963	**0.835**	**0.854**

To study the effect of batch size, analysis was carried out on two different batch sizes, 64 and 128 (Table IV and Table V), keeping the image dimensions 224×224×3. Batch size refers to the number of images the model works on before updating its parameters and since the difference between the classes could be very subtle, a higher batch size would enable the model to learn better. The sensitivity values improved from 0.67 to 0.79 with respect to COVID-19 class on increasing the batch size for CNN1, indicating the model indeed learned better on considering higher number of images at a given time.

TABLE V. SENSITIVITY AND PPV OF CNN1 AND CNN2 MODELS USING TWO IMAGE DIMENSIONS AND BATCH SIZE 128

Model	Metric	COVID-19	Normal	Pneumonia
Image Size: 224×224×3				
CNN1	Sensitivity	0.79	0.91	0.88
	PPV	0.963	0.858	0.786
CNN2	Sensitivity	0.62	0.90	0.89
	PPV	0.939	0.818	0.718
Image Size: 150×150×3				
CNN1	Sensitivity	0.77	0.93	0.89
	PPV	0.963	0.877	0.781
CNN2	Sensitivity	0.70	0.95	0.86
	PPV	0.921	0.812	0.804

TABLE VI. ACCURACY OF CNN1 AND CNN2 MODELS USING TWO IMAGE DIMENSIONS AND BATCH SIZE 128

Model	Accuracy		
	Training	Validation	Testing
Image Size: 224×224×3			
CNN1	0.9303	0.8702	0.8600
CNN2	0.9144	0.8915	0.8033
Image Size: 150×150×3			
CNN1	0.925	0.875	0.883
CNN2	0.914	0.878	0.837

Below we present comparison of our results with some recent studies using deep learning architectures for detecting COVID-19 from CXR images. COVID-Net [4] proposed by Wang and Wong has a very complex architecture based on ResNet with over 11.75 million parameters. The model achieved an accuracy of 93% for 3-class classification and reported 91% sensitivity and PPV 98.9% for COVID-19 class. For both the CNN models, accuracy and PPV values are comparable with this model. However, the sensitivity is lower for the simple CNN model (Table V). Khan et al [5] proposed CoroNet based on Xception architecture with 33 million parameters. It achieved an accuracy of 89.6% for 3-class classification. In the study by Rajaraman et al [11], an ensemble of architectures, *viz*., VGG, Inception, Exception, etc., is used and the best performing models pruned iteratively to improve the performance of classifiers. In this case an accuracy of 99% was reported, which is one of the best performing models reported. However, it involves a selection of multiple models, pre-trained on ImageNet, resulting in very large number of parameters and very high training costs. Most other studies that achieved high classification accuracies were binary classification models for COVID-19 *vs* other classes (*viz*., healthy/pneumonia/other non-covid infections) [6], [7], [14]. Compared to these very complex and deep architectures, the simple CNN model proposed here gives comparable accuracies with fewer layers and much fewer parameters. These results clearly show that one does not always require very deep layers to achieve good classification accuracies.

VII. CONCLUSION

In this study we show that a simple CNN model with fewer parameters can have performance comparable to a deep learning network in the classification of CXR images, making it possible to deploy on small hand-held devices such as smart phones. Here a detailed analysis of the performance of two CNN models for multi-class classification of CXRs in the context of COVID-19 is presented. Performance analysis of DeCoV-CNN with very deep networks shows its ability in identifying COVID-19 CXR images on par with other complex arcitectures. Further, in this model three-class classification (COVID-19 *vs* Other Pneumonis *vs* Normal) is carried out, and indicates the ability of the model to distinguish between pneumonia like symptoms due to COVID-19 from those due to other bacterial and viral infections with a high precision of 0.96. Our analysis showed no significant difference in the performance of the model on reducing the image dimensions. However, it was observed that the performance improved for larger batch sizes. To summarize, a simple CNN with fewer layers but larger batch sizes performed very well in classifying the COVID-19 CXR images. It is expected that the performance of the proposed model would further improve with the availabilty of more COVID-19 CXR images for training.

REFERENCES

[1] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in False-Negative Rate of Reverse Transcriptase Polymerase Chain Reaction–Based SARS-CoV-2 Tests by Time Since Exposure," *Annals of Internal Medicine*, May 2020.

[2] J. Cleverley, J. Piper, and M. M. Jones, "The role of chest radiography in confirming covid-19 pneumonia," *BMJ*, vol. 370, Jul. 2020.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255.

[4] L. Wang and A. Wong, "COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images," *arXiv:2003.09871 [cs, eess]*, May 2020, Accessed: Jul. 30, 2020.

[5] A. I. Khan, J. L. Shah, and M. M. Bhat, "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105581, Nov. 2020.

[6] A. Sedik *et al*., "Deploying Machine and Deep Learning Models for Efficient Data-Augmented Detection of COVID-19 Infections," *Viruses*, vol. 12, no. 7, Art. no. 7, Jul. 2020.

[7] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, "Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105608, Nov. 2020.

[8] S. Khobahi, C. Agarwal, and M. Soltanalian, "CoroNet: A Deep Network Architecture for Semi-Supervised Task-Based Identification of COVID-19 from Chest X-ray Images," *medRxiv*, p. 2020.04.14.20065722, Apr. 2020.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *arXiv:1512.00567 [cs]*, Dec. 2015.

[11] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-Rays," *IEEE Access*, vol. 8, pp. 115041–115050, 2020.

[12] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 Image Data Collection," *arXiv:2003.11597 [cs, eess, q-bio]*, Mar. 2020.

[13] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3462–3471, Jul. 2017.

[14] D. Das, K. C. Santosh, and U. Pal, "Truncated inception net: COVID-19 outbreak screening using chest X-rays," *Phys Eng Sci Med*, pp. 1–11, Jun. 2020.

[15] S. Suba and N. Parekh, "Machine Learning approaches in detection and diagnosis of COVID-19", Artificial Intelligence and Machine Learning in Healthcare Perspective, A. Saxena and S. Chandra, Eds. Springer Nature Singapore (accepted for publication)

Firefly Algorithm Based Maximum Power Point Tracking for Photovoltaic System under Partial Shading Condition

Hasan Basri KARAKAYA
Department of Electrical and Electronics, Faculty of Engineering, Kahramanmaras Sutcu Imam University, Kahramanmaras, Turkey
hbasrikarakaya@gmail.com

O. Fatih KECECIOGLU
Department of Electrical and Electronics, Faculty of Engineering, Kahramanmaras Sutcu Imam University, Kahramanmaras, Turkey
fkececioglu@ksu.edu.tr

***Abstract*— It is important that a photovoltaic (PV) panel under partial shading condition (PSC) be operated around the global maximum power point (GMPP) in order to the increasing efficiency of the PV system. Recently, meta-heuristic algorithms are used to determine and track the GMPP of PV systems. The firefly algorithm (FA) is preferred because of its basic structure and fast speed. In this study, the firefly algorithm is used to maximum power point tracking (MPPT) of PV system under PSC. Three simulation studies were carried out in MATLAB / Simulink environment to evaluate the performance of the proposed MPPT method. The results show that the performance of FA based MPPT under the PSC scenarios is satisfactory.**

Keywords*— *firefly algorithm, MPPT, partial shading condition

I. INTRODUCTION

Decreasing fossil fuel reserves, global warming, and environmental worries have been increased the demand for renewable energy sources. Photovoltaic (PV) systems have become the most favored renewable energy system in the last decades because of their advantages which are noiseless operation, reduced maintenance costs and high modular structure. However, the major disadvantage of PV systems is their high dependence on environmental conditions. Therefore, it is decisive to operate the PV array systems around the maximum power point to increase the efficiency of the PV system [1-2].

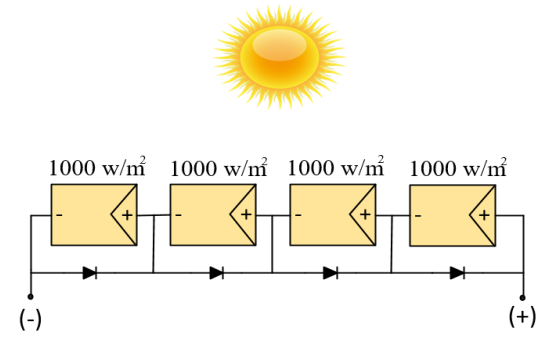
The current-voltage and power-voltage characteristic curves of PV arrays are nonlinear. The PV arrays with uniform insolation are formed a single maximum power point on the characteristic curves. Generally, the classical MPP tracking methods that consist of incremental conductance (IC) algorithm, perturb and observe (P&O) algorithm and fuzzy logic controls are successful in drawing maximum power from PV panels with uniform solar irradiance distribution [3-4].

Due to the environmental factors such as trees, clouds, or tall buildings around the PV power plant, PV panels in an array may receive different solar irradiance. This situation is called partial shading conditions (PSC). Characteristic curves of PV arrays under PSC have more than one MPPs. One of these points is called the global MPP (GMPP) and others are called local MPPs (LMPP). The above-mentioned classical MPPT methods are insufficient to track global MPP in a PV system under PSC. Different meta-heuristic optimization methods such as firefly and cuckoo search, Genetic, Particle Swarm optimization algorithms are used in order to overcome this problem [5]. In this study, the GMPP convergence performance of the Firefly algorithm is evaluated by simulation studies that were carried out in MATLAB / Simulink environment.

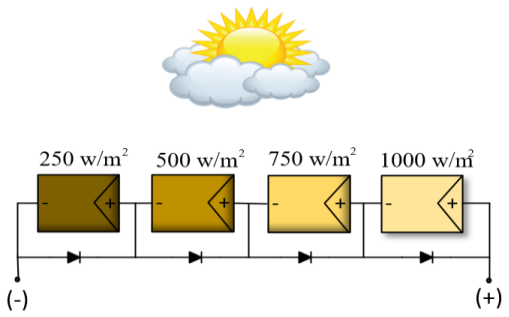
The present paper is organized as follows. The modelling PV system under PSC is given in Section II. The proposed firefly algorithm is explained in Section III. The performance of the FA based MPPT method is examined in IV. Finally, the conclusions of the study are explained in Section V.

II. MODELLING PV SYSTEM UNDER PSC

A PV array will only have one MPP in its characteristic curves unless it is under environmental factors such as clouds, rain, dust, trees, and building shadows [6]. Although, when the mentioned environmental conditions occur, the characteristic curves of the PV array will have multiple MPP. Uniform insolation and partial shading condition on a PV array are illustrated in Figure 1a and 1b respectively. The characteristic curves of the PV array under PSC are shown in Figure 2.

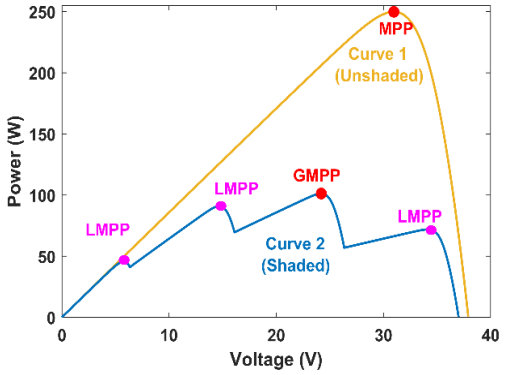


(a)

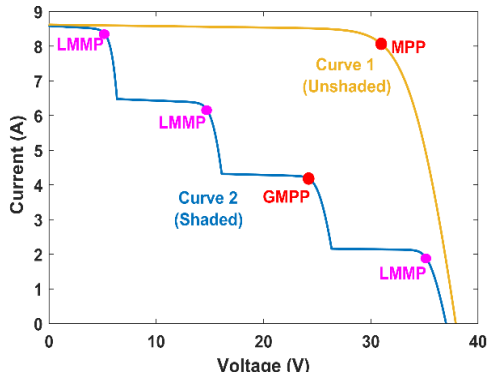


(b)

Fig. 1. PV array under uniform insolation (a) and PSC (b)



(a)



(b)

Fig. 2. The characteristic curves of the PV array under PSC

The curves illustrated with yellow color in Figure 2 belong to the PV array with uniform insolation (i.e. unshaded array). As shown in Fig. 2, The partial shading condition causes more than one maximum power point on power-voltage and current-voltage curves of the PV system. Whole MPP values illustrated with pink color on this figure are called as local MPP (LMPP) and the highest value among these LMMPs is called the global MPP (GMPP) [7-8].

A. Firefly Algorithm

Among the meta-heuristic optimization algorithms, the firefly algorithm (FA) which is inspired by the flashing 7behavior of fireflies and was introduced by Yang [9-10] is used for many optimization problems such as MPPT under PSC. This algorithm is managed by two main operators, brightness and degree of attraction [11].

The FA is based on three assumptions. First, all fireflies are unisex and any firefly will be attracted to the brighter one. Second, based on the space between fireflies itself and any other, the attractive one is determined to its brightness. If there is no brighter one in a firefly colony, each one will move randomly. Finally, the brightness of a firefly is determined by the value of the objective function of a given problem [12-13]. The FA is referred by three equations. Distance between two fireflies is defined as follows:

$$r_{kl} = \left\| x_k - x_l \right\| = \sqrt{\sum_{m=1}^{s} (x_{k,m} - x_{l,m})^2} \quad (1)$$

where x_k and x_l are positions of the fireflies k and l. $x_{k,m}$ and $x_{l,m}$ are the m th components of the fireflies. s is the number of dimensions. The attractiveness (or brightness), β is a function of fireflies distance and is given by

$$\beta(r) = \beta_0 \exp\left(-\gamma (r_{kl})^n\right), \quad n \geq 1 \quad (2)$$

where β_0 is initial attractiveness, γ is absorption coefficient and n is constant. The movement of a firefly is defined by the following equation:

$$x_k^{t+1} = x_k^t + \beta(r)(x_k - x_l) + \alpha(\varepsilon - 0.5) \quad (3)$$

where α is random movement factor to be in [0, 1] and ε is a random number uniformly distributed [0, 1]. The steps of the implementing of FA to proposed controller are illustrated in Figure 3.

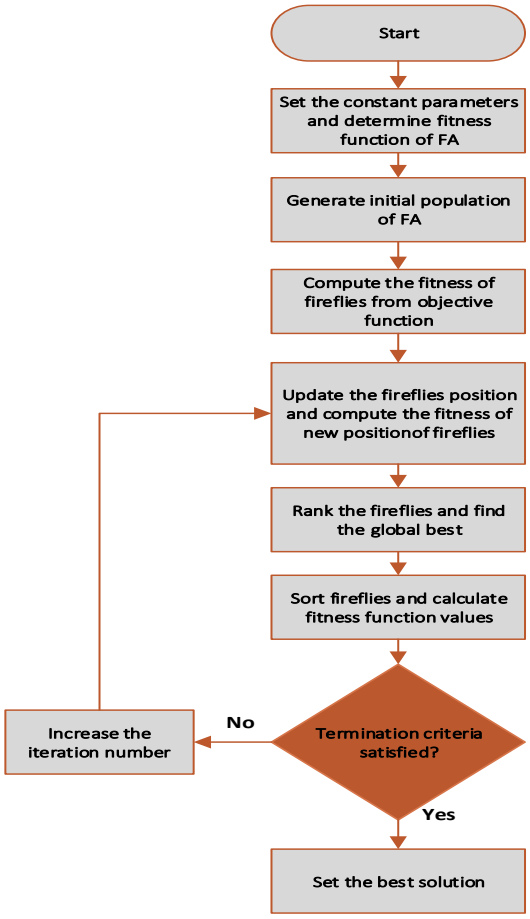


Fig. 3. Flowchart of Firefly Algorithm

III. Simulation Results

In this section, the MPPT system based on Firefly Algorithm is modeled using MATLAB/Simulink environment and Sim Power System Toolbox. The model of proposed MPPT control method for PV system has been shown in Figure 4. Whole simulation parameters that are consist of values of the DC-DC converter, firefly algorithm MPPT, and Matlab/Simulink are given in Table 1. In order to evaluate the performance of the proposed MPPT control method, the simulation studies are implemented under three PSC scenarios. The panel temperature values of all studies are selected as 25°C.

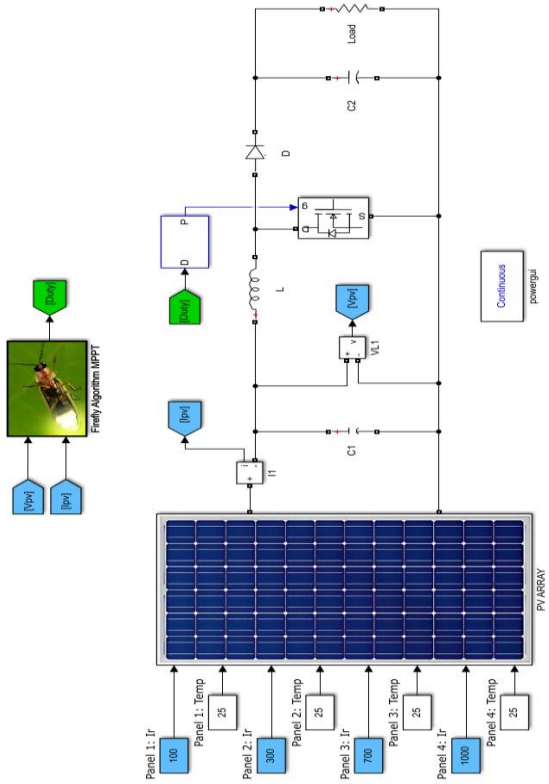


Fig. 4. Simulation model

TABLE I. WHOLE SIMULATION PARAMETERS

DC-DC Converter		
Inductor	L	3.3 mH
Capacitor	C1	220 µF
Capacitor	C2	150 µF
Load Resistance	RL	30 Ω
Switching Frequency	f	20 kHz
Firefly Algorithm		
Number of Fireflies	N	4
Firefly Attractiveness	β_0	1.5
Light absorption coefficient	γ	10
Random Parameter	α	0.2
Matlab/Simulink		
Solver	ode23tb	
Sampling time	0.01 s	

The insolation levels, P-V and I-V curves of PV panels under PSC scenarios are illustrated in Figure 5a-c. The figures show that the variations of GMPP values against different insolation levels. The GMPP values of PSC scenarios have been located in different regions of the characteristic curves of the PV array in order to evaluate the effectiveness of the proposed MPPT method. The GMPP values of patterns are calculated as 85.36 W, 101.2W, and 140W respectively. Total simulation time is determined 0.4s for all scenarios in order to investigate MPP tracking efficiencies and speeds. Simulation results of PV power and duty cycle of DC-DC converter obtained from FA based MPPT for PSC scenarios are shown in Figure 6.

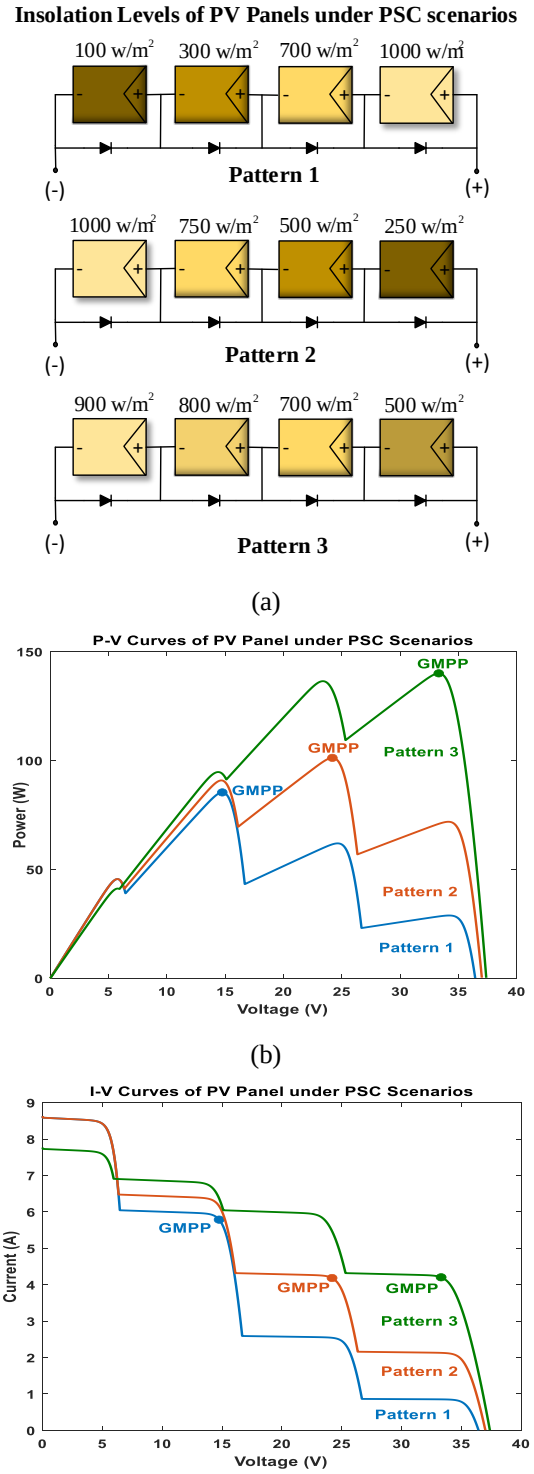


Fig. 5. The insolation levels (a) P-V curves (b) and I-V curves (c) of PV panel under PSC scenarios.

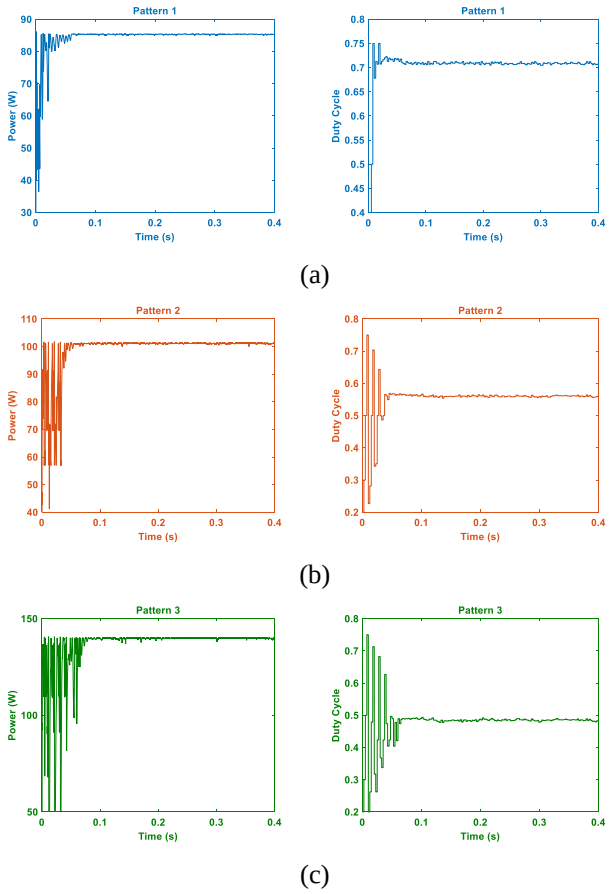


Fig. 6. The insolation levels (a) P-V curves (b) and I-V curves (c) of PV panel under PSC scenarios.

As Figure 6(a) is examined, the tracking speed and efficiency of FA based MPPT for pattern-1 have been calculated at 70ms and 99.55 % respectively. In this scenario, the output power value of PV panel is 84,98W after the settling time. The GMPP value was 85,36 W and the proposed method caught the MPP successfully. As Figure 6(b) is investigated, proposed method has 70 ms tracking speed and 99.95 % tracking efficiency. In this scenario, the output power value of the PV panel is 101,15 W after the settling time. The GMPP value of pattern-2 was 101,2 W, and the performance of FA-based MPPT is satisfactory. As shown in Figure6(c), the tracking speed and efficiency of proposed MPPT method for pattern-3 have been calculated at 90ms and 99.42 % respectively. In addition to this, when the duty cycle changes of three PSC scenarios are examined, the proposed FA based MPPT algorithm is more effective against GMPP located on the left side of characteristic curves than other sides.

IV. Conclusion

This paper examines the performance of the firefly algorithm based MPPT for PV system under partial shading conditions. In order to assess the performance of the proposed FA based MPPT method, three PSC patterns are developed by using MATLAB/Simulink environment. The performance results of FA based MPPT controller obtained from simulation studies show that the proposed controller has high tracking speed and efficiency for MPPT of the PV system. The simulation results are obviously shown the effectiveness of the proposed MPPT algorithm under PSC scenarios.

References

[1] B. H. Kwon, K. H. Nam, “Three-PhasePhotovoltaic System with Three-Level Boosting MPPT Control,” IEEE Transactions on Power Electronics, vol. 23, pp. 2319-2327, 2008.

[2] S. Qin, M. Wang, T. Chen, and X. Yao, “Comparative Analysis of Incremental Conductance and Perturb-and-Observation Methods to Implement MPPT in Photovoltaic System,” ICECE, pp. 5792 – 5795, 2011.

[3] R. Faranda, S. Leva, and V. Maugeri, “MPPT techniques for PV Systems: energetic and cost comparison,” Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1-6, 2008.

[4] A. Safari, and S. Mekhilef, “Simulation and hardware implementation of incremental conductance MPPT with direct control method using cuk converter,” IEEE Trans Ind Electron, vol.58, pp. 1154-61, 2011.

[5] P. Selvapriyanka, and G.Vijayakumar, “Partical swarm optimization based MPPT for PV system under partial shading conditions,” IJIRSET Inter. Conf. on Eng. Tech. and Science, vol. 3, pp. 856- 861, January 2014.

[6] S. Hesari, “Design And Implementation Of Maximum Solar Power Tracking System Using Photovoltaic Panels,” International Journal of Renewable Energy Research, vol. 6, pp. 1221-1226, 2016.

[7] A. Singh, “MATLAB / SIMULINK Simulation of PV System based on MPPT in Variable Irradiance with EV Battery as Load,” IEEE Int. Conf. Comput. Intell. Comput. Res., pp. 4–7, 2017.

[8] D. Pera, J. A. Silva, S. Costa and J. M. Serra, "Investigating the impact of solar cells partial shading on photovoltaic modules by thermography," IEEE 44th Photovoltaic Specialist Conference (PVSC), pp. 1979-1983, 2017.

[9] X. S. Yang, “Firefly algorithms for multimodal optimization,” Stochastic Algorithms: Foundations and Appplications, Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 3-6, 2009.

[10] X. S. Yang, “Nature-Inspired Metaheuristic Algorithms,” Luniver Press, pp. 81-84, 2008.

[11] E. Mostafa, and N.K. Bahgaat, “A Comparison Between Using A Firefly Algorithm and A Modified PSO Technique for Stability Analysis of a PV System Connected to Grid,” International Journal of Smart Gridij Smart Grid., Vol. 1, pp. 1-8, December 2017.

[12] T.T. Yetayew, T. R. Jyothsna, and G. Kusuma, “Evaluation of Incremental conductance and Firefly algorithm for PV MPPT application under partial shade condition,” IEEE 6th International Conference on Power Systems (ICPS), 2016

[13] D. F. Teshome, C. H. Lee, Y. W. Lin, and K. L. Lian, “A modified firefly algorithm for photovoltaic maximum power point tracking control under partial shading”, IEEE Journal of Emerging and Selected Topics in Power Electronics, pp. 661-671, 2017.

Developing a Protective – Preventive and Machine Learning Based Model on Child Abuse

Fatih MERT
Department of Computer Engineering
Istanbul Commerce University
Istanbul, Turkey
fatih.mert@istanbulticaret.edu.tr

Muhammed Ali AYDIN
Department of Computer Engineering
Istanbul University - Cerrahpasa
Istanbul, Turkey
aydinali@istanbul.edu.tr

Abdül Halim ZAIM
Department of Computer Engineering
Istanbul Commerce University
Istanbul, Turkey
azaim@ticaret.edu.tr

***Abstract* — Online grooming is an ever-increasing problem in societies and the time spent online is recently started to rise drastically. People can become anonymous whilst posting, sharing his/her own opinion, and being a part of online chatting. Option to be anonymous also brings together the chance for hiding personal identity when making an attempt on illegal activities. Online grooming is one of the significant areas of aforementioned actions and sexual predators can easily use online chatting platforms to quickly build a friendly relationship with children or teenagers to gain their trust and make them share their obscene media files. These sexual predators mostly try to convince their victims to meet and it may lead to having sexual intercourse with a minor. In order to draw attention to the huge challenge that most societies face, this study mainly aims to identify predators in the early stage of online communication. The objective is to do an investigation to detect child grooming through online chat records by using Machine Learning techniques. In the first part of the study, it has been achieved to make a multi-label classification on a Wikipedia dataset with more than 97 percent accuracy, where a given text gets classified based on the toxicity types. The outcome of this work is also used in the second stage and herein PAN12 dataset has been used to train and test our model. We have ended up with more than 92 percent accuracy, where suspicious conversation messages from the chat records get identified and sexual predators can be recognized.**

Keywords— Child Abuse Detection, Online Sexual Predator Identification, Multi-Label Text Classification, Machine Learning

I. INTRODUCTION

As of October 2020, the number of active internet users has been reached around 4.66 billion people all over the world, amounting to nearly 50 percent of the whole population globally [1]. The number of messaging application users also exceeds billions world-wide. As people tend to use social applications with the spread of internet usage, it comes with unwanted troubles as well. More than 80 percent of the youth who resided in the USA could reach the internet and children whose ages were between 5 – 16 were spending nearly 7 hours a day on the devices having internet access. Even though the internet is a fabulous source of information, it may also become an environment full of danger, especially for children. One of the major reasons behind this issue is that there is no recognized way to regulate the usage of the internet. People can become anonymous whilst posting, sharing his/her own opinion, and being a part of online communication. Option to be anonymous also brings together the chance for hiding personal identity when making an attempt on illegal activities. Hence, any malevolent person can easily attempt to the solicitation, both in virtual and real life. Online solicitation addresses the moment when an adult asks for having sexual intercourse, being a part of undesired sexual actions, or having sexual talks in the online areas. When youth are compared to adults, their level of sense for making an inference regarding having an inkling of potential threats waiting after interaction with people who have ill-will against themselves. Based on the outcomes of a study, nearly 20 percent of the youths have been subjugated to sexual content without their acquiescence and nearly 10 percent of them have experienced unwelcome online sexual abuse [2]. Herein, we should not overlook the unreported solicitations, since most of the children feel quite ashamed and guilty preventing them to explicitly declare the situation they have been going through. Moreover, they may even not be aware of that the fact that they were abused. Online-facilitated child abuse could be done through many ways: The production, dissemination or possession of CSAM (Child Sexual Abuse Materials), also knowns as ‘’child pornography’’ in the general acceptance, sexting (sending or receiving of sexual texts or media files such as pictures or videos through technology usage), revenge pornography, online child grooming (befriending and building an emotional bridge with children to heighten their exiting curiosity about sex, with the ultimate aim of meeting them in real, by considering sexual benefits), active sexual harassment, sexual extortion (also known as sextortion), abuse of children over online prostitution, live streaming of sexual incident, and etc. [3] Online grooming is one of the significant ways of aforementioned sexual abuse actions and sexual predators can easily use online chatting platforms to quickly build a friendly relationship with children or teenagers to gain their trust and make them share their obscene media files. These sexual predators mostly try to convince their victims to meet and it may lead to having sexual intercourse with a minor. In order to draw attention to the huge challenge that most societies face, this study mainly aims to identify predators in the early stage of online communication. The objective is to do an investigation to detect child grooming through online chat records by using Machine Learning techniques.

Structure of the rest of this paper is given below:

Section 2, introduces our project and mainly gives the background with the basic understanding and explanations of online child abuse. A detailed summary of the related work conducted in the literature is explained throughout this chapter. Section 3, describes the methodology that has been used throughout this study. Section 4, gives a presentation for the results of the conducted research and whole study. Section 5, concludes with the suggestions and describes possible future works.

II. LITERATURE REVIEW

A. Related Work

A study focusing on child abuse identification in the public health sector with the examination of medical records, used feature extraction from the word clouds, with the help of classifiers such as SVM. The overall performance of this study has been stated as good-enough for daily usage. [4]

Another study conducted in the collaboration with the Swedish Financial Coalition targeted to make a classification for illegal advertisement on Dark Net. In order to perform the algorithm by evaluating several classification models and feature extraction techniques, deep learning was used and it was seen that these deep learning models outperformed the standard methods. [5]

State-of-the-art technologies were presented for analyzing internet crimes against children in a study, where the main purpose was to protect children from being abused by online predators, by developing automated tools. As a result, a program was developed, helping to correctly identify the online sexual predators 60 percent of the time. Following several updates in the second experiment, the identification ratio has been reached 93 percent. [6]

In another study, each line in a conversation has been labeled and communication theories with computer algorithms were used for the identification of predatory messages. After using different machine learning algorithms that classified the lines based on a rule-based approach and phrase matching, the approach labeled the lines with 83.11 percent accuracy where the experiment included 33 unique conversations. [7]

The last sample work in the literature provided an overview of the PAN12 competition focused on sexual predator identification task internationally. This contest was considered a combination of two sub-challenges, the first, being the challenge to identify possible whole predators from the given chat logs, which consisted of both predatory and non-predatory data within. The latter challenge was the task to make an identification for which of the predators' lines in the conversations can be marked as a moment for abusive behaviors to take place. [8]

B. Types of Machine Learning Models

Below figure gives an overview for the categorization of machine learning algorithms and both of our tasks fall into the category of supervised learning.

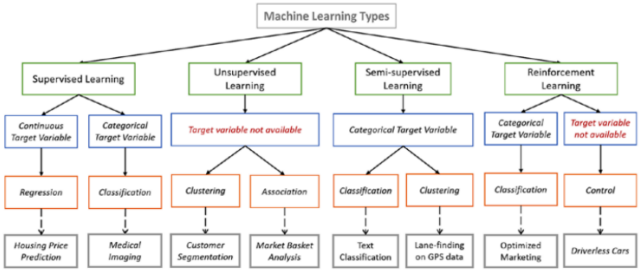


Fig. 1. Types of machine learning algorithms [9]

1. Supervised Learning

Given a set of data points as $\{x^{(1)}, \ldots, x^{(m)}\}$ and this dataset is associated with a set of outputs as $\{y^{(1)}, \ldots, y^{(m)}\}$. Then we would like to come up with a classifier which will learn how to make prediction (predict y from x). In our work, we used 5 different supervised learning models in order to implement binary and multi-label classification tasks.

a) Naïve Bayes

Naive Bayes is a probabilistic machine learning model that is used for classification tasks. It is supposed that the features of each data point will be independent of each other. It is mostly used in sentiment analysis, recommendation systems, and spam filtering tasks. It is also very widely used for text classification tasks. The advantage is speed and being easy-to-use. However, there is also a disadvantage related to the requirement of predictors to be independent. Typical types of the Naive Bayes classifiers are as follows: Multinomial Naive Bayes, Bernoulli Naive Bayes, Gaussian Naive Bayes, and so on.

b) Support Vector Machine (SVM)

Support Vector Machine (SVM) is a type of classifier which is described by a hyperplane which is a V-1 dimensional subspace of a V-dimensional vector space (Christopher M.). The main goal is to find the line which will maximize the minimum distance to the line.

With the use of kernels, SVM is more powerful and helpful for solving classification problems. Some of the kernel options are Linear, Sigmoid, Gaussian, Polynomial... If the Linear kernel is used, learning of the hyperplane gets performed by making the problem transform into a linear algebra problem. SVM Classifier has a regularization parameter called C, in order to detect how much of the misclassification is tolerated for each and every data given as an input. Another parameter of SVM is gamma and it describes how far the influence of a particular input from training reaches. Majority of the time SVMs are chosen for classification tasks, especially binary classification.

c) Logistic Regression

Logistic Regression is one of the most commonly used methods for solving classification problems. This model works for computing the logarithm of the odds as a linear combination of predictors (independent variables). Logistic Regression is mainly a combination of the Sigmoid function and linear regression equation. The advantageous aspect of logistic regression is that high computational power is not needed. It is very easy to use and mostly used by data scientists. In contrast, there is a disadvantage of not being able to deal with a big number of features, and this classifier is not powerful when it comes to overfitting.

d) K-Nearest Neighbors (K-NN)

The K-Nearest Neighborhood (KNN) algorithm is one of the easy-to-implement supervised learning algorithms. It is used in the solution of both classification and regression problems but mostly used in the solution of classification problems in the industry. First, the k parameter is determined. This parameter is the number of neighbors closest to a given point. For example: Let k = 2. In this case, the classification will be made according to the closest 2 neighbors. With the help of the relevant distance functions, the distance of the new data to be included in the sample data set is calculated one by one according to the existing data. The nearest neighbors from the relevant distances are considered. It is assigned to the class of k neighbors or neighbors according to the attribute values.

The selected class is considered to be the class of the observation value expected to be estimated. In other words, the new data is labeled.

e) AdaBoost

AdaBoost, in other words, Adaptive Boosting, is a commonly used machine learning method and it is known as one of the boosting algorithms. Boosting algorithms are used as a collection of classifiers with low accuracy, in order to build a highly accurate classifier. Boosting algorithms are not that much affected by the problem of overfitting. AdaBoost, Gradient Tree Boosting, and XGBoost are the most commonly used boosting algorithms and in this study we used AdaBoost. The main logic behind AdaBoost is about setting the classifier weights and sampled training data in each and every iteration. That way we can make sure of the accuracy of unusual records.

2. Unsupervised Learning

The main goal of unsupervised learning is to find the hidden layers in unlabeled data, $\{x^{(1)}, \ldots, x^{(m)}\}$. Here, the algorithm tries to identify patterns by studying the data. Unlike supervised learning, the machine makes a determination regarding the correlation and relationships checking the available data. The task for making the dataset convert into an organized version, the machine groups the data into clusters so that it will look more organized.

3. Semi-supervised Learning

It is quite similar to supervised learning. However, it combines the work on both unlabeled and labeled datasets. That way, the machine learns how to label the unlabeled data.

4. Reinforcement Learning

The main focus is to provide a set of actions and processes that can be considered as regimented learning. After monitoring and evaluating each and every result for the aim of determining the optimal one, this learning type defines a set of rules in the beginning. In this approach, the machine is taught by trial and error. By learning from the previous experiences, the algorithm adopts as a response to the situation and tries to get the possibly best result.

C. Evaluation Metrics

Loss Function: It is defined as a function that takes the predicted values of z to correspond to the real value of y as input and shows how different they are. Some of the most commonly used loss functions are least-square error, logistic loss, hinge loss, cross-entropy, hamming loss and etc.

$$L : (z , y) \in R \times Y \longmapsto L (z , y) \in R \qquad (1)$$

Confusion Matrix: It is used in order to have a complete representation for the model performance assessment. The figure showing a simple confusion matrix is given as below:

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

Fig. 2. Binary confusion matrix

Accuracy Score: It is the proportion of correctly classified predictions over the total number of predictions.

Precision Score: It is the proportion of correctly predicted inputs over the total number of samples which belongs to that particular class.

Recall Score: It is the proportion of correctly predicted inputs given all existing samples of that class.

F- Score: It refers to the harmonic mean of Precision and Recall scores.

TABLE I. MOST COMMONLY USED EVALUATION METRICS

Evaluation Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FN}$
F1-Score	$\frac{2TP}{2TP + FP + FN}$
True Positive Rate	$\frac{TP}{TP + FN}$

Receiver Operating Curve (ROC: It refers to the plot representation of True Positive Rate (TPR) with respect to False Positive Rate (FPR).

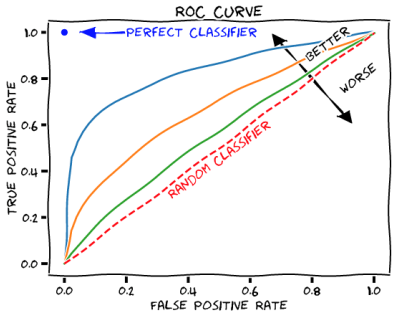


Fig. 3. Evaluation of ROC curves [10]

Precision-Recall Curve: It is the summarization of the trade-off between the True Positive Rate (TPR) and positive predicted value. It is more useful for imbalanced datasets.

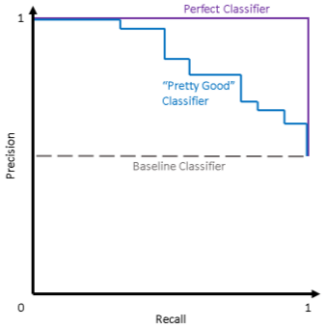


Fig. 4. Evaluation of Precision-Recall curves [11]

K-Fold Cross Validation: It is a widely used method for performance assessment. When the data is scarce, it is most of the time helpful to split the dataset several times creating multiple validations, as well as multiple training and test sets for making the assessment. A sample representation of K-Fold cross-validation is shown in Fig. 5.

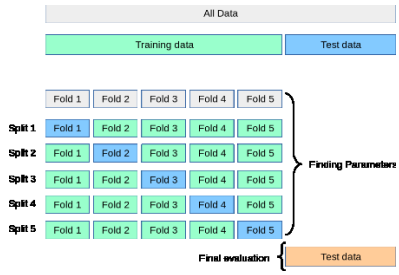


Fig. 5. K-Fold cross validation [12]

III. METHODOLOGY

A. Data Gathering

Wikipedia Comment Dataset: This dataset is provided with a large number of Wikipedia comments that have been labeled by human raters for examining toxic behaviors. Dataset has been obtained via kaggle.com.

PAN12 Dataset: This dataset contains the training and test corpus for the ''Sexual Predator Identification'' task of the Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN) Lab. Dataset has been obtained via zenodo.org.

B. Pseudocode for Toxic_Comment_Classifier

```
1.df_train, df_test: Dataframes for train and test data
2.label_list: [toxic, severe_toxic, obscene, threat, insult, identity_hate]
3.contractions: List of abbreviations for normal and abusive languages, as well as encrypted sexting speech
4.test_size: 0.33
5.classification_models:[MultinomialNB, LinearSVC, LogisticRegression, K-NN, AdaBoost]
6.for each comment_text in df_train and df_test:
        #Apply cleaning
        6.1.Remove HTML tags
        6.2.Remove punctuations
        6.3.Remove non-alphanumeric characters
        6.4.Expand contractions
        6.5.Apply stop-words removal
        6.6.Apply stemming
        6.7.Remove most commonly used words
        6.8.Remove most rarely used words
7.Apply oversampling to deal with data imbalance
8.train,test: Split df_train into 2 dataframes with proportional to test_size
9.for each model in classification_models:
        9.1.X_train: vectorized train
        9.2.X_test: vectorized test
        9.3.ngram_range: (1,2)
        9.4.for each label in label_list:
                9.4.1.model.fit(X_train, train[label])
                9.4.2.prediction: model.predict(X_test)
        9.5.Display performance evaluation results comparing prediction and test[label]
```

C. Pseudocode for Sexual_Predator_Identifier

```
1.train_data, test_data: get raw data in XML format
2.for each conversation in train_data, test_data:
        2.1.if number_of_authors == 1:
                2.1.1. remove conversation
        2.2.if number_of_messages < 5:
                2.2.1 remove conversation
        2.3.if ratio_of_unrecognized_chars > 0.65
                2.3.1 remove conversation
3.for each XML tag in train_data:
        3.1. convert tag into list
4.df_train, df_test: Dataframes converted from tags for train and test data
5.label_list: [sexual_predator]
6.contractions: List of abbreviations for normal and abusive languages, as well as encrypted sexting speech
7.test_size: 0.20
8.classification_models:[MultinomialNB, LinearSVC, LogisticRegression, K-NN, AdaBoost]
9.for each chat_message in df_train and df_test:
        #Apply cleaning
        9.1.Remove chat_message if it has 1 word
        9.2.Remove HTML tags
        9.3.Remove punctuations
        9.4.Remove non-alphanumeric characters
        9.5.Expand contractions
        9.6.Apply stop-words removal
        9.7.Apply stemming
        9.8.Remove most commonly used words
        9.9.Remove most rarely used words
        9.10. Apply spell-check
10.Apply oversampling to deal with data imbalance
11.Identify abusive chat messages using previously built Toxic_Comment_Classifier and mark them in the newly added column called abusive_message
12.X_train,X_test, y_train, y_test: Split df_train using test_size
13.for each model in classification_models:
        13.1.X_train: TF-IDF vectorized train
        13.2.X_test: TF-IDF vectorized test
        13.3.ngram_range: (1,2)
        13.4.model.fit(X_train, y_train)
        13.5.prediction: model.predict(X_test)
        13.6.Apply hyper-parameter tuning to find best parameters
14.Display performance evaluation results comparing prediction and y_test
```

IV. RESEARCH FINDINGS & DISCUSSION

In the first stage of our work, we have concentrated on classification of toxic comments, while in the second stage we focused on the task of sexual predator identification. For both of the specified sub-tasks, we basically used ROC curves and Precision-Recall curves, in order to measure and compare the performance results. Since our datasets are highly imbalanced, meaning that the distribution of the labels are not homogeneous, we could not rely on the Accuracy score. Most of the time, it needs to be clearly defined which classification metrics should be chosen, in the light of the problem domain and priorities. For example; we can better choose depending on what we would like to predict (class labels or probabilities). Assume that we want to predict the probabilities and we need the class labels; Precision-Recall curves would be more useful if the positive class label is more important for us, whereas ROC curves would be more useful if both of the labels are equally important. In our case checked both of them. Depending on further scenarios, both metrics would be giving an idea about which classification model should be used. For the purpose of a clearer interpretation, we also compared the basic metrics (Accuracy, F-Score, Precision, Recall) and displayed the graph considering their average values.

A. *Results of Toxic Comment Classification*

ROC Curves:

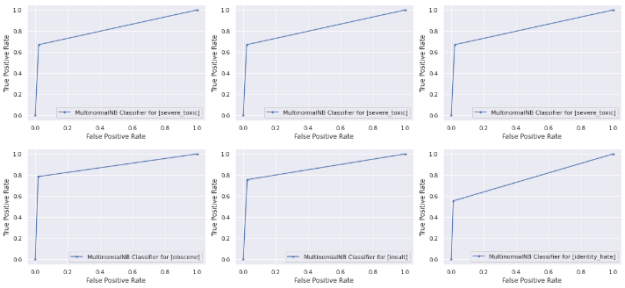


Fig. 6. ROC curves of Multinomial Naïve Bayes

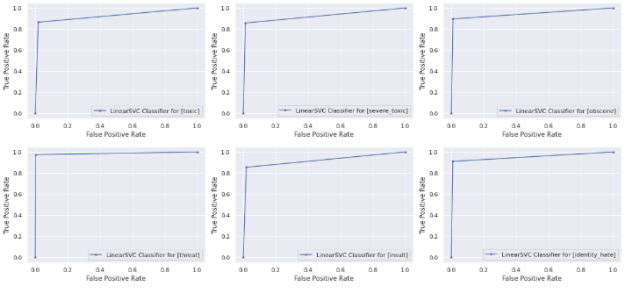


Fig. 7. ROC curves of Linear SVC

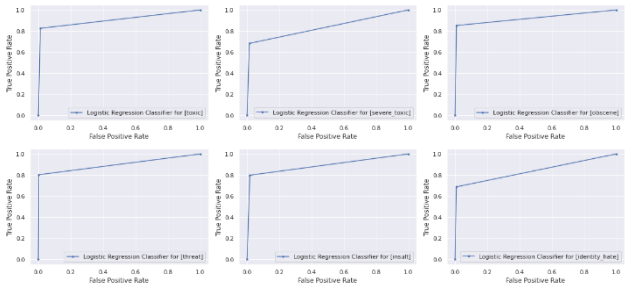


Fig. 8. ROC curves of Logistic Regression

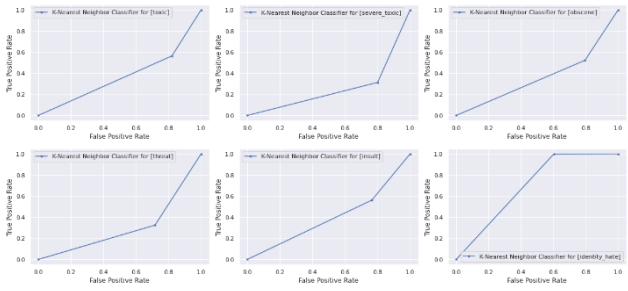


Fig. 9. ROC curves of K-NN

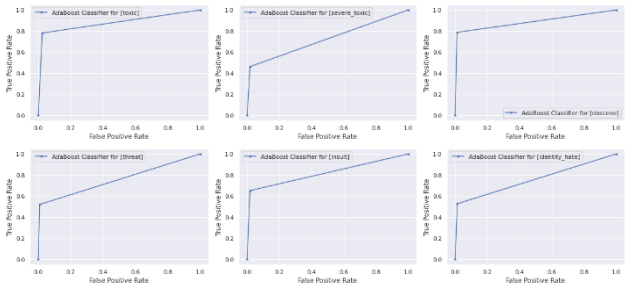


Fig. 10. ROC curves of AdaBoost

ROC curves are basically drawn using False Positive Rates and True Positive Rates. As we checked the ROC curves of our classifier models, we have seen that LinearSVC gives the best results, since the curves for all of the 6 labels are closest to the perfect curve given in Figure 3. We can also see that all of our classifier models performed better than normal case (random classifier) but K-NN classifier. As we checked the literature, it was seen that K-NN was not highly preferred as the others, as well.

***Precision - Recall Curves**:*

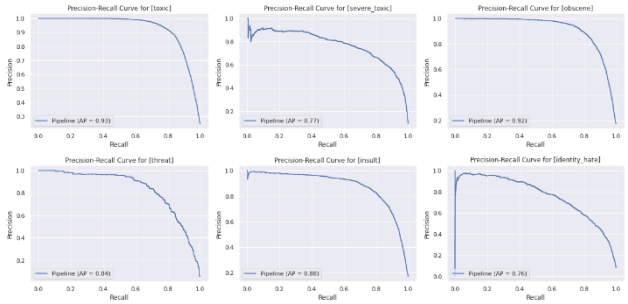


Fig. 11. Precision-Recall curves of Multinomial Naïve Bayes

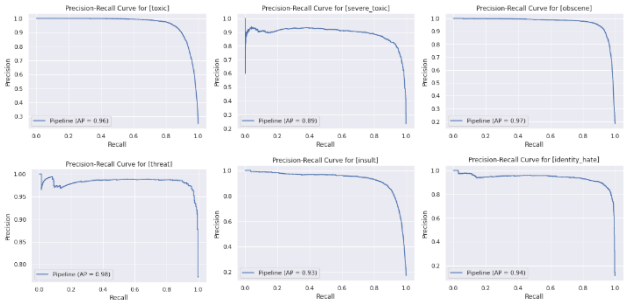


Fig. 12. Precision-Recall curves of Linear SVC

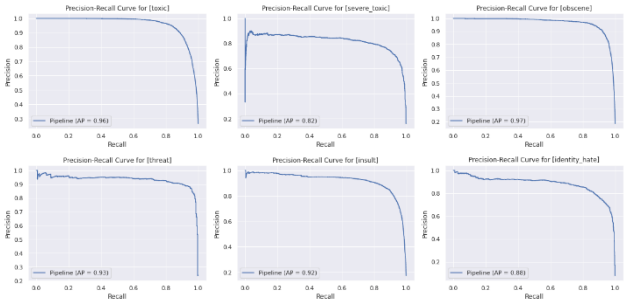


Fig. 13. Precision-Recall curves of Logistic Regression

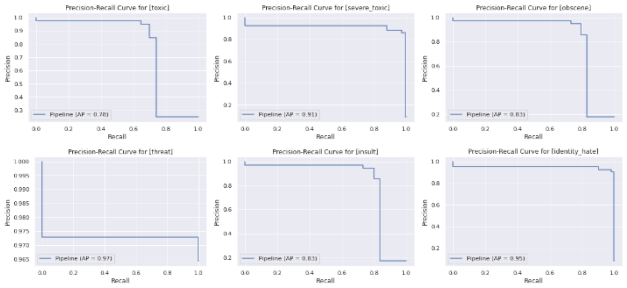


Fig. 14. Precision-Recall curves of K-Nearest Neighbors

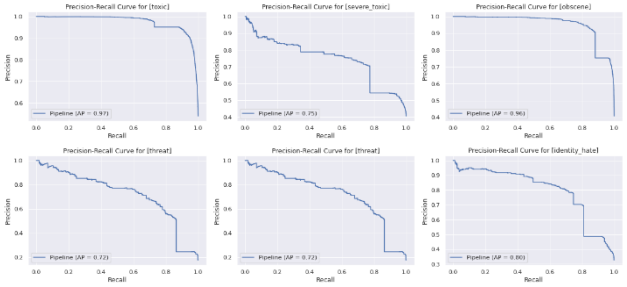
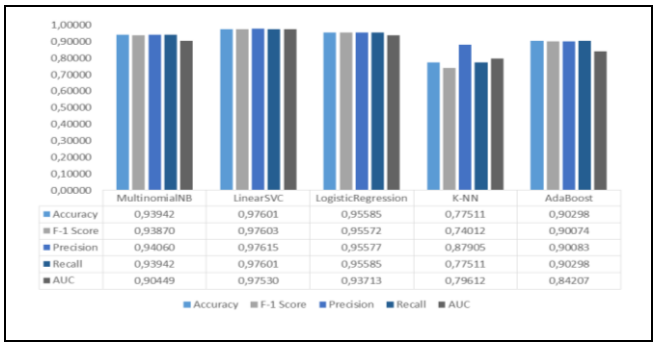


Fig. 15. Precision-Recall curves of AdaBoost

Precision-Recall curves are used to depict the relationship between precision (also known as the positive predicted value) and recall (also knowns as sensitivity). These curves often tend to have frequent zigzags with up and downs in the shape and for that reason, if we combine several classifiers in a single representation, it is more possible for them to intercept with each other when compared to ROC Curves. In our case, the Precision-Recall curves of the Linear SVC model are the best outputs, since their shape is closest to the perfect representation given in Fig 4. We can also see that Multinomial Naïve Bayes and Logistic Regression models have performed quite well.



	MultinomialNB	LinearSVC	LogisticRegression	K-NN	AdaBoost
Accuracy	0,93942	0,97601	0,95585	0,77511	0,90298
F-1 Score	0,93870	0,97603	0,95572	0,74012	0,90074
Precision	0,94060	0,97615	0,95577	0,87905	0,90083
Recall	0,93942	0,97601	0,95585	0,77511	0,90298
AUC	0,90449	0,97530	0,93713	0,79612	0,84207

Fig. 16. Performance metrics comparison of classifier models

We can also compare the overall performance results in the light of the table given in Fig 16. For each metric, it is clear that Linear SVC gives the best results with more than 97 percent achievement. According to the table, Logistic Regression would be our second preferrence and it could be followed by Multinomial Naïve Bayes and AdaBoost. However, K-NN is not a good option to use as our classifier model.

B. *Results of Sexual Predator Identification*

While performing sexual predator identification task, we again used 5 different classification models. These are Multinomial Naïve Bayes, Linear SVC, Logistic Regression, K-NN and AdaBoost. Unlike the previous task, this time our target label was only sexual_predator. Since it is not a multi-label classification problem as before, the overall result was not similar to the toxic comment classification and this time K-NN outperformed the other models.

ROC Curves:

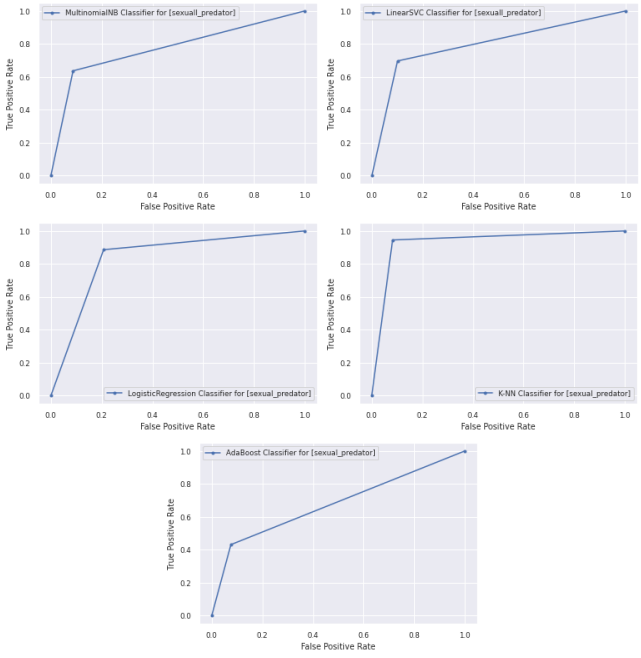


Fig. 17. ROC curves of classifier models

The ROC curves given above tells us that K-NN classifier outperforms the other classification model as its shape is much closer to the perfect case and the area under the curve is higher than the other models. We can also note that AdaBoost is not performing as expected. K-NN is followed by Logistic Regression, Linear SVC and Multinomial Naïve Bayes models as well.

Precision - Recall Curves:

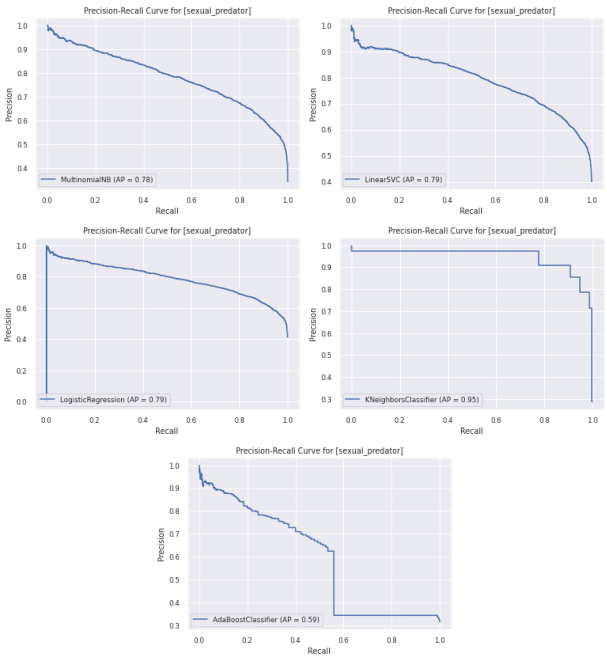
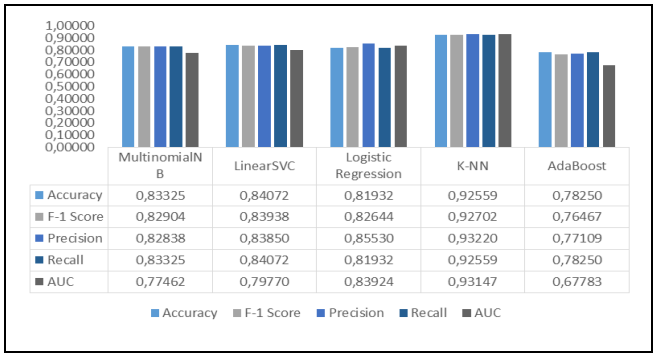


Fig. 18. Precision-Recall curves of classifier models

Based on the generated Precision-Recall curves, we can clearly see that K-NN model gives us the best results and it has the highest area under the curve. Multinomial Naïve Bayes, Linear SVC, Logistic Regression could be also considered as a good model, but AdaBoost did not give us a shape as expected, based on the perfecrt classification curve shown in Fig 4.



	MultinomialNB	LinearSVC	Logistic Regression	K-NN	AdaBoost
Accuracy	0,83325	0,84072	0,81932	0,92559	0,78250
F-1 Score	0,82904	0,83938	0,82644	0,92702	0,76467
Precision	0,82838	0,83850	0,85530	0,93220	0,77109
Recall	0,83325	0,84072	0,81932	0,92559	0,78250
AUC	0,77462	0,79770	0,83924	0,93147	0,67783

Fig. 19. Performance metrics comparison of classifier models

Based on our performance metric comparison table, we can see that our K-NN classification model outperformed the other models for all of the metrics given above (Accuracy, F-1 Score, Precision, Recall and AUC). It is also seen that Multinomial Naïve Bayes and Linear SVC models perfomed almost the same. However, AdaBoost classifier was a poor model when compared to others.

V. CONCLUSION

In light of the graphical representations of the most commonly used evaluation metrics, we can infer that Linear SVC is the most appropriate classifier model for our toxic

comment classification task. When compared to the other ones, it is clear that Linear SVC has the best scores for Accuracy, Precision, Recall, F1, and AUC values. As for the sexual predator identification task, we can see that K-NN outperforms the other classifier models since it has the best results for the same metrics. Once we checked the training and prediction times, K-NN became the slowest one due to dealing with neighborhood selection, whereas Logistic Regression is the fastest model. We also calculated hamming loss scores for each and every classifier and the more accuracy we had for our model, the less loss score we observed. All of the drawn graphs are consistent with each other. For the ROC and Precision-Recall curves, we checked our results based on Fig 3 and Fig 4. We believe that we have successfully implemented and combined the concept of toxicity classification and sexual predator identification in our work, by generating our classifier models following a series of Natural Language Process tasks. As a result, we are able to identify sexual predators for a given set of conversations, as well as we can highlight the abusive messages with the help of our toxic comment classifier.

VI. Future Work

In the current work, we could not concentrate on image data, due to time limitations. Instead, we started with the text type of dataset. However, most of the Child Sexual Abuse Materials are images and media files. Hence, image processing algorithms could be inserted into our work for the betterment of the predator identification task. Then, a more in-depth version of pre-filtering and text processing activities could be done and new abbreviations that recently became popular but not unofficial yet could be discovered. As a more common trend in the literature, Deep Learning algorithms could be tried to come up with better results. Lastly, in the upcoming revisions, it would be helpful to focus on the Turkish language, since there is no sufficient number of academic research studies in the domain of preventing online child abuse through Machine Learning methodologies.

References

[1] https://www.statista.com/statistics/265147/number-of-worldwide-internet-users-by-region/

[2] C. Azzopardi, R. Eirich, C. L. Rash, S. MacDonald, S. Medigan, A meta-analysis of the prevalence of child sexual abuse disclosure in forensic settings, 2018, Elsevier.

[3] https://www.ecpat.org/what-we-do/online-child-sexual-exploitation/

[4] C. Amrit, T. Paauw, R. Aly, M. Lavric, Identifying child abuse through text mining and machine learning, Expert Systems with Applications, 2017.

[5] H. Adamsson, Classification of illegal advertisement working with imbalanced class distributions using machine learning, DiVA, 2017.

[6] A. Kontosthathis, L. Edwards, A. Leatherman, Text mining and cybercrime, Wiley Online Library, 2010.

[7] I. McGhee, J. Bayzick, A. Kontostathis, L. Edwards, A. McBride, E. Jakubowski, Perverted Justice: Learning to identify internet sexual predation, International Journal of Electronic Commerce 15, 3, 103-122, 2011.

[8] G. Inches, F. Crestani, Overview of the international sexual predator identification competition at PAN-2012, CLEF, 2012.

[9] https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

[10] https://analyticsindiamag.com/beginners-guide-to-understanding-roc-curve-how-to-find-the-perfect-probability-threshold/

[11] https://medium.com/@douglaspsteen/precision-recall-curves-d32e5b290248

[12] https://scikit-learn.org/stable/modules/cross_validation.html

[13] C. Morris, G. Hirst, Identifying online sexual predators by SVM classification with lexical and behavioral features, CLEF, 2012.

[14] M. A. Saif, A. N. Medvedev, M. A. Medvedev, T. Atanasova, Classification of online toxic comments using the logistic regression and neural networks models, AIP Conference Proceedings, 2018.

[15] M. Ibrahim, M. Torki, N. El-Makky, Imbalanced toxic comments classification using data augmentation and deep learning, ICMLA, 2018.

[16] C. Cardei, T. Rebedea, Detecting sexual predators in chats using behavioural features and imbalanced learning, Cambridge University Press, 2015

[17] M. Ebrahimi, C. Y. Suen, O. Ormandijeva, A. Krzyzak, Recognizing predatory chat documents using semi-supervised anomaly detection, Society for Imaging Science and Technology, 2016.

[18] H. J. Escalante, E. Villatoro-Tello, A. Ju´arez, L. Villaseñor, M. Montes-y-G´omez, Sexual predator detection in chats with chained classifiers, Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2013.

[19] J. I. Rodríguez, S. R. Durán, D. Díaz-López, J. Pastor-Galindo, F. G. Mármol, C3-Sex: A conversational agent to detect online sex offenders, Electronics, 2020.

[20] M. W. RahmanMiah, J. Yearwood, S. Kulkarni, Detection of child exploiting chats from a mixed chat dataset as a text classification task, Australasian Language Technology Association Workshop, 2011.

Automated Biometrical Fingerprint Recognition Scheme using Synthesized Images

Erdal Alimovski
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-0909-2047

Jawad Rasheed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-3761-1641

***Abstract*—The evolution of digitization has engulfed various methods of forensic sciences, such as fingerprint detection, recognition or recovery of partial prints. Prior to computerization, huge fingerprint repositories were manually maintained and involve humans for classification. But the advent of artificial intelligence-based tools performs the fingerprint recognition much faster and easier. Therefore, this study proposed CNN-based deep learning technique to extract effective features from ridges and valleys of skin impression for accurate recognition. The experimental work is based on FVC2020 dataset to train and test the proposed model. Moreover, ResNet50 framework is also tested on this dataset, and results shows that proposed model achieved an accuracy of 81.25%, whereas ResNet50 attained 79.25% accuracy. Furthermore, the incorporation of convolutional auto-encoder (CAE) based model for enhancing the dataset by generating synthetic fingerprint images, improved the recognition accuracy of proposed CNN-based model to 86.0%.**

Keywords— Fingerprint, deep learning, convolutional neural network, convolutional auto-encoders

I. INTRODUCTION

With growing demand for security all around the world, biometrics systems become the necessity of institutional life. Biometrics is a science that describes people in terms of physiological and behavioral terms. Physiological features cover characteristics such as palmprint, iris, fingerprint, facial part, while behavioral features include gait, gestures, signature and voice. The advancement in computerization has eased the usage of biometrics technology, thus it is widely adopted in vicinities of airports, banks, schools, or official buildings. Moreover, smart devices have also incorporated biometric features to secure the devices from intruders [1] [2].

Fingerprints are one of the most preferred solutions to authenticate people in biometric technology. A fingerprint is the presentment of the epidermis of a finger: it includes a sample of intermittent edges and valleys [3]. Each person has a matchless fingerprint which is mostly composed of few components such as ridges, grooves, and direction of lines. Ridges consist of three fundamental patterns namely, arch, loop and whorl. In general, patterns of fingerprint are determined by features such as ridge, minutiae and spots. Despite of the fact that fingerprints are distinct and highly distinguishable, but the performance accuracy of fingerprint recognition systems are associated with factors like quality of the image and applied matching algorithm [4].

Machine learning (ML) is a field of computer science that uses classification algorithms to identify patterns in large data for effective possible prediction. Beside ML tools, scientists have widely used Deep learning (DL) techniques, based on complex structures created with hierarchical modules to learn data representations, for tasks like image classification [5], segmentation [6], text detection [7], fingerprint recognition [8], face recognition [9], object detection [10], and fault prediction [11].

DL-based models require huge amount of data for proper training and effective prediction, however, very few and limited datasets related to fingerprint images are publicly available. Thus, this study exploited CAE to generate synthesized fingerprint images that are later used to train and generalize the proposed CNN model. The proposed CNN-based classification model is then evaluated on test set for accurate fingerprint recognition task. Moreover, CNN-based ResNet50 architecture is also implemented for comparative analysis. In addition to this, the effect of data synthesis on classification models is also analyzed.

The organization of this paper is as follows. In Section II, related work on fingerprint recognition task are provided. Section III describes dataset and methodology of the paper, while Section IV analysis the experimental results. Finally, paper ends with Section V by outlining the concluding remarks.

II. RELATED WORK

Finger recognition is key research area since last decade. Computer vision scientists have proposed various artificial intelligence based tools for fast and accurate finger classification, such as Stojanovic et al. [12] proposed DL-based framework for fingerprint recognition using CNN to extract the core of fingerprint Region of Interest (RoI). Experiments were performed in two variations with and without Gaussian noise. The proposed model obtained promising results while tested on FVC2002 dataset. Similarly, Shrein [13] suggested CNN-based Lenet-5 model for fingerprint image classification. Moreover, some image pre-processing techniques were applied to boost the performance while sufficiently reducing the training time. Authors evaluated the model using NIST-DB4 dataset to get an accuracy of 95.9%.

Darlow et al. [14] proposed deep CNN-based minutia extraction network (MENet), that consists of five convolutional layers, followed by 2 fully connected layers with softmax activation function. Later, some post-processing is performed to identify minutiae locations from the output of proposed MENet model. It is noted from experimental section that MENet outperformed previous minutiae extractor.

An end-to-end fingerprint recognition framework is proposed by Minaee et al. [15]. Before passing the PolyU dataset to proposed CNN-based network, affine transformation is employed to increase the data for each class. Thus, the data augmentation empowered the model to fit better that accomplished an accuracy of 95.7%. Likewise, Fanfeng et al. [16] proposed DL-based recognition framework to

recognize partial fingerprint images. To improve the recognition of partial fingerprints, authors used two loss functions, one for training while another for feature extraction. The model is trained and evaluated on NIST-DB4 and self-built NCUT-FR datasets that performed better than many existing techniques developed for partial fingerprint classification.

Authors in [17] blended a novel discriminative restricted Boltzmann machines (DRBM) and deep Boltzman machines (DBM) model to examine fingerprints. DBM is used to extract features from grayscale images, which are then feed to k-nearest-neighbors (kNN) classifier to analyze spoof forgeries. A novel method inspired from VGG-16 is proposed in [18] for aligned fingerprints matching. The model is tested using two publicly available datasets; NIST SD04, and NIST SD14. Besides various developed frameworks, there is still room to suggest more methods with higher performance accuracy.

III. Materials and Methods

The publicly available dataset is downloaded and later data augmentation technique is employed using CAE to produce synthesized images of fingerprints. Both datasets (original and augmented) are then used to train the proposed method. This section outlines the dataset and proposed methods in details.

A. Dataset

In this study FVC2002 [19] dataset is used as original dataset. FVC2020 dataset contains four classes and each class consist of 80 images. Hence the total number of images is 320. Except images in one class all the images were obtained from different sensors. Some of sample images of FVC2002 dataset are demonstrated in Fig. 1.

Fig. 1. Fingerprint samples of FVC2002 dataset.

B. Data Augmentation using Convolutional Auto-encoders

A data augmentation technique is employed to generate synthetic images using CAE. Auto-encoders are unsupervised machine learning algorithms that aims to rebuild the input data back using lower-dimensional representation [20]. Usually, auto-encoders consist of two parts: encoder, and decoder. The encoder has the ability to transform the input x into a representation h, also called as code, by using deterministic function as below

$$h = f_\theta(x) = \sigma(Wx + b) \quad (1)$$

with parameters $\theta = \{W, b\}$, where W is matrix's weight, b represents the bias of vector, while σ refers to the activation function. Similarly, reverse mapping of f is done by:

$$r = g_{\theta'}(h) = \sigma'(W'h + b') \quad (2)$$

While dealing with images, CAE can be more effective [20]. Generally, the structure of CAE is very similar to any other auto-encoder. Compared with normal auto-encoders, CAE uses convolutional and pooling layers to extract the features and minimizes the size of the input image. Therefore, a CAE architecture is proposed, considering the suggestion described in [21] and [22]. A detailed network topology of proposed CAE is outlined in Table I.

TABLE I. Network Topology of Proposed Convolutional Auto-Encoders (CAE)

CAE Architectural Details		
Layer type	***Output shape***	***No. of Prams***
input_1 (InputLayer)	None, 224, 224, 1	0
conv2d_1 (Conv2D)	None, 224, 224, 32	320
max_pooling2d_1	None, 112, 112, 32	0
conv2d_2 (Conv2D)	None, 112, 112, 64	18496
max_pooling2d_2	None, 56, 56, 64	0
conv2d_3 (Conv2D)	None, 56, 56, 128	73856
conv2d_4 (Conv2D)	None, 56, 56, 128	147584
up_sampling2d_1	None, 112, 112, 128	0
conv2d_5 (Conv2D)	None, 112, 112, 128	73792
up_sampling2d_2	None, 224, 224, 64	0
conv2d_6 (Conv2D)	None, 224, 224, 1	577

FVC2002 dataset is used to train the proposed model to generate synthetic images of fingerprints. The training is performed to reduce the differences between input and its reconstructed images. Fig. 2 shows few synthetic images of fingerprints generated by proposed CAE after 300 epochs. The augmented dataset constitutes of combination of original images and synthetic images, thus raised the dataset tally to 430 samples.

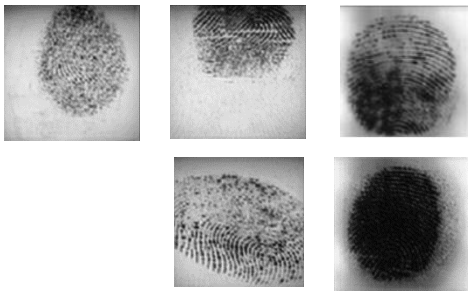


Fig. 2. Samples of generated sythetic fingerprint images.

C. Classification using Convolutional Neural Network

CNN is deep learning based powerful tool for recognizing local patterns in data instances [23]. As images are interrelated data, CNN based architectures are widely adopted for image classification task. CNNs detect local patterns by creating feature maps through conducting element wise multiplication using kernel. As data at hand can't be described with linear functions thus a nonlinear function, such as rectified linear unit (ReLU), is usually used to normalize the extracted feature map values. A pooling layer, which comes in variations of maximum, average and sum pooling, is also sometimes applied for dimensionality reduction that results in less network parameters. Later, the resultant is flattened, converted to one long vector. Afterwards, a regular feed forward backpropagation neural network methodology is applied. Usually, at end of architecture, a fully connected layer is

placed to calculate the probabilities for different classes using the features detected from prior steps. Finally, the network is back propagated based on the selected optimizer function for adjusting the weights.

The proposed CNN model consist of four convolutional layers, followed by flattening layer and fully connected layer. For this study, max pooling is utilized as it has been found more effective in previous studies [24]. It reduces the dimensions of the feature map while maintaining the most important identity values [25]. Between each convolutional layers, max pooling layers are placed. In order to eliminate overfitting among third convolutional layer and fully connected layers dropout layers are used. ReLU is used as activation function in each convolutional layer except last that uses Softmax. Adam is used as optimizer while batch size is set to 10.

In order to compare proposed CNN model, as a next phase we apply ResNet50 model. ResNet50 model is also trained and tested in both original and augmented datasets. ResNet50 architecture were performed with distinct hyper-parameters in order to reach the best accuracy. So, with following hyper-parameters achieved the best performance: optimizer Adam, learning rate 0.002, number of epoch 9 and batch size 10.

IV. Experimental Results

In this study, all the experiments were performed using Jupyter Notebook with Python (3.7) programming language. In addition, we used Tensorflow, Keras, Sklearn and Seaborn libraries.

As described earlier, original dataset and augmented datasets are used separately for experimental work. Note that, for each experiment 90% of both datasets were used for training and remaining 10% for testing. In order to analyze the effect of data augmentation and the performance of proposed CNN model, various performance metrics have been measured, such as accuracy, confusion matrix, precision, recall and f1-score, using Equations (3)-(6)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - score = \frac{2 * (Recall + Precision)}{(Recall + Precision)} \tag{6}$$

where TP refers to true positive, TN refers to true negative, FP refers to false positive and FN is false negative.

The proposed CNN model is first trained and evaluated on original dataset, which secured an accuracy of 81.25%. The accuracy and loss curves of model on original dataset is depicted in Fig. 3(a)-(b), respectively. During experiments, the model is trained with 5, 9, 10 and 15 number of epochs, respectively, thus it accomplished the highest accuracy epochs were set to 9. Moreover, experiments were repeated with

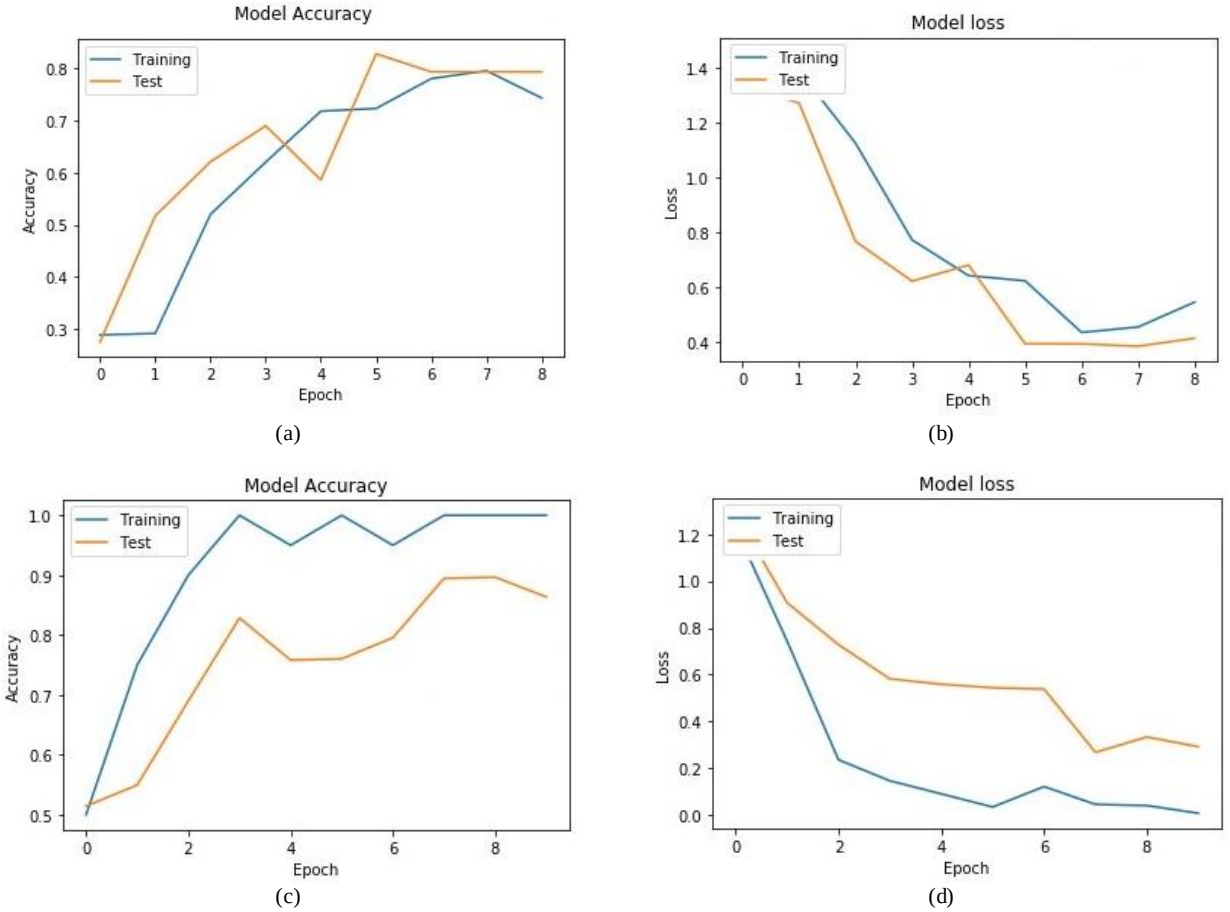
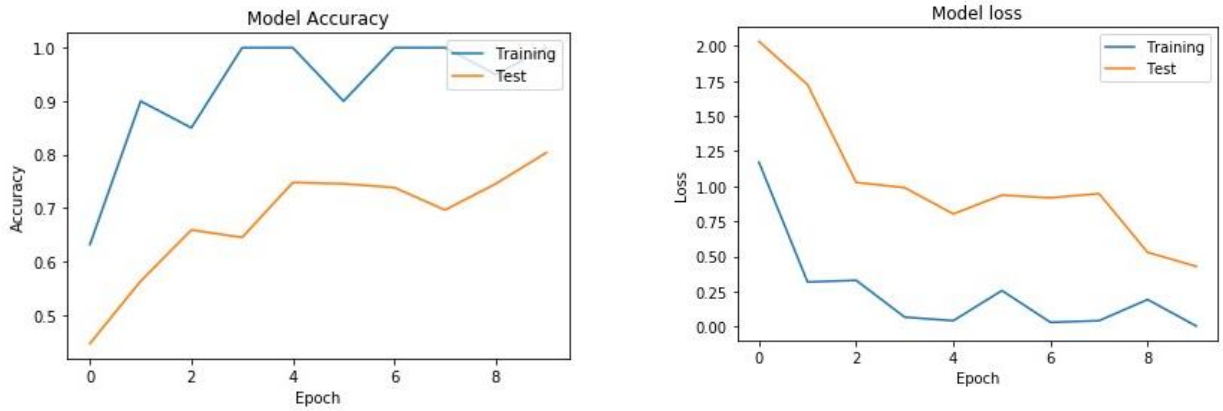


Fig. 3. Training and loss curves of proposed model, (a) accuracy on original dataset, (b) loss on original dataset, (c) accuracy on augmented dataset, and (d) loss on augmented dataset.

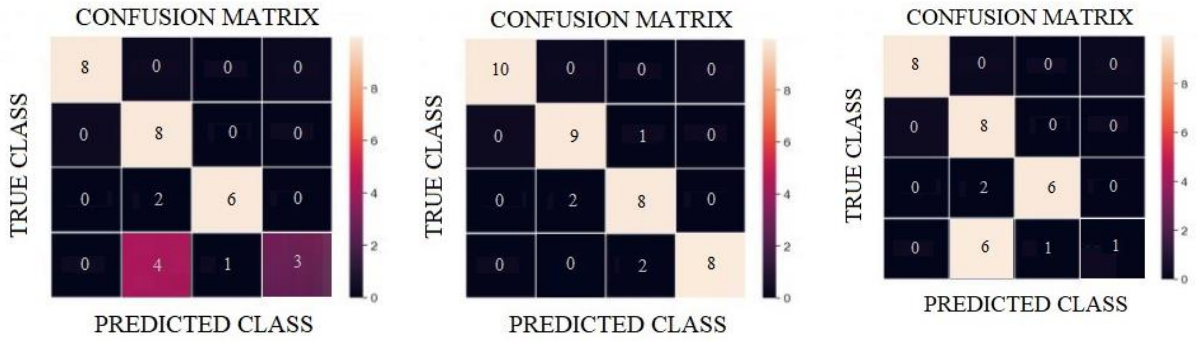


(a) (b)

Fig. 4. Training and loss curves of ResNet50 on original dataset.

TABLE II. COMPARATIVE ANALYSIS OF PROPOSED MODEL WITH RESNET50

Performance Analysis					
Method	***Dataset***	***Accuracy***	***Precision***	***Recall***	***F1-score***
Proposed	Original	0.81	0.72	0.78	0.75
	Augmented	0.86	0.78	0.83	0.80
ResNet50	Original	0.79	0.70	0.75	0.72



(a) (b) (c)

Fig. 5. Confusion matrix of (a) proposed model on original dataset, (b) proposed model on augmented dataset, and (c) ResNet50 on original dataset.

different combinations of convolutional layers without pooling layers, but the network's performance degraded. In addition to this, experiments were repeated for augmented dataset. The proposed classification CNN-based model is trained on augmented dataset that achieved accuracy of 86% on respective test set. The accuracy and loss curves for augmented dataset is shown in Figure 3(c)-(d), respectively. Evidently, the experimental results shows that the incorporation of convolutional auto-encoders for data augmentation has significantly increased the performance accuracy of proposed recognition network.

Besides the proposed classification model, a state-of-the-art DL based image classification framework, known as ResNet50 is also trained and evaluated for comparative analysis. For the problem at hand, fingerprint recognition task, ResNet50 performed less accurately when compared with proposed model of this study. ResNet50 model hardly attained an accuracy of 79.25% when trained and evaluated on original dataset, as shown in Figure 4(a), while Figure 4(b) depicts its loss curve. A detailed comparative analysis is presented in Table II.

For better visualization, confusion matrix for each experiment is also composed. It can be observed from Figure 5(a), which depicts the confusion matrix of proposed model when trained and test on original dataset that first two classes are correctly distinguished but 2 fingerprint images of Class-3 are misclassified. Similarly, half of the images in Class-4 are also misclassified by proposed model. On the other hand, with the addition of data augmentation technique, the model performed better while distinguishing Class-4 samples as shown in Figure 5(b). The ResNet50 performed worse than others while recognizing Class-3 and Class-4 instances as illustrated in Figure 5(c). It wrongly classified seven fingerprint images of Class-4, whereas only one image is recognized correctly. Thus, the experimental results shows

that the proposed model outperformed other models when trained on augmented dataset.

V. Conclusion

In this paper we proposed a CNN based model to perform fingerprint recognition task. At first, the proposed model is trained on publicly available FVC2002 dataset, which is then evaluated on remaining 10% set that achieved an accuracy of 81.25%. As DL models are data hunger, thus this study also proposed a model to increase the fingerprint images samples. Therefore, the dataset is enhanced using convolutional auto-encoder that generated synthetic fingerprint images. The proposed model is then trained on augmented dataset, which contains original images as well synthetized images. Evidently, the synthesized images technique increased the overall performance of classification network by 4.75%. Moreover, ResNet50 framework is also implemented and trained on downloaded dataset, which attained an accuracy of 79.25%, thus performed worse than the proposed model. Even though, the proposed model with data augmentation approach attained promising results, but still there is room for further enhancement which can be done in future works by training on other datasets.

References

[1] D. Bouchaffra and A. Amira, "Structural hidden Markov models for biometrics: Fusion of face and fingerprint," *Pattern Recognition*, vol. 41, no. 3, pp. 852–867, 2008, doi: 10.1016/j.patcog.2007.06.033.

[2] M. D. Security, "A Mobile Biometric System-," no. April, pp. 13–19, 2010.

[3] E.- Book, S. Vishwas, and L. D. Resolution, "E-Handbook Sabka Vishwas (Legacy Dispute Resolution) Scheme , 2019," vol. 2019, 2019.

[4] W. J. Wong and S. H. Lai, "Multi-task CNN for restoring corrupted fingerprint images," *Pattern Recognition*, vol. 101, p. 107203, 2020, doi: 10.1016/j.patcog.2020.107203.

[5] J. Rasheed, H. B. Dogru, and A. Jamil, "Turkish Text Detection System from Videos Using Machine Learning and Deep Learning Techniques," in *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, Aug. 2020, pp. 116–120, doi: 10.1109/DSMP47368.2020.9204036.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "A deep convolutional encoder-decoder architecture for image segmentation}," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] J. Rasheed, A. Jamil, H. B. Dogru, S. Tilki, and M. Yesiltepe, "A Deep Learning-based Method for Turkish Text Detection from Videos," in *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, Nov. 2019, pp. 935–939, doi: 10.23919/ELECO47770.2019.8990633.

[8] H.-R. Su, K.-Y. Chen, W. J. Wong, and S.-H. Lai, "High-Resolution Fingerprint Recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2017*, pp. 2057–2061, 2017, [Online]. Available: https://ieeexplore.ieee.org/document/7952518/authors#authors.

[9] J. Rasheed, E. Alimovski, A. Rasheed, Y. Sirin, A. Jamil, and M. Yesiltepe, "Effects of Glow Data Augmentation on Face Recognition System based on Deep Learning," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2020, pp. 1–5, doi: 10.1109/HORA49412.2020.9152900.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] H. Rashid, E. Khalaji, J. Rasheed, and C. Batunlu, "Fault Prediction of Wind Turbine Gearbox Based on SCADA Data and Machine Learning," in *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*, Sep. 2020, pp. 391–395, doi: 10.1109/ACIT49673.2020.9208884.

[12] B. Stojanovic, O. Marques, A. Neskovic, and S. Puzovic, "Fingerprint ROI segmentation based on deep learning," *24th Telecommunications Forum, TELFOR 2016*, pp. 5–8, 2017, doi: 10.1109/TELFOR.2016.7818799.

[13] J. M. Shrein, "Fingerprint classification using convolutional neural networks and ridge orientation images," *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings*, vol. 2018-Janua, pp. 1–8, 2018, doi: 10.1109/SSCI.2017.8285375.

[14] L. N. Darlow and B. Rosman, "Fingerprint minutiae extraction using deep learning," *IEEE International Joint Conference on Biometrics, IJCB 2017*, vol. 2018-Janua, pp. 22–30, 2018, doi: 10.1109/BTAS.2017.8272678.

[15] S. Minaee, E. Azimi, and A. Abdolrashidi, "FingerNet: Pushing the limits of fingerprint recognition using convolutional neural network," *arXiv*, 2019.

[16] F. Zeng, S. Hu, and K. Xiao, "Research on partial fingerprint recognition algorithm based on deep learning," *Neural Computing and Applications*, vol. 31, no. 9, pp. 4789–4798, 2019, doi: 10.1007/s00521-018-3609-8.

[17] D. M. Uliyan, S. Sadeghi, and H. A. Jalab, "Anti-spoofing method for fingerprint recognition using patch based deep learning machine," *Engineering Science and Technology, an International Journal*, vol. 23, no. 2, pp. 264–273, 2020, doi: 10.1016/j.jestch.2019.06.005.

[18] Y. Liu, B. Zhou, C. Han, T. Guo, and J. Qin, "A novel method based on deep learning for aligned fingerprints matching," *Applied Intelligence*, vol. 50, no. 2, pp. 397–416, 2020, doi: 10.1007/s10489-019-01530-4.

[19] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," *Proceedings - International Conference on Pattern Recognition*, vol. 16, no. 3, pp. 811–814, 2002, doi: 10.1109/icpr.2002.1048144.

[20] K. G. Kim, "Deep learning book review," *Nature*, vol. 29, no. 7553, pp. 1–73, 2019.

[21] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6791 LNCS, no. PART 1, pp. 52–59, 2011, doi: 10.1007/978-3-642-21735-7_7.

[22] F. Chollet, "Building Autoencoders in Keras," 2016. https://blog.keras.io/building-autoencoders-in-keras.html (accessed Dec. 15, 2020).

[23] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Cnn实际训练的," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2012.

[24] D. C. Cires, U. Meier, J. Masci, and L. M. Gambardella, "IJCAI11-210.pdf," *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence Flexible*, pp. 1237–1242, 2003, [Online]. Available: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI11/paper/viewFile/3098/3425.

[25] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *ArXiv e-prints*, no. November, 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.

Remedial Directed Topic Map On Personalized Scaffolding Adaptive Learning Management System

Yulia Wahyuningsih
Department of Informatics,
Institut Teknologi Sepuluh Nopember
yulia.19051@mhs.its.ac.id

Arif Djunaidy
Department of Information Systems,
Institut Teknologi Sepuluh Nopember
adjunaidy@is.its.ac.id

Daniel Oranova Siahaan
Department of Informatics, Institut
Teknologi Sepuluh Nopember
danielos@cs.its.ac.id

Abstract*— The problem arises behind the mastery learning model, that is some students fail to achieve the minimum standard of completeness. The existing solutions have been ineffective in helping the study plan. The existing remedial method does not consider the previous formative test answers, but all material that has been mastered must also be tested. This remedial method did not give students more time to learn misconceptions or the material they had not mastered yet. We propose a framework for building a scaffold on a remedial learning path that is personal and adaptive to the material needs that students must learn. Initial exploration in this study aims to look the possibility in implement the proposed framework by utilizing machine learning on mastery module using neural network and knowledge-based recommendation technique to process the input of student's answers, domain expert's topic maps, and the relationship among answers to build a computer-based scaffold. This novelty of exercise learning path could be an effective remedial path for the failed student to learn just according to topics have not mastered.

Keywords—Scaffolding, Topic Map, Self-Regulated Mastery-Learning, Assessment-Based Learning, Machine Learning

I. INTRODUCTION

The COVID-19 pandemic has caused nearly all schools worldwide to close. More than 530,000 schools in Indonesia have closed, which mean that 68 million Indonesian students are forced to no longer be in class. These school closings adverse effects on academic learning because for a distance learning to be effective, it required new skills for both teachers and students. Regarding this pandemic, SDI argues that Indonesia needs a strategy to support better face-to-face teaching and improve the quality of distance learning, to restore and improve its human resources [1]. The World Bank stated that drastic action was needed to support Indonesian students' learning as part of the ongoing recovery and re-opening process. Some actions are recommended related to learning recovery to prevent the permanent impact of "learning loss" on students. The recommendations include 1. using formative classroom assessment to identify learning losses; 2. integrated use of blended-learning and technology to ensure that all students receive the support they need to become effective learners. The United Nations mentioned that preventing the learning loss from becoming a generational disaster needs to be a top priority to protect millions of students' rights and promote economic progress, sustainable development, and lasting peace [2].

In face-to-face learning, it is easier for teachers to assist students in need. Teachers need to provide scaffolding to students in need during learning activities to improve their understanding, affective, and skills [3]. Unfortunately, learning during the pandemic lacks personal assistance from teachers to students who do not understand the material. It potentially causes problems, especially for students who need assistance and need an extraordinary approach to complement and prepare tools for students and teachers related to this problem. Research projects related to scaffolding proven to be effective in science learning are, for example in physics learning, which uses the PhET simulation (physics simulation portal) to increase student's conceptual understanding and independent learning [4]. Inquiry-based learning uses an agent-based framework where students can check their understanding of science and the concept of Computational Thinking (CT) by taking formative tests managed by a mentor agent. The mentor agent assesses students' responses to multiple-choice type questions and provides feedback on correct responses along with suggested learning resources pages to read if there is an incorrect response [5]. Although this research has provided feedback for students' mistakes, it is limited to recommendations for subject matter that must be read. Simultaneously, physics lessons' characteristics require problem-solving skills that cannot be met in these studies. Problem-solving skills in physics lessons require continuous practice compared to mere theoretical reading.

The solution approach we offer is adapted to the learning process in Indonesia, which applies mastery learning model. Each student must achieve a minimum completeness score in a unit before entering the next study unit in mastery learning. There are always students who fail to achieve the minimum completeness score; unfortunately, there is no solution ineffective remedial handling. These findings indicate that the scaffolding approach can be considered for application to improve students' understanding of concepts and useful. This work plan explicitly adds a scaffolding assessment-based learning process for students who do not pass the complete learning process. This process is an opportunity to correct misconceptions or material that has not been understood to help students effectively achieve mastery adaptively according to their needs. Fig.1 is the recommendation for the mastery learning diagram presented by Thomas, 2008 [6].

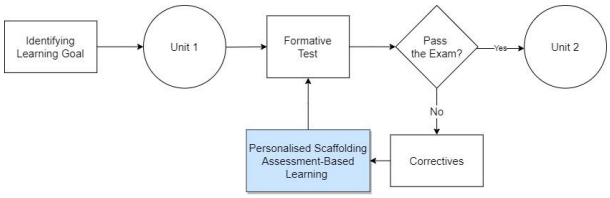


Figure 1 Mastery with Personalized Scaffolding

This study is a proactive step in responding to the World Bank's recommendation to Indonesia regarding the re-opening of schools with a proposed framework that implements computer-based scaffolding in flipped classrooms by applying assessment-based learning. Assessment-based learning method deserves to be taken considering that the assessment has a "test effect", useful for long-term retention, and practice questions can increase understanding of the subject of physics [7]. The Assessment-based learning method is the guidelines for school opening policymakers, namely schools need to conduct an assessment, a computer-based diagnostic assessment [8]; It is a rational basis that assessment-based learning is proven to increase learning completeness. This paper's composition is as follows: Section 2 contains previous research work, followed by a proposed method in Section 3, initial investigation is in Section followed by discussion in Section 5, and Section 6 is a conclusion.

II. PREVIOUS RESEARCH WORK

In preparing for returning to class after the pandemic, it is necessary to pay attention to the zone proximal development (ZPD) referring to the area between what one knows and what one does not know during learning. In this area, one needs guidance from other knowledgeable people, so that in the end, they will be able to develop their understanding [8]. The scaffolding strategy has been integrated into e-learning. One of the studies that identify and determine the attributes of a mixed learning environment that supports students' self-regulated learning ability found scaffolding to be one of the seven key attributes (authenticity, personalization, learner control, scaffolding, interaction, reflection, and calibration) [9]. Meanwhile, recent research on adaptive scaffolding is an efficient strategy to support and improve student performance when learning new concepts, and guide them progressively towards a better understanding when answering questions [10]. Other scaffolding research uses breaking a problem into smaller items or directing students to other problems similar to the current problem and helping students through the new problem. When students incorrectly answer the main questions, scaffolding items are activated according to the following four types: (a) procedural, (b) conceptual, (c) metacognitive, and (d) strategic [11].

The study's selection of objects also refers to the World Bank report, which focuses explicitly on the pandemic's Indonesian PISA test results. It appears that science is one of the subjects that need special attention. Learning science does have its peculiarities, for example, in Physics, apart from requiring practice in problem-solving skills, Physics have a close relationship among knowledge topics. The challenge in finding out which concepts are less mastered by students causes them to fail to achieve learning completeness. When students learn to build new physics concepts, they must remember previous concepts and increase students' cognitive load and impact learning outcomes. A concept map may be a solution to the problems because a concept map represents various concepts. Its relationships can also organize students' cognitive structures and are expected to encourage in-depth, integrated knowledge. Concept mapping is used as a scaffold to support adequate information problem-solving— using questioning as a facilitator for constructing students' previous knowledge structure. A domain expert will provide the initial topic map because not all students know how to benefit from a concept map [12]

Research related to machine learning is more common nowadays. Some research uses machine learning algorithms to deal with several problems, such as predicting student performance [13], predicting student performance based on learning styles [14]. So, in the future, when implementing this proposed framework, it will utilize the development of machine learning in e-learning [15] [16]. The evaluation tool for this proposed framework will emphasize the level of satisfaction and user preferences, referring to the weakness of the existing recommendation system's (RS) evaluation, which is stated in a review that RS in e-learning is based on students; however, system evaluation still focuses on measuring algorithm accuracy through Mean, Precision, Recall and F-measure rather than evaluating its impact on user satisfaction and preference levels [17].

This study offers a Remedial Directed Topic Map framework on Personalized Scaffolding Adaptive implemented in the Learning Management System (PSALMS) to be a novelty of remedial learning path. The last published research provides the exercises learning based on a sequence of students' previous answers, but they do not consider the internal connection between knowledge concepts [18]. A remedial directed topic map will be developed by paying attention to students' formative test answers, expert's initial topic map as the internal connection between knowledge concepts and the failed topic related question. This personalised remedial learning path that effectively helps students achieve learning mastery. By applying the state-of-the-art computer-based scaffolding to giving practice questions, it is expected that failed students receive the learning support they need to achieve mastery learning and increase resilience in self-regulated learning.

III. PROPOSED METHOD

Our study selected the Flipped-learning to support the re-opening of face-to-face classes after the pandemic and the World Bank direction regarding blended learning development. This framework will be applied to the learning management system (LMS) because of the wide use of LMS (Moodle, canvas, another open-source LMS) [19]. This paper is a step development of initial research idea [20]. Fig. 2 is the Flipped Classroom method proposed to adjust to the "new normal" learning conditions.

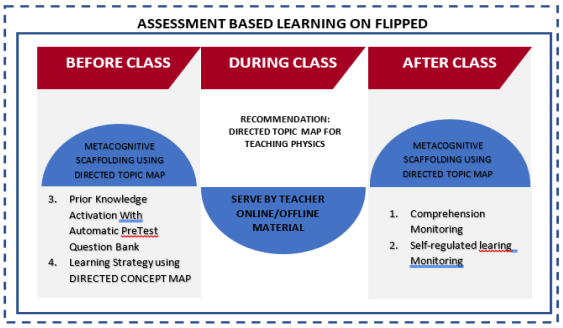


Figure 2 Assessment-based Learning on Flipped the Classroom

In the new normal state at the re-opening school, students take turns attending classes, so that it's needed to provide a system that can cover the three main learning activities. Namely, activities are before students into the classroom, the second part is when the students are in the class during the face-to-face learning, and the third part is the activities after the face-to-face class ends.

This initial research focuses on the after-class activities from our general framework PSALMS research ideas [19]. This study's primary purpose is to provide practical solutions to students' remedial problem based on real-world problems (see Table 1). The result shows that not all students exceed the minimum completeness score standard. It is necessary to develop practical computer-based tools to help students personally prepare for remedial tests. Therefore, we proposed 2 (two) main stages with three modules: The Topic module on the first stage, mastery module, and scaffolding module on the second stage, Fig. 3.

A. Topic Module and Scaffolding Module

The first stage is a Topic module, left side of Fig 3. There are two inputs at this stage: a topic map (topic knowledge) based on information from domain experts and the single-labelled teacher's questions. The domain expert's topic map is used at the topic module to examine teacher's questions based on various inputs, such as keywords, certain meta-keywords related to the questions to identify whether the questions need to be assigned and classified multi-label Topic. This process involves cleaning and preliminary processing the available questions in the dataset for further processing using a classification algorithm; the questions are classified and labelled appropriately if they are multi-label questions and stored back in the database. At first, each question is manually classified on one Topic by the Teacher.

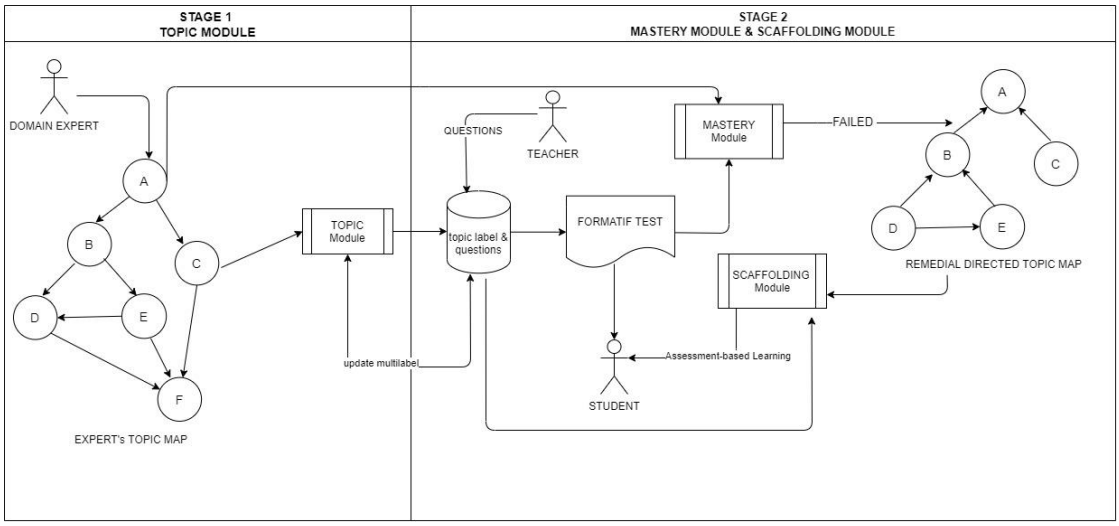


Figure 3 Remedial Directed Topic Map on Personalized Scaffolding Adaptive Learning Management System

A second stage, there are two modules, namely the scaffolding module and the mastery module. The scaffolding module helps provide recommendations based on question queries provided to students as scaffolding for students preparing to re-test. The scaffolding module extracts information from the mastery module and the question database to provide coherent questions according to the personal remedial path. First, the personalized question request is forwarded to run a recommendation algorithm. The association rule algorithm, Apriori, is used to mine and extract the appropriate problem patterns from the repository of questions and then test the relationship's accuracy with the questions. To validate the data of the identified questions will using recall, f1-score, and precision using matrix factorization to observe the questions' suitability and estimate the recommendation result's accuracy.

B. Mastery Module

The focus of this paper is the initial investigation related to the mastery module. The illustration in Fig 4 shows the flow of the mastery module process. The purpose of the mastery module is to extract students who do not meet the minimum completeness standards taken from the data of the formative test. The formative test is a test given at the end of a topic unit.

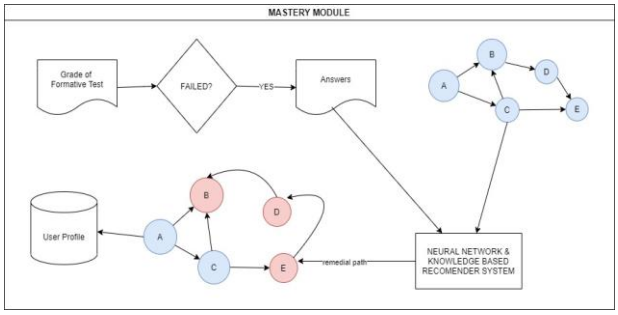


Figure 4 Mastery Module

After obtaining the data, a remedial learning path is designed for students who failed based on the wrong and correct answers from the formative test results. These answers will be correlated into an expert topic map to obtain any topics that had not been mastered, causing this student failed using knowledge-based technique [18]. This personal information remedial learning path may be constructed using neural network algorithms [21] . The user information repository is formed with the students' topic map profiles and stored in a database to be used to the scaffolding module.

Therefore, the PSALMS framework is expected to provide personal recommendations for self-regulated learning and gradually overcome material misunderstandings until mastery. The objectives of this proposed work are, (i) update for multilabel classification of practice questions under the related topics to remedial students that have not been mastered; (ii) based on the knowledge topic that students did not master and the need for practice questions on a particular topic, a new query of practice question as a personalized scaffold was formed; and (iii) by formulating classification and personalization, it allows personal recommendations that are accurately diagnosed and given to students to reinforce self-regulated learning.

General proposed framework is, if the student score fails to achieve the minimum completeness score, their test answers will be applied to the expert topic map to building an effective remedial learning path acts as an adaptive scaffolding. It will be personal guidance to direct students to retrace the previous Topic's misconceptions and using practice questions only on material that had not been mastered coherently as a form of assessment-based learning.

IV. INITIAL INVESTIGATION

In this paper, the researchers use real-world data as a study case. The data were taken from the formative test results for Physics, Newton's laws, which was conducted online with Moodle as its LMS. The number of students in this class was 23 students; the test questions were 10 (ten) questions and were given 40 minutes to finish. Subject materials and practice questions were delivered synchronously as much as 4 x 40 minutes and asynchronously within two weeks.

The initial topic map, obtained from experts, and to build a sequence on the topic map used a relation table as in Table 1. Table 1 and a collection of topics on Newton's laws indicate the topic map's order is obtained as in Fig 5. The test result obtained by 23 students on Newton's laws can be seen in Table 2.

Fig 6 is the position of the question topic on the initial topic map by the expert. The figure shows that the formative test question topics are spread out on the domain expert's initial topic map. Table 3 is the initialization of the Topic for each question made by the Teacher.

Table 1 Example of PERT table to built relation between the Topic

No	Present State	About	Previous State
1	equilibrium	equilibrium, in physics, the condition of a system when neither its state of motion nor its internal energy state tends to change with time	-
2	net force	The net force is defined as is the sum of all the forces acting on an object. Net force can accelerate a mass. Some other force acts on a body either at rest or motion. The net force is a term used in a system when there is a significant number of forces.	equilibrium
3	inertial reference frame	Within the realm of Newtonian mechanics, an inertial frame of reference, or inertial reference frame, is one in which Newton's first law of motion is valid. An inertial frame of reference is one in which the motion of a particle not subject to forces is in a straight line at a constant speed.	force
4	inertia	Inertia is the resistance of any physical object to any change in its velocity. This includes changes to the object's speed or direction of motion. An aspect of this property is the tendency of objects to keep moving in a straight line 4at a constant speed when no forces act upon them.	First Newton Law

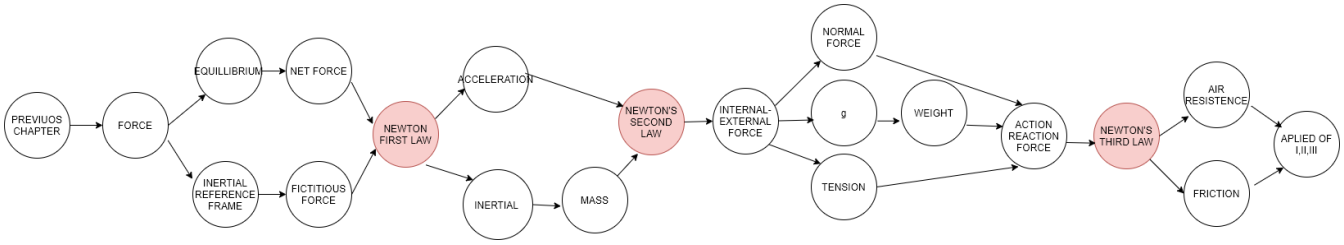


Figure 5 Topic Map Dynamic of Motion

Table 2 Students Grade and Answer Form Formative Test

Student	Grade	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
S01	30	1	1	0	0	1	0	0	0	0	0
S02	30	1	1	0	0	1	0	0	0	0	0
S03	40	1	1	1	0	1	0	0	0	0	0
S04	50	1	1	0	0	1	0	1	0	1	0
S05	50	0	1	0	1	0	1	1	0	0	1
S06	60	1	1	1	0	0	0	1	0	1	1
S07	60	1	0	0	0	1	1	1	0	1	1
S08	60	0	1	0	1	1	0	1	0	1	1
S09	60	1	1	1	1	0	1	0	0	1	0
S10	70	1	1	1	1	0	1	0	0	1	1
S11	70	1	1	1	1	0	1	0	0	1	1
S12	70	1	1	1	1	1	0	1	0	0	1

Student	Grade	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
S13	70	1	1	1	1	1	0	0	1	1	0
S14	70	1	1	0	1	1	1	1	0	0	1
S15	70	0	1	1	1	1	1	1	1	0	0
S16	70	0	1	1	1	1	0	1	0	1	1
S17	80	1	1	1	1	0	1	1	0	1	1
S18	80	1	1	1	1	1	0	1	0	1	1
S19	80	1	1	1	0	1	0	1	1	1	1
S20	80	1	1	0	1	1	1	1	1	0	1
S21	90	1	1	0	1	1	1	1	1	1	1
S22	100	1	1	1	1	1	1	1	1	1	1
S23	100	1	1	1	1	1	1	1	1	1	1

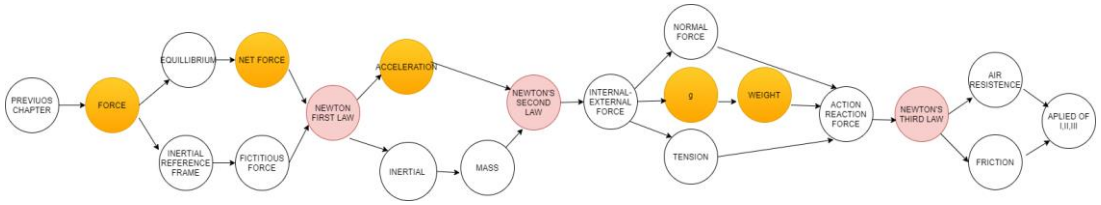


Figure 6 Question Topic Position on Initial Topic Map Based on Single Topic by Teacher

Table 3 Single Topic defined by Teacher

	T0 (Previous)	T1 (force)	T2 (Net Force)	T3 (Acceleration)	T4 (Gravitation)	T5 (Weight)
Q1	0	0	0	0	0	1
Q2	0	0	0	0	1	0
Q3	0	0	0	1	0	0
Q4	0	0	1	0	0	0
Q5	0	0	1	0	0	0
Q6	0	1	0	0	0	0
Q7	0	0	1	0	0	0
Q8	0	0	0	1	0	0
Q9	1	0	0	0	0	0
Q10	1	0	0	0	0	0

V. DISCUSSIONS FOR FURTHER WORKS

This Section will discuss several aspects related to the initial investigation of this research. Based on data are presented in Table 2, If the complete score is 70, it can be observed that students who fail to achieve the minimum completeness score in the table. For example, namely: Student 1 (S01) with a score of 30, Student 4 (S04) with a score of 50, and Student 6 (S06) with a score of 60, see Table 4. Table 3, which refers to all the topics of questions, Table 4 shows which topics cause these students to fail. For example, Question 4, Question 6, and Question 8; it turned out that these three students made the same mistake on the question items. Meanwhile, suppose we pay attention to the Topic (concept) on that question from Table 3. In that case, we see that these questions contain the Topic Net Force for Question 4, the topic force for Question 6, and topic acceleration for Question 8, which can be seen in Table 4.

Table 4 Student's Failed Topics

Question	T0	T1	T2	T3	T4	T5
Q4	0	0	1	0	0	0
Q6	0	1	0	0	0	0
Q8	0	0	0	1	0	0

In the initial topic map made by experts, we can observe that the sequence of lessons in this unit is Topic Force followed by Net Force then acceleration, as presented in Fig 7.

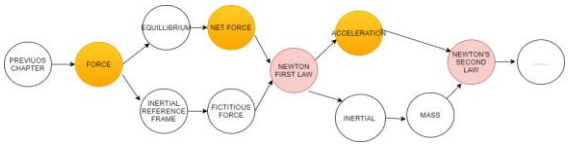


Figure 7 Position Failed Topic

Table 5 will observe another student, S09, S10 and S11 they have a similar answer. Poorly, S09 is a failed student. If we combine S09 answer with the topic map, we will know the topics which caused S09 failed.

Table 5 Example of one student who failed

Student	Grade	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
S9	60	1	1	1	1	0	1	0	0	1	0
S10	70	1	1	1	1	0	1	0	0	1	1
S11	70	1	1	1	1	0	1	0	0	1	1

Based on Table 2 and Fig. 6, we will notice, just two topics caused S09 to fail. Based on Fig.8, this proposed framework will be an effective scaffolding for the failed students because this remedial learning path directs forward to the Topic who mastered yet and gives them an adaptive question for fulfilling, they needed.

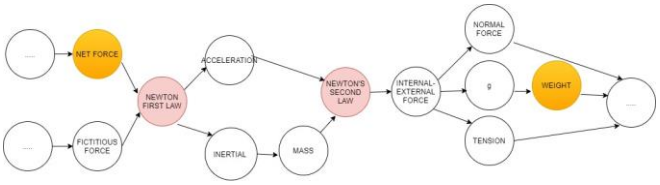


Figure 6 Personal Map for Student 9 (S09)

The inputs which are students answers, the relationship between the Topic and the questions as well as the initial topic map of the expert domain can retrace which material must be re-studied by students preparing for the formative remedial test so that students do not need to learn all the materials but focus on the material which causes them to fail in mastery.

We observed failed topics for students S05, S08 S15 in Table 6, especially on questions Q1, Q6 and Q8. The Topic for those questions, namely Weight, Force and acceleration.

Table 6 Not Matches Path

Student	Grade	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
S05	50	0	1	0	1	0	1	1	0	0	1
S08	60	0	1	0	1	1	0	1	0	1	1
S16	70	0	1	1	1	1	0	1	0	1	1

They have a similar path on Q1, Q3, Q6 and Q8; when S08 and S16 doing wrong but S05 correctly answered Q6 regarding Force. So too S16 correctly answered Q3 regarding Acceleration when S05 and S08 have a false answer.

This is something out of our thought before. There are several possibilities related to this problem, one of which is hidden relations between topics [22], to solve this problem we can try using Deep Neural Network who use of multiple layers in the network [23]

VI. CONCLUSION

In this paper, the preliminary investigations regarding PSALMS have been discussed. The discussion-section found that test answer, topic maps, and topic-labelled questions can be used to build an effective learning path remedy. The Failed Student will be directly forward to the Topics causing failed and learn with assessment-based learning. With the ideas presented as an initial investigation, it is expected to provide adaptive scaffolding solutions that can specifically recommend learning materials for a unit. The adaptive scaffold modelling approach with assessment-based learning using directed topic maps has advantages and challenges. The main challenge is developing the scaffolding model on the cognitive knowledge students have had comprehensively. We suggest a Deep Neural Network Algorithm solve the hidden correlation problem between Topics. The proposed Topic Module needs more investigation whether using a multi-label classification provided by the topic module can help with hidden relational concept problems with update the questions profiling. The proposed PSALMS model is expected to overcome preparation for returning to school after the pandemic, especially for remedial problems. Choosing a suitable machine learning algorithm for designing remedial directed topic maps is a basic challenge to solve.

References

[1] N. Yarrow, E. Masood and R. Afkar, "Estimates of COVID-19 Impacts on Learning and Earning in Indonesia: How to Turn the Tide," IBRD - IDA World Bank Group, Indonesia, August 2020.

[2] U. Nation, "Policy Brief:Education during COVID-19 and beyond," United Nation, Agust 2020.

[3] M. Y. G, B. L, d. J. T and Lazonder, "Scaffolding learning by modelling: The effects of partially worked-out models," *J. of research in science learning*, vol. 53(3) , pp. 502-523, 2016.

[4] E. E. e. al., "The Effect of Scaffolding Approach Assisted by PhET Simulation on Students' Conceptual Understanding and Students' Learning Independence in Physics," *J. Phys.: Conf. Ser. 1233 012036*, 2019.

[5] S. Basu, G. B. J. S. and Kinnebrew, "Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment," *User Model User-Adap Inter* , vol. 27, pp. 5-23, 2017.

[6] T. R. Guskey, "Formative Classroom Assessment and Benjamin S. Bloom:Theory, Research, and Implications," in *Annual Meeting of the American Educational Research Association*, Montreal, Canada. , April 2005.

[7] P. E. R. Cutri, "Ten Ways to Improve Learning Physics as Part of an Engineering Course," in *ASEE Confference & Exposition*, New Orland, LA, , 2016.

[8] M. F. Jumaat and Zaid, "A Model To Predict Students' Performance Based On Instructors' Pattern Of Metacognitive Scaffolding Through Data Mining Analysis," *Journal Of Theoretical And Applied Information Technology* , vol. Vol.98. No 21, November 2020.

[9] S. V. Laer and J. Elen, "In search of attributes that support self-regulation in blended learning environments," *Educ Inf Technol*, vol. 22, p. 1395–1454, 2016.

[10] J. D. P. K. S. e. a. Kukkonen, "Pre-service teacher's experiences of scaffolded learning in science through a computer supported collaborative inquiry," *Education and Information Technologie*, vol. 21, p. 349–371, 2016.

[11] A. Korhonen, S. Ruhalahti and M. Veermans, "The online learning process and scaffolding in student teacher's personal learning environments," *Education and Information Technologies*, vol. 24, p. 755–779, 2019.

[12] W.-W. A., L. N, W. D. and G. D, "Exploring student information problem solving behaviour using fine-grained concept map and search tool data," *Computers & Education*, vol. 145, 2020.

[13] Z. Iqbal, J. Qadir, A. N. Mian and F. Kamiran, "Machine Learning Based Student Grade Prediction: A Case Study," *arXiv:1708.08744v1*, 2017.

[14] H. T. Binh and B. T. Duy, "Predicting Students' performance based on Learning Style by using Artificial Neural Network," in *International Conference on Knowledge and Systems Engineering (KSE)*, Hue, Vietnam, 2017.

[15] D. O. S. a. C. F. S. F. Kusuma, "Automatic Question Generation In Education Domain Based On Ontology," in *International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Surabaya, Indonesia, 2020.

[16] S. Kusuma, D. Siahaan and U. Yuhana, "Automatic Indonesia's Questions Classification Based On Bloom's Taxonomy Using Natural Language Processing," in *International Conference on Information Technology Systems and Innovation (ICITSI)*, Bali, Indonesia, 2015 .

[17] S. S. Khanal, P. Prasad, A. Alsadoon and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning," *Education and Information Technologies*, vol. 25, p. 2635–2664, 2020.

[18] Z. Wu, M. L. Y. Tang and Q. Liang, "Exercise recommendation based on knowledge concept prediction," *Knowledge-Based Systems*, vol. Volume 210, p. 12, 27 December 2020.

[19] Y. Wahyuningsih and A. Djunaidy, "Framework Blended Learning Personalisasi Scaffolding Adaptif LMS (PSALMS) Pada Pelajaran Matematika," in *SEMNASTIK*, Palembang - Indonesia, 19 Oktober 2018.

[20] A. Tenriawaru, A. Djunaidy and D. Siahaan, "Mapping Metric Between Meaningful Learning Characteristics and Moodle Activities," *International Review on Computers and Software*, Vols. Vol. 11, N. 12, December 2016.

[21] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi and S. Ali, "Using machine learning to predict student difficulties from learning session data," *Artificial Intelligent Review*, 2018.

[22] J. Grundspenkis, "USAGE OF GRAPH PATTERNS FOR KNOWLEDGE ASSESSMENT BASED ON," *APPLIED COMPUTER SYSTEMS*, 2009.

[23] S. Zhang, L. Yao, A. Sun and Y. Tay, "Deep Learning based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys*, Vols. Vol. 1, No. 1, 2018. .

Application of Machine Learning methods for Prediction and Diagnosis of Urology Diseases

Yasemin Hande Sıtkı
R&D Software
Sisoft Health Information Systems
Ankara, Turkey
yasemin.sitki@sisoft.com.tr

Erhan Gökçay
Software Engineering Department
Atılım University
Ankara, Turkey
erhan.gokcay@atilim.edu.tr

Yusuf Şevki Günaydın
Computer Engineering Department
Ankara Yıldırım Beyazıt University
Ankara, Turkey
ysgunaydin@ybu.edu.tr

***Abstract*—Machine learning techniques combined with powerful computation hardware have been widely used to solve many problems such as image classification, object detection, image segmentation etc. Motivated from other fields, it has also played vital role for prediction and diagnosis of various diseases to help the medical practitioners for making better and timely decisions by exploring patterns and relationships in data. In this study, different machine learning algorithms were investigated for various decision making in medical data analysis. We specifically focused on diseases related to Urology. Data for 18 different diseases was collected from four different state hospitals. We found that most of the machine learning techniques produced satisfactory results and can be used as an auxiliary tool for diagnosis of various diseases.**

***Keywords*—Urology, machine learning, Random Forest, Neural networks,**

I. INTRODUCTION

Data mining used to reveal confidential, useful information for strategic decision making; Finding answers to the problem areas related to big screen data has created a different perspective in the use of other health data and has become a method that continues to increase rapidly. The purpose of this article is to create an infrastructure for the use of Data Mining in health and to provide a new perspective in decision-making processes by offering medicines related to the use of data mining to healthcare professionals in the health sector. [7] Urology disease prediction is studied with data mining studies. To select the successful algorithm, models will be created and the algorithm with the highest result will be selected.

It has increased in machine learning method in the early diagnosis of diseases in the field of health [1]. Biomedical, health bioinformatics, and medical machine imaging are among the most popular methods of learning [2], [3]. Learning methods that are successful in analyzing big data with different network architectures and learning algorithms are seen to help both early diagnosis of diseases and early disease workers in the field of health [4]. Data mining has become a common method, especially in the use of health data. Data collection tools and developments in data technologies require the storage and analysis of much information in information stores [5], [6].

This classification and regression titles according to data mining models functions will be examined and comparatively used algorithms will be given. Using the target WEKA, the most common representation for the urology branch is diagnosis and estimation for 25 diseases.

Object-based classification experts separate the knowledge base and then perform the statistical classification of an object. In recent years, different learning-based classification algorithms have been developed to quickly access the most accurate and reliable information from satellite images. Commonly used learning classifiers, Random Forest classification method, Bagging, Boosting, Decision Trees, Artificial Neural Networks, Support Vector Machines, K-Nearest Neighbor are included. These algorithms are also called machine learning methods. Machine learning methods can automatically extract rules and restrictions that users cannot see, using sufficiently large data and parameters. These methods try to find the most suitable model for new data with decision rules created by using input data. [8]

Data mining method has been applied in many areas. Some of the studies on these methods are summarized as follows:

In the study of AKKAYA [2] and his colleagues, after giving brief information about Breast Cancer, which is one of the cancer types and among his friends, they obtained general information about the tissue with the help of Xcyt pattern recognition program and breast cancer prediction and diagnosis was made using the Weka program.

In the studies of ISLER [3] and colleagues, the k-Mean clustering algorithm performance in the WEKA software structure with ongoing heart rate variability (HRV) analysis was examined to distinguish the patient with congestive heart failure (CHF) from the control group.

TAPKAN [1] and her colleagues carried out applications in WEKA package program by examining the algorithms of data pre-processing and classification, clustering and association rules.

KAYA [4] and colleagues tried to reveal the working principles of deep learning methods applied in the field of health and in which diseases they are used in the light of

the relevant literature. As a result of this study, the deep learning method suitable for the data used in the diagnosis of the disease was preferred and it was thought that it would increase the success of early diagnosis of the disease.

In their studies, KOYUNCUGIL [7] and colleagues wanted to create an infrastructure for the use of Data Mining in health and to provide health professionals with a new perspective in terms of decision-making processes by presenting examples of the use of data mining in the health sector.

In their studies, OZDARICI [8] and colleagues wanted to create the product classification success of the RO classification method was examined by comparing the Random Forest classification method with the SPOT 5 Maximum Similarity (EÇB) method.

In their studies, ERALDEMIR [9] and colleagues the performances of Random Forest and J48 decision trees were compared to classify EEG signals recorded during mathematical operations and text reading processes.

In their studies, CIHAN [10] and colleagues, Yıldız Teknik University (YTU), Computer Engineering department, Performed in the System Analysis and Design course from the questionnaires made to students about the projects data set was created. To this data set obtained most successful by applying classification algorithms algorithm has been detected. The purpose of the study is the project project success in the lessons to increase student projects The most successful classification in classification algorithm is determined.

In their studies, ALTINSOY [11] and colleagues, using artificial intelligence techniques in higher education institutions making evaluations about the effect of training methods on success Providing useful information to both the university administration and students is targeted.

In their studies, COŞKUN [12] and colleagues, in WEKA (Waikato Environment for Knowledge Analysis) program and SEER (Surveillance Epidemiology and End Results), models were created using algorithms belonging to various classification methods on the data source, and by comparing the estimation performance of the obtained models, it was examined which algorithm created more successful models in the data source used.

II. METHODOLOGY

Classification method is one of the main methods of data mining and is based on learning algorithm. It is applied to discover a pattern hidden in large-scale data. As part of data mining, the pattern is recorded digitally for an asset; It is expressed as observable, measurable and repeatable information. The classification algorithms applied to obtain the desired information ensure that the data set is divided into certain classes (discretization) according to the common feature of the data it contains. After this process, a classification model is obtained. The classification model obtained is applied on a new data set and the existence of similar classes in the data set of the classes determined by the model is investigated. For this purpose, various algorithms have been applied on the data set and the results have been examined. Random Forest algorithm, J48, Randomtree, NaiveBayes and MultilayerPerception algorithms were used. Detailed analysis results are given in the last section.

A. Random Forest Algorithm

Nowadays, Random Forest algorithm is frequently preferred because it performs very well in classification according to collective learning methods. The Random Forest classifier, which has been developed in recent years, provides advantages over acceleration (Freund and Schapire, 1996) and bagging (Breiman, 1996) methods, which are known as two very good methods in collective learning, in terms of both high accuracy and fastness.

Compared to the Random Forest classifier learning methods, especially the acceleration method, it is much faster in the training phase. It is a very useful classifier with its accuracy and efficiency. Because of the high computational load of it the acceleration method is slower than the bagging method, however, in most cases it gives more accurate results than this method.

Despite the disadvantages of the acceleration method such as being very slow and sensitive to noise, and the possibility of repetitive training, Random Forest classifier is computationally much simpler than the acceleration classifier, and it is not sensitive to noise.

First, the algorithm is a supervised classification algorithm. Because the Random Forest classifier performs very well in classification, it is an algorithm that is frequently preferred today compared to collective learning methods. The Random Forest classifier, which has been developed in recent years, is preferred both because of its speed and high accuracy. Compared to learning methods, the Random Forest classifier is much faster in the training phase, especially compared to the acceleration method. It is a very useful classifier with its efficiency and accuracy.

Random Forest is a supervised learning algorithm. As the name suggests, it creates a forest and somehow randomly does it. The "forest" it has established is a collection of decision trees that are often trained by the "bagging" method. The general purpose of the bagging method is that a combination of learning models increases the overall result. In simple terms; The Random Forest creates multiple decision trees and combines them to obtain a stable and more accurate forecast.[9]

B. Decision Tree Algorithm

Developed by Quinalan, J48 is a C4.5 decision tree developed for the classification process of nonlinear and small size data.

The decision tree approach is important in solving classification problems, with this method, a tree is created to model the classification process. After the tree is created, the classification process is performed by applying it to each data group in the database. Missing values are ignored when creating the tree. [10]

C. Random Tree Algorithm

Random Tree is a classification algorithm that generates a new tree by taking a certain number of randomly selected features in each node. There is no pruning in this classification algorithm. Therefore it has an option that allows estimation of class probabilities based on the stored dataset. [12]

D. Naive Bayes Algorithm

Naive Bayes algorithm is based on Bayes theorem. This theorem shows the relationship between conditional probabilities and marginal probabilities in probability for a random variable. A simple classification, the NB classification, can be used when it occurs when its own class in the data set has a classification problem and the contributions are assumed to be independent. While classifying, the highest probability target class was chosen.

Naive Bayes algorithm is based on Bayes theorem. This theorem shows the relationship between conditional probabilities and marginal probabilities within the probability distribution for a random variable. Assuming that every feature in the data set contributes equally to the classification problem and the contributions are independent from each other, a simple classification NB can be used. While classifying, the state with the highest probability is chosen as the target class.

E. Multilayer Perception Algorithm

Multi-Layer Sensor; Includes classifier that uses backpropagation to classify MLP samples. This network can be configured manually, created by the algorithm, or both. The network can also be viewed and modified during the training period [13].

The multilayer perceptron is applied to successfully solve some difficult and different problems. It is based on the hugely popular error propagation logic. Multilayer network has three important properties: (1) Each neuron model in the network contains nonlinearity at the exit. An important point here is that non-linearity is smooth transitive with respect to the sharp-transitive function used in Rosenblatt's perceptron. (2) The network has one or more hidden neurons that do not belong to the exit or entrance. Hidden neurons enable the network to learn complex tasks. (3) In the network, each neuron is interconnected. A change in connections causes a change in synaptic connections and weights. The development of the backpropagation algorithm is a turning point in neuronal networks. [11]

III. DATA SET

1555 of the 1758 data used as input in this article are classified correctly. The data set was divided into 2 parts as training data and test data and 10- fold Cross-validation has been selected in Weka.

A. Urology Unit Diseases and Symptoms

Data collected from 3 different hospitals first went through data mining preprocessing stages. Data collected from different hospitals were also found to be incomplete and incorrect. This data has been cleaned up and tagged to some variables and made into a single file and the data is ready for model building.

B. Data Set and Conversions:

The data set used includes 25 diagnostic main groups. There are 45 symptoms related to these diagnoses. Symptoms contain important information in making an accurate diagnosis of patients.

Gender is divided into 2 categories and labeled as female: F and male: E.

Location data are divided into 3 categories and labeled as 0: Location Unknown, 1: Urban and 2: Rural.

Age data are divided into 6 categories:

1: 0-10 childhood,
2: 11-24 youth,
3: 25-40 adult age,
4: 41-64,
5: 65-79 old age,
6: 80-above labeling was made.

• Data parameter created by weighting 25 diagnoses

(Number of Data of Relevant Diagnosis / Total Number of Data) x10 (1)

TABLE I Diagnostic List According to ICD-10 Main Codes

NO	CODE	DIAGNOSIS NAME
1	N23	Renal colic, unspecified
2	N20	Kidney and ureteral stones
3	N39	Other disorders of the urinary system
4	N30	Cystitis
5	N35	Stricture of the urethra
6	R52	Pain, not elsewhere
7	R31	Hematuria, unspecified
8	N31	Neuromuscular dysfunction of the bladder, not elsewhere classified
9	C67	Bladder malignant neoplasm
10	I86	Varicose veins, other sites
11	R32	Urinary incontinence, unspecified
12	R10	Abdominal and pelvic pain
13	R30	Pain with urination
14	N28	Other disorders of the kidney and ureter, not elsewhere classified
15	M79	Other soft tissue disorders, not elsewhere classified
16	N43	Hydrocele and spermatocele
17	N41	Inflammatory diseases of the prostate
18	N46	Male infertility
19	N45	Orchitis and epididymitis
20	N13	Obstructive and reflux uropathy
21	N48	Other disorders of the penis
22	M54	Dorsalgia
23	N21	Lower urinary tract calculus
24	K40	Inguinal hernia
25	N34	Urethritis and urethral syndrome

There are 45 symptoms in the data obtained from hospital records. Some of these can be listed as "BLOODY EJACULATION", "SAND IN URINE" and "DIZURI".

IV. EXPERIMENTAL RESULTS

A. Determination of Artificial Learning Method & Revealing Models

In our studies, to the data set added on Weka; Decision trees (Random Forest algorithm, Randomtree algorithm and J48), Bayes algorithm (NaiveBayes), Artificial neural network algorithms (Multilayer Perception) were applied. The data set was divided into 2 parts as training data and test data, and models were created. When we compared the models as a result of the analysis, the Random Forest Algorithm gave the highest result.

Many factors can be used to find the most accurate model in data mining studies. Criteria such as accuracy, precision and precision are some of them. The key factor here is to best measure the success of the algorithms. Choosing a model using only the accuracy criteria is not considered sufficient in scientific studies. Along with the accuracy criteria, sensitivity and precision criteria should also be taken in the model decision-making process [8].

a) Model Performance Metrics

Basic concepts in use while evaluating the model performance are error rate, certainty, sensitivity, and F-Measure.

Accuracy- Error Rate: The simplest and up to date method used in the evaluation of the model performance is the rate of accuracy.

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN) \quad (2)$$

$$\text{Error Rate} = (FP+FN)/(TP+FP+FN+TN) \quad (3)$$

Precision: It is the ratio of the number of true positive samples estimated as precision class 1 to the total number of samples estimated to be class 1.[13]

$$\text{Precision} = TP/(TP+FP) \quad (4)$$

Accuracy: Number of correctly classified positive samples is the ratio of the total number of positive samples.

$$\text{Accuracy} = TP/(TP+FN) \quad (5)$$

F-measure: The precision and sensitivity criteria alone are not sufficient to make a meaningful comparison. Evaluating both criteria together gives more accurate results. For this, the F-measure has been defined. The F-measure is the harmonic average of precision and sensitivity. [13]

$$F\text{−Measure} = (2\times\text{Accuracy}\times\text{Precision})/(\text{Accuracy} + \text{Precision}) \quad (6)$$

V. RESULTS AND DISCUSSIONS

In this study, 10- fold Cross-validation has been selected in Weka. Model success is given in Table III

TABLE II Model Success

Random Forest Algorithm Model Results	
Correctly Classified Instances	%88,45
Kappa Statistics	0,87
Mean Absolute Error	0,023
Root Mean Squared Error	0,092
Relative Absolute Error	32,404
Root Relative Squared Error	48,60
Total Number of Instances	1758

TABLE III Comparison of the Alternative Supervised Learning Model

	Random Forest	J48	Random Tree	Naive Bayes	ANN
Correctly Classified Instances	1555	598	1555	493	1548
Correctly Classified Rate	88,45	87,96	88,45	82,44	88,05
Precision	0,704	0,571	0,704	0,75	0,649
Kappa Statistics	0,869	0,864	0,8694	0,8012	0,8652
Mean Absolute Error	0,023	0,013	0,023	0,0181	0,0096
Root mean squared error	0,0915	0,0893	0,0915	0,0997	0,0832
ROC Area	0,991	0,973	0,991	0,988	0,99
F-Measure	0,633	0,533	0,633	0,5	0,686

According to these results, Random Forest and Random Tree algorithms have the highest success rate in the diagnosis of the most common diseases with a rate of 88.452%. These algorithms are followed by Multilayer

perception and J48 algorithms, respectively. In addition to the accuracy rate, the table includes other criteria such as mean absolute error, mean root square error, kappa statistic, relative absolute error, root relative square error, precision, F criterion and ROC Area value.

Kappa statistical value is a statistical method that compares the observed accuracy with the expected accuracy and shows the agreement between assignments made to existing classes. Random Forest and Randomtree algorithms have emerged as the algorithms giving the highest kappa statistic value with a value of 0.869. The kappa statistic value between 0.6 and 0.8 indicates that there is a significant fit, and that there is no random guess.

According to the Random Forest algorithm, the F Criterion was 0.633 and the ROC Area value was 0.991. These variables are desired to be close to 1. In addition, precision, sensitivity and F criterion values among the model performance criteria are required to be close to 1. When the comparison table is examined, it is seen that the highest success belongs to the Random Forest and Random Tree algorithms in the light of this value.

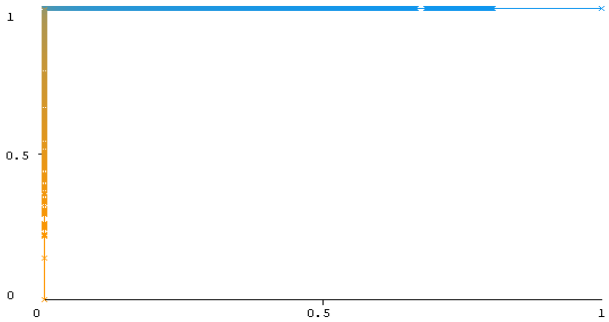


Figure 1 ROC curve plot for N39 diagnosis of Random Forest algorithm.

The ROC curve drawing of the Random Forest algorithm, which is the most successful algorithm for the most common diagnosis "N39 Other disorders of the urinary system", is given in Fig.1

VI. CONCLUSIONS

The data of urology branch collected from different hospitals were used to predict and diagnose diseases. 1555 of the 1758 data used as input in this article are classified correctly. According to these results, Random Forest and Random Tree algorithms have been the algorithms that give the highest success rate with a rate of 88.452% in the diagnosis of the most common diseases. These algorithms are followed by Multilayer Perception and J48 algorithms, respectively. Being able to deduce the meaningful relationships in the data obtained from the methods used in this diagnosis affects the diagnosis and treatment processes of the disease positively. It is believed that in the future, the results of these studies will also help doctors diagnose and make faster decisions.

REFERENCES

[1] Tapkan P., Özbakır L., Baykasoğlu A., "Weka İle Veri Madenciliği Süreci Ve Örnek Uygulama", Endüstri Mühendisliği Yazılımları ve Uygulamaları Kongresi, 30 Eylül-01/02, 2011.

[2] Danaci M., Celik M., Erhan Akkaya A., "Veri Madenciliği Yöntemleri Kullanılarak Meme Kanseri Hücrelerinin Tahmin ve Teşhisi", Akıllı Sistemlerde Yenilikler ve Uygulamaları Konferansı, ASYU, Haziran, 2010.

[3] İşler Y., Narin A., "WEKA Yazılımında k-Ortalama Algoritması Kullanılarak Konjestif Kalp Yetmezliği Hastalarının Teşhisi", SDU Teknik Bilimler Dergisi, Cilt 2, Sayı 4, Sayfa 21-29, 2012.

[4] Kaya U., Yılmaz A., Dikmen Y., "Sağlık Alanında Kullanılan Derin Öğrenme Yöntemleri", Avrupa Bilim ve Teknoloji Dergisi, Sayı 16, Sayfa 792-808, Ağustos 2019.

[5] Akman M., Genç Y., Ankaralı H., "RandomForests Yöntemi ve Sağlık Alanında Bir Uygulama", Turkiye Klinikleri Journal of Biostatistics, Sayı 3(1), Sayfa 36-48, 2011.

[6] Silahtaroğlu G., Veri Madenciliği, Papatya Bilim Yayınevi, 2016.

[7] Koyuncugil A.S., Özgülbaş N., "Veri Madenciliği: Tıp ve Sağlık Hizmetlerinde Kullanımı ve Uygulamaları" Bilişim Teknolojileri Dergisi, Cilt: 2, Sayi: 2, Mayıs 2009

[8] Silahtaroğlu, G., Veri madenciliği. Papatya Yayınları, İstanbul ,2008.

[9] Özdarıcı Ok A., AKAR Ö. ve Güngör, O. "Rastgele Orman Sınıflandırma Yöntemi Yardımıyla Tarım Alanlarındaki ürün Çeşitliliğinin Sınıflandırılması." TUFUAB 2011 VI. Teknik Sempozyumu, Sayfa 2, (2011)

[10] Eraldemir S.G., Arslan M. T. ve Yıldırım E., "EEG'de HHT Tabanlı Özellikleri Kullanan Rastgele Orman ve J48 Karar Ağacı Sınıflandırıcılarının Karşılaştırılması", International Advanced Researches & Engineering Congress-2017, sayfa 1253, 16-18 November 2017

[11] Cihan P., Kalıpsız O., "Öğrenci Proje Anketlerini Sınıflandırmada En Başarılı Algoritmanın Belirlenmesi", TBV - Türkiye Bilişim Vakfi Bilgisayar Bilimleri Ve Mühendisliği Derg isi, Sayfa 43, 2015 Cilt: 8 - Sayı: 1

[12] Altinsoy F., Uzaktan Eğitim Öğrencilerinin Başarilarinin Yapay Zeka Teknikleri İle Tahmini, Yüksek Lisans Tezi Bilgisayar Mühendisliği Anabilim Dali, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta – 2019, Sayfa 15.

[13] Coşkun C., Baykal A., "Veri Madenciliğinde Sınıflandırma Algoritmalarının Bir Örnek Üzerinde Karşılaştırılması" Akademik Bilişim'11 - XIII. Akademik Bilişim Konferansı Bildirileri, sayfa 4, 2- 4 Şubat 2011, İnönü Üniversitesi, Malatya

Impact of Local Histogram Equalization on Deep Learning architectures for Diagnosis of COVID-19 on Chest X-rays

Suleyman Serhan NARLI
Department of Computer Engineering
Iskenderun Technical University
Hatay, Turkey

Gokhan ALTAN
Department of Computer Engineering
Iskenderun Technical University
Hatay, Turkey

***Abstract*—Deep Learning (DL) is one of the most popular Machine Learning (ML) algorithms with feature learning capabilities. Its use is becoming widespread day by day due to its high-performance in classification in the various fields, including medical image processing. DL is inspired by an advanced neural network structure and includes many parameters. In consequence of its high performance, it is used in the classification of many diseases. DL algorithms, which are frequently used in the field of image processing, classify the pixels on the images by convolutional progress in different layers. Before learning the significant pixels in supervised ways, it can be ensured that the classification is more successful with different preprocessing methods. In this study, the effect of DL architectures on COVID-19 was investigated using Local Histogram Equalization (LHE). Chest x-ray images were examined with- and without-LHE to determine the effect of disk factor on transfer learning. The dataset consists of COVID-19, pneumonia, and normal chest x-ray images. Chest x-rays were segmented into two parts as of right lung lobe and left lung lobe. The effect on classification performance of transfer learning was observed by applying different disk value for LHE. The experiments were evaluated on the different pre-trained DL architectures, such as VGG16, AlexNet, and Inception model.**

***Index Terms*—Deep Learning, Convolutional Neural Networks, Chest X-Ray, Local Histogram Equalization, Medical Image Analysis.**

I. Introduction

Conventional image processing tries to obtain the significance by characterizing the meaningful features in the image. Handcrafted method has been developed to solve the problem of feature extraction for classification. Afterwards, the elaborated handcrafted techniques used in classification and segmentation. However, applying pattern recognition systems with handcrafted techniques for large datasets is a challenging process. There is no need to use handcrafted techniques in deep learning [1]; It uses many (deeper) unit layers with highly optimized algorithms, architectures, and regularization. Feature learning and transfer learning, which are novel techniques in Deep Learning (DL), are used in the pattern recognition without a separate feature extraction stage. In conventional image processing techniques, Convolutional Neural Networks (CNN) has started to be used more than the handcrafted techniques, typical classifier deep neural networks, such as AlexNet [2], VGGNet [3] or GoogLeNet [4] are fed directly by images. The sequence in DL generates class probability. Whereas extracting the probabilistic prediction, the experiments are performed by using distinct datasets for training and validation.

Deep learning has a widespread-use in a wide variety of machine learning tasks, including image classification, speech recognition, medical image processing, natural language processing, and more. Many researchers examined deep learning techniques from different perspectives, with the applications of medical image analysis [5], natural language processing [6], speech recognition systems and remote sensing [**?**]. Deep Neural Networks (DNN) consist of several layers of nodes, different architectures have been developed to solve various issues in different fields of use. Whereas the CNN is often used for computer vision and image recognition, recurrent neural network [7] is often used for time-series.

In DL, pruning is mainly used to developing a smaller and more efficient neural network model. The purposes of this technique are to optimize the model by eliminating the values of weight tensors and to obtain a less time consuming, computationally cost-effective model in training. Many studies have been performed in the field of medical image processing, such as CT images. 3-dimensional Deep architectures were used to automatically segment abdominal CT to identify arteries, portal veins, liver, spleen, stomach, gallbladder, and pancreas in each multi-organ image [8]. Deep convolution and deep belief networks, a type of RBM network, conditional random field, and structured support vector machine were utilized to separate breast masses from mammograms on mammograms [9]. Various methods were presented in the literature to detect pneumonia using chest X-ray images. Some of these methods use a machine learning algorithm as the classification technique and deep learning techniques for feature extraction and classification [10]. Similarly, [11] used logistic regression as a basic model for pneumonia detection using x-rays using a 121-layer dense convolutional network (DenseNet). In another study, Rajpurkar et al. developed a model that diagnoses pneumonia with a high accuracy rate, ChexNet developed a 121-layer CNN model that analyzes chest x-ray image and

classified the image bilaterally and determined the probability of pneumonia [12]. Many studies were performed on DL-based medical image applications. Examples include SegNet [16], DenseU-Net [17] and CardiacNet [18], [19] in detecting pneumonia in CXRs.

In this study, pre-processing was performed using the Local Histogram Equalization (LHE) method to make the tissues and masses in the lung images more visible before feeding the CNN for pathology detection on chest x-rays. Afterward, the performance of CNN and the effect of LHE with CNN on the detection of COVID-19, pneumonia, and normal images were evaluated. After LHE, it is aimed to increase the performance by combining similar tissues with CNN in the training stage. Lung images were cropped and separated as right and left lobe images, and only lung sections were inhaled. Thus, non-lung sections were eliminated from chest x-rays, and enabled the CNN architecture to be trained only on the lung sections for experimented architectures.

II. Materials and Methods

A. Convolutional Neural Networks

CNNs are a deep network model consisting of sequential convolutional layers and fully connected layers. The convolutional layers serve as feature extractors and learn the various representations of the input images. Output of convolutional layers are arranged in feature maps. Each neuron in a feature map has a receptive field that connects to a layer of neurons in the previous layer via a trainable set of weights [26]. Inputs are combined with learned weights to create a new feature map, and folded results are transferred via rectified linear units (ReLU) [24], which is a nonlinear activation function. The weights of each neuron in a feature map are equally constrained; however, different feature maps within the same convolution layer have different weights so that various features can be extracted at each location. The convolution layer consists of using various dimensions and numbers of filters to represent different properties of the inputs. The depth of the CNN is defined by the number of filters in each transform layer [25]. Deep learning models also include the pooling layer apart from the convolution layer, the purpose of this layer is to reduce the spatial resolution of the feature maps and thus achieve spatial invariance to enter distortions and translations. It is used to propagate the average of all values of a small region of an image to the next layer [26], [27]. After various numbers of sequential convolution layers and pool layers, the feature maps are stacked on top of each other, and the fully connected layers connected to flattened feature map. It interprets these layer representations and perform the classification function [28], [29]. It is standard to use the softmax function as output layer of fully connected layers [2].

B. Chest X-Ray Database

Chest x-ray images are a dataset obtained from real patients by chest x-ray examination and available in the clinical PACS database at the National Institutes of Health Clinical Center. The chest x-ray dataset includes 112,120 frontal view medical images of 30,805 unique patients with fourteen pulmonary diseases (each image can have multiple labels). Fourteen common thoracic pathologies are atelectasis, consolidation, infiltration, pneumothorax, edema, emphysema, fibrosis, effusion, pneumonia, pleural thickening, cardiomegaly, nodule, mass, and hernia [20]. This dataset is publicly available on the Kaggle platform [1]. Similar datasets: Montgomery County X-ray Set [21], Shenzhen Hospital X-ray Set [21], ChexNet [12] etc. Pneumonia-diagnosed lung images and healthy lung images were selected from the NIH Chest X-Ray dataset. The chest x-rays with COVID-19 were selected from the COVIDx dataset. This dataset is the largest open-access dataset in terms of the number of COVID-19 positive patient cases[2]. Many datasets are also publicly available in [13], [14], [15]. Chest x-ray can show areas such as edema, nodule, and infiltration in the lungs according to different intensity colors, making diagnosis difficult in the early stages due to the noisy image. In such cases, thanks to the filters applied on the image and by localizing the lung regions by segmentation on the images, the noise can be reduced by selecting certain regions and the performance of the dataset trained with deep learning can be increased. When we examine the publications in this area, the medical images can be processed with machine learning methods and the diagnosis of pathology for many lung diseases can be performed with the sensitivity of the doctor. Automatized techniques can also play an important role in identifying many pathologies and for early diagnosis with the enhancements of DL algorithms and architectures.

C. AlexNet and Transfer Learning

AlexNet is a DL architecture created by Alex Krizhevsky et al. [2] for ImageNet large visual recognition in 2012. It is a basic, simple, and an effective CNN architecture consisting mainly of gradual stages such as convolution layers, pool layers, corrected linear unit (ReLU). Specifically, AlexNet has five convolutional layers: first layer, second layer, third layer, and fourth layer, followed by pool layer and fifth layer, then three fully connected layers, respectively [2]. In this study, transfer learning was applied using the AlexNet architecture and the chest x-ray images were classified into multi classes. Transfer learning is a method of retraining a pre-trained model on a new dataset. The model is applied to the new dataset by fine-tuning the model in transfer learning. In this study, we adapted transfer learning into the chest x-ray database using the AlexNet architecture. The number of filters in the convolution layers of the model was reduced by pruning, thus performance improvement was reported on pruned AlexNet architecture.

D. Local Histogram Equalization

Automatic contrast enhancement is one of the common operations performed on visual data to reveal confidential detail. A histogram is a graph showing pixel density. Global

[1] https://www.kaggle.com/nih-chest-xrays/data

[2] www.kaggle.com/tawsifurrahman/covid19-radiography-database

histogram equalization (GHE) is generally a global tone mapping process, allowing the gray level of each pixel to be recreated by calculating the overall histogram of the image. However, these processes fail to increase contrast in both dark and bright image regions at the same time. Especially, small bright spots are hardly visible after such a global operation. To solve this problem, local histogram equalization (LHE) was proposed to perform in a floating window with this adjustable window size (radius or disk). Histogram equalization is applied independently to small areas of the image, thus preserving the contrast adjustment for different regions of the image [30], [31], [32].In this study, chest x-ray images were preprocessed using LHE. Figure(1) indicates the lung images for different LHE window sizes (disk) values.

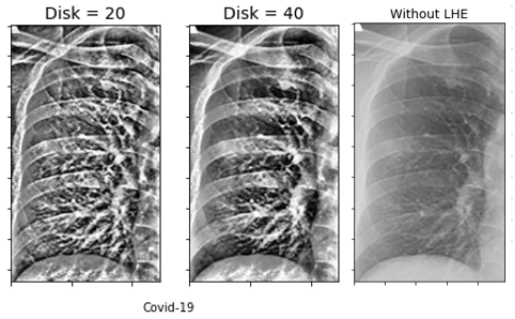


Fig. 1. Different LHE (disk) values

III. Experimental Results

AlexNet architecture was proposed by Alex Krizhevsky et al. in 2012. The total data size in this study consists of 2856 images for each model training, which is very small-scale dataset for the AlexNet architecture. Therefore, while the data is adapted to the AlexNet architecture with transfer learning, the number of filters in the convolution layers has been changed and the model has been optimized according to the training performance. By cropping the chest x-ray images, 2 lung lobes were cropped from each chest x-ray, separately. Thus,1000 healthy lung images, 1000 lung images diagnosed with Pneumonia, and finally 856 lung images diagnosed with COVID-19 were experimented in the analysis. Two floating window parameters (disk) were used for LHE with 20 and 40. The model has been optimized by fine-tuning the parameters on the AlexNet model. Instead of the last three fully connected layers in the AlexNet architecture, one fully connected layer is used, thus we prevented overfitting and reducing the number of the classification parameters. As in the AlexNet architecture, 5 convolution layers were used in the pruned model and the number of filters for each layer are determined in this way for 32,64,128,256 and 512, respectively. The dataset is split into three folds with 20%, 20%, and 60% for model testing, validation, and training, respectively. The batch size is 32 and the epoch number is 50 in the training of the fully connected layers. An early stopping was used to prevent overfitting during the supervised learning. In the last layer, 3 multi-classes are used for classification using the softmax function (COVID-19, Pneumonia, and Normal). We set 0.85e-5 for the learning rate in the adam optimizer. In this manner, the hyper-parameters in the model were arranged. The total number of classification parameters was 2,072,899. Tensorflow framework with python programming language was chosen to train the model (GPU: Nvidia Rtx 2060). We calculated accuracy, precision, recall, and f1 Score were used to evaluate the performance of the model [2].

TABLE I
Confusion Matrix results for different LHE Disk values

	Confusion Matrix	Normal	Pneumonia	Covid-19
Without LHE	**Normal**	170	3	7
	Pneumonia	4	200	1
	Covid-19	25	10	150
LHE (disk = 20)	**Normal**	150	8	24
	Pneumonia	0	190	20
	Covid-19	18	14	150
LHE (disk = 40)	**Normal**	160	6	13
	Pneumonia	0	200	5
	Covid-19	15	4	160

IV. Discussion

COVID-19 analysis of many researching areas has gained importance to ease the diagnosis and detection rate of disease for a pandemic. Consequently, advancing image processing techniques and proposing robust CADx systems are the main focus of researchers. The manuscripts for assessments of COVID-19 and using automatized techniques for abnormality detection within various diagnostic tools indicate the popularity of the field in 2020. Whereas high classification performances for diagnosis of COVID-19 were reported using Deep models with complex Deep learning architectures, applying simple pre-processing stages to the evaluation process reached high enough achievements using pruned and simple Deep architectures. The main ideas are using the generated representations instead of many convolutional layers, transmitting feature learning within resembled instances with LHE, and adapting the capability of feature transferring between adjacent layers for identification of chest x-rays with healthy, COVID-19, and pneumonia pathology.

TABLE II
Comparison of relevant DL-based disease classification studies in terms of different Disk values for LHE

Disk Value	Data(Train/Test)	Accuracy	Precision	Recall	F1 Score
Without LHE	2284/572	91%	91%	91%	91%
20	2284/572	85%	85%	85%	85%
40	2284/572	92%	92%	92%	92%

Table III presents the comparison of our study with the deep learning algorithms developed for the diagnosis of COVID-19. Wang et al. reported an accuracy rate of 93.3% for their proposed COVID-Net model using cropping, translation, rotation, horizontal rotation, zoom on chest x-rays [33]. Karthik et al. proposed a different deep learning model to learn specific filters within a single convolutional layer to identify specific

TABLE III
COMPARISON OF RELEVANT DL-BASED COVID-19 DISEASE DETECTION STUDIES IN TERMS OF ALGORITHMS AND PERFORMANCES

Related Works	Methods	Classifier	Covid-19 Test Sample	ACC (%)	F1 Score (%)	F1-score for COVID-19 (%)
Wang et al. [33]	COVID-Net CNN	CNN	100	93.3	93.13	94.78
Karthik et al. [34]	Customized CNN with distinctive filter learning module	CNN	112	97.94	96.90	97.20
Ozturk et al. [35]	DarkNet-19	CNN	25	87.02	88.0	88
Khan et al. [36]	CoroNet net	CNN	70	89.6	89.8	95.61
Apostolopoulos et al. [37]	Transfer learning with MobileNetV2	CNN	222	94.72	93.80	90.50
Farooq et al. [38]	Transfer learning with ResNet50	CNN	68	96.23	96.88	100
Hemdan et al. [39]	COVIDX-Net	CNN	25	90	-	-
Narin et al. [40]	ResNet-50	CNN	100	98	-	-
Our Study	AlexNet Model without LHE	CNN	572	91	91	87
Our Study	AlexNet Model With LHE Disk 20	CNN	572	85	85	80
Our Study	AlexNet Model With LHE Disk 40	CNN	572	92	92	90

classes of pneumonia. Their proposal achieved an F1 score rate of 97.20% for 112 chest x-rays with COVID-19 . Moreover, the lung regions were divided into sections by applying a pre-trained algorithm and the segmentation process was applied [34]. Ozturk et al. presented the DarkNet model, which is a novel model for the detection of COVID-19 which consists of 17 convolutional layers for binary classification(COVID-19 and no finding), and multi-class classification (no finding, COVID-19, and pneumonia). The model achieved the classification accuracy rates of 98.08% and 87.02% for binary classification and multi-class classification [35]. Khan et al. reported the CoroNet model, which is based on the pre-trained Xception CNN architecture and achieved classification accuracy of 95% for 3 classes (COVID-19, pneumonia, and normal) [36]. Apostolopoulos et al. reached an accuracy rate of 96.78% using transfer learning on MobileNetV2 [37]. Farooq et al. developed the COVID-ResNet model by fine-tuning the ResNet50 model and with transfer learning and achieved an accuracy rate of 96.23% in the COVIDx dataset [38]. Hemdan et al. proposed a novel model COVIDX-Net that is based on seven different architectures of DCNNs; VGG19, DenseNet201, InceptionV3, ResNetV2, InceptionResNetV2, Xception, and MobileNetV2. They reported performance rate range for various pre-tained architectures between 80-90% [39]. Narin et al. evaluated five pre-trained convolutional neural network-based models including ResNet50, ResNet101, ResNet152, InceptionV3 and Inception-ResNetV2. He applied three different binary classifications with four classes (COVID-19, normal (healthy), viral pneumonia, and bacterial pneumonia) using 5-fold cross-validation. Among the other four models used, the pre-trained ResNet50 model had the highest classification performance for different datasets with accuracy rate of 99.5% [40].

The proposed pruned CNN architecture with LHE is a powerful trajectory to get high enough results with easy-adaptable real-time applications over to the state-of-art. The proposed model possesses the capabilities of pre-trained detailed CNN architectures by feeding the input with resembled chest X-rays before the layer-wise feature learning. Using LHE for generating smoothed representation provided high enough classification performances using pruned AlexNet architecture. Therefore, simplifying the deep architectures enables reducing the training time of the COVID-19 identification architectures and integrating the shallow architectures for even real-time applications in embedded systems.

V. CONCLUSION

The study contains the novelty of applying LHE for generating resembling representations for increasing the generalization capability of CNN with shallow architectures. Whereas many studies are focusing on just classification performances in supervised training and modeling feature learning stages, the proposal is a pioneer study with LHE, which has a common use on medical images. The necessity of big data for the training of CNN architectures is overcome using LHE as a resembling procedure using a middle-scale chest x-ray dataset. The proposed model has reached high identification rates for multi-class diseases using is a basic, pruned architecture. It is easy-to-integrate architecture for various types of medical image analysis. Using novel visualization techniques and localization algorithms on chest x-rays has a possible contribution to Deep Learning algorithms. Although CNN architectures don't need additional feature extraction and pre-processing procedures, it has limitations in detecting small pathologies with low generalization capabilities.

REFERENCES

[1] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015, arXiv:1409.1556v6 [cs.CV].

[4] C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[5] Litjens, G., Kooi, T., Bejnordi, B., Setio, A., Ciompi, F., Ghafoorian, M., et al. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88.

[6] Young, T., Hazarika, D., Poria, S., Cambria, E. (2018). Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, 13(3), 55–75.

[7] K. Cho. (2014). "Learning phrase representations using RNN encoderdecoder for statistical machine translation." [Online]. Available: https://arxiv.org/abs/1406.1078

[8] H.R. Roth, et al. Deep learning and its application to medical image segmentation (2018), pp. 1-6

[9] Deep learning and structured prediction for the segmentation of mass in mammograms, N. Navab, J. Hornegger, W. Wells, A. Frangi (Eds.), Medical image computing and computer-assisted intervention – MICCAI 2015. MICCAI 2015. Lecture notes in computer science, vol. 9349 (2015), pp. 605-612

[10] Luján-García, J.E.; Yáñez-Márquez, C.; Villuendas-Rey, Y.; Camacho-Nieto, O. A Transfer Learning Method for Pneumonia Classification and Visualization. Appl. Sci. 2020, 10, 2908.

[11] Antin, B.; Kravitz, J.; Martayan, E. Detecting Pneumonia in Chest X-rays with Supervised Learning; Semanticscholar Org.: Allen Institute for Artificial intelligence, Seattle, WA, USA, 2017.

[12] Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv 2017, arXiv:1711.05225.

[13] Cohen, J. P., Morrison, P.,Dao, L. COVID-19 image data collection. arXiv:2003.11597 (2020).

[14] Chung, A. Figure 1 COVID-19 chest x-ray data initiative. https://github.com/agchung/Figure1-COVID-chestxray-dataset (2020).

[15] Radiological Society of North America. COVID-19 radiography database. https://www.kaggle.com/tawsifurrahman/covid19-radiography-database (2019).

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[17] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, "HDenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes," IEEE Trans. Med. Imaging, vol. 37, no. 12, pp. 2663–2674, Dec. 2018, doi: 10.1109/TMI.2018.2845918.

[18] A. Mortazi, R. Karim, K. Rhode, J. Burt, and U. Bagci, "CardiacNET: Segmentation of left atrium and proximal pulmonary veins from MRI using multi-view CNN," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017, vol. 10434 LNCS, pp. 377–385, doi: 10.1007/978-3-319-66185-8-43.

[19] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, "Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification," Sci. Rep., vol. 9, no. 1, pp. 1–10, Dec. 2019, doi: 10.1038/s41598-019-42294-8.

[20] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, MohammadhadiBagheri, Ronald Summers, ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common ThoraxDiseases, IEEE CVPR, pp. 3462-3471,2017

[21] Jaeger S, Candemir S, Antani S, Wáng YX, Lu PX, Thoma G. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. Quant Imaging Med Surg. 2014 Dec;4(6):475-7. doi: 10.3978/j.issn.2223-4292.2014.11.20. PMID: 25525580; PMCID: PMC4256233.

[22] Roosa K, Lee Y, Luo R, Kirpich A, Rothenberg R, Hyman JM, and et al. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. Infectious Disease Modelling, 5:256-263, 2020.

[23] Coronavirus. World Health Organization: https://www.who.int/healthtopics/coronavirus,2020.

[24] Nair, V., Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (pp. 807–814). N.p.: International Machine Learning Society.

[25] Cireşan DC, Meier U, Masci J, Gambardella LM, Schmidhuber J,(2011) Flexible, high performance convolutional neural networks for image classification. In: IJCAI International Joint Conference on Artificial Intelligence. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210.

[26] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

[27] Ranzato, M. A., Huang, F. J., Boureau, Y., LeCun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Proceedings IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–8). Los Alamitos, CA: IEEE Computer Society.

[28] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv 1207.0580.

[29] Zeiler, M. D., Fergus, R. (2014). Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision (pp. 818–833). Berlin: Springer.

[30] Digital Image Processing-Concepts, Algorithms, and Scientific Applications, Jaehne. B, Springer, Berlin 1991.

[31] V. Caselles, J. L. Lisani, J. M. Morel, and G. Sapiro, "Shape preserving local histogram modification," IEEE Trans image proc,vol. 8, no. 2, pp. 220-230, Feb. 1999.

[32] J. Y. Kim, L. S. Kim, and S. H. Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," IEEE Trans circuits systems and video technology, vol. 11, no. 4, pp. 475-484, Apr. 2001.

[33] Wang, L., Lin, Z.Q. Wong, A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep 10, 19549 (2020). https://doi.org/10.1038/s41598-020-76550-z.

[34] Karthik R, Menaka R, M H. Learning distinctive filters for COVID-19 detection from chest X-ray using shuffled residual CNN. Appl Soft Comput. 2021;99:106744. doi:10.1016/j.asoc.2020.106744.

[35] Ozturk T., Talo M., Yildirim E.A., Baloglu U.B., Yildirim O., Rajendra Acharya U. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput. Biol. Med. 2020;121.

[36] Khan A.I., Shah J.L., Bhat M.M. Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. Comput. Methods Programs Biomed. 2020.

[37] Apostolopoulos I.D., Mpesiana T.A. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys. Eng. Sci. Med. 2020.

[38] Farooq Muhammad, Hafeez Abdul. 2020. Covid-resnet: a deep learning framework for screening of covid19 from radiograph. arXiv preprint arXiv:2003.14395.

[39] Hemdan E.E.-D., Shouman M.A., Karar M.E. COVIDX-Net: A framework of deep learning classifiers to diagnose covid-19 in X-ray images (2020) arXiv preprint arXiv:2003.11055.

[40] Narin A., Kaya C., Pamuk Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks (2020).

Hybridizing a Conceptual Hydrological Model with Neural Networks to Enhance Runoff Prediction

Zeynep Beril Ersoy
Department of Civil Engineering, Hydraulic Division
Balikesir University
Balikesir, Turkey
zeynepberilersoy@gmail.com

Umut Okkan
Department of Civil Engineering, Hydraulic Division
Balikesir University
Balikesir, Turkey
umutokkan@balikesir.edu.tr

Okan Fıstıkoğlu
Department of Civil Engineering, Hydraulic Division
Dokuz Eylül University
İzmir, Turkey
okan.fistikoglu@deu.edu.tr

***Abstract*—In this paper, a hybrid rainfall-runoff model is developed by embedding an artificial neural network (ANN) into a monthly lumped conceptual model termed the dynamic water balance model (dynwbm). Based on this approach, groundwater storage parameters are eliminated, and direct runoff and deep percolation outputs derived with three conceptual parameters are served as input to the ANN part. In this serial hybridization, the dynwbm parameters and ANN weights are calibrated simultaneously. The proposed hybrid model, dynNN, equipped with an automatic calibration algorithm, has been applied at a key location in the Gediz River Basin of western Turkey. From the performance measures assessed, it has been proven that a robust model has taken the strengths of conceptual and artificial neural networks while it has disregarded their drawbacks.**

Keywords—Conceptual Rainfall-Runoff Modeling, Artificial Neural Network, Gediz River Basin, Automatic Calibration

I. INTRODUCTION

The transition of rainfall to runoff over a watershed is a dynamic and nonlinear process due to hydrological components heterogeneously distributed in the basins and their spatial-temporal changes. Conceptual rainfall-runoff (CRR) models were commonly used in this concept for modeling. Interconnection between physical and hydrological processes through a series of conceptual soil moisture and groundwater stores can be represented with simplified mathematical expressions in CRR models [1]. On the other hand, related literature has pointed out that machine learning (ML) models that do not require detailed hydrological process knowledge can be better options against CRR models [2]. Concerning the use of artificial neural networks (ANNs) in rainfall-runoff modeling, it is clear that the use of this method did not lose popularity after the mid-1990s. In the related literature, the main beneficial aspects of ANNs, which are compensating for the missing hydro-meteorological data information, providing relatively more flexible predictions using nonlinear mapping, and simulating hydrological processes only through various observations, were accentuated [3].

In addition to the application of CRR and ML modeling, some researchers have tried to discover different alternatives to combine the strengths of those techniques as they recognize the capability of traditional conceptual models in representing the physical properties of the basins and the dominance of the data-driven models in exploring the nonlinear relationship between input and output series.

The first study on this subject was profoundly handled by [4]. They incorporated the simulated soil moisture index values obtained from the GR4J model (in French, modèle du Génie Rural à 4 paramètres Journalier) with the certain parameter estimations into the ANNs to improve short-term streamflow simulation capability. Inspired by their study, other researchers have recently introduced similar implementations [1,2,5,6]. In those coupled model exercises, after the parameter calibrations of the CRR models are made, their several outputs constitute auxiliary inputs for the ML model training. However, in this parallel hybridization comprising two consecutive processes, performing the cascade calibration of CRR and ML models increases the computational complexity. Moreover, the mutual interaction between the parameters governing CRR and ML models is also not considered.

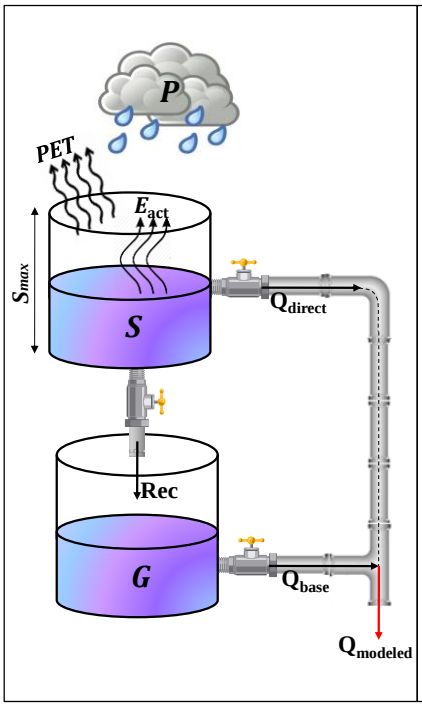
In this study, to overcome the several drawbacks addressed, the embedding of multi-layered ANN into a monthly CRR model called the dynamic water balance model (dynwbm) has been provided, and all defined parameters (dynwbm parameters and ANN weights) have been calibrated simultaneously. Unlike most studies, we also eliminated the groundwater store of the dynwbm (with baseflow parameters) and forwarded only the direct runoff and deep percolation outputs of the dynwbm to the input layer of ANN to enhance streamflow simulation. To detect any benefits achieved by the hybridization of the conceptual and ANN-based methods, a comparison, which involves relative performances of the single ANN, dynwbm with linear and nonlinear baseflow functions, and the hybrid model termed as dynNN, has been made for monthly runoff prediction regarding an exemplary subbasin in western Turkey.

II. METHODOLOGY

A. Dynamic Water Balance Model

Although there are many CRR models in the hydrology literature, the dynamic water balance model (dynwbm) proposed by [7] was chosen for this study because it is a parsimonious model and performs well in various climatic conditions [8,9]. The model is also efficient because it only requires potential evapotranspiration (PET) and monthly total precipitation (P) as input variables to simulate monthly runoff.

In the study, two variants of the dynwbm model were examined separately against the hybrid model proposed. First version contains the linear baseflow function (after this referred to as dynLGWS) as applied in [7]. The other version uses a nonlinear baseflow function (after this referred to as dynNLGWS). Schematic representation of the conceptual flow chart and the related computation steps is given in Fig. 1. The parameter definitions are also denoted in this figure. We had to provide a brief content of the model due to the page limitation. Further details can be found in [7].



Computation steps

(E1): $C(t) = P(t) \times \left[1 + \frac{S_{\max} - S(t-1) + PET(t)}{P(t)} - \left[1 + \left(\frac{S_{\max} - S(t-1) + PET(t)}{P(t)} \right)^{1/(1-\alpha_1)} \right]^{1-\alpha_1} \right]$

(E2): $Q_{direct}(t) = P(t) - C(t)$

(E3): $wc(t) = C(t) + S(t-1)$

(E4): $eop(t) = wc(t) \times \left[1 + \frac{PET(t) + S_{\max}}{wc(t)} - \left[1 + \left(\frac{PET(t) + S_{\max}}{wc(t)} \right)^{1/(1-\alpha_2)} \right]^{1-\alpha_2} \right]$

(E5): $\mathrm{Rec}(t) = wc(t) - eop(t)$

(E6): $E_{act}(t) = wc(t) \times \left[1 + \frac{PET(t)}{wc(t)} - \left[1 + \left(\frac{PET(t)}{wc(t)} \right)^{1/(1-\alpha_2)} \right]^{1-\alpha_2} \right]$

(E7): $S(t) = \max(0, eop(t) - E_{act}(t))$

(E8): for dynLGWS $Q_{base}(t) = d \times G(t-1)$, for dynNLGWS $Q_{base}(t) = (G(t-1)/d)^{1/e}$

(E9): $G(t) = G(t-1) + \mathrm{Rec}(t) - Q_{base}(t)$

(E10): $Q_{\mathrm{modeled}}(t) = Q_{direct}(t) + Q_{base}(t)$

Parameter definitions:

S_{max}: the maximum capacity of soil moisture storage.
α_1: the parameter controlling the rainfall retention and the direct runoff.
α_2: the parameter relating to the evapotranspiration efficiency.
d, e: the baseflow parameters.

Fig 1. Schematic diagram of the dynwbm, the related calculation steps, and the parameter definitions.

B. *Artificial Neural Network*

Despite several ANN architectures in the related literature, feed-forward neural networks, which are sometimes referred to as multi-layer perceptron, are the commonly known type. In the ANN training, the weights and bias terms are calibrated through an optimization algorithm, which can also calibrate CRR models. As is known, the complexity of ANN can be configured by the number of nodes in the hidden layer (*nnh*), and herein a trial-and-error approach is usually preferred to find the convenient *nnh* [1]. The log-sigmoid transfer function was used at both the hidden layer nodes and the output layer node. In the study presented, the ANN model, referred to as the ANN1, was trained with two inputs: precipitation (P) and potential evapotranspiration (PET). On the other hand, to represent the initial catchment wetness, the model referred to as ANN2 was trained by adding one-month-ahead precipitation data as the third input.

C. *Hybrid Rainfall-Runoff Model*

As seen in Fig.2, the hybrid dynNN model consists of conceptual and ANN parts. The expressions from E1 to E7, as shown in Fig. 1, are performed with the S_{max}, α_1, and α_2 parameters in the conceptual part that needs inputs such as P, PET, and initial soil moisture content (S_o) and the simulated Rec and Q_{direct} outputs are transmitted to the ANN part. In this case, the baseflow computation of the conceptual model in which the groundwater storage is governed with a one-parameter variant dynLGWS or a two-parameter variant dynNLGWS is eliminated by means of ANN part structured. This nested configuration converted the mentioned conceptual outputs to the runoff at the drainage basin outlet. In this model, by the log sigmoid activation function, the normalization function scales new inputs (I_1=Rec and I_2=Q_{direct}) and observed runoff values (Q_{obs}) so that they fall in the range [0,1]. After compiling the input matrix as in Eq.(1), the weighted aggregation matrix formed between the input layer and the hidden layer, and the hidden layer outputs are obtained by Eq.(2) and Eq. (3), respectively.

$$[Inp]_{L\times 2} = \begin{bmatrix} \mathrm{Rec}(t=1) & Q_{direct}(t=1) \\ \mathrm{Rec}(t=2) & Q_{direct}(t=2) \\ \vdots & \vdots \\ \mathrm{Rec}(t=L) & Q_{direct}(t=L) \end{bmatrix} \quad (1)$$

where *Inp* is the network input matrix consisting of normalized values of the soil-water fluxes simulated from the conceptual part; *L* is the data length for the calibration or validation period.

$$\left[net^{(1)}\right]_{L\times nnh} = [Inp]_{L\times 2} \begin{bmatrix} W_{1,1}^{(1)} & W_{1,2}^{(1)} & \dots & W_{1,nnh}^{(1)} \\ W_{2,1}^{(1)} & W_{2,2}^{(1)} & \dots & W_{2,nnh}^{(1)} \end{bmatrix}_{2\times nnh} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{L\times 1} \begin{bmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_{nnh}^{(1)} \end{bmatrix}_{1\times nnh} \quad (2)$$

$$[\textit{hidden layer outputs}]_{L\times nnh} = f^{(1)}\left(\left[net^{(1)}\right]\right) \quad (3)$$

where $W_{i,j}^{(1)}$ and $b_j^{(1)}$ are the weight and the bias term, respectively, existed between the input layer and hidden layer (i=1 to 2; j=1 to *nnh*); $f^{(1)}(.)$ is the activation function converting the related aggregation matrix $net^{(1)}$ to hidden layer outputs.

After that, hidden layer outputs are turned into the scaled outputs of the output layer using Eq. (4) and Eq. (5).

$$\left[net^{(2)}\right]_{L\times 1} = [\textit{hidden layer outputs}]_{L\times nnh} \begin{bmatrix} W_1^{(2)} \\ W_2^{(2)} \\ \vdots \\ W_{nnh}^{(2)} \end{bmatrix}_{nnh\times 1} + \begin{bmatrix} b^{(2)} \\ b^{(2)} \\ \vdots \\ b^{(2)} \end{bmatrix}_{L\times 1} \quad (4)$$

$$\left[Q_{m,s}\right]_{L\times 1} = f^{(2)}\left(\left[net^{(2)}\right]\right) \quad (5)$$

where $W_j^{(2)}$ is a weight that existed between the hidden layer and the output layer (j=1 to *nnh*); $b^{(2)}$ is the bias term connecting to the output node; $f^{(2)}(.)$ is the last utilized activation function transforming the aggregation matrix $net^{(2)}$ to the scaled model outputs $Q_{m,s}$.

In the final stage of the hybrid model, the reverse-scaling converts the scaled network outputs $Q_{m,s}$ back into the runoff predictions ($Q_{modeled}$), having the same unit as the original targets.

As can be seen from Fig.2, not only three parameters managing the conceptual part but also the weights and bias terms providing inter-layer connection in the ANN part need to be tuned. In the study, the hybrid model whose total number of parameters is equal to 4 × *nnh* + 4 was trained with an optimization algorithm named hybrid particle swarm optimization (HPSO) algorithm. Further details about the algorithm run can be accessed from [9].

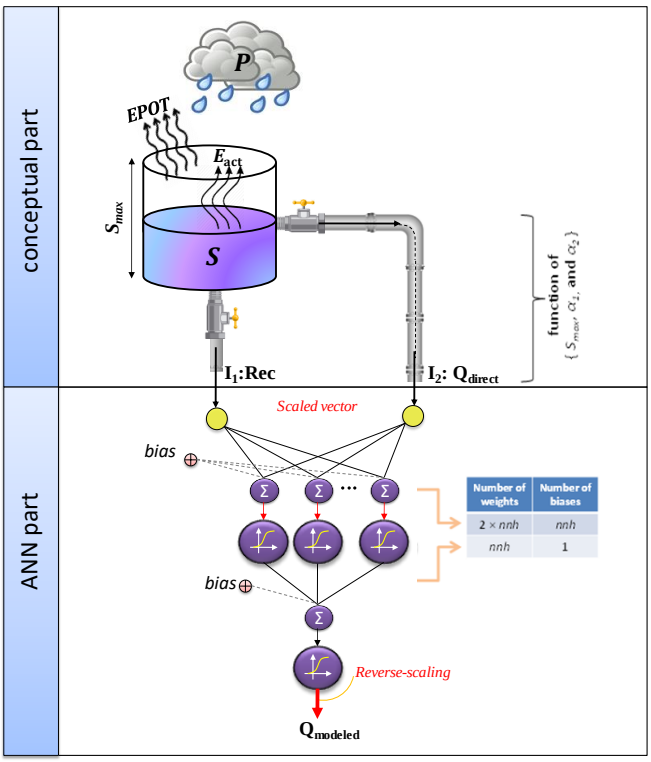


Fig. 2 The illustrative flowchart of the hybrid model

D. *Criteria Used For Evaluating The Performances Of Models*

In this study, as the objective (cost) function within the HPSO algorithm, the conventional sum-of-squared errors (SSE) function was used. The calibration of all models requires minimizing SSE. In addition to questioning SSE values in training and validation periods, different performance metrics are also utilized for analyzing the quality of the predictions. In a review about model assessment, [10] have recommended some quantitative statistics, namely, Nash–Sutcliffe efficiency (NS), the ratio of the root mean square error (RMSE) to the standard deviation of the observed runoff (RSR). The models were graded in terms of their NS and RSR performance, and the classification system was given in Table 1 [10,11,12].

Table 1. Classification criteria for hydrological models [10]

Performance Rating	NS	RSR	Grading for each
Vey Good	0.75<NS≤1.00	0.00≤RSR≤0.50	3
Good	0.65<NS≤0.75	0.50<RSR≤0.60	2
Satisfactory	0.50<NS≤0.65	0.60<RSR≤0.70	1
Unsatisfactory	NS≤0.50	RSR>0.7	Unsatisfactory

III. STUDY AREA AND DATA

In this study, the Acisu station in the Gediz Basin, which has a drainage area of approximately 3,272.4 km2, was selected to apply the hybrid model described previously. This station is located in the north-east part of the basin and feeds the Demirköprü reservoir. In the subbasin with regard to this station, the mean annual temperature is about 15 ^oC, while the total annual areal precipitation calculated from the Thiessen polygon method is about 505 mm.

As a first model input, the monthly time-scale Thiessen-weighted precipitation series (P) obtained from the Turkish State Meteorological Service (MGM) data was prepared. While monthly temperature and relative humidity data were compiled from the meteorological stations operated by MGM, ERA-Interim reanalysis data sets having 0.75^0 × 0.75^0 resolution were used for other variables (wind speed and solar radiation) needed for the Penman-Monteith equation to obtain areal mean PET estimations used as second model input series.

IV. RESULTS

A. *Optimization Algorithm Setting And Calibrated Parameters*

In the study, the data obtained for the Acisu station were divided into two equal parts for calibration and validation. As the first month of the water year started in October, and the rainfall regime is weak in September, the initial storage values in the models having conceptual reservoirs were set to zero at *t*=0. Then, control variables of the HPSO algorithm, which calibrate the models automatically, have been selected. The acceleration coefficients c_1 and c_2 were set to 2.0. The chaotic random inertia weight was operated as the velocity updating rule of the algorithm. The Levenberg-Marquardt (LM) part of HPSO that exalts the calibration speed and solution stability was also adjusted as in [9]. The population size of the algorithm also was defined as 50, which is adequate for all modeling processes. A broader estimation bound was assumed for the three common parameters in the two dynwbm variants and the dynNN. Accordingly, the lower-upper bound of parameter S_{max} is 50-1000 mm, while those of α_1 and α_2 are 0.01-0.90. Besides, the range of 0.01-1.0 was chosen for the parameter *d* in dynLGWS, while the range of 0.01-10.0 was taken for parameters *d* and *e* in dynNLGWS. After the initializing lower and upper bound of the dynwbm and the water balance part in the dynNN model, bounds of weights and biases, which are the free parameters of the ANN part, were also set to -1 and 1. Five simulations were done for both models. Additionally, the dynNN model with a range of 2–20 hidden nodes was sensitively trained by HPSO. After all trials for simulations, the solution which provides the smallest SSE in the validation period was accepted as the best one, and the calibrated parameters were given in Table 2 (Since sharing the calibrated ANN weights of dynNN would take up space in the paper, only optimal *nnh* numbers were specified).

Table 2. Calibrated parameters of the original and modified models

Models	S_{max}	α_1	α_2	d	e
dynLGWS	224.29	0.62	0.76	0.50	-
dynNLGWS	230.03	0.62	0.76	0.76	1.32
dynNN (nnh = 3)	783.39	0.72	0.52	-	-

B. Calibration and Validation Performances

In the study, the calibration period findings indicate that the increase in the number of parameters has also reflected on the model prediction precision. While the dynNLGWS shows a small improvement compared to the original one (dynLGWS), the situation in the dynNN is unsurprisingly much more pronounced. For the station, dynNN achieved a 13% reduction in the training RMSE compared to its closest competitor. In return for this determination, the perception that parsimonious models can produce more reasonable predictions in the validation period should be questioned. So, it is thought that it will be more consistent with scrutinizing the relative success of the dynNN by means of the validation outputs. Table 3 shows that the best results for Acisu station according to each metric are highlighted in bold, and the grading regarding the NS and RSR is given in Table 4.

Moreover, it is immediately obvious from Table 3 that the two-input ANN was the worst among all models as it gave the lowest NS and the highest RSR. Without using any antecedent data, it produced predictions that could only fall into *satisfactory* and *unsatisfactory* (US) categories. According to the same table, it is apparent that the dynNN provided the most accurate and reliable predictions in terms of all criteria considered. Besides, it supplied the necessary conditions for the *very good* category whose limit values are given in Section 2.D and got six points. Besides, dynNN is the only model that got the maximum point during the validation period. On the other hand, as can be seen from Table 4, dynLGWS and dynNLGWS models were able to produce results that got involved in *good* or *very good* categories in NS and RSR.

Table 3. Performance of the models during calibration and validation periods

Models	Calibration			Validation		
	RMSE (mm)	NS	RSR	RMSE (mm)	NS	RSR
dynLGWS	3.52	0.84	0.40	4.83	0.69	0.56
dynNLGWS	3.49	0.84	0.40	4.71	0.70	0.54
ANN1	6.47	0.45	0.74	6.31	0.47	0.73
ANN2	4.29	0.76	0.49	4.71	0.70	0.54
dynNN	3.03	0.88	0.35	3.50	0.84	0.40

Table 4. Grading of the models during calibration and validation periods

Models	Calibration		OVERALL	Validation		OVERALL
	NS	RSR		NS	RSR	
dynLGWS	3	3	6	2	2	4
dynNLGWS	3	3	6	2	2	4
ANN1	US	US	US	US	US	US
ANN2	3	3	6	2	2	4
dynNN	3	3	6	3	3	6

As can be seen in Fig. 3, even if there were small diversions on high and low flow predictions produced by the hybrid model, it was more robust and reliable when compared with the closest competitor dynNLGWS. While the desired value of the slope of the trendline is 1, the validation period results of the dynNN and dynNLGWS model are 0.96 and 0.81, respectively. Besides, in Fig. 4, the time series of the observed data and the modeled data using dynNN visually support the simulations derived.

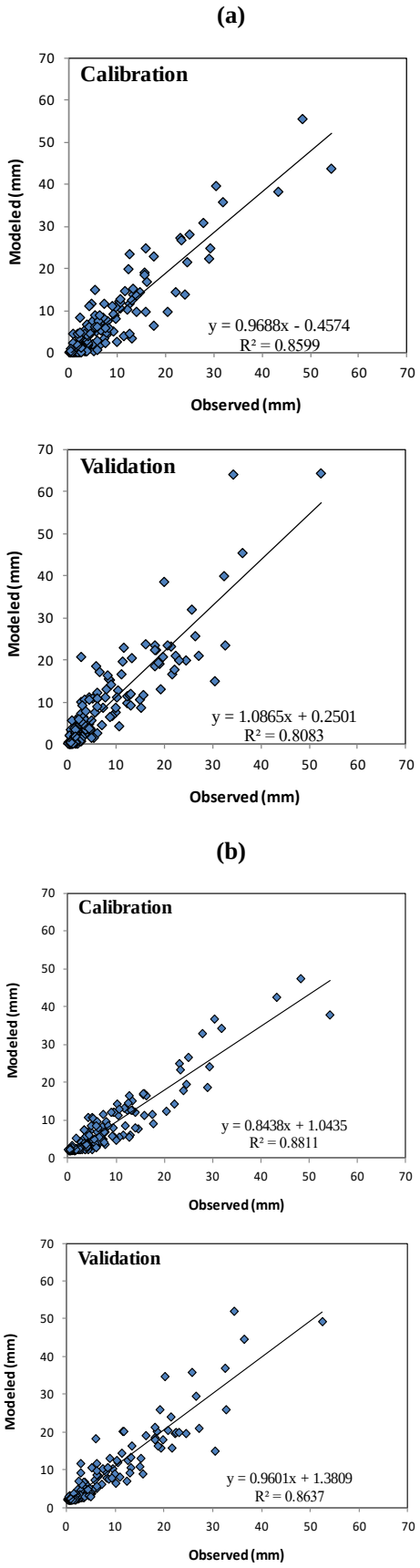


Fig.3 Scatter plot for (a) dynNLGWS and (b) dynNN.

V. CONCLUSION

In this study, it was questioned whether a hybrid-style rainfall-runoff model, in which ANN was directly embedded into a known conceptual model, provided better monthly runoff predictions. The model abbreviated as dynNN was applied to a critical station located at the Gediz Basin in western Turkey. To test the idea, which argues that combining the prominent strengths of a conceptual model and the ANN into a single model will perform well, the dynNN performances were compared to those obtained from dynwbm variants and ANN individually. The advantages and some limitations of the hybrid model proposed and other detections made are listed below.

1) The primary mastery of the offered hybridization over the ANN-based black-box models is that it better reflects the initial catchment conditions because the conceptual outputs processed in its ANN part are the functions of the simulated soil moisture content.

2) Besides, the fact that the dynNN does not require any antecedent data makes it sufficient to be used in the studies pertaining to streamflow projections under climatic change scenarios.

3) While an improvement in monthly runoff prediction was achieved with the dynNN, the main conceptual structure of the reference model was preserved so that the major fluxes, including actual evapotranspiration, recharge (except for baseflows), and soil moisture that may then serve as an additional information for the other exercises such as hydro-meteorological drought analyzes.

4) Although hybridization approaches in the literature usually exploit machine learning models like ANNs for the output updating correction of CRR models, the hypothesized approach refers to serial hybrid modeling, where conceptual and black box models are intertwined in one structure. In this respect, the study sheds light on the further evaluation of different conceptual models (daily and monthly models) and much more sophisticated machine learning techniques within the scope of serial hybridization. Such a comprehensive study is under review.

5) On the other hand, a limitation of dynNN is that it needs the expertise to code the coupled serial structure and the time to successively train the algorithm and pick out the ultimate solution among numerous trials.

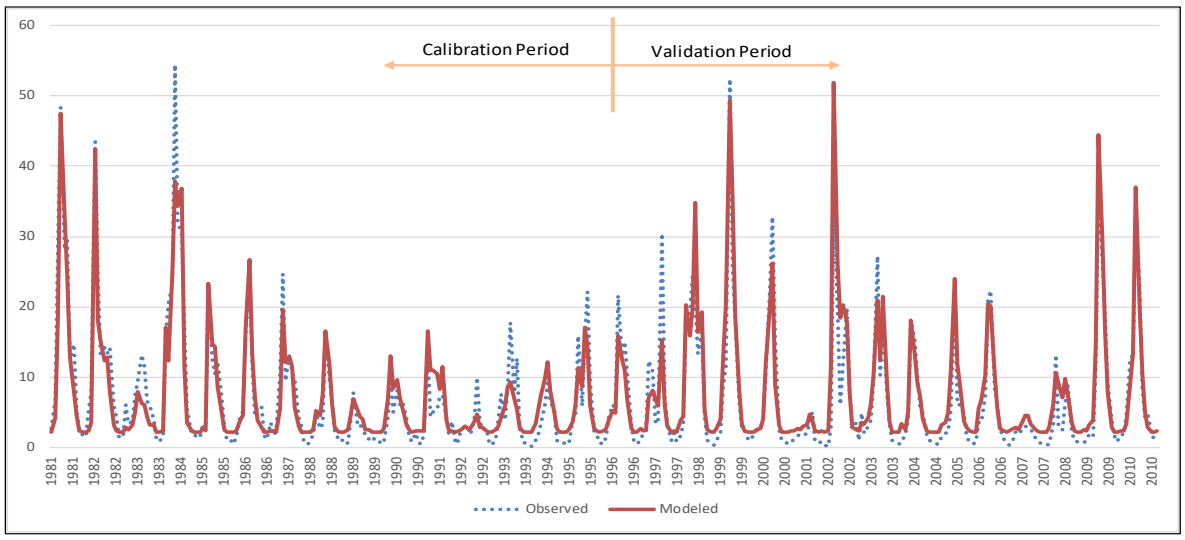


Fig. 4 Time series representation for simulations derived from dynNN model (in millimeters).

REFERENCES

[1] N. Noori and L. Kalin, "Coupling SWAT and ANN models for enhanced daily streamflow prediction" Journal of Hydrology, 533, 2016, pp.141-151.

[2] W.W. Ren, T. Yang, C.S. Huang, C.Y. Xu, Q.X. Shao, "Improving monthly streamflow prediction in Alpine regions: integrating HBV model with Bayesian neural network." Stochastic Environmental Research and Risk Assessment, 32(12), 2018, pp. 3381–3396.

[3] B. Zhang and R.S. Govindaraju, "Geomorphology-based artificial neural networks (GANNs) for estimation of direct runoff over watersheds." Journal of Hydrology, 273(1-4), 2003, pp.18-34.

[4] F. Anctil, C. Michel, C. Perrin, V. Andréassian, "A soil moisture index as an auxiliary ANN input for stream flow forecasting." Journal of Hydrology, 286 (1-4), 2004, pp.155–167.

[5] G.B., Humphrey, M.S. Gibbs, G.C. Dandy, H.R. Maier, "A hybrid approach to monthly streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial neural network." Journal of Hydrology, 540, 2016, pp.623–640.

[6] A.A. Kumanlioglu, O.Fistikoglu, O., "Performance enhancement of a conceptual hydrological model by integrating artificial intelligence." Journal of Hydrologic Engineering, 24 (11), 2019.

[7] L. Zhang, N. Potter, K. Hickel, Y. Zhang, Q. Shao, "Water balance modeling over variable time scales based on the Budyko framework - Model development and testing." Journal of Hydrology, 360(1-4), 2008, pp. 117–131.

[8] S. Tekleab, S. Uhlenbrook, Y. Mohamed, H.H.G. Savenije, M. Temesgen, J. Wenninger, "Water balance modeling of Upper Blue Nile catchments using a top-down approach." Hydrology and Earth System Sciences, 15(7), 2011, pp. 2179–2193.

[9] U. Okkan, U. Kirdemir, "Towards a hybrid algorithm for the robust calibration of rainfall-runoff models." Journal of Hydroinformatics, 22(4), 2020, pp. 876-899.

[10] D.N. Moriasi, J.G. Arnold, M.W. Van Liew, R.L. Bingner, R.D. Harmel, T.L. Veith, "Model evaluation guidelines for systematic quantification of accuracy in watershed simulations." Transactions of the ASABE, 50(3), 2007, pp. 885-900.

[11] D. de Almeida Bressiani, R. Srinivasan, C.A. Jones, E.M. Mendiondo, "Effects of spatial and temporal weather data resolutions on streamflow modeling of a semi-arid basin, Northeast Brazil." International Journal of Agricultural and Biological Engineering, 8(3), 2015, pp.125-139.

[12] N. Noori, L. Kalin, S. Isik, "Water quality prediction using SWAT-ANN coupled approach." Journal of Hydrology, 590, 125220.,2020.

Development of a weighted ensemble approach for prediction of blood glucose levels

Shashank Bhargav
Applied cognitive science lab
Indian Institute of Technology Mandi
Mandi, India
shashankbhargav55@gmail.com

Shruti Kaushik
RxDataScience, Inc
Durham, NC, United states
shrutikaushik15@gmail.com

Abhinav Choudhury
Applied cognitive science lab
Indian Institute of Technology Mandi
Mandi, India
abhinavchoudury.0490@gmail.com

Varun Dutt
Applied cognitive science lab
Indian Institute of Technology Mandi
Mandi, India
varun@iitmandi.ac.in

***Abstract*— Type 1 diabetes (T1D) is a serious global problem. Several differential equation models have been proposed to model the blood glucose levels (BGLs) in the human body. However, a comparison of prediction of BGLs via non-recurrent models (multilayer perceptron or MLP), recurrent models (long short-term memory or LSTM), and statistical models (vector auto-regression or VAR) has received relatively little attention. This research's primary objective is to address this literature gap and propose BGL predictions via non-recurrent, recurrent, and statistical models. A T1D dataset was compiled from a differential equation-based mathematical diabetes simulator, Automated Insulin Dosage Advisor (AIDA). A non-recurrent model (MLP), a recurrent model (LSTM), a statistical model (VAR), and a weighted ensemble model of the individual models were developed. In all models, the time-series BGLs were divided into 80% training and 20% test. Results revealed that the recurrent LSTM model outperformed the non-recurrent MLP and statistical VAR models during training and test to account for the BGLs from the AIDA model. Furthermore, the weighted ensemble model performed the best to account for the BGLs from the AIDA model among all models. Overall, machine learning models may provide an alternative to differential equation models for simulating healthcare variables. We illustrate the significance of using ensemble models for blood glucose predictions.**

Keywords*— *Long short-term memory (LSTM) model, time-series forecasting, blood glucose levels, multilayer perceptron (MLP), vector autoregression (VAR), weighted ensemble model, Automated Insulin Dosage Advisor (AIDA).

I. INTRODUCTION

Type 1 diabetes (T1D) is among the most prevalent diseases globally, primarily widespread in both developed and developing countries [1]. In India alone, approximately 30 million people are affected by T1D [2], induced by an adverse reaction that eliminates the cells that make insulin in the pancreas. Basic indications associated with this disorder are a fluctuation of blood levels that may lead to the subconscious or blackout-like disorders, cardiac arrest, permanent blindness, diabetic foot, and mental illness [2]. Diabetes complications can be controlled when patients regularly monitor their blood glucose levels while changing their insulin dosage. With the help of prior records consisting of blood glucose levels (BGLs), it may be possible to predict the future BGLs. This BGL prediction may help patients, and healthcare workers maintain the BGLs as close as possible to their usual levels. The BGL predictions may be influenced by lifestyle factors, complications, and response to treatment among individual patients [2].

For the prediction of BGLs, several computer-assisted differential equation models have been proposed [3]. For example, for patient and medical staff education, an Automated Insulin Dosage Advisor (AIDA) model of glucose-insulin interaction in T1D patients has been proposed [3]. This model aims to represent the underlying physiology of insulin action and carbohydrate absorption in quantitative terms, such as insulin sensitivity. Struble et al. [4] trained a preliminary feed-forward neural network on the AIDA models' predictions using a window model. For a 15-minute prediction horizon, these authors obtained an optimal model for each of the ten patients developed using AIDA.

An alternate approach to predicting BGLs could be via machine learning or statistical models [5]. Various prior approaches for glucose prediction have been suggested in the literature [6]. These approaches have focused on statistical models like autoregression (AR), moving average (MA), and ARMA models [7]. Some of the machine learning models investigated were artificial neural networks (ANNs) and support vector machines (SVMs) [7]. For example, Razavian et al. compared neural networks and logistic regression models with several hand-engineered and clinically relevant features [8]. Razavian et al. Used ANNs 15-minutes ahead of time for predicting BGLs. These authors used actual continuous glucose monitoring (CGM) data for making predictions for 12 patients. The feasibility of NN models to predict BGLs was also demonstrated by Razavian et al. [8]. These authors used a predictive window that varied in the 50-180 minute range [8]. Uh, Sandham et al. [8] Implemented an ANN model for predicting BGLs for two female patients [9]. These authors' objective was to explore the effectiveness of ANNs to educate and counsel patients with T1D for short-term therapy. A related study [9] was conducted to provide short-term therapeutic advice where RNN models predict BGLs. Karim et al. [10] used LSTMs to predict BGLs for up to one hour. The model requested only the T1D patient's glucose history [10].

Although the literature has suggested using statistical, ANN, and RNN models, a comprehensive evaluation of statistical, non-recurrent, and recurrent models has lacked in literature. Moreover, the investigation and benchmarking of ensemble models that combine the predictions of different statistical, non-recurrent, and recurrent models is yet to be performed. This research's primary objective is to address such gaps in the literature and formulate a comprehensive study of the statistical, non-recurrent, recurrent, and ensemble models in their ability to predict BGLs. Specifically, we develop and compare a statistical vector auto-regression or VAR model, a non-recurrent multilayer perceptron (MLP)

model, an LSTM model, and a weighted ensemble model consisting of the individual models. We approach the BGL prediction as a time-series forecasting problem since BG measurements have a natural temporal ordering. The data used for model prediction is obtained from the AIDA model. It comprises BGL measurements taken at 15-minute intervals by a CGM system involving insulin injections, meals, exercise, and sleep. The ensemble model is developed using a grid search method that takes a weighted average of individual statistical, non-recurrent, and recurrent models. Here, ensembling is performed on a large dataset by considering both statistical, non-recurrent, and recurrent models. In the next section, we discuss the background literature relevant to this study. The background is followed by data, models, results, and discussion.

II. Background

Prior research has investigated certain statistical models for BGL predictions [11]. For example, Zhao et al. [11] developed a statistical model named latent variable (LV) to predict the glucose levels and forecast future glucose concentrations. The LV model was compared to an AR model and an AR exogenous input model. However, no benchmarking against different machine learning models was investigated. Similarly, Zhao et al. [11] analyzed the inter-individual variability of the underlying glucose levels and the relative predictive power of different frequency band exogenous inputs for online subcutaneous glucose predictions. However, no benchmarking against ML models was investigated.

Certain researchers have proposed non-recurrent and recurrent models for glucose predictions. For example, Karim et al. [10] proposed end-to-end training of recurrent neural network-based models, needing nothing but the history of the patient's glucose level. The models generated the prediction and estimated its certainty, enabling users to interpret the levels expected. However, benchmarking was limited, and no ensemble approaches were investigated.

Jianwei et al. [12] proposed a new deep learning technique based on a dilated recurrent neural network (DRNN) model. The model had predicted future glucose levels. However, benchmarking with some other models and evaluation of an ensemble model was still absent.

Similarly, Quchani et al. [13] compared MLP and Elman neural networks' performance to predict glucose level in T1D. They proposed that the Elman model might be effective in predicting the T1D in long-term blood glucose levels. Also, an improvement in prediction accuracy was shown by RNN models. Furthermore, Quchani et al. [14] built a method that uses some variables from previous intervals and blood glucose level as input and correctly predicted blood glucose level for the next interval. Chouikhi et al. [15] compared the radial basis function (RBF) neural network with the other-implemented ANNs such as MLP. Results revealed that the RBFNN technique provided excellent results over the MLP neural network, and the RBFNN provided better generalization. However, again the benchmarking was limited, and no ensemble approach was investigated.

This research aims to find the best performing model for predicting BGLs to overcome the gaps in the literature. Thus, a new ensemble model is developed using the non-ensemble models in this paper. The predictions from MLP, LSTM, and VAR models are provided as inputs to a final ensemble model.

III. Data

A. Data description

Data include the blood glucose levels of 40 T1D patients over a 24-hour time-period at 0.25-hour time steps. Data were split into a training data set (80 percent) and test data set (20 percent) for each patient. To obtain the collection of multi-patient data, training, and test samples for all patients were stitched together, respectively. Table 1 gives details about the 13 input features, including time, present in the data. The attribute to be predicted was the blood glucose level of patients over time. We first converted our time-series into X, Y pairs, such that for a vector X of past values passed into a model, we got a 0.25-hour look ahead prediction for the glucose values vector Y for a patient at a certain time-step.

Fig. 1 shows the stitched time-series of BGLs (overall 40 patients). The x-axis in Fig. 1 shows the time-series value of all 40 patients from the dataset. In the dataset's construction in Fig. 1, we had to first stitch the 80% training values across all 40 patients and then stitch the remaining 20% test values across all 40 patients. In Fig. 1, the vertical line shows the training and test data split.

TABLE I. INPUT FEATURES FROM AIDA MODEL

Attribute	Attribute Description
Weight (Kg)	Weight of every patient measured in kilograms. (Min: 60; Max: 99)
Time (hours)	Multiples of 0.25, starting from 0 to 24.
Carbohydrates (grams)	Carbohydrates are the sugars, starches, and fibers found in the meal intake of patients. (Min: 0; Max: 80)
Short-acting injections (units)	The injection took 15-minutes before the meal by the patient. The injection has its max effect of two to five hours, and the effect lasts for six to eight hours. (Min: 0; Max: 25)
Intermediate & long-acting injections (units)	The injection took 15-minutes before the meal by the patient. These injections are to lower blood glucose levels. They have max effects from 24 to 48 hours. (Min: 0; Max: 40)
Type of medicines (units)	These are the type of medication used by patients for T1D control.
Kidney function RTG (mmol/l)	The renal glucose threshold (RTG) is the blood glucose concentration at which the kidneys start to excrete glucose into the urine. (Min: 7; Max: 11)
Kidney Renal function (mL/min)	Renal activity is essentially a test of how well the kidneys operate. (Min: 40; Max: 100)
Insulin Sensitivities Liver (mmol/l)	The normalized values reflect the sensitivity of the liver and periphery to insulin.
Glucose Upper/lower Limit (mmol/l)	These higher and lower limits have Glucose values (Min: 4; Max: 10).
Blood glucose level (mmol/l)	Variable to be predicted for every patient over time. (Min: 2; Max: 15.7)

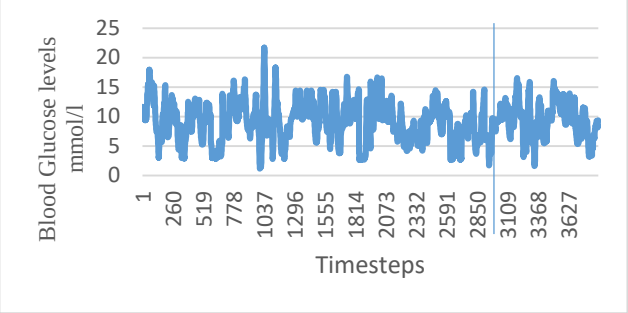


Fig.1. Time-series plot of BGLs across patients over 24 hours. The vertical line separates the training and test datasets.

B. Models

MLP The multilayer perceptron is a state-of-the-art model that has been used for both classification and regression [15]. A basic MLP model was constructed with an input layer, an output layer, and a hidden layer. Except for the input node, every node in the network took the input from the previous layer and forwarded the output to the next layer after applying the activation function. It utilized the backpropagation technique in training to minimize the error in the prediction [15]. The node inputs were multiplied with the weights (W), which were optimized based on the error (the difference between the actual value and predicted value) of the network and the backpropagation technique during the model's training. We varied the number of layers, lookback period, and nodes per layer in the MLP model.

VAR Model the VAR is an algorithm used for multivariate forecasting when two or more time-series affect each other. The time-series is modeled as a linear combination of its lags in autoregression models. To predict future values, the present and past values in a sequence are used.

$$Y_t = \alpha + \beta_1 Yt_{-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t \cdot \quad (1)$$

Where the intercept is represented by α and the coefficients of lags of Y are represented by the β1, β2 ..., βp. Y's lookback period is represented by 'p' in the equation, and the number of 'p' values represents the number of predictors in the equation. The ϵ_t was the error term. We varied the p order in the VAR model.

LSTM The LSTM is a type of RNN model with the ability to remember values from earlier stages. The cell state in an LSTM acts as a conveyor belt (ct), which helps unaltered information flow through the units with only a few linear interactions [16]. Each unit has an input (it), output (ot), and a forget (ft) gate, which can add or remove information to the cell state. The forget gate decides what information from the previous cell state needs to be overlooked, for which it uses a sigmoid function. The input gate controls the information flow to the current cell state using a point-wise multiplication operation of sigmoid and tanh. Finally, the output gate decides which information should be passed on to the next hidden state (ht). The sigmoid function outputs numbers between zero and one, describing how much each component should be let through [16]. We varied the LSTM model's three parameters, namely lookback period, number of layers, and number of nodes per layer.

Ensemble model the ensemble model utilized various individual models to forecast an outcome [17]. Using a weighted average approach in the ensemble model, we ensembled the individual models (such as MLP, LSTM, and ·VAR) [17]. Each model was assigned a weight to decide the contribution of the model in the prediction. To optimize the hyperparameter in the ensemble model, a grid search method was implemented, where the weights of individual models were estimated. Thus, weights were assigned to individual models such that the ensemble model produced the lowest training error. Equation 6 illustrates how the prediction was achieved using weights in the ensemble model.

$$y_t = W_{MLP} + m_{MLP} + W_{LSTM}\, m_{LSTM} + W_{VAR}\, m_{VAR} \quad (2)$$

Where, y_t was the output (prediction) of the ensemble model; W_{MLP}, W_{LSTM}, and W_{VAR} were the values of weights (between 0 and 1) of each model MLP, LSTM, and VAR, respectively; and, m_{MLP}, m_{LSTM}, and m_{VAR} were the predictions of the MLP, LSTM, and VAR models, respectively.

C. Optimization of Model Parameters

Machine learning models were optimized using a grid search method. The grid search method used an exhaustive search to find the best parameter within the given range. Table 1 displays the different set of parameter values adopted in different models. The hyperparameters for MLP were varied in following ranges: lookback period (1, 2, 3, 4, 5, 6, and 12); number of hidden layers (1, 2, 4, 6, 8, 16, 32, and 64); and, nodes per layer (1, 3, 6, 12, 25, 50, 75, and 100). In LSTM, parameters were varied as: lookback period (1, 2, 3, 4, 5, 6, and 12); number of hidden layers (1, 2, 4, 6, 8, 16, 32, and 64); and, nodes per layer (1, 3, 6, 12, 25, 50, 75, and 100). In the VAR model, the p-parameter was varied as integers between 0 and 12. In the ensemble model, the weights were varied between 0 and 1 in steps of 0.01.

TABLE II. PARAMETER OPTIMIZATION OF MODELS

Model	Parameter	Range of values
MLP	Lookback period	1, 2, 3, 4, 5, 6, 12
	Hidden layers	1, 2, 4, 6, 8, 16, 32, 64
	Nodes per layer	1, 3, 6, 12, 25, 50, 75, 100
LSTM	Lookback period	1, 2, 3, 4, 5, 6, 12
	Hidden layers	1, 2, 4, 6, 8, 16, 32, 64
	Nodes per layer	1, 3, 6, 12, 25, 50, 75, 100
VAR	p-order	[0, 12]
Ensemble	wMLP	[0, 1] in steps of 0.01
	wLSTM	[0, 1] in steps of 0.01
	wVAR	[0, 1] in steps of 0.01

Input and outputs of the model We first converted our time-series into X, Y pairs. For a vector X of past values passed into a model, we got a 0.25-hour look ahead prediction for

the glucose values vector Y for a patient at a certain time-step. Thus, corresponding to the vector Y, the model's prediction was based upon past values X. The X vector consisted of initial n time values of glucose for the corresponding patient.
An X, Y pair was defined as a packet, and several such packets were created across different model attributes. These packets were shuffled before inputting them into a model.

Accuracy measure The problem of glucose prediction is a regression problem, where a floating value is predicted for the resulting BGL. Thus, between the real value and the predicted value of BGLs, the root means square error (RMSE) was estimated. The RMSE was computed as:

$$RMSE = \sqrt{\sum_{i=1}^{n}(Pred_i - Real_i)^2 / n} \qquad (3)$$

Where, the $Real_i$ is the actual value from the AIDA model, the $Pred_i$ is the value predicted by the model under investigation, and n is the total number of data points.

IV. Results

The best hyperparameter values have been reported for different models in training data in Table III. The best parameters for MLP were lookback period: 4, hidden layers: 4, and nodes per layer: 75. The optimized parameters for LSTM were defined as: lookback period: 4, hidden layers: 4, and nodes per layers: 75. Furthermore, the best value of p in the VAR model was 2. Lastly, the ensemble model's best weights were: W_{MLP}: 0.0, W_{LSTM}: 0.43, and W_{VAR}: 0.57.

TABLE III. Optimized parameters for models

Models	**Optimized parameters**
MLP	Lookback period: 4, hidden layers: 4, nodes per layer: 75
LSTM	Lookback period: 4, hidden layers: 4, nodes per layers: 75
VAR	p-value: 2
Ensemble Model	W_{MLP}: 0.0, W_{LSTM}: 0.43, W_{VAR}: 0.57

Table IV shows the RMSEs from different models in training data. The ensemble model obtained the lowest RMSE among all models, which was observed to be 3.9511. Among all models, the LSTM performed the second-best with an RMSE of 3.9808. The LSTM was followed by the VAR model (RMSE of 4.7294) and MLP model (RMSE of 7.4951).

TABLE IV. RMSEs in different models in the training dataset.

Model	**RMSE in the training dataset (mmol/l)**
MLP	7.4951
LSTM	3.9808
VAR	4.7294
Ensemble Model	3.9511

Table V shows the RMSEs from different models in test data. The ensemble model again obtained the lowest RMSE among all models (= 4.5871). Among all models, the LSTM performed second-best (RMSE of 5.6673), and it was followed by the MLP model (RMSE of 6.4065) and VAR model (RMSE of 7.1578). Fig. 2 and 3 show the predictions from ensemble and LSTM models in test data.

TABLE V. The RMSEs of different models in the test dataset.

Model	**RMSE in test dataset (mmol/l)**
MLP	6.4065
LSTM	5.6673
VAR	7.1578
Ensemble Model	4.5871

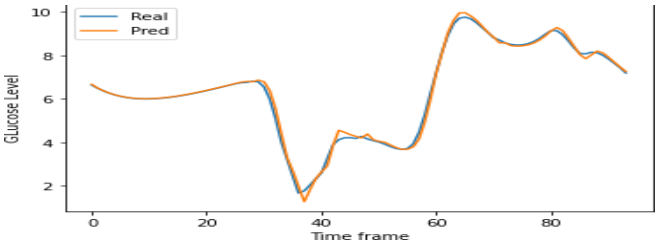


Figure 2: Predicted and the actual value obtained from the ensemble model in test data.

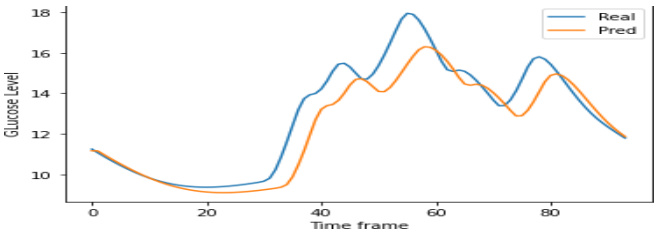


Figure 3: Predicted and the actual value obtained from LSTM model in test data.

V. Discussion and Conclusion

The number of people with type 1 diabetes (T1D) is growing worldwide, and due to vascular complications, most people with T1D may suffer from severe consequences [18]. This research aimed to develop ensemble forecasting models that could combine individual machine learning predictions from statistical, non-recurrent, and recurrent models with a high degree of accuracy and predict blood glucose levels (BGLs). We used the T1D data set provided by the AIDA model for our model development, training, and testing. Our study revealed that BGL prediction could be carried out by different statistical (VAR), non-recurrent (MLP), recurrent (LSTM) models. Also, both in the training and test datasets, the weighted ensemble model performed the best, integrating the individual prediction models. The Root Mean Square Error (RMSE) for both the training and test data were small in the ensemble model.

First, we found that compared to the statistical (VAR) model, the machine learning models (e.g., MLP, LSTM) performed better in test data. A probable explanation for this outcome may be the presence of non-linearity (MLP and LSTM) and memory capabilities (LSTM) available in machine learning models. In contrast, the statistical (VAR) model possessed a linear structure.

Second, the weighted ensemble model performed better than all other models (LSTM, MLP, and VAR). This finding agrees with prior literature, where ensemble models have also

been found to perform better than individual models [18]. It may also be that the weighting of predictions in the ensemble model enabled this model to take the best characteristics of individual models in the ensemble models.

For the real world, this work has some implications. For example, if the BGLs exceed predefined thresholds, then using the ensemble model, one could promptly warn doctors and patients about upcoming fluctuations in BGLs. This timely notice could enable T1D patients to avoid many dangerous diseases and strokes. The timely warning may also help the doctors reduce the patient's problem by early diagnosing the problem. Furthermore, the models built can be implemented in a production environment at a minimal cost in health centers, where BGLs are assessed via sensors in real-time.

Future work could capitalize on this research and development and compare multivariate models to predict BGLs. Values of other variables such as carbohydrates, insulin injections, and meal intake time can be used in such models and previous glucose values (mmol/l) to achieve higher predictions. Here, it is possible to compare the established multivariate individual and ensemble models with univariate individual and ensemble models. In addition to connectionist models for predicting BGLs, another priority may be improving longer-term blood glucose prediction several hours ahead of time. We would like to try some of these ideas in our research program concerning the prediction of BGLs.

ACKNOWLEDGMENT

This research was possible because of funding by RxData-Science Inc., USA, on project IITM/CONS/RxDSI/VD/33 to Varun Dutt. We are also thankful for the computational support from the Indian Institute of Technology Mandi, Himachal Pradesh, India.

REFERENCES

[1] Borchers, Andrea T., Raivo Uibo, and M. Eric Gershwin. "The geoepidemiology of type 1 diabetes." Autoimmunity reviews 9, no. 5 (2010): A355-A365.

[2] Bluestone, Jeffrey A., Kevan Herold, and George Eisenbarth. "Genetics, pathogenesis and clinical interventions in type 1 diabetes." Nature 464, no. 7293 (2010): 1293-1300.

[3] Magni, Paolo, and Riccardo Bellazzi. "A stochastic model to assess the variability of blood glucose time series in diabetic patients self-monitoring." IEEE Transactions on biomedical engineering 53, no. 6 (2006): 977-985.

[4] Struble, Nigel. "Measuring Glycemic Variability and Predicting Blood Glucose Levels Using Machine Learning Regression Models." PhD diss., Ohio University, 2013.

[5] Bagherian, Maryam, Elyas Sabeti, Kai Wang, Maureen A. Sartor, Zaneta Nikolovska-Coleska, and Kayvan Najarian. "Machine learning approaches and databases for prediction of drug–target interaction: a survey paper." Briefings in bioinformatics (2020).

[6] Andreassen, Steen, Jonathan J. Benn, Roman Hovorka, Kristian G. Olesen, and Ewart R. Carson. "A probabilistic approach to glucose prediction and insulin dose adjustment: description of metabolic model and pilot evaluation study." Computer methods and programs in biomedicine 41, no. 3-4 (1994): 153-165.

[7] Shi, Jing, Jinmei Guo, and Songtao Zheng. "Evaluation of hybrid forecasting approaches for wind speed and power generation time series." Renewable and Sustainable Energy Reviews 16, no. 5 (2012): 3471-3480.

[8] Razavian, Narges, Jake Marcus, and David Sontag. "Multi-task prediction of disease onsets from longitudinal laboratory tests." In Machine Learning for Healthcare Conference, pp. 73-100. 2016.

[9] Sandham, William, Dimitra Nikoletou, David Hamilton, Ken Paterson, Alan Japp, and Catriona MacGregor. "Blood glucose prediction for diabetes therapy using a recurrent artificial neural network." In 9th European Signal Processing Conference (EUSIPCO 1998), pp. 1-4. IEEE, 1998.

[10] Karim, Rebaz AH, István Vassányi, and István Kósa. "After-meal blood glucose level prediction using an absorption model for neural network training." Computers in Biology and Medicine 125 (2020): 103956.

[11] Zhao, Chunhui, and Yongji Fu. "Statistical analysis based online sensor failure detection for continuous glucose monitoring in type I diabetes." Chemometrics and Intelligent Laboratory Systems 144 (2015): 128-137.

[12] Chen, Jianwei, Kezhi Li, Pau Herrero, Taiyu Zhu, and Pantelis Georgiou. "Dilated Recurrent Neural Network for Short-time Prediction of Glucose Concentration." In KHD@ IJCAI, pp. 69-73. 2018.

[13] Quchani, S. A., and Ehsan Tahami. "Comparison of MLP and Elman neural network for blood glucose level prediction in type 1 diabetics." In 3rd Kuala Lumpur International Conference on Biomedical Engineering 2006, pp. 54-58. Springer, Berlin, Heidelberg, 2007.

[14] Chouikhi, Naima, and Adel M. Alimi. "Adaptive extreme learning machine for recurrent beta-basis function neural network training." arXiv preprint arXiv:1810.13135 (2018).

[15] Dreiseitl, Stephan, and Lucila Ohno-Machado. "Logistic regression and artificial neural network classification models: a methodology review." Journal of biomedical informatics 35, no. 5-6 (2002): 352-359.

[16] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991 (2015).

[17] Kamal, Imam Mustafa, Hyerim Bae, Sim Sunghyun, and Heesung Yun. "DERN: Deep Ensemble Learning Model for Short-and Long-Term Prediction of Baltic Dry Index." Applied Sciences 10, no. 4 (2020): 1504.

[18] Susan van, Dieren, Joline WJ Beulens, Schouw Yvonne T. van der, Diederick E. Grobbee, and Bruce Nealb. "The global burden of diabetes and its complications: an emerging pandemic." European Journal of Cardiovascular Prevention & Rehabilitation 17, no. 1_suppl (2010): s3-s8.

Underlying Concepts and Understandings of Internet of Things (IoT): Case Study

Md Sarwar Morshedul Haque
Dept. of PYP - Computer Science
King Fahd University of Petroleum & Minerals
Dammam, Kingdom of Saudi Arabia
smhaque@kfupm.edu.sa
ORCID 0000-0003-4916-0788

Md Rafiul Hassan
College of Arts and Sciences
University of Maine at Presque Isle
Presque Isle
04769 ME, USA
md.hassan@maine.edu

Mohammad Kamal Hossain
Center of Research Excellence
King Fahd University of Petroleum & Minerals
Dammam, Kingdom of Saudi Arabia
kamalhossain@kfupm.edu.sa
ORCID 0000-0001-9264-3828

Sk Md Mizanur Rahman
Dept. of Info. and Comm. Eng. Technology
Centennial College
Toronto, Canada
srahman@centennialcollege.ca
ORCID 000-0002-9166-6830

Md Arifuzzaman
Dept. of Civil and Env. Engineering
King Faisal University
Hofuf, Kingdom of Saudi Arabia
marifuzzaman@kfu.edu.sa
ORCID 0000-0002-5069-0447

***Abstract*—Internet of Things (IoT) is an emerging technology. Many different standardization bodies, national initiatives, research projects and industries, etc, discussed the concept of IoT. In this study, we did a thorough analysis of different studies that discussed the concepts of IoT. We found that there are some common aspects between studies in addition to differences. Some of the studies foresee the IoT as a network of things or items that will be available anywhere and anytime. Things will work as a node of the network and can be accessed by anyone from any place. As different studies discussed different perspectives of IoT, we believe that a sound and standard understanding of the concept of IoT is important to develop and improve the technologies of IoT.**

***Index Terms*—Internet of Things, Events, Connectivity, Data Sensing Protocols, Availability of Network.**

I. Introduction

Internet of Things (IoT) is an arising technology. From last decade it has been in the spotlight. It is regarded as one of the innovative technologies of this century [1]. Near future IoT technology will be used by billions of devices. It is predicted that there will be 30 billion IoT devices by 2020 [2]. In the year 2017, the number of IoT-enabled devices increased by 31% in number 8.4 billion. The market value is expected to reach $7.1 trillion by 2020 [3].

IoT has got the attention of academy, industry and society because this will technologically enhance our daily activities, create new business models, services and products. Many alliances, enterprises, organizations and governments have recognized its significance and recognized its potential benefits, which leads them to embark on initiatives and undertake strategic projects targeting to grow this field for generating profits out of it and undertake strategic projects [4, 5, 6, 7, 8].

The fast multiplication of web associated gadgets with ascent of the IoT provides extraordinary expectation. These recently associated gadgets bring the guarantee of upgraded business efficiencies and expanded consumer loyalty. IoT includes broadening Internet network past standard gadgets, for example, work areas, workstations, cell phones and tablets, to any scope of customarily stupid or non-web empowered actual gadgets and ordinary articles. Installed with innovation, these gadgets can impart and cooperate over the Internet, and they can be distantly checked and controlled. With the appearance of driver less vehicles, a part of IoT, for example the Internet of Vehicle begins to acquire consideration [9].

People use IoT as a general term. But what does the IoT really means? Many different bodies / studies tried to describe IoT in their papers. They tried to define IoT in different ways. For example, IEEE discusses, IoT will connect billions of item or things, which will be embedded with sensors and connects to internet. IETF considers, IoT will connect all kinds of objects or things, which will be embedded with RFID tags, actuators or sensors. These objects will be available anywhere and anytime.

If we notice carefully, there are lot of similarities, in the understanding of IoT among these bodies / studies. However, few studies have extensively described what elements could be considered as part of the IoT. Few studies have only briefly described about IoT. Eventually they meant to be similar (their understandings of IoT have lots of similarities) but, there are some vocabulary differences among the understanding of IoT.

In this study, we have studied many different literatures originated by academics, industrial bodies, books etc. Initially, we group the studies based on organization Standardization bodies, National Initiatives, Research Projects, White Papers,

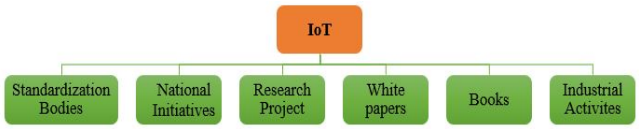


Fig. 1. Grouping of different bodies/studies that discussed about Iot

Books, Industrial Activities. We understand that this organizations have extensively done research on IoT, analyze the IoT and they tried to attempt to characterize what elements or things can make IoT infrastructure. In this study, we attempt to summarize different studies. How they have defined IoT and in what aspect they are similar and not similar. We also find the main important aspects of IoT, the common understanding of IoT. In this paper, we conduct an extensive analysis of the understanding of IoT in terms of what elements and/or objects can be considered as part of the IoT. The contribution of this paper could be pointed out as follows:

- Analysis of common features of IoT as characterized by different studies and organization.
- Analysis of the understanding of IoT by different studies and organization that contrasts in terms of features of different technologies, protocols.
- Grouping of different studies and organizations based on similar and dissimilar aspects of IoT as discussed by them.
- Identification of underlying four key characteristics of IoT namely Events, Connectivity, Data sensing protocol and Availability of network.

II. LITERATURE REVIEW

In this section, we discuss the understanding of IoT given by different Standardization bodies, National Initiatives, Research Projects, White Papers, Books, Industrial Activities.

Many standardization bodies, research projects, national projects and industrial activities are conducting research in the field of Internet of Things [10, 11].

Figure 1, lists the names of the different studies, that we consider in this paper.

Standardization bodies: IEEE, ITU, ITU-T Study Group 13, ISO/IEC, IETF, ETSI, NIST, OASIS, W3C and ZigBee Alliance

National Initiatives: UK - Future Internet Strategy Group (UK FISG), Australia – CSIRO, Finland - Internet of Things Strategic Research Agenda (IoT-SRA), India and Malaysia – Digital Lifestyle Malaysia (DLM)

Research Projects: CASAGRAS Project, Berkeley University, IoT-A Project, IoT Special Interest Group, CERP-IoT Project, IERC, ETP EPoSS Project and iCore.

White Papers: “From the Internet of Computers to the Internet of Things”, “Future Internet”, “The Internet of Things: Networked objects and smart devices”, “The Internet of Things”, “The Software Fabric for the Internet of Things”, “The Internet of Things: In a Connected World of Smart Objects”, “China’s Initiative for the Internet of Things and Opportunities for Japanese Business”.

Books: Architecting the Internet of Things, The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications, Internet of Things: Legal Perspectives, 6LoWPAN: The Wireless Embedded Internet, Internet of Things: Global Technological and Societal Trends from Smart Environments and Spaces to Green ICT.

Industrial Activities: SAP, CISCO, HP and Google.

After analyzing all the literature of different studies and organization as mentioned above, a taxonomy tree is drawn which is shown in figure 2. The tree summarizes the key aspects of IoT as per the respective studies.

As shown in the tree 2, the understanding of IoT provided by different bodies/studies, varies in terms of different keywords. For example, IEEE emphasized understanding of IoT in terms of network of items, embedded with sensors and internet [12]. According to IEEE, IoT’s true value lies in the data that the interconnected items share. IoT can improve the operations of many things, e.g., highways, hospital managements, shipments managements. However, to get the benefit of IoT that they consider, several challenges are there: e.g., much intelligent sensor is required with its own IP address along with the location, big data analysis, data privacy preservation, etc. They demand to redefine the current definition of privacy to include collecting our daily life data, who owns the data etc. However, the current sensors and network capabilities can delineate the requirement as IEEE demands. Eventually, the full potential of IoT’s benefit is still to be explored/enjoyed with the existence of a very large network containing all the small networks and the large network can accommodate connections with products, systems, and machines, throughout the world [12].

International Telecommunication Union (ITU) is the United Nations specialized organization for information and communication technologies (ICTs). ITU defined IoT in terms of ubiquitous network and availability of anything with its location and the specific time. A ubiquitous network means the network will be available anytime and everywhere [13]. According to ITU, anywhere means on the move, outdoors, indoors and at the PC. Anytime can be on the move, outdoors and indoors, night and daytime. Anything might be between PCs, Human-to-Human(H2H), not using PC, Thing-to-Thing(T2T), Human-to-Human(H2T). The challenge is to design a network (ITU refer as next generation network) that will be able to connect billions of devices/things where computer devices will be embedded to the everyday objects. In this report, ITU also described different technologies for the realization of the IoT. For example: RFID for “tagging things”, sensor technology for “feeling things” and nanotechnology for “shrinking things”. ITU addresses some challenges, i.e., Standardization and harmonization, Harmonizing the Internet of Things and privacy implications [13].

We further notice that ITU-T study group defined IoT as a worldwide infrastructure with focus towards connecting physical things and virtual things for the information society. According to them, the physical and virtual things have identities, physical attributes and virtual personalities with smart interfaces and these things are integrated into the information

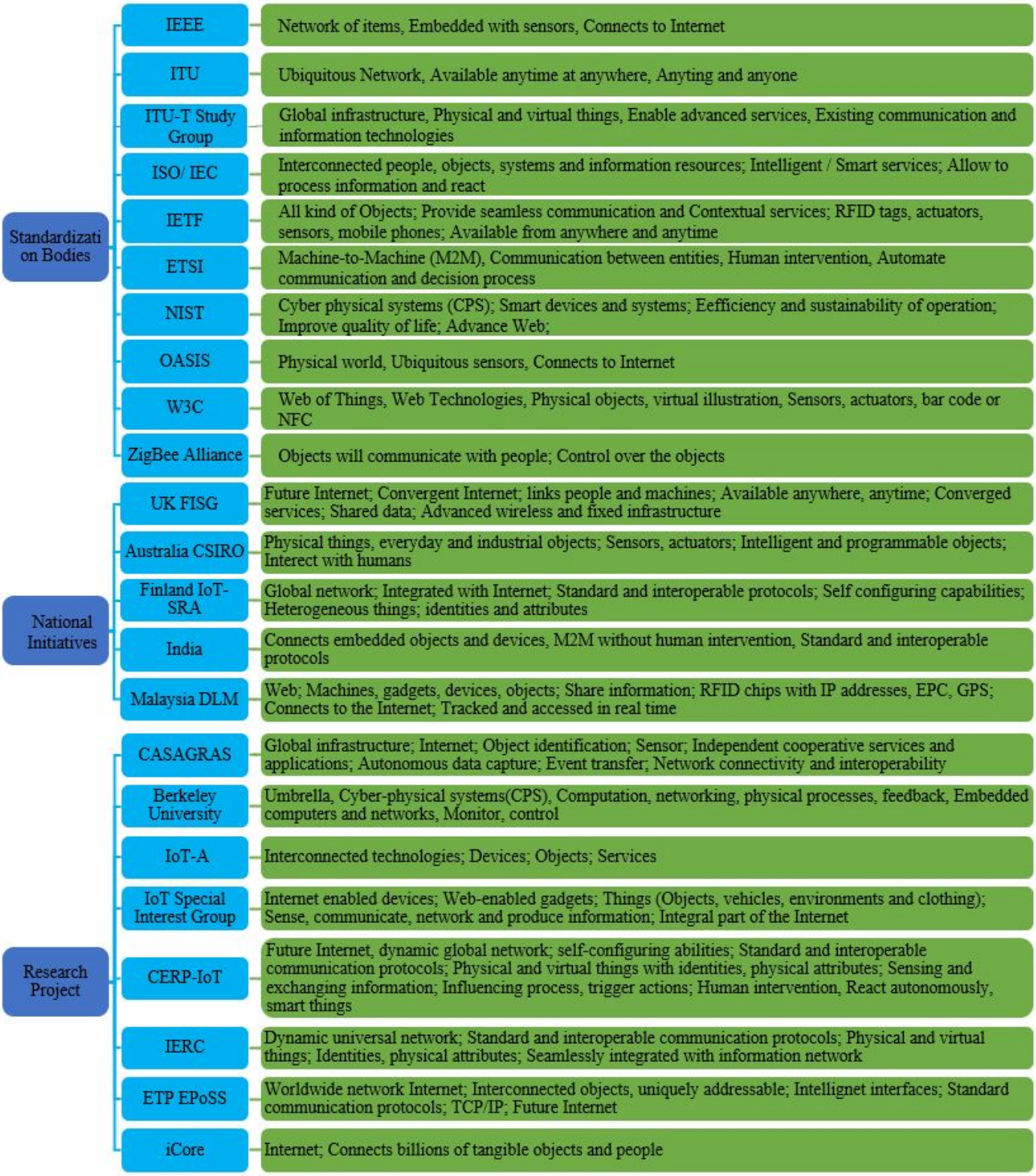


Fig. 2. Tree of the understanding of IoT by different bodies/studies - part 01.

Fig. 3. Tree of the understanding of IoT by different bodies/studies - part 02.

network flawlessly. This standardization body also puts focus towards enabling the advanced services. Advanced services refer to the process of identification exploitation, capturing and processing data etc. According to the ITU-T, IoT also involves communication and information technology [14].

From the tree 2 we also identify that, in terms of the understanding of IoT by ISO/IEC, IoT interconnects people, objects, systems and information resources through intelligent services. The information services facilitate processing of information collected through physical world and takes necessary action accordingly [15].

IETF considers IoT as a means to connect all kinds of objects/things (e.g., electronic, electrical, non-electrical). The things have RFID tags, actuators or sensors and can also represent people, machine, and information etc. An individual thing must possess a unique address. These objects are accessible anytime from anywhere [16].

ETSI defines IoT as Machine-to-Machine (M2M). In such environment, there is always communication among entities and the communication can be initiated following an automatic decision. These automated M2M concepts can be applied in different applications, for example smart metering, fleet management and home automation etc [17].

The NIST is one of the oldest laboratories in the area of physical science situated in USA. This laboratory is a part of the U.S. Department of Commerce. NIST defined IoT as

Cyber-physical systems (CPS). IoT and CPS connect smart devices and systems in different areas such as manufacturing, energy, healthcare and transportation. As per NIST, IoT enhances the efficiency and sustainability of operation to improve the quality of life [18].

Advancing open standard for the information society (OASIS) introduced IoT as the system where the Internet is connected to the physical world by using ubiquitous sensors (i.e., sensors that are available in all real-world things e.g. all household things, every device, door, room, every vehicle, every building in city and village, every office, every mobile on earth and every sensor in every device such as in every bed, chair or bracelet, in every home, office, building or hospital room, in every city and village on Earth) [19].

The tree in figure 2 reveals that, W3C considers IoT as an event pertaining to Web of Things and connects physical objects and their virtual illustration which consist of Sensors, actuators, bar code or NFC. As they define IoT, it uses Web Technologies (e.g., HTTP and JavaScript APIs) to connect things. These web technologies are used to facilitate the development of applications and Services for IoT [20].

According to ZIGBEE ALLIANCE, objects communicate with each other and with people. Objects can refer to variety of devices e.g., battery-operated device, bulbs, shopping carts, thermostats, smart meters, refrigerators and more. According to them, people have more control over the objects around them and objects communicate and control each other to help people [21].

UK FISG defines IoT as "future internet". The term "future internet" generally refers to the new architecture of the Internet. UK FISG also mentions that, IoT is a developing convergent Internet of things and services which links people and machines and offers converged or advanced services to people and businesses. According to UK FISG, the network is made up of services, data and an advanced wireless and fixed infrastructure. This network is also available anywhere at any time [22].

From the tree 2 we also notice that, in terms of the understanding of IoT by CSIRO (Australia), IoT interconnects physical things e.g. industrial objects that are used on daily basis. According to them these physical things should be embedded with sensors and actuators. They also mention that, the things should be intelligent and programmable, are capable of interacting each other as well with humans [23].

As shown in the tree 2, IoT-SRA refers the IOT as a worldwide network structure that connects heterogeneous things. Such as personal devices those use wearable wireless sensors or home appliances those integrate sensors. These heterogeneous things must pose physical and virtual attributes, and identities. Standard and inter-operable protocols should be used in the worldwide network and the system is capable of self-configuration (like plug and play) [24] [25].

As per the government of India (Ministry of Communication and Information Technology), IoT should interconnect embedded objects and devices. The objects and devices that are considered as part of the IoT should have identities. Within the IoT infrastructure, the communication between M2M can be established automatically, i.e., there is no need of human intervention. Similar to IoT-SRA, Government of India also defined IoT such that, the existing standard and interoperable communication protocols are used in connecting things with the IoT infrastructure. They consider few popular electronic devices should be part of the IoT. Examples of such devices are: PCs, tablets and phones [26].

Malaysia (DLM) defines IoT as a web which interconnects machines, gadgets, devices, daily appliances and inanimate objects. These interconnected things should have individual RFID tags. These tags are linked with unique IP addresses. Moreover, electronic product codes (EPC), GPS systems and near-field communication technologies should also be used in IoT things. The objects/things should be connected to the existing Internet. There should be technology to trace and record the IoT things and their behavior, in real time. The recorded data should be analyzed for further decision in real time [27].

As shown in the figure 2, CASAGRAS project defines IoT as global infrastructure which includes existing and evolving Internet and network developments [28]. It links physical and virtual objects. Specific object-identification, sensor and connection capability, autonomous data capture, event transfer, network connectivity and interoperability technologies are used. Independent services and applications are developed using these technologies.

Berkeley University project considers IoT under the umbrella of Cyber Physical Systems (CPS) [29]. The project defines IoT as integrated network of physical process, computation and networking, where computers are embedded. Physical processes are controlled and monitored by these computers and network. Feedback system is included and computation is affected by physical process and vice versa.

IoT-A project mentions: objects, devices, services and technologies are interconnected in IoT. According to IoT special interest group, things are connected e.g. objects, vehicles, environments and clothing, which are internet enabled. Things have the ability to communicate, sense, produce information and netwok. Things have more information associated with them and are an integral part of the Internet [30].

As shown in figure 2, CERP-IoT project defines IoT as a worldwide dynamic network which is a part of Future Internet with self-configuring abilities. Standard and interoperable communication protocols are the basis for IoT [31] [32]. Smart things (Physical and virtual) have identities, physical attributes, and virtual personalities. These things are able to communicate and interact with each other and with the surroundings autonomously sensing the environment, exchanging data and information, influencing the process and triggering actions and creating services with or without direct human intervention. Understanding of IoT by IERC is similar to that of understanding by CERP-IoT [33].

According to ETP EPOSS project, IoT means a global network where objects are interconnected [34]. These things are uniquely identifiable. Standard communication protocols

for example the Internet suite (TCP/IP) are used. Objects have active role in the Future Internet. Internet Connected Objects for Reconfigurable Ecosystems (iCore) states that in future billions of physical objects will be connected to the Internet [35].

After analyzing different White Papers, a summary chart can be drawn as figure 3. As listed in the figure 3, Mattern & Floerkemeier defines IoT as a network where real world physical items or everyday objects are connected to the Internet [36]. These objects are parts of virtual world and can be remotely accessed and controlled. They work as access point of the Internet services. So, the computing is ubiquitous. According to Society for Brain Integrity [37], IoT means a self-configuring, dynamic and complex network which connects PCs, human to thing, human to human and between things to the Internet. RFID tags, sensor technologies are used to collect and process data and to detect changes. Using these technologies, network becomes more powerful and processing power becomes more useable.

According to figure 3, Hammersmith group [38], discusses the relationship between digital information and physical world object. Location becomes an important part of object in IoT. Information and GPS coordinates location get connected. On the other hand, objects can be embedded with sensors and transmitters which help to address the objects using Internet protocols. Also objects are able to communicate with each other and user, and sense the environments and respond. Michael Chui, et al. [39] mentions, IoT changes the physical world to information system. Sensors and actuators are embedded in physical objects, which are connected to wired, wireless networks and the Internet. IoT generates huge amount of data, which can be used for analysis. Objects have the capability to sense and communicate with environment. These physical information systems work mostly without human intervention. Rellermeyer et al. [40] describes IoT as a concept where everyday objects are capable to identify communicate and compute themselves. Accenture & Bankinter Foundation of Innovation [41] defines, everyday objects can be connected to the Internet which are integrated with sensors and devices. So, the things or objects are available anytime and anywhere using the Internet over wired or wireless networks. Also the objects can work as data source. In figure 3, Taiichi Inoue et al. [42] provides IoT as a system, which recognizes information about things. Things use sensors and cameras and connects to the Internet. The information consists unique attributes, time and location of the things.

As listed in figure 3, Dieter Uckelmann et al. [43] mention, IoT links things to virtual representations in the Internet. Things are identifiable with additional information on their identity, location, status and any other relevant information. This information can be accessed by non-predefined participants at the right time, price and place with right quantity and condition. IoT uses a combination of many aspects and technologies of Internet Protocol (IP), embedded devices, communication technology, Intranet/Extranet of Things and Internet of People. Giusto et al. [44] defines IoT as a new paradigm which includes an extensive set of technologies, applications and visions. IoT highlights smart objects and their virtual identity and their ability to interact with each other and with humans and environment intelligently. IoT also focuses on service-oriented architecture of the future Internet.

Weber et al. [45] mentions IoT as a world where physical objects are integral part of information network. These objects are accessible over the Internet with available services. These smart objects can participate actively in business processes. Considering security and privacy issues, anyone can query about object's state and any associated information. Shelby and Bormann [46] describes all the devices and networks are embedded and IP enabled which are connected to the Internet. These devices can be monitored and controlled with the Internet services. Vermesan and Friess [47] explains IoT as a global dynamic network. This network is based on standard and interoperable communication protocols with self-configuring abilities. Physical and virtual things are flawlessly connected into the information network and use smart interfaces. They have identities, attributes (physical) and personalities (virtual).

As shown in figure 3, SAP's understanding is similar to the understanding of Weber et al. [45]. SAP mentions IoT as a world where physical objects are integral part of information network [48]. These objects are accessible over the Internet with available services. These smart objects can participate actively in business processes. Considering security and privacy issues, anyone can query about object's state and any associated information. CISCO considers IoT under the Internet of everything [49]. Things, people, data and processes are connected and make networks. IoT turns information into actions. It creates better experiences, new capabilities and economic opportunity for individuals, business and countries. HP describes IoT as a system, where everyday objects are connected to the Internet and can be uniquely identified [50]. These things can be accessible from anywhere in the world and allows people to interact and control these things. Also, objects are able to interact each other without human participation.

III. Major criteria that define IoT

In summary, from the above discussion it is apparent that the understanding of IoT given by different studies differ in terms of many aspects. For example, IEEE discusses IoT as interconnected items where sensors are embedded in the items while IETF discusses IoT as a system where all kind objects (electronic, electrical, non-electrical) are connected through RFID tags, actuators and mobile phones. In this section, we analyzed the studies given by different standardization bodies, literatures, books, research projects, countries and industries etc. After careful analysis of each and every aspect of IoT, we identified that IoT understanding typically varies in terms of the following four criteria:

1) Events
2) Connectivity
3) Data sensing protocol
4) Availability of Network

TABLE I
MAJOR CRITERION OF IOT

Different Bodies or Studies	Events - Criteria	Connectivity - Criteria	Data sensing protocol - Criteria	Availability of Network - Criteria
IEEE	Items or things	Connects to the Internet	Embedded with sensors	Through the Internet
ITU	Anything and anyone	Ubiquitous Network		Anytime at anywhere
ITU-T	Physical and virtual things		Use current and evolving communication and information technologies	
ISO/ IEC	People, objects, systems, and information resources			
IETF	All kinds of objects, e.g. electronic, electrical, non-electrical		RFID tags, actuators, sensors, mobile phones	
ETSI	Machine-to-Machine (M2M), e.g. entities	Machine to Machine (M2M)	Do not need any direct human intervention	
NIST	Smart devices and systems	Cyber-physical systems (CPS)		Anywhere and anytime
W3C	Physical objects	Web of Things	Web Technologies like HTTP and JavaScript APIs, Sensors, actuators bar code or NFC	
OASIS	Real world things	Physical world connects to the Internet	Ubiquitous sensors are available in real world things	
ZigBee Alliance	Everyday objects and people		People have more control over the objects around them	Through the Internet
UK FISG	People and machines	Advanced wireless and fixed infrastructure	converged services, shared data and an advanced wireless and fixed infrastructure	Through the Internet
Australia CSIRO	Everyday and industrial objects		Networks of sensors, actuators and smart objects with physical things	
Finland IoT-SRA	Heterogeneous things	Seamlessly and securely combined with the Internet	Standard and interoperable protocols and formats	Available anywhere, anytime
India	Embedded objects and devices		Standard and interoperable communication protocols	
Malaysia DLM	Machines, gadgets, everyday products, devices and inanimate objects	The web where objects will be connected to the Internet	Embedded RFID chips linked with IP addresses, EPC, GPS system	A universal network
CASAGRAS	Physical and virtual objects	Includes existing and evolving Internet	Object-identification, sensor and connection capability, autonomous data capture, even transfer, network connectivity, and interoperability	
Berkeley University	Physical process, computation and networking	Cyber-physical systems(CPS)	Embedded computers	Objects can be tracked and accessed, in real time
IoT-A	objects, devices, services and technologies			Global network using Internet
IoT Special Interest Group	Objects, vehicles, environments and clothing	Integral part of the Internet	Sense, network communicate and produce information	Through Networking
CERP-IoT	Physical and virtual things or smart things	Integrated part of Future Internet	Use standard and interoperable communication protocols, sensing, react autonomously, trigger action	
IERC	Physical and virtual things	Seamlessly integrated into the information network	Use standard and interoperable communication protocols, self-configuring capabilities	Through Internet
ETP EPoSS	Things or objects	Future Internet Based on standard communication protocols, the Internet suite (TCP/IP)	Through Future Internet, global network	
iCore	Billions of tangible objects will be connected to people	Connects to Internet		Dynamic universal network infrastructure
[28]	Physical items, everyday objects	Connects to the Internet	Remotely controlled, access point of Internet	Through Future Internet
[29]	Physical things, PCs, human	Connects to the Internet	RFID tags, sensor technologies, collect and process data, detect changes	Through the Internet
[30]	Objects and locations	Connects to the Internet	GPS, Sensors, transmitter, Internet protocols	ubiquitous network
[31]	Every physical object will connect environment	Connects to the Internet	Sensors, actuators, Internet Protocol (IP), data	Through Internet
[32]	Everyday objects will be connected		Identify, communicate, compute	Through Internet
[33]	Everyday objects	Connects to the Internet	Sensors, devices, feasible technology, data source	Through Internet
[34]	Things will be connected	Connects to the Internet	Sensors and cameras	
[35]	Uniquely identifiable things	Connects to the Internet	Internet Protocol (IP), embedded devices, communication technology, Intranet/Extranet	Through Internet, anytime, anywhere
[36]	Smart objects, humans, environment	The future Internet	Extensive set of technologies, applications	Through Internet

TABLE II
MAJOR CRITERION OF IOT (CONTINUE)

Different Bodies or Studies	Events - Criteria	Connectivity - Criteria	Data sensing protocol - Criteria	Availability of Network - Criteria
[37]	Physical or Smart objects	Interact with the Internet	Security, privacy, business process, state and information	Right time place, price
[38]	Connects all the embedded devices and networks	Connects to the Internet	IP-enabled, embedded devices	Through Internet
[39]	Physical and virtual things		Standard and interoperable communication protocols, self-configuring abilities, smart interfaces	Through Internet
SAP	Physical or Smart objects	Interact with the Internet	Security, privacy, business process, state, information	Through Internet
CISCO	Things, people, data and process	Internet of Everything	Information into actions	
HP	Everyday objects, devices, human	Connects to the Internet	Unique identity, Internetization, human involvement	Through Internet

Based on these criteria we further analyze the similarities and dissimilarities among them. Table I and II show this analysis in detail.

A. Events

According to the table I and II, based on the events criteria, it is observed that, few bodies / studies considered that IoT should connect things/items. For example, IEEE and ITU. In contrast to these, UK FISG defines that the connection should be established among people and machines. Thereby, the studies differ in terms of events like people, items, devices, objects etc.

B. Connectivity

The table I and II illustrates, the criterion "connectivity" which shows how different studies discuss the different ways of connecting IoT things/items/objects/smart devices each other and how these events connect to internet. For example, IEEE defines IoT where items should be connected to the Internet. In contrast to this, ITU considers IoT as a ubiquitous network (i.e., the network is available everywhere). We notice that, there are differences between the concept of Internet and ubiquitous network thereby, the studies of IoT by IEEE and ITU differs in terms of connectivity. As per IEEE, items are considered as part of the existing Internet. On the contrary ubiquitous network which is explicitly mentioned by ITU, is created from wide-ranging use of networked devices and networks.

C. Data sensing protocol

The column three in table I and II lists different "data sensing protocols" that are used in IoT by different standardization bodies or studies considered in this study. For example, IEEE mentioned that items in IoT should be embedded with sensors. It is also noticed that, according to IETF objects have RFID tags, and also are embedded with actuators and sensors. Similarly, OASIS considered real world things that should use ubiquitous sensors. In a nutshell, IEEE, IETF and OASIS considered IoT as a system where sensor technology should be used.

D. Availability of Network

The column four in table I and II shows the how Iot devices will be connected and get access to the network. This means, is the network always available to the events; or does it depend on the contemporary location of the event; or does it depend on the time when the respective event wants to connect to the network to communicate to other events? As per the table I and II, ITU defined IoT, where events are referred to as anything that are accessible from anywhere and anytime. It should be noted here that, the terms anywhere, anytime are very sloppy words and hence this concept may not be clear enough to identify an IoT. However, ITU clarifies that, anywhere can represent events on the move, outdoors, indoors (i.e., either away from the PC or in front of the PC). Anytime can refer to either night or daytime. Furthermore, anything can represent PCs, Human-to-Human(H2H), Thing-to-Thing(T2T), Human-to-Things (H2T) etc.

IV. CONCLUSION

In this paper, We did extensive analysis of different studies that discussed the underlying concept of IoT. We found that the understanding of IoT by different studies have many common aspects as well as differences. According to different studies, it can be predicted that IoT will be considered as a ubiquitous network of things that will be available anytime, anywhere, anything and anyone. These things will be a node of a network and it can be physical and virtual. Diverse exploration are giving various perspective of IoT. We think having a standard and sound understanding of the concept of IoT is critical to encourage a superior comprehension of IoT. This will assist various associations with running after the improvement of IoT. This will likewise open the entryway for a future research.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of King Fahd University of Petroleum & Minerals for providing all the support to accomplish the research presented in this paper.

REFERENCES

[1] Alkhatib, H.; Faraboschi, P.; Frachtenberg, E.; Kasahara, H.; Lange, D.; Laplante, P.; Merchant, A.; Milojicic, D.; Schwan, K. IEEE CS 2022 Report; IEEE Computer Society: Washington, DC, USA, 2014.

[2] Nordrum, Amy (18 August 2016). "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated". https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated (access on 18 December, 2020)

[3] Hsu, Chin-Lung; Lin, Judy Chuan-Chuan (2016). "An empirical examination of consumer adoption of Internet of Things services: Network externalities and concern for information privacy perspectives". Computers in Human Behavior. 62: 516–527. doi:10.1016/j.chb.2016.04.023.

[4] International Telecommunication Union—Telecommunication Standardization Sector (ITU-T). Recommendation ITU-T Y.2060—Overview of the Internet of Things; ITU-T: Geneva, Switzerland, 2012.

[5] Cisco. The Internet of Things Reference Model; Cisco and/or its Affiliates: San Jose, CA, USA, 2014.

[6] OECD Committee for Digital Economy Policy. OECD Technology Foresight Forum 2014—The Internet of Things. Available online: http://www.oecd.org/internet/ieconomy/technology-foresight-forum-2014.htm (accessed on 01 December 2020).

[7] IBM Connects Internet of Things to the Enterprise. Available online: http://www-03.ibm.com/press/us/en/pressrelease/46453.wss (accessed on 17 December 2020).

[8] IEEE IoT Technical Community—About Internet of Things. Available online: http://iot.ieee.org/about.html accessed on 18 December 2020)

[9] Umar Zakir Abdul, Hamid; et al. (2019). "Internet of Vehicle (IoV) Applications in Expediting the Implementation of Smart Highway of Autonomous Vehicle: A Survey". Performability in Internet of Things. Retrieved 28 August 2018.

[10] Roberto Minerva et al, "Towards a definition of the Internet of Things (IoT)", IEEE Internet of Things, Revision 1 – Published 27 MAY 2015

[11] http://www.postscapes.com/internet-of-things-definition/

[12] In its "Special report: Internet of Things" issued in March 2014 (IEEE, "Internet of Things," March 2014), http://theinstitute.ieee.org/static/special-report-the-internet-of-things]

[13] [In its 2005 IoT report, ITU, SERIES Y, 2005]

[14] ITU, SERIES Y, 2005, http://www.itu.int/en/ITU-T/about/groups/Pages/sg13.aspx

[15] ISO/IEC JTC 1 Information technology Internet of Things (IoT) Report 2014, http://www.iso.org/iso/internet_of_things_report-jtc1.pdf

[16] [IETF, "Internet of Things – Concept and Problem Statement," by Gyu Myoung Lee, Jungsoo Part, Ning Kong, Noel Crespi 2010]

[17] [ETSI, "Machine-to-Machine communications (M2M); M2m service requirements," ETSI TS 1-2 689 V1.1.1, 2010 http://www.etsi.org]

[18] NIST, "Global City Teams," 2014, https://pages.nist.gov/cpspwg/

[19] OASIS, "Open Protocols," 2014, https://www.oasis-open.org/org

[20] W3C, "Web of Things," https://www.w3.org/WoT/ (accessed on 5 December 2020)

[21] ARTICLE "ZigBee: The Language of the Internet of Things" January 9, 2015, By Ryan Maley, Director of Strategic Marketing, ZigBee, Alliance, http://www.sensorsmag.com/internet-things/zigbee-language-internet-things-16778

[22] UK FISG, "Future Internet Report," 2011

[23] http://www.csiro.au/en/Research/DPF/Areas/Autonomous-systems/IoT

[24] Tarkoma, S., and Katasonov, A.: Internet of Things Strategic Research Agenda. Finnish Strategic Centre for Science, Technology and Innovation (2011), http://www.internetofthings.fi/

[25] Dr. Ovidiu Vermesan et. al. IoT-SRA, "Internet of Things Strategic Research," 2011

[26] http://meity.gov.in/content/internet-things

[27] Ramalingam, "Engage and Interact," 2013 https://www.skmm.gov.my/skmmgovmy/media/General/pdf/Malini-Ramalingam-Digital-Lifestyle-Malaysia.pdf

[28] RFID and the Inclusive Model for the Internet of Things by CASAGRAS, Final Report, http://www.rfidglobal.eu/userfiles/documents/CASAGRAS26022009.pdf

[29] Lee, E.A., 2008, May. Cyber physical systems: Design challenges. In 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC) (pp. 363-369). IEEE.

[30] Bassi, A., Bauer, M., Fiedler, M., van Kranenburg, R., Lange, S., Meissner, S. and Kramp, T., 2013. Enabling things to talk (p. 379). Springer Nature.

[31] "Internet of Things (IoT) and Machine to Machine Communications (M2M) Challenges and opportunities: Final paper" May 2013, by Technology Strategy Board – IoT Special Interest Group

[32] CERP-IoT. "Visions and Challenges for Realising the Internet of Things," European Commission (2010), ISBN 9789279150883.

[33] European Research Cluster on Internet of Things (IERC), "Internet of Things," http://www.internet-of-things-research.eu/about_iot.htm

[34] Internet of Things in 2020 – A Roadmap for the Future, by INFSO D.4 Networked Enterprise & RFID INFSO G.2 with EPoSS, 05 September 2008

[35] Frank Berkers, Wietske Koers, Katia Colucci, Oskar Kadlec, Dan Puiu, Marc Roelands, Stephane Menoret, iCore Deliverable D1.3, "Vision of the future business ecosystem, new roles and models of acceptance, 2013.

[36] Friedemann Mattern and Christian Floerkemeier, "From the Internet of Computers to the Internet of Things", In: Kai Sachs, Ilia Petrov, Pablo Guerrero (Eds.): From Active Data Management to Event-Based Systems and More. LNCS, Vol. 6462, Springer, pp. 242-259, 2010

[37] Society for Brain Integrity, Sweden, "Future Internet" 2010, http://www.svegritet.se/emergin-technologies/future-internet/

[38] The Hammersmith group research report February 2010 "The Internet of Things: Networked objects and smart devices" https://www.theinternetofthings.eu/sites/default/files/Rob/

[39] Michael Chui, Markus Löffler, and Roger Roberts, "The Internet of Things", McKinsey & Company, March 2010, http://www.mckinsey.com/industries/high-tech/our-insights/the-internet-of-things

[40] Jan S. Rellermeyer et al., "The Software Fabric for the Internet of Things" "The Internet of Things", First International Conference IOT 2008, P.87-104, ISBN: 987-3-540-78730 https://www.duller.net/michael/fileadmin/pubs/Rellermeyer2008.pdf

[41] G. Feller, "The Internet of Things: In a Connected World of Smart Objects," Accenture & Bankinter Foundation of Innovation, 2011.

[42] Taiichi Inoue et al., "China's Initiative for the Internet of Things and Opportunities for Japanese Business," P.4, Normura Research Institute (NRI) Papers, No. 165, 2011

[43] Dieter Uckelmann et al. editors, "Architecting the Internet of Things", Springer, P. 8, ISBN 978-3-642-19156-5, 2011

[44] Giusto et al., "The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications" Springer-Verlag, ISBN: 1-4419-1673-3, 978-1-4419-1673-0, 2010

[45] Weber et al., "Internet of Things: Legal Perspectives", Springer-Verlag, ISBN: 3-642-11709-0, 978-3-642-11709-1, 2010

[46] Zach Shelby, Carsten Bormann, "6LoWPAN: The Wireless Embedded Internet", Publisher: Wiley, ISBN 978-0-470-74799-5

[47] Ovidiu Vermesan, Peter Friess, "Internet of Things - Global Technological and Societal Trends From Smart Environments and Spaces to Green ICT", River Publishers, ISBN: 978-87-92329-67-7, 2011

[48] Internet of Things: An Integral Part of the Future Internet, By Stephen Haller, SAP Research, 2009, http://services.future-internet.eu/images/1/16/A4_Things_Haller.pdf

[49] Bradley, "Internet of Everything," 2013, https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoE_Economy.pdf (accessed on 10 December 2020)

[50] Miessler, "HP Security," 2014

Controlling Driver Behaviour in ADAS with Emotions Recognition System

1st Oleg Evstafev
Faculty of Control Systems and Robotics
ITMO University
Saint Petersburg, Russia
oaevstafev@itmo.ru

2nd Vladimir Bespalov
Faculty of Control Systems and Robotics
ITMO University
Saint Petersburg, Russia
vvbespalov@itmo.ru

3rd Sergey Shavetov
Faculty of Control Systems and Robotics
ITMO University
Saint Petersburg, Russia
s.shavetov@itmo.ru

4th Mikhail Kakanov
Faculty of Control Systems and Robotics
ITMO University
Saint Petersburg, Russia
makakanov@itmo.ru

***Abstract*—This work is devoted to the creation of a non-intrusive vehicle accident warning system, which is a part of ADAS. The proposed algorithm uses technical vision, implemented on the basis of the Viola-Jones algorithm to assess the age and psychological state of the driver, by determining emotions. In case of non-compliance with the specified conditions, the system sends a warning signal to the user.**

***Index Terms*—Computer Vision, Viola–Jones, Haar-like feature, Face Detection, Emotion Recognition, Target Tracking, Matlab, Machine Learning, ADAS.**

I. Introduction

Thanks to the growth of the computing power of modern technology, more and more areas of human life are being automated. This trend did not pass by the automotive sector. One of the new and promising areas — *Intelligent Autonomous Vehicles* (IAV) — an integrated technical and technological complex of systems that combines the safety subsystems of individual vehicles and the organization of safe road traffic in general, as well as the provision of information services for road users and potential subjects of the transport process, see in (1).

A distinctive feature of modern IAV is the change in the status of a transport unit from an independent, autonomous and largely unpredictable subject of traffic towards an "active", predictable subject of transport and information space. In this regard, one of the critical tasks is the development of the telematic complex of road infrastructure.

The operational task of IAV is the implementation and support of the possibility of automated and automatic interaction of all transport entities in real-time on adaptive principles.

To successfully achieve the IAV goal, the operational task can be divided into two main areas — the IAV situational management, based on the interaction of the vehicle with external data, the other — management in relation to a single vehicle, which is carried out by reading roadside information, monitoring the profile of the road, monitoring traffic in the stream and monitoring the condition of the driver.

*This work is supported by the Russian Science Foundation grant (project No19-19-00403).

As a control for a single tool, one of the essential areas of development is *ADAS (advanced driver-assistance systems)* (2). This is a set of certain algorithms that, to varying degrees, help the driver to make quick decisions for safe and comfortable driving, which reduces the number of accidents on the road. The basis of such systems is the totality of the work of many systems and sensors: cameras, accelerometers, gyroscopes, GPS-navigation, microphones, temperature sensors, etc. The processed data that was received from the sensors can be used to implement the ADAS functions: an intelligent parking system, cruise control, adaptive lighting system, the "smart mirror" function, a system for recognizing traffic signs and pedestrians, and an exit warning sign lane or on a one-way road, etc.

Some premium cars often have built-in ADAS features in on-board computers, for example, a system from NVIDIA can distinguish and identify different types of objects when driving. Or the well-known safe braking system from VOLVO when quickly approaching an object (pedestrian or car), in fact — also an element of the ADAS system.

There are many algorithms for detecting, localizing, and recognizing objects from a video stream in real-time. Each algorithm has its advantages and disadvantages, however, the most popular today is the Viola-Jones algorithm see in (3), (4) and (5), since it guarantees high detection accuracy and detection speed, and is also widely used in various automation systems for the detection and identification processes.

To prevent a road accident on time, it is necessary not only to detect the face but to assess the position of the driver's eyes and also evaluate the emotion, based on which to determine whether the user is in a stressful state and whether he maintains attention on the road. These works (6), (7) and (8) are devoted to improving the Viola Jones method for recognizing human emotions. The articles examined and solved the problem of assessing the attention of students in lectures in the audience.

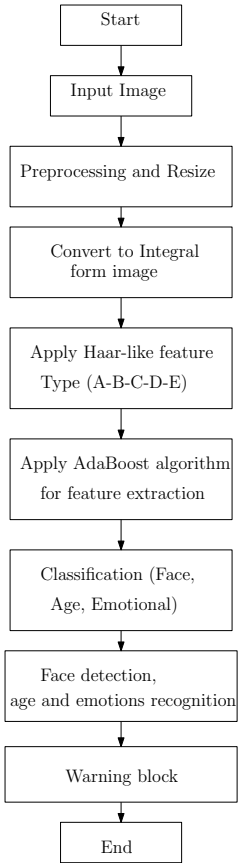


Fig. 1. A block diagram of the algorithm

In (9) and (10) non-intrusive vehicle accident warning systems have been developed, which is an example of ADAS to prevent accidents on the road due to driver drowsiness or carelessness. Approaches offer to evaluate the position of the head and eyes, and in case of loss of user attention, the system gives a warning signal. Other examples of ADAS implementation can be found in the following papers — (11) and (12).

In this work, a system was developed that combines early approaches and allows, according to certain parameters, such as age, head position, and emotions, to assess the psychological and physical condition of the driver and, in case of negative assessments, to give a warning signal to the user.

II. Algorithm implementation steps

The block diagram is given in fig.1 represents the developed algorithm and the steps required for implementing the face recognition and emotion using MATLAB.

A. Viola–Jones object detection

The first step of the algorithm is to detect a face from the video stream in real-time. In this paper, this is done by using the Viola-Jones algorithm.

The Viola-Jones algorithm allows you to detect objects in the image in real-time. For a better understanding of the method, it is divided into separate points:

- Haar-like features, search for the desired object;
- Integrated image representation for quick calculation of Haar-like features;
- Adaptive Boosting (AdaBoost), which allows you to weed out signs that carry the least amount of information, as well as get strong classifiers from many weak ones.

Thus, the implementation of the detection algorithm is reduced to the implementation of its individual methods, which, in turn, can be implemented in any order, and then combine them into one algorithm.

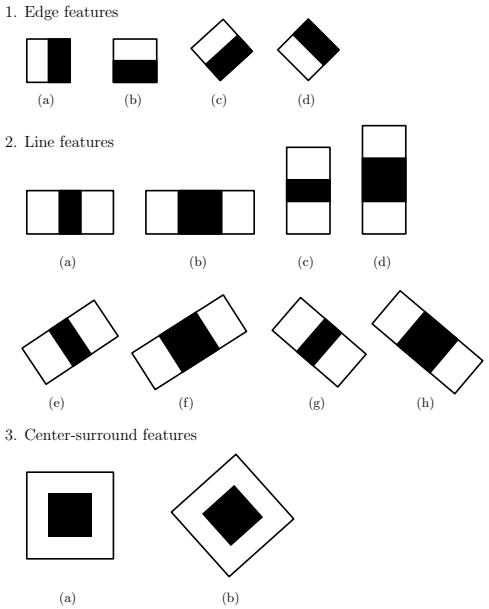


Fig. 2. Haar-like features

B. Haar-like features

Haar-like features are a number characterized by the difference of pixel sums between a black and white area fig. 2. In a face detection system, the set of all features is given by the shape, size, and position of the image.

The Viola-Jones algorithm uses Haar-like features (13) to calculate the difference between the sums of pixels in the black and white areas of an image. The resulting scalar value characterizes a particular feature. The image must be normalized fig. 3, i.e. have zero expected value and 1 variance, to compensate the effects of different lighting conditions.

$$X = X - \overline{X}, \tag{1}$$

where X is the image matrix, and $\overline{X}$ is mathematical expectation calculated by the

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{2}$$

with n — image matrix dimension

Fig. 3. Not normalized image (left), normalized image (right), (14)

To obtain a unit variance we use

$$X = \frac{X}{\sigma}, \tag{3}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X - \overline{X})^2}, \tag{4}$$

where σ — standard deviation.

C. Integral representation of the image

In order to perform any actions with the data, the integral representation of images (15) is used in the Viola-Jones method. The integrated representation allows you to quickly calculate the total brightness of an arbitrary rectangle in a given image, and no matter what the rectangle is, the calculation time is constant. The integral representation of an image is a matrix that is the same size as the original image. Each element of it stores the sum of the intensities of all pixels to the left and above this element. Matrix elements are calculated using the following formula:

$$L(x, y) = \sum_{i=0,j=0}^{i \le x, j \le y} I(i, j), \tag{5}$$

where $I(i, j)$ is the brightness of the pixel of the original image.
Each element of the matrix $L(x, y)$ represents the sum of the pixels in the rectangle from $(0, 0)$ to (x, y), i.e. the value of each pixel (x, y) is equal to the sum of the values of all pixels to the left and above the given pixel (x, y). The calculation of the matrix takes linear time, proportional to the number of pixels in the image, so the integrated image is calculated in one pass. The calculation of the matrix is possible by the formula:

$$\begin{aligned} L(x, y) =& I(x, y) - L(x - 1, y - 1) + \\ &+ L(x, y, -1) + L(x - 1, y). \end{aligned} \tag{6}$$

Using such an integral matrix, you can calculate the sum of the pixels of an arbitrary rectangle very quickly.

Further, the approach based on the *scanning window* (16) is used: the image is scanned by the search window, and then the classifier is applied to each position. The training system and the selection of the most significant features is fully automated and does not require human intervention, so this approach works quickly.

The task of finding and finding faces in the image using this principle contributes to the recognition of characteristic features, for example, verification of a person by a recognized face or recognition of facial expressions.

D. AdaBoost

AdaBoost, short for Adaptive Boosting (17) is a machine learning approach based on creating a highly accurate prediction rule by combining relatively weak and inaccurate rules.

The main task of adaptive acceleration is to build a strong classifier based on weak ones. The weak classifier, in this paper, is the Haar-like features. A strong classifier is the set of weak classifiers that best define the face in the image.

As input are used (x_1, y_1), ..., (x_n, y_n), where x_i — is for Haar-like features, and y_i is appropriate classification.
The initialization process takes place according to the following expressions

$$\begin{cases} D_1(y_i = 1) = \frac{1}{k} \\ D_1(y_i = -1) = \frac{1}{r} \\ k + r = n. \end{cases} \tag{7}$$

Then for $t = 1, ..., T$, where T is for number of stages of classifier training, we start the following procedure:

- weak classifier calculation h_t
- calculation ε_t using

$$\varepsilon_t = \sum_{i=1}^{n} D_t(i)[y_i \neq h_t]; \tag{8}$$

- if $\varepsilon_t = 0$ then stop the procedure and return the value of h_t
- calculation α using

$$\alpha = \frac{1}{2} ln(\frac{1 - \varepsilon_t}{\varepsilon_t}); \tag{9}$$

- weight reduction

$$D_{t+1} = \frac{D_t}{2}(\frac{1}{\varepsilon_t}[y_i \neq h_t] + \frac{1}{1 - \varepsilon_t}[y_i = h_t]). \tag{10}$$

And so we can get the final classifier:

$$H(x) = sign(\sum_{T}^{t=1} \alpha_t h_t(x)). \tag{11}$$

Thus, having a strong classifier, it is very likely to determine the face in the image without calculating all the signs, which saves processor time and increases productivity. This process can also be called the *"principle of constructing a decision tree."* (18) In each iteration, a weak classifier is formed with its threshold value and parity sign. To train such a cascade, the following actions will be required:

- Error level values are set for each stage (they must first be quantified when applied to the image from the training set) — they are called *detection* and *false positive rates* — the *detection level* must be high and the false-positive rates low;
- Features are added until the parameters of the calculated stage have reached the set level, here such auxiliary steps as are possible:

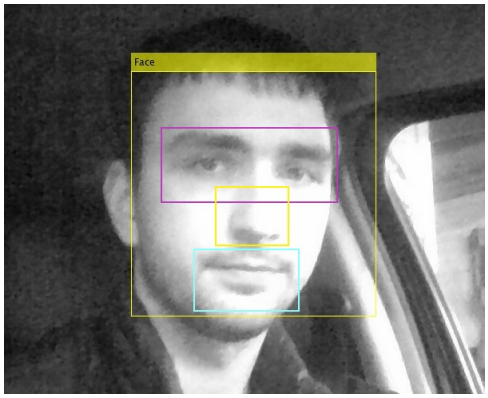


Fig. 4. Facial recognition algorithm

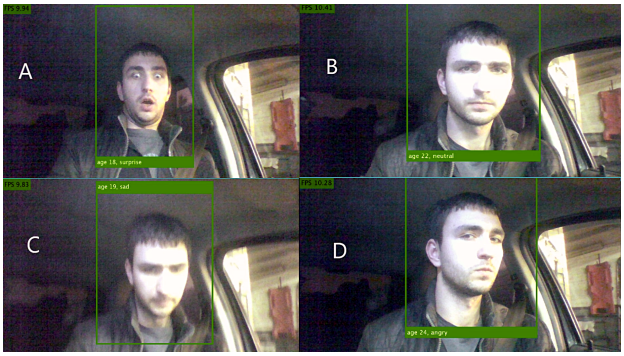


Fig. 5. The result of the algorithm. Where: A — surprise, B — neutral, C — sad, D — angry

  - Testing an additional small training set;
  - The AdaBoost threshold is deliberately lowered in order to find more objects, but in connection with this a larger number of inaccurate definitions of objects is possible;
- If *false positive* rates remains high, the next step or layer is added;
- False detections at the current stage are used as negative ones already at the next layer or stage.

The result of this step is ready-to-use, trained classifiers. In the module, we developed a lot of classifiers will be used (for each facial features that are distinctive for a given emotion). Classifier training is very slow, but facial search results are very fast. For simplicity of training, ready-made databases from sources were used AffectNet database (19). The algorithm works and recognizes facial features at a slight angle, up to about 30 degrees. With an inclination angle of more than 30 degrees, the percentage of detections drops sharply (20).

Thanks to this feature — the algorithm detects the position of the head and warns the driver in case of long non-detection of the head. Once the classifiers have been correctly trained, detection, capture and tracking of a person's face in the video stream will be performed when the algorithm is fully passed. This is followed by detection of the age of the driver. It was done for additional verification of the driver's age of majority. Then the algorithm recognizes emotions: anger, disgust, fear, happiness, sadness, surprise, neutral emotion. Additionally, it is determined whether the eyes are closed or not. As a result, the algorithm records the time during which the driver experiences negative emotions, head deviation and eye conditions, based on which the system warns the driver with a warning sound signal.

III. Work result

The Viola-Jones algorithm is implemented in the *MATLAB* environment, using similar *OpenCVs* toolboxes. The implementation was carried out using the technical characteristics of the laptop, i.e. computing power, camera and speaker.

As shown in the fig. 4, the face's features are detected by using their own classifiers and are distinguished by the *borders of bbox* (16). Therefore, from this, we would capture nose, eyes and mouth profiles for the face tracking task. Thus, whenever a face is in motion, the bbox will highlight facial features using their borders, along with real-time motion of the streaming video.
The result of the algorithm for assessing driver emotions is presented in the table I.

TABLE I
PERFORMANCE ANALYSIS FOR DIFFERENT EMOTIONS

Emotion	Accuracy(%)	Time Process(s)
Angry	93	0.675
Disgust	90	0.704
Fear	88	0.597
Happy	95	0.485
Neutral	98	0.422
Sad	89	0.603
Surprise	94	0.535

In the fig. 5, you can observe the result of the entire algorithm for determining emotions and age in a car, where A — surprise, B — neutral, C — sad, D — angry. It was revealed that the proposed method will be performed at a low number of frames per second (FPS), because of the camera's low technical characteristics. Video was captured in BGRA (Blue Green Red Alpha) format to increase speed and get a higher FPS, which was 11-20 FPS at 1280x480 resolution. As one optimization option, OpenCV along with CUDA support can be used to increase this figure.

It was also revealed that in conditions of insufficient lighting, large merging with the background of the desired objects, the presence of noise in the image, the module works more slowly. The image preprocessing mechanism was proposed to improve the recognition process in conditions of poor visibility of the object (mixing with the background).

Another problem of the proposed algorithm relates to the eyes' occlusion. Sunglasses and other items of clothing that visually close your eyes impair the definition of emotions because the classifier is trained to recognize faces and emotions

through eye detection. Therefore, when one of the classifiers does not work correctly, a reliable assessment of emotions is not guaranteed. So, the accuracy of the system decreases.

IV. Conclusion

As a result of the work, a non-intrusive driver accident alert system has been developed and proposed that can be used in ADAS systems. Crash and accident prevention, with the help of the developed algorithm, is carried out in several stages. First, based on the Viola-Jones algorithm, which includes Haar cascades to detect facial parts such as eyes, nose, chin in the video stream and AdaBoost classifier amplifier, the driver's age is estimated, in order to prevent underage persons from driving vehicles. Then, head position and emotion are simultaneously monitored to ensure the driver remains focused on the road. In the event of inconsistent head position or prolonged exposure to negative emotions, the warning system issues a warning signal to the driver to, for example, regain control of the road in case of overexertion or to stop driving too aggressively.

In the future, it is planned to improve the work of already implemented functions to increase the applicability of the developed system, introduce algorithms for estimating traffic density and distance estimation to moving objects.

References

[1] N. Tsolakis, D. Bechtsis, and J. S. Srai, "Intelligent autonomous vehicles in digital supply chains: From conceptualisation, to simulation modelling, to real-world operations," Bus. Process Manag. J., vol. 25, no. 3, pp. 414–437, 2019.

[2] A. Simic, O. Kocic, M. Z. Bjelica, and M. Milosevic, "Driver monitoring algorithm for advanced driver assistance systems," 24th Telecommun. Forum, TELFOR 2016, 2017.

[3] T. Paul, U. A. Shammi, S. Kobashi, and M. F. Detection, "A Study on Face Detection Using Viola-Jones Algorithm in Various Backgrounds, Angles and Distances," Int. J. Biomed. Soft Comput. Hum. Sci. Off. J. Biomed. Fuzzy Syst. Assoc., vol. 23, no. 1, pp. 27–36, 2018.

[4] I. Gusti Ngurah Made Kris Raya, A. N. Jati, and R. E. Saputra, "Analysis realization of Viola-Jones method for face detection on CCTV camera based on embedded system," Proc. 2017 Int. Conf. Robot. Biomimetics, Intell. Comput. Syst. Robionetics 2017, vol. 2017-December, pp. 1–5, 2017.

[5] A. Mohsen Abdul Hossen, R. Abd Alsaheb Ogla, and M. Mahmood Ali, "Face Detection by Using OpenCV's Viola-Jones Algorithm based on coding eyes" Iraqi J. Sci., vol. 58, no. 2A, pp. 735–745, 2017.

[6] D. Reney and N. Tripathi, "An efficient method to face and emotion detection," in Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015, 2015, pp. 493–49.

[7] P. Nair and V. Subha, "Facial Expression Analysis for Distress Detection," Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018, no. Iceca, pp. 1652–1655, 2018.

[8] S. Sahoo, "Emotion Recognition from Text," Int. J. Res. Appl. Sci. Eng. Technol., vol. 6, no. 3, pp. 237–243, 2018.

[9] M. Kahlon and S. Ganesan, "Driver Drowsiness Detection System Based on Binary Eyes Image Data," IEEE Int. Conf. Electro Inf. Technol., vol. 2018-May, pp. 209–215, 2018.

[10] O. Rizwan, H. Rizwan, and M. Ejaz, "Development of an efficient system for vehicle accident warning," ICET 2013 - 2013 IEEE 9th Int. Conf. Emerg. Technol., 2013.

[11] S. M. Sarala, D. H. Sharath Yadav, and A. Ansari, "Emotionally adaptive driver voice alert system for advanced driver assistance system (ADAS) applications," Proc. Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2018, no. Icssit, pp. 509–512, 2018.

[12] S. M. Iranmanesh, H. Nourkhiz Mahjoub, H. Kazemi, and Y. P. Fallah, "An adaptive forward collision warning framework design based on driver distraction," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 12, pp. 3925–3934, 2018.

[13] P. Viola, "The Viola / Jones Face Detector Classifier is Learned from Labeled Data," Procedia Comput. Sci., pp. 22–23, 2001.

[14] K. G. Derpanis, "Integral image-based representations," Dep. Comput. Sci. Eng. York Univ. Pap., 2007.

[15] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, vol. 1.

[17] R. E. Schapire, "Theoretical, views of boosting and applications," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1999, vol. 1720, pp. 13–25.

[18] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 904, pp. 23–37, 1995.

[19] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," IEEE Trans. Affect. Comput., 2019.

[20] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," Image Process. Line, vol. 4, pp. 128–148, 2014.

Comparative Analysis of Deep Learning and Traditional Machine Learning Models for Turkish Text Classification

Hasibe Büşra Doğru
Department of Computer Engineering
Istanbul Sabahattın Zaim University
Istanbul, Turkey
hasibe.dogru@izu.edu.tr

Alaa Ali Hameed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
alaa.hameed@izu.edu.tr

Sahra Tilki
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
sahra.tilki@izu.edu.tr

Akhtar Jamil
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
akhtar.jamil@izu.edu.tr

***Abstract*— In this study, using the word embedding method Doc2vec, the Turkish Text Classification 3600 (TTC-3600) dataset consisting of Turkish news texts was classified based on deep learning. Most commonly used classifiers were selected: Convolutional Neural Network (CNN), Gauss Naive Bayes (GNB), Random Forest (RF), Naïve Bayes (NB) and Support Vector Machine (SVM). While investigating the effect of text preprocessing steps on the success rate in the study, the results are compared with the previous studies with the TTC-3600 dataset. In the proposed model, a better accuracy rate was achieved with a result of 94.17% compared to the studies in the literature.**

Keywords—Turkish Text Classification, Doc2Vec, Text Preprocessing, Machine Learning, Deep Learning

I. INTRODUCTION

Internet usage continues to grow day by day [1]. This increase also causes data to be produced. It is very difficult to manually classify the data due to its unstructured nature. Fields such as Natural Language Processing [2], Machine Learning [3] allow us to automatically classify data. Classification, which enables to analyze data containing text, is the process of separating the data into predefined classes. There are many studies in the literature on text classification, but most of these studies have been done with English texts. Therefore, although there are fewer datasets, tools and study resources to be used in text classification compared to the English language, studies in other languages have been increasing in recent years.

One of the studies on the classification of Turkish texts, a system called NECL was developed by Çatal et al. [4]. This system, developed by using N-grams, was used in classification of documents. Amasyalı and Diri [5] suggested that n gram-based approaches performed better with Support Vector Machine (SVM), J48 and Random Forrest classifications. Çataltepe et al. [6] investigated the effect of root length on classification. As a result of the research, it was concluded that Centroid classification using shortened roots is more successful. In a study conducted by Güran et al. [7], the best success rate among the Naive Bayes (NB), Decision Tree (J48) and K - Nearest Neighbor (K - NN) classification algorithms on Turkish datasets was obtained in the Decision Tree algorithm. Amasyalı and Beken [8] divide Turkish words into different categories semantically and suggest classification with a different approach. The best result was obtained by Linear Regression classification method. Torunoğlu et al. [9] made an important study on text representation and text classification in terms of preprocessing. In this study, data cleaning, root separation, word feature weighting stages were tested on the Turkish dataset. According to the results, they stated that while the word root is beneficial in Knowledge Acquisition problems, it does not contribute to text classification. Tüfekçi and Uzun [10] investigated the effect of term weighting methods on the determination of text texts and the best result was obtained with SVM classification. Uysal and Gunal [11] stated that preprocessing is important for text classification by using a dataset consisting of English and Turkish e-mails and news. They examined how the SVM classification method and preprocessing stages affect the accuracy rate, and as a result, it is seen that while some preprocessing methods decrease the accuracy rate in text classification, conversion to lowercase and removing stop-words increase the accuracy rate. Levent and Diri [12] conducted a study on recognizing the authors of Turkish texts with Artificial Neural Networks, and the study obtained close results in terms of success rates compared to the algorithms used previously. Kılınç et al. [13] created a Turkish dataset containing news texts named TTC-3600 and shared it for use in academic studies. At the same time, they applied the model they developed on the dataset they created. In the model they proposed, they used word bag, n-gram model and feature selection models for text representation. As the classification methods, 6 different algorithms were applied and feature selection models were used. They classified the text representations obtained by feature selection using Zemberek library to separate word roots and ARFS (Attribute ranking-based Feature Selection) for feature selection. Conclusion It is emphasized that the RF classifier gives the best result. Kılınç [14] evaluated the effect of collective learning models on Turkish text classification. Text classification process was carried out on TTC-3600 dataset with NB, SVM, K-Nearest Neighbor (KNN), J48 Decision tree and their Boosting, Bagging and Rotation Forest community learning models. As a result of the study, it has been shown that the basic classification methods of collective learning models increase the success rate. Başkaya and Aydın [15] reduced the size of a dataset with 4 categories and 20 news texts belonging to each category taken from different news sites and newspapers with the CfsSubset Algorithm and then classified the dataset with the NB, DVM, J48 and RO methods. Kaynar and Aydın [16] used autocoder and deep learning network as feature reduction method for emotional analysis and compared with other common feature reduction

techniques. Acı and Çırak [17] were classified on TTC-3600 dataset using CNN and Word2Vec word embedding method and success rates were compared with previous studies using the same dataset. In the study, both the original and the body version of the TTC-3600 dataset are trained with two different CNNs. Compared to previous studies, a higher success rate was achieved with the method they recommended. Yıldırım et al. [18], using two different datasets, TTC-4900 and TTC-3600 [13], which have 700 text documents under 7 different categories shared by the Bone DDI Group, in their study, using neural network-based text representations and a method of classifying traditional text representations. compared with. Knowledge Gain and chi-square approach are used in traditional text representation, PV-DM, PV-DBOW, PV-DM + PV-DBOW, and vector averages are used in artificial neural network-based architecture. Knowledge Gain and chi-square approach is more successful than other text representation. has been found. With the PV-DM method Logistic Regression classifier, 89.0 in the TTC-4900 dataset, 92.3 F1 in the TTC-3600 dataset, the Information Gain (IG) is 90.0 in the TTC - 4900 dataset with the multi-nominal NB (m-NB) approach with feature selection. 93.1 F1 success rate was obtained in 3600 datasets. Using the Doc2Vec word embedding method, Safalı et al. [19] classifies academic documents belonging to 9 different categories using RNN and LSTM architectures. Aydoğan and Karcı [20] created two different unlabeled Turkish datasets and trained using Word2Vec method. CNN, RNN, LSTM and GRU methods are used in the study. The variations of the architectures created in terms of depth are compared and their effects on the accuracy rate are analyzed. Köksal et al. [21] used the TTC-4900 dataset in their experiments. This dataset is similar to the TTC-3600 dataset. The TTC-4900 dataset consists of 700 examples of both Turkish and English texts belonging to 7 different classes, and has a total of 4900 news documents. Data correction was applied primarily in the study. Then stop-words in Turkish and then English are removed. Finally, the root separation (lemmatization) process is applied. Correcting the original data improved the f1 score while lemmatizing decreased it. Accordingly, 90% f1 score was obtained for the original dataset, while correcting the data without applying lemmatizing, the f1 score increased to 91.77%.

The aim of this study is to compare the success rates of classifying Turkish news texts by using Deep Learning and Doc2Vec methods with the methods studied so far in the literature. In this context, the TTC-3600 [13] news dataset has been recorded as 4 different datasets according to the preprocessing steps applied. After the Doc2Vec training model of each dataset was created, it was classified with CNN, GNB, RF, NB and SVM. Better accuracy rates have been achieved in the developed model compared to studies in the literature.

The remainder of the article is organized as follows: In Chapter 2, information is given about the methods used, and in the material and method section in Chapter 3, details about the dataset, preprocessing stages and the models created are given. The results of the method suggested in Chapter 4 were compared with previous studies and the article was finalized.

II. METHODOLOGY

A. Doc2Vec

Word Embedding method has been developed so that the texts can be perceived by the computer [22]. It is based on artificial neural networks and words are represented as vectors. Doc2Vec model was used as word embedding method in the study. Doc2Vec, developed by Quoc Le and Tomas Mikolov, generates a vector representing the document to predict the target word [23]. When doing this, the length of the document is not counted. It has two different methods. One of them is the Distributed Memory Model of Paragraph Vectors (PV-DM) and the other is the Distributed Bag Of Words of Paragraph Vector (PV-DBOW).

In the PV-DM method, each paragraph is accepted as a word and each paragraph has a special identity, namely a vector representation. First, vectors are started randomly. It acts as a moving memory, taking into account what is missing in the current context. While the document vector represents the concept of the document, the word vector represents the concept of the word [23]. PV-DBOW uses a paragraph vector to classify words in the document instead of guessing the target word. It is a structure that consumes little memory and less resources because it does not need to save word vectors.

B. Convolutional Neural Network (CNN)

Deep Learning [25] is a set of methods consisting of artificial neural networks based on deep architecture, the number of hidden layers is increased and a feature of the problem is learned in each layer. In this architecture, the attribute learned in each layer creates an input to the upper layer. Thus, a structure in which the simplest to the most complex feature is learned from the lowest layer to the top layer is established [26]. The main purpose of deep learning is to transform the input data into a state that can provide a more effective learning with various transformations and then operate the learning algorithm [27].

Although CNN, which is a specialized architecture of deep learning, is very successful especially in image processing, it has been frequently used in text classification studies in recent years. A CNN architecture can be studied in three parts, basically the convolutional layer, the pooling layer and the fully connected layer. In the convolutional layer, the input is filtered and feature maps are obtained. Feature maps are sampled in the pooling layer and a more general and faster learning of the network is provided. Finally, each neuron in the fully connected layer generates an output based on all inputs from the previous layer. Each layer extracts attributes based on the result of the previous layer and can learn the attribute hierarchy by combining and training all layers. The aim here is to achieve effective learning starting from low level details to high level details.

C. Naive Bayes

Naive Bayes is one of the simplest, understandable and easily applicable machine learning algorithms used in classifying text created using Bayes' theorem. With this method, the probability that the target attribute of a sample belongs to the class value can be found [28].

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (1)$$

Where, $P(c|x)$ is the probability of instance x being in class c, $P(x|c)$ is the probability of generating instance x given class c, $P(c)$ is the probability of occurrence of class c and $P(x)$ is the probability of instance x occurring.

D. Gauss Naive Bayes

Gauss Naive Bayes enables classification of numerical data with Gaussian distribution as well as categorical data. Working with Gauss (Normal distribution) is easiest because it is only necessary to estimate the mean and standard deviation from the training data. We can calculate the mean and standard deviation of input values (x) for each class.

$$mean\ (\mu) = \frac{\sum x_i}{N} \quad (2)$$

Where N is the number of samples and x_i is the value for each input variable in the training data.

$$standard\ deviation\ (\sigma) = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}} \quad (3)$$

Where N is the number of samples, $x_i$is the i th sample, and μ is the mean value. The difference of each sample from the mean is squared and added. It is then divided by the total number of samples. By taking the square root of this, the standard deviation is obtained.

When making predictions, these parameters can be added to the Gaussian Probability Density Function with a new entry for the variable, and in return an estimate of the probability of this new input value for that class is provided.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4)$$

$f(x)$ is the Gaussian Probability Density Function. Here and above is the mean and standard deviation we calculated. is the numeric constant, the numeric constant or the number of Euler raised to power, and is the input value for the input variable..

E. Random Forest

Random forest algorithm is a supervised classification algorithm. The algorithm randomly creates a forest. There is a direct relationship between the number of trees in the algorithm and the result it can achieve. As the number of trees increases, a precise result can be obtained. There are several reasons why the random forest classification method is preferred. It can be used in both classification and regression tasks. For this algorithm, if there are enough trees in the forest, the probability of overfitting problem is reduced. Over-learning is a critical problem that negatively affects results. Another advantage is that the classifier can be modeled for categorical values.

F. Support Vector Machine

Support Vector Machine is capable of separating data into two or more classes with separation mechanisms in linear form in two-dimensional space, planar in three-dimensional space and hyperplane in multi-dimensional space [29]. The method, which is frequently used in determining the classes that can be separated linearly, is successfully used in the classification of nonlinear data by moving the input space that cannot be separated linearly through kernel functions to this higher dimensional linearly separable space.

III. Materials and Methods

A. Dataset

The TTC-3600 dataset, which was prepared to be used widely in Turkish news classification studies, was compiled by Kılınç et al. [13] in 2015. TTC-3600, an easy-to-use and well-documented dataset published in Turkish news datasets in recent years, is accessible [30]. The dataset consists of 3600 documents containing 600 news / texts in 6 categories: economy, culture and arts, health, politics, sports and technology. News texts were collected from relevant news portals via Rich Site Summary (RSS) between May and July 2015 [13].

TABLE I. TTC-3600 Dataset [13]

Category	Total Number of Data (Documents)
Economy	600
Culture and Arts	600
Health	600
Politics	600
Sports	600
Technology	600
Total	**3600**

Some important preprocessing steps were applied on the TTC-3600 dataset. In order to investigate the effect of these stages on the success rate, 4 different datasets were created according to the preprocessing steps applied. These datasets were determined as the original dataset (Orig-DS), cleaned dataset (C-DS), dataset prepared by reducing words to their roots using Zemberek (Zemb-DS) and both cleaned and Zemberek applied dataset (Clean+Zemb-DS).

B. Text Preprocessing

Data preprocessing is one of the most important factors affecting the success rate. Therefore, the following text preprocessing steps were applied before the TTC-3600 dataset was vectorialized. Before applying the preprocessing stage to the dataset, the word clouds with the most repetitive first 50 words belonging to the classes are shown in Table 2.

As discussed in word clouds, stop words are used quite a lot in each class. These words were removed from the dataset because they did not have any distinguishing features and could negatively affect the success rate.In addition, all words were converted to lowercase, all characters such as numbers, symbols and punctuation marks except letters were cleared. After these steps, the original TTC-3600 dataset was saved as C-DS.

TABLE II. WORD CLOUDS OF CLASSES IN DATASET

Economy	Culture and Arts	Health
Politics	**Sports**	**Technology**

Zemberek [31] library was used for the separation process, which is another important preprocessing step. For this, firstly, the words in the original dataset were divided into root form. This was recorded as Zem-DS. Finally, both data cleaning and rooting processes were applied to the original dataset and Clean+Zem-DS was created. The created datasets are ready for Doc2Vec training model.

C. Doc2Vec Model

The datasets created are first transformed into vector by creating the Doc2Vec training model. There are some important parameters when creating the Doc2Vec model. These; feature vector size (vector_size), Dov2Vec methods (dm), maximum distance (window) between the current and predicted word in a sentence, ignoring all words whose total frequency is less than the specified value (min_count), and the number of iterations. The parameters and values determined in this study are shown in Table 3.

TABLE III. DOC2VEC MODEL PARAMETERS

Parameters	Value
vector_size	100
dm	1
window	3
min_count	5
iteration	50

D. CNN Model

After Doc2Vec model was created for each dataset, each one was ready for classification. The proposed CNN model has a maximum pooling operation. After each convolution layer, the feature maps are pooled and their dimensions are reduced, thus reducing the variation in features. Then flatten and dense layers are used. ReLU function is used for activation in hidden layers and Softmax activation function is used in the output layer of the model. The CNN architecture used in the study is shown in Table 4.

TABLE IV. CNN ARCHITECTURE USED IN THE STUDY

CNN Layers
Convolution2D - 16 (3x3 Filter)
MaxPooling - (1x1 Filter)
Convolution2D - 32 (3x3)
MaxPooling - (1x1 Filter)
Convolution2D - 64 (3x3)
MaxPooling - (1x1 Filter)
Convolution2D - 128 (3x3)
MaxPooling - (1x1 Filter)
Flatten
Dense – 4096 (Activation Fonc. = ‘ReLu’)
Dense – 4096 (Activation Fonc. = ‘ReLu’)
Dense (4, Activation Fonc. = ‘SoftMax’)

IV. EXPERIMENTAL RESULTS

In this study, our aim is to compare the success rates as a result of classifying the datasets created according to the preprocessing stages applied to the TTC-3600 dataset by creating the Doc2Vec model. In order to classify in the proposed method, documents expressed as vectorial with Doc2Vec model training are divided into 90% training and 10% test. Then, the datasets were classified using the deep learning model CNN and traditional machine learning methods GNB, RF, NB and SVM. When classifying with CNN, Python libraries Tenserflow and Keras [32-33] are used. While making machine learning classifications, Knime software, which is a data analysis platform, was used [34].

In the method we propose in terms of classifying Turkish news texts, the highest accuracy rate was obtained as 94.17% as a result of the CNN classification of the PV-DM model of the Clean + Zemb-DS dataset. The accuracy rates obtained by classifying each dataset after creating the Doc2Vec training model are given in Figure 3.

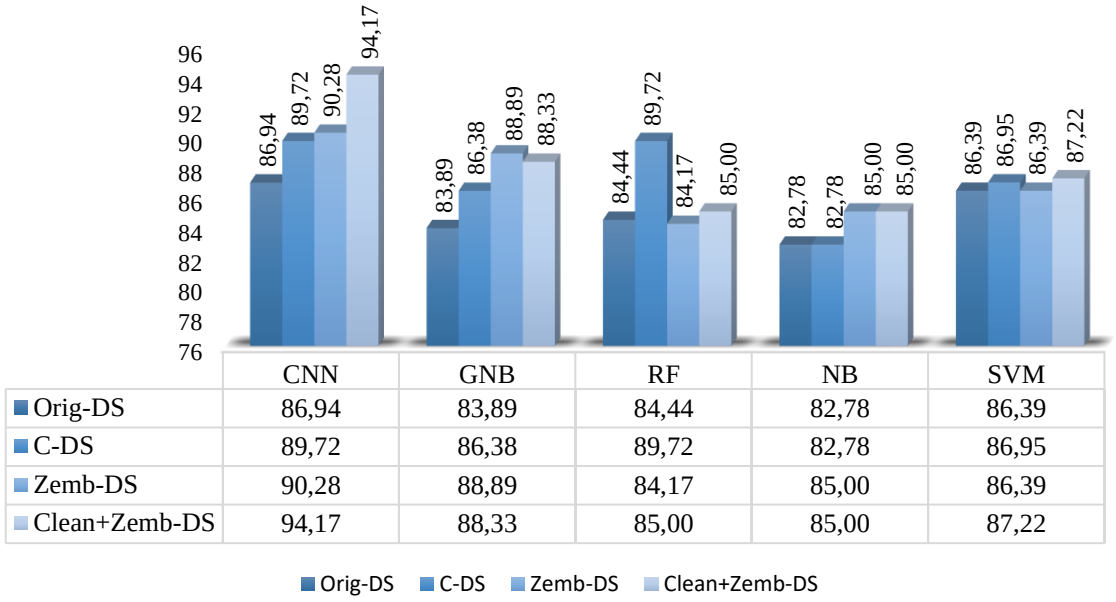


Fig. 1. Comparison of accuracy rates of CNN, GNB, RF, NB and SVM classification methods for each dataset.

When the results are examined according to the accuracy rates, in each dataset, CNN gives better results than other machine learning classification methods. While the accuracy rate obtained with CNN increases when the text preprocessing steps are applied, it is seen that some text preprocessing stages decrease the success rate in some machine learning methods.

Basically, accuracy can immediately tell us whether a model is properly trained and how it can perform overall. However, it does not give detailed information about its application to the problem. Therefore, we need to know the precision, sensitivity and f1 score to get a better answer. Therefore, for all classification procedures, other success criteria were also looked at.

Accuracy value is calculated by the ratio of the areas that we correctly estimated in the model to the total dataset. Precision shows how many of the values we estimate as Positive are actually Positive. The precision value is particularly important in situations where the cost of False Positive estimation is high. Recall is a metric that shows how much of the transactions we need to predict as Positive. Recall value is also a metric that will help us in situations where the cost of estimating as False Negative is high. It should be as high as possible. F1 Score value shows us the harmonic mean of Precision and Recall values. The reason why it is a harmonic average instead of a simple average is that we should not ignore extreme cases.

TABLE V. ORIG-DS SUCCESS MEASURES (%)

Classification	Accuracy	Precision	Recall	F1 Score
CNN	**86.94**	86.67	87.17	86.83
GNB	83.89	83.15	83.60	83.10
RF	84.44	84.98	84.45	84.42
NB	82.78	82.70	82.80	82.60
SVM	86.39	86.30	86.40	86.20

TABLE VI. C-DS SUCCESS MEASURES (%)

Classification	Accuracy	Precision	Recall	F1 Score
CNN	**89.72**	89.50	89.50	89.50
GNB	86.38	86.23	86.35	86.07
RF	**89.72**	89.80	89.73	89.72
NB	82.78	82.50	82.80	82.40
SVM	86.95	86.90	86.90	86.80

TABLE VII. ZEMB-DS SUCCESS MEASURES (%)

Classification	Accuracy	Precision	Recall	F1 Score
CNN	**90.28**	89.67	90.17	90.00
GNB	88.89	89.42	88.87	89.00
RF	84.17	85.17	85.00	84.80
NB	85.00	85.10	85.00	84.80
SVM	86.39	86.60	86.40	86.30

TABLE VIII. CLEAN+ZEMB-DS SUCCESS MEASURES (%)

Classification	Accuracy	Precision	Recall	F1 Score
CNN	**94.17**	94.17	94.19	94.00
GNB	88.33	88.17	88.20	88.13
RF	85.00	84.18	84.18	84.05
NB	85.00	85.00	85.00	84.90
SVM	87.22	87.20	87.20	87.20

When the success criteria are evaluated, the rankings in precision, sensitivity and f1 score are exactly the same as the accuracy criteria order. In addition, below, the graphs of training and test accuracy and loss according to the CNN training model results are given in Figure 4 and Figure 5.

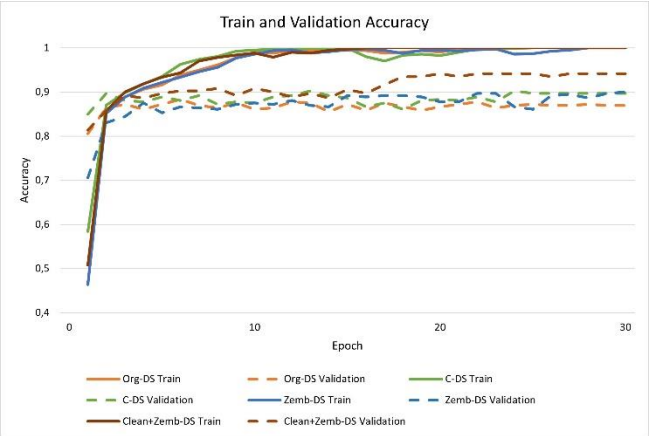


Fig. 2. CNN training and validation accuracy chart for each dataset.

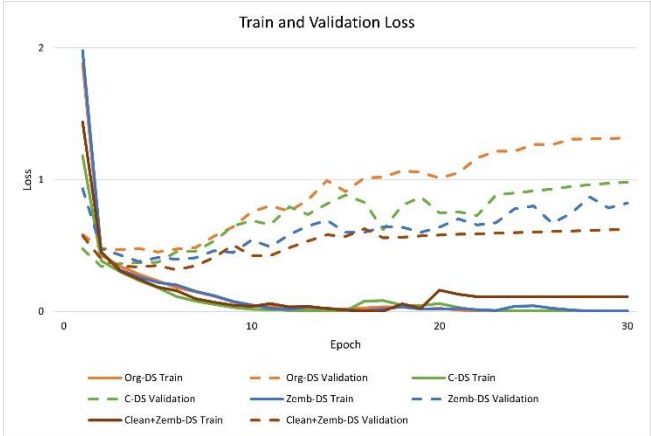


Fig. 3. CNN training and validation loss graph for each dataset.

TABLE IX. COMPARISON TABLE

Study	Dataset	Model	Accuracy (%)	F1 Score (%)
Kılınç, D. et. al. [13]	TTC-3600	RF + Zemberek + ARFS	91.03	-
Kılınç, D. [14]	TTC-3600	J48 + Boosting	85.52	-
Acı, Ç. İ. [17]	TTC-3600	Word2Vec + CNN + Zemberek	93.30	-
Yıldırım, S. and Yıldız, T. [18]	TTC-3600	M-NB + IG	-	93.33
Yıldırım, S. and Yıldız, T. [18]	TTC-4900	M-NB + IG	-	90.00
Köksal [21]	TTC-4900	SW + No Lem.	91.77	-
Proposed Method	TTC-3600	Doc2Vec + CNN + (Clean+Zemb-DS)	**94.17**	**94.00**

The summary of the results of the proposed system and the results obtained in previous studies with TTC-3600 and TTC-4900 datasets are given in Table 9. When compared with the F1 score and accuracy of previous studies, it is seen that the model we suggested gives better results with a success rate of 94.00% and 94.17%, respectively.

V. CONCLUSION

After the TTC-3600 dataset consisting of Turkish news texts belonging to 6 different categories was recorded as 4 different datasets according to the text preprocessing stages, the Doc2Vec training model of each dataset was created. Then, the accuracy rates obtained as a result of classification with deep learning classification method CNN and traditional machine learning classification methods GNB, RF, NB and SVM scores were compared. When the accuracy rates are compared, the result of classifying the Clean + Zemb-DS dataset with CNN is 94.17%. It was noted that better results were obtained when comparing the proposed method with the previous studies.

REFERENCES

[1] Internet: World Internet Statistics. https://www.internetworldstats.com/stats.htm, 12.12.2020.

[2] N. Indurkhya, F.J. Damerau, Handbook of Natural Language Processing, Chapman & Hall/CRC, 2010.

[3] E. Alpaydin, Machine learning : The New AI, The MIT Press, 2016

[4] Ç. Çatal, K. Erbakırcı, Y. Erenler, "Computer-based Authorship Attribution for Turkish Documents", Turkish Symposium on Artificial Intelligence and Neural Networks, 2003.

[5] Amasyali, M.F.; Diri, B. Automatic Turkish text categori-zation in terms of author, genre and gender. In: Natural Language Processing and Information Systems, Berlin: Springer. 2006; pp. 221-226.

[6] Çataltepe, Z.; Turan, Y.; Kesgin, F. Turkish document classification using shorter roots. In: Proceedings of IEEE Signal Processing and Communications Applications Con-ference (SIU), Newyork: IEEE, Eskisehir, Turkey. 2007; pp. 1-4.

[7] Guran, A.; Akyokus, S.; Guler, N.; Gurbuz, Z. Turkish text categorization using n-gram words. In: Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA). 2009; pp. 369-373.

[8] Amasyalı, M.F.; Beken, A. Measurement of Turkish word semantic similarity and text categorization application. In: Proceedings of IEEE Signal Processing and Communications Applications Conference, Newyork: IEEE. 2009; pp. 1-4.

[9] Torunoğlu D, Çakırman E, Ganiz MC, Akyokuş S, Gürbüz MZ. "Analysis of preprocessing methods on classification of Turkish texts.". International Symposium on Innovations in Intelligent Systems and Applications (INISTA), İstanbul, Türkiye, 15-18 June 2011.

[10] Tufekci, P.; Uzun, E. Author detection by using different term weighting schemes. In: Proceedings of IEEE Signal Processing and Communications Applications Conference (SIU), Newyork: IEEE, Trabzon, Turkey. 2013; pp. 1-4.

[11] Uysal AK and Gunal S. The impact of preprocessing on text classification. Information Processing and Management 2014; 50: 104-112.

[12] V.E. Levent, B. Diri, "Türkçe Dokümanlarda Yapay SinirAğları ile Yazar Tanıma", 15. Akademik Bilişim Konferansı, 735–741, Mersin, 2014.

[13] Kılınç D, Özçift A, Bozyigit F, Yıldırım P, Yücalar F, Borandag E. "TTC-3600: A new benchmark dataset for Turkish text categorization". Journal of Information Science, 43(2), 174-185, 2015.

[14] Kılınç, D. Topluluk Öğrenme Modellerinin Türkçe Metin Sınıflandırmasına Etkisi. Celal Bayar Üniversitesi Fen Bilimleri Dergisi, 2016, 12.2.

[15] F. Baskaya, I. Aydin, "Haber metinlerinin farklı metin madenciliği yöntemleriyle sınıflandırılması", International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, 1–5, 2017.

[16] O. Kaynar, Z. Aydın, Y. Görmez, "Sentiment Analizinde Öznitelik Düşürme Yöntemlerinin Oto Kodlayıcılı Derin Öğrenme Makinaları ile Karşılaştırılması", Bilişim Teknolojileri Dergisi, 10(3), 319 - 326, 2017.

[17] Çiğdem, A. C. I., and Adem ÇIRAK. "Türkçe Haber Metinlerinin Konvolüsyonel Sinir Ağları ve Word2Vec Kullanılarak Sınıflandırılması." Bilişim Teknolojileri Dergisi 12.3 (2019): 219-228.

[18] Yıldırım, Savaş; Yıldız, Tuğba. Türkçe için karşılaştırmalı metin sınıflandırma analizi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 2018, 24.5: 879-886.

[19] Safali, Yaşar, et al. "Deep Learning Based Classification Using Academic Studies in Doc2Vec Model." 2019 International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE, 2019.

[20] Aydoğan, Murat, and Ali Karci. "Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification." Physica A: Statistical Mechanics and its Applications 541 (2020): 123288.

[21] Köksal, Ömer. "Tuning the Turkish Text Classification Process Using Supervised Machine Learning-based Algorithms." 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). IEEE, 2020.

[22] O. Levy and Y. Goldberg, "Neural Word Embedding as Implicit Matrix Factorization," in Advances in Neural Information Processing Systems 27 (NIPS 2014), 2014.

[23] Lau, Jey Han, and Timothy Baldwin. "An empirical evaluation of doc2vec with practical insights into document embedding generation." arXiv preprint arXiv:1607.05368 (2016).

[24] Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. 2014.

[25] L. Deng, D. Yu, "Deep Learning: Methods and Applications", Foundations and Trends in Signal Processing, 7(3–4), 197–387, 2014.

[26] G. Isik, H. Artuner, "Recognition of radio signals with deep learning Neural Networks", 24. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Zonguldak, Türkiye,16-19 Mayıs 2016.

[27] H. Yalçın, "Derin Anlama Ağları ile İnsan Aktiviteleri Tanıma", Türkiye Robotbilim Konferansı, İstanbul, 26 - 27 Ekim 2015.

[28] Kartal, Elif, Enformatik Programı, and M. Erdal BALABAN. "Sınıflandırmaya Dayalı Makine Öğrenmesi Teknikleri ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama," Doktora Tezi, Haziran 2015, pp 19-20.

[29] Güran, Aysun, Mitat Uysal, and Özge Doğrusöz. "Destek vektör makineleri parametre optimizasyonunun duygu analizi üzerindeki etkisi," DEÜ Mühendislik Fakültesi Mühendislik Bilimleri Dergisi 48, 2014,pp. 87- 88.

[30] Internet: UCI-Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/TTC3600%3A+Benchmark+dataset+for+Turkish+text+categorization, 12.12.2020.

[31] Akin A, Akin MD. "Zemberek, an open source NLP framework for Turkic Languages". Structure, 10, 1-5, 2007.

[32] Internet: Tensorflow. https://www.tensorflow.org/, 12.12.2020.

[33] Internet: Keras. https://keras.io/, 12.12.2020.

[34] Internet: KNIME Open for Innovation, End to End Data Science, https://www.knime.com, 12.12.2020.

Comparative Analysis of Different Algorithms for Image Denoising

Mohammad Ikhsan bin ZAKARIA
Computer Science and Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkiye
zakaria.mohammad@std.izu.edu.tr

***Abstract*—Image denoising attends to be the way to restore image from noise during the image was taken. There are quite a lot of algorithms on how we manage to improve broken image into better quality. In this paper we are going to compare few algorithms such as Non-Linear Means, Wavelet Transform, BM3D and Total Variation (TV) minimization algorithms. We also measure the Mean Square Error to estimate or compare original image with denoised image. Such that we can use them to improve the image quality using image denoising technique. Our experiments we use variety of noises and its level and take that image to denoise and then calculate PSNR as a result. The noises that we used are Gaussian, Speckle, Salt Pepper and Poisson.**

Keywords—Comparative, Denoising Algorithms

I. Introduction

Digital image processing is one of main part in machine learning or Computer Vision. In order to have precise result, images should be processed before analyzing into more specific research in machine learning. Images with noise is the challenges before processing with data analyzing. Without removing noise from the image, the result would be no good. Briefly noise is unwanted pixel that exists in any image taken from camera or other kind of devices. So to restore the image into a better visual quality we need denoising techniques. Difference noises, noisy level and denoising techniques will be discussed in this paper.

Generally, image noises are divided into two models. There are additive and multiplicative model.

A. Additive Model

Corrupted signal image noise can be presented by adding noise to original image. Simply defined as follow.

$$w(x,y) = s(x,y) + n(x,y) \tag{1}$$

where $s(x,y)$ in the noisy image is the original bit and $n(x,y)$ is the noise which produce the corrupted signal result $w(x,y)$ locates the pixel location.

B. Multiplicative Model

Another kind of noise model is multiplicative noise. This noise present when we multiply noise signal with original signal. It is defined in the following algorithm.

$$w(x,y) = s(x,y) \times n(x,y) \tag{2}$$

Definition in the algorithm above tells us the same, but what make different is that $s(x,y)$ is multiplied by $n(x,y)$ so the noise signal will be different than additive model.

To prove that denoising algorithms are working well, we need to have make up noise using noise algorithms. These noises have their own purposes and function. There are default noises will be described as follow.

A. Gaussian Noise

Gaussian noise also known as normal noise in predefine density function. It is widely used for adding noise in image. Gaussian noise can generated randomly and separate in image and defined by the following.

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2} \tag{3}$$

where z defined as intensity, $\bar{z}$ is the average (mean) value of z and σ is standard deviation. Variance of z is the standard deviation square σ^2. This function can be plot as describe in Fig.1.

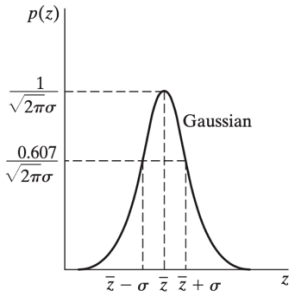


Figure 1 Gaussian Noise

B. Salt and Pepper Noise

Salt and pepper noise or impulse noise which can be either positive or negative. bipolar impulse noise also is called *salt-and-pepper* noise[1]. Impulse noise generally convert to digital as pure white and black values in an image. The values are maximum and minimum as white and black. For that reason the appearance of noise is black as 0 and white as 255 pixel.

C. Speckle Noise

Speckle noise [2][3] can be described as multiplicative noise. This noise can degrade original image into high signal noise image. This noise commonly appears in any coherent systems for instance SAR images or ultrasound images. When there is random fluctuations in signal from an object conventional

radar which is not bigger than image-processing element, the speckle can appeared.

II. Denoising Algorithms

Image denoising defined as technique to restore or degrade the image, reducing the noise content of the image. In the following, we will describe some denoising algorithms basic which will be used in experimentation.

A. Spatial Domain Filtering

When there is additive noise spatial filtering can be solution to remove the noise. Spatial domain techniques are very efficient computation and require less processing resources to be implemented. This technique can be denoted by expression below.

$$g(x,y) = T[f(x,y)] \tag{4}$$

where $f(x,y)$ is the input image, $g(x,y)$ is the output image, and T is an operator on f defined over a neighborhood of point (x,y). The operator can be applied to a single image or multiple images by summing pixel by pixel of a sequence of an image noise reduction.

Basically spatial domain filtering can classified by linear and non-linear filtering techniques.

1) Linear Filters

This type of filtering will be only when there is additive noise is present[4]. The optimal filter is a mean filter for noise signal such as Gaussian noise which can be measured by mean square error. This filter will blur edges, remove lines and the other fine the detail. It specified in two sub filter such as Mean filter and Wiener filter.

a) Mean Filter

Mean filter can formed as reducing the intensity variation between one pixel to next pixel. It takes the average of pixel surrounds it. As the result it will make a pixel under mask and the image become smooth.

b) Wiener Filter

This filter remove noise from corrupted signal. Image restoration with this filter require Fourier transform of frequency-domain. To perform this filtering operation we need to know the spectral properties of original signal and the noise itself and the result will be as close as original signal. This filter can formed as the following.

$$f(x,y) = \left[\frac{H(u,v)^*}{H(u,v)^2 + \left[Sn(u,v)/Sf(u,v)\right]}\right] G(u,v) \tag{5}$$

Where $H(u,v)$ is the degradation function and $H(u,v)^*$ is its conjugate complex. $G(u,v)$ as the degraded image. $Sn(u,v)$ and $Sf(u,v)$ are the power of spectra of original image and the noise. It assumes noise and power spectra priori object.

2) Non-linear Filter

When there is a multiplicative noise, this filter can restore signal. With this filter the noise can be removed without identifying it explicitly. Median neighborhood pixel is determined by the value of output pixel. With spatial filters we use low pass filtering of group pixels and the noise covers the higher region of frequency spectrum. Basically, noise is removed and as the result image will be blurred or edge loss.

a) Median Filter

Image restoration with median filter can be done by finding median value by across window and replacing each entry in the window. Median filter frankly describe as moving window principle and use 3x3 or odd number matrices.

B. Transform Domain Filtering

Generally transform domain filtering can be divided into data adaptive and non-adaptive filters. Transform domain includes wavelet based filtering techniques and spatial frequency filtering techniques.

1) Wavelet Transform

The transform constructed by a set of building blocks which represents a signal or function[7]. The expansion of this system returns time frequency localization of signal.

C. Total Variation

This algorithm is implemented in infrared images, medical images, remote hyperspectral and multispectral images. In medical diagnosis it requires precise detection. Li and Que[6], they found that total variation (TV) filter removes noise effectively.

D. Block Matching and 3D (BM3D) Filtering

This denoising technique based on the local image sparse representation in transform domain. The sparsity is grouped 2D image patches into 3D groups. BM3D grouping and filtering is named as collaborative filtering. This denoising method can be implemented in four steps[6].

- Get image patches similar to given image patch and grouping them in 3D block
- 3D linear transform of 3D block
- Shrinkage of the transform spectrum coefficients
- Inverse 3D transformation

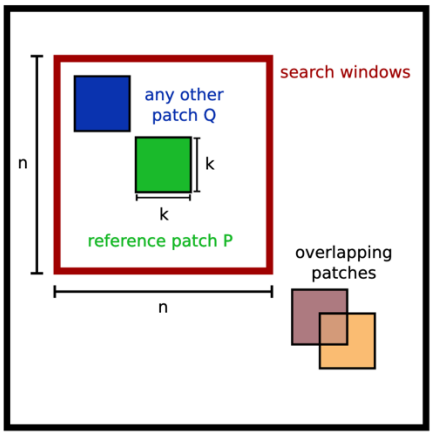


Figure 2 Patches, search windows and overlapping [6]

The process of finding similar block or finding patch can be form as follow.

$$\rho(p) = \{Q: d(P,Q) \leq \tau^{hard}\} \quad (7)$$

Where P denotes as patch whose size is $k^{hard} \times k^{hard}$ of image loop. $d(P,Q)$ is the Euclidean distance between blocks.

III. Measurements

A. Mean Square Error

Mean square error is the method where we can compare restored image to original image, calculate the different error between them. The mean square error generally defined as cumulative squared error between the restored image and original image. It is form as follow.

$$MSE = \frac{\sum_{M,N}[I_1(m,n)-I_2(m,n)]^2}{M*N} \quad (6)$$

Where I_1 is original image and I_2 is noisy image, m and n are the height and weight of the respectively images. In our experiment the average or MSE are less than 0.1.

B. Peak Signal to Noise Ratio

Basically peak signal noise ratio defined as the expression for ratio between maximum power of signal and power of signal image noised[8]. The result or expression should be in decibel (dB) scale.

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right) \quad (7)$$

Where R is the maximal power of the signal image. The PSNR is calculated based on MSE

IV. Experimental Results

In our experiment we tried to implement various of denoising techniques. Every algorithm aims to produce better solution to remove noise from the image. Also the algorithms sometimes are not used in same fields. However we are going to implement it in script. By calculating the mean square error (MSE) and Peak Signal Noise Rate (PSNR) we try to present the best result. Moreover we also add variant of noise level and type for each experiment. We can see the result of image denoising technique as following. In Fig 3, we made experiment by taking original image (image without noise), then we add Gaussian noise (sigma) and processed it with denoising algorithms which we already described it. During denoising computation we also take MSE and PSNR to finalize the result. As a result of this denoising technique, Wavelet VisuShrink came out with blur which caused by noise elimination. On the other hand block matching 3D came out with better result and closer to original image (noise free).

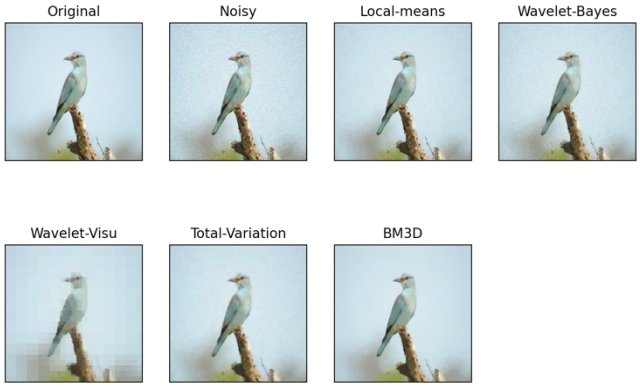


Figure 3 Testing Result with Gaussian Noise

A comparative analysis has been processed between Non-linear mean, Wavelet Bayes-Shrink, Wavelet Visu-Shrink, Total Variation and Block Matching and 3D techniques. As results in table I, we found out that BM3D gave us lesser MSE with Poisson noise than Gaussian noise. On the other hand, Total Variation came out lesser using Salt and pepper noise.

TABLE I. MEAN SQUARE ERROR(MSE)

Denoising Algorithms	Noises			
	Gaussian	***Salt and Pepper***	***Speckle***	***Poisson***
Non-Linear Mean	0.02887	0.16898	0.02311	0.00172
Wavelet-Bayes Shrink	0.03414	0.13345	0.02956	0.00254
Wavelet-Visu Shrink	0.05134	0.11457	0.04504	0.00648
Total Variation	0.02854	**0.09982**	0.02811	0.02186
BM3D	**0.02446**	0.16642	**0.01976**	**0.00161**

Based on our observation during experimentation, in table II, BM3D gave best result or got highest PSNR which is 57.9996. so based on this result we conclude that BM3D can perform best. We also present the color channel of the image which is [175 201 214] and image size is 3686400 pixels, we resize to 400 by 400 and the image size become 480000 pixel.

TABLE II. PEAK SIGNAL NOISE RATIO(PSNR)

Denoising Algorithms	Noises			
	Gaussian	***Salt and Pepper***	***Speckle***	***Poisson***
Non-Linear Mean	32.9298	17.5848	34.8649	57.3961
Wavelet-Bayes Shrink	31.47461	19.63475	32.7249	54.0410
Wavelet-Visu Shrink	27.93106	20.95985	29.0679	45.9027
Total Variation	33.03135	**22.1572**	33.16168	35.3461
BM3D	**34.37153**	17.71746	**36.2236**	**57.9996**

In fig. 4 we plot sample of one our experiment using Gaussian noise, MSE value changed based on different denoising techniques. BM3D returned lowest result as 0.025. In fig.5 we calculated the

PSNR and the plotting is same sample we used in first experiment which is Gaussian noise. BM3D returned PSNR as 34.37.

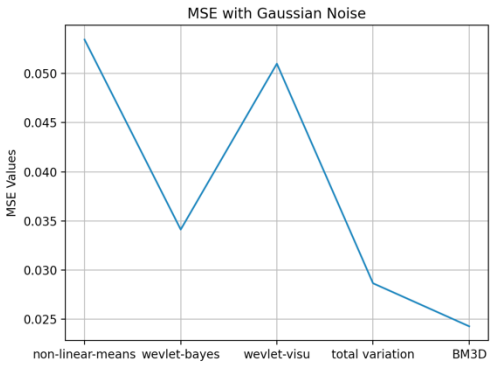


Figure 4 MSE Result

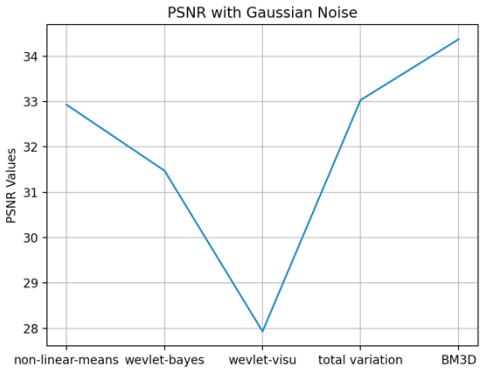


Figure 5 PSNR Result

CONCLUSION

According to out experimentation, we can conclude that returns better PSNR and has lesser MSE on BM3D than the other algorithm. Not to mention on what purpose the is analyzed for removing noises from the image. After experiment and we compare the denoising algorithms we can see that the more we high in peak signal noise ratio we performed better restoration for the noisy image and lesser peak signal noise ratio the restoration will not gave better quality. We intent to perform multiple images in future work, because with this way we can make more comparison on the denoising algorithms.

V. FUTURE WORK

As we know that computer vision has broad fields study on image processing which can be developed by using deep learning algorithms. So in this field we will try to use deep learning or artificial intelligent method to improve denoising techniques more accurately and to improve time consumed for each experiment and maybe we can manage to denoise multiple images at same time with deep learning.

REFERENCES

[1] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing 3rd Edition", 2008

[2] Umbaugh, S. E. 1998, Computer Vision and Image Processing, Prentice Hall PTR, New Jersey.

[3] Gagnon, L. 1999. Wavelet Filtering of Speckle Noise- ome Numerical Results, Proceedings of the Conference Vision Interface, Trois-Reveres.

[4] Pankaj Hedaoo and Swati S Godbole, "Wavelet Thresholding Approach for Image Denoising", International Journal of Network Security & Its Applications (IJNSA), July 2011 ,Vol.3, No.4.

[5] Qian Xiang, Xuliang Pang, "Improved Denoising Auto-encoders for Image Denoising". 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics. CISP-BMEI 2018

[6] March Lebrun, "An Analysis and Implementation of the BM3D ImageDenoising Method". Image Processing Online. 2012–08–08. Submitted on 2012–00–00

[7] Sachin D Ruikar, Dharmpal D Doye, "Wavelet Based Image Denoising Technique". IJACSA. Vol. 2, No.3, March2011

[8] D.Poobathy, Dr. R.Manicka, "Edge Detection Operators: Peak Signal to Noise RatioBased Comparison". I.J. Image, Graphics and Signal Processing,2014, 10, 55-6

Skin Lesions Segmentation and Classification for Medical Diagnosis

Merve Gun
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0001-5190-9466

Alaa Ali Hameed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-8514-9255

Mirsat Yesiltepe
Department of Computer Engineering
Yildiz Technical University
Istanbul, Turkey
0000-0003-4433-5606

Akhtar Jamil
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-2592-1039

***Abstract*— Classification and segmentation of various skin lesions play a very important role in the field of dermoscopy. Using computer-aided applications to detect cancerous cells and predict the lesion as benign and malignant can yield better results. Automatic estimation of skin disease from skin lesion images help practitioners to perform rapid diagnosis, provide early treatment and quick decision making. In this paper, Convolution Neural Network (CNN) is used to identify cancer-prone skin lesions from dermoscopy images. Experiments were performed on ISIC 2016 data set with two lesion classes (Malignant and Benign). The training was carried out with the Multiple Residual Neural Network (ResNet) architecture, where the data is pre-processed with different methods. Finally, the comparative analysis with other methods was also performed. The results indicated that the performance of our proposed method is also in line with state of the art methods.**

Keywords— lesion classification, Resnet, convolutional neural network, medical image analysis.

I. INTRODUCTION

Advances in deep learning methods in recent years have also led to the development of intelligent medical imaging-based diagnostic systems that can help the analysis of medical images and help professionals make better decisions for patient health.

Some lesions on the skin can be the harbinger of serious cancer types. Especially early detection of melanoma is very important for the success of the treatment process. In this project, skin lesion classification is made. A deep learning-based approach is presented to classify a dermoscopic image containing a skin lesion as malignant or benign.

Malignant is a type of skin cancer that occurs in the skin tissue and can cause death. The disease can be treated with early diagnosis. Benign is a more common but not dangerous type of skin lesion. The dermoscopic images of these two species are similar and may not be distinguished by human vision. For this reason, Computer Aided Diagnosis (CAD) systems, which are frequently used in the detection of diseases, can help patients and physicians in the detection of skin cancer [1]. Even if expert dermatologists use dermatology images for diagnosis, the accuracy rate of the expert diagnosis is estimated at 75–84% [2].

In the study conducted by Hameed et al. [3] in 2019, they used AlexNet CNN model for feature extraction on 9144 skin images. For classification, they reached an accuracy of 86.21% with SVM classifier. They showed that the features obtained from CNN models by using a 10-fold cross-validation technique to prevent overcompliance increased the classification performance of multiple skin lesions. In the study conducted by Ahmad et al. [4] in 2020, they proposed a new frame by fine-tuning the layers of ResNet152 and InceptionResNet-V2 models to the skin lesion images they received from Wuhan Hospital in China and achieved an accuracy of 87.42%.

Studies conducted with Deep Learning methods have shown that better prediction success has been achieved in many areas such as image classification compared to machine learning methods [1]. In the study of Rodrigues et al. [5] for the year 2020, it was seen that using CNN and classical machine learning methods together gave better predictions according to the results obtained by comparing them.

Deep Learning models do not need extra preprocessing to extract features that represent image data. CNN is the most widely used architecture among deep learning models. In 2015, He et al. [6] proposed a new CNN model called ResNet with a low computational cost. ResNet achieved a great success in the 2015 ImageNet competition with an error rate of 3.57% [1]. However, very deep neural networks are difficult to train as the number of parameters needed to training exponential increase with increasing number of layers increase. In addition, highly powerful computational resources are also required to train the algorithm faster. In theory, training error is expected to decrease as the depth increases. However, in reality, the training error increases with adding more layers to CNN. ResNet architecture has brought a solution to the problem of training error that gets worse / disappears as its depth increases.

Fig. 1 represents the training error (%) and test error (%) of a 20-layer and 56-layer flat network. It was observed that as the depth increased, that is, the number of iterations, training and test data errors increased.

In this paper we investigate the power of deep neural network using CNN for classification of skin lesion from images. Specifically, we employed ResNet on ISIC 2016 data set with for classification of lesion into malignant or benign.

The rest of the paper is organized as follows. The data set used in this study is described in section II. The proposed methodology for classification and segmentation is described in Section II. Section IV summarized the results obtained and paper was completed with concluding remarks.

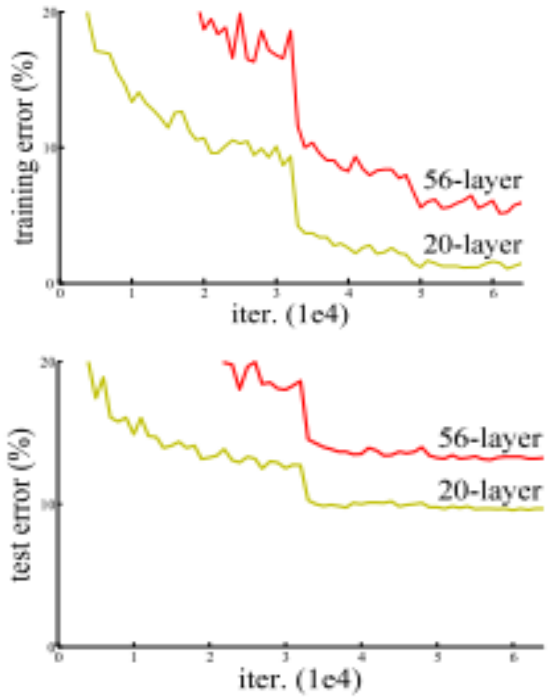


Fig. 1. Training error (top) and test error (down) [6]

II. DATA SET

Dermoscopic images are commonly used by dermatologists for various disease identification. In this study, data was obtained from International Skin Imaging Collaboration (ISIC) [7] which is freely available online. This data set includes total 1279 skin lesion images in RGB format which hare further divided into training (900) and testing (379) subsets. There are two classes in this data set: Malignant and Benign. Fig. 2 shows some samples for malignant tissues (a) and benign tissues (b).

(a)

(b)

Fig. 2. Sample images a) Malignant Tissue Image, b) Benign Tissue Image

TABLE I. DATA SET USED IN THIS STUDY

Class	Training	Test	Total
Malignant	173	75	248
Benign	727	304	1031
Total	**900**	**379**	**1279**

III. PROPOSE METHOD

A. *Deep Learning (DL)*

Deep Learning is a new machine learning method based on artificial neural networks that can automatically extract features from data and generally produces higher accuracy. Unlike classical machine learning methods, it does not require manual feature engineering. CNN is one of the most successful deep learning models for image data which is described in following section.

B. *Convolutional Neural Network (CNN)*

There are many CNN models such as ResNet, GoogleNet, VGGNet and AlexNet. CNN has a multiple neurons arranged layer structure [1]. These include Input Layer, Convolution Layer, Pooling Layer, Fully-Connected Layer and Output Layer.

In Convolution Neural Network, neurons are expressed in 3 dimensions; width, height and depth. Depth is equal to 3 due to the RGB image format, for example if the image data is color.

The basic architecture of Convolutional Neural Network is shown in Fig. 3 while fig.4 shows the architecture of ResNet18. In addition to these basic layers, the layers of many different CNN architectures vary.

1. *Convolutional layer:* They are the most important elements of deep learning architectures. Each layer consists of a certain size and fixed number of filters selected before the training process. Each filter detects a specific feature and produces a feature map in response [9].

   Filters create convolutional layers and are trainable feature extractors that are learned from data. Filters have the ability to learn properties of images such as color, size, edge during training.

2. *Pooling layer:* The Pooling layer, a subsampling layer, reduces the size of feature maps. As a result, only the activated properties are transmitted to the next layer [1].

3. *Fully connected layer:* It acts as a classifier. It usually consists of 1-2 fully bonded layers. Next comes the classification layer for multi-class data.

C. *Residual Network (ResNet)*

Residual blocks form the basis of the restoration architecture. One of the problems of deep learning is the problem of overfitting due to increasing depth. Overfitting means good training accuracy but poor test accuracy. Blocks are now suggested to solve this problem. Now in the block the input x is directly added to the output of the network i.e. it becomes f(x) + x.

In Figure 5, the output generated when a block is added to the CNN architecture is now summarized. In this case, output is given as in (1);

$$H(x) = F(x)+x \quad (1)$$

When the input of the network is equal to its output, F (x) = 0 '. So H (x) = x. The aim is to skip certain layers using jump links or leftover blocks to improve performance in the deeper layers of deep learning. Thus, improvement is provided to the problem of optimization and distortion in a deep network.

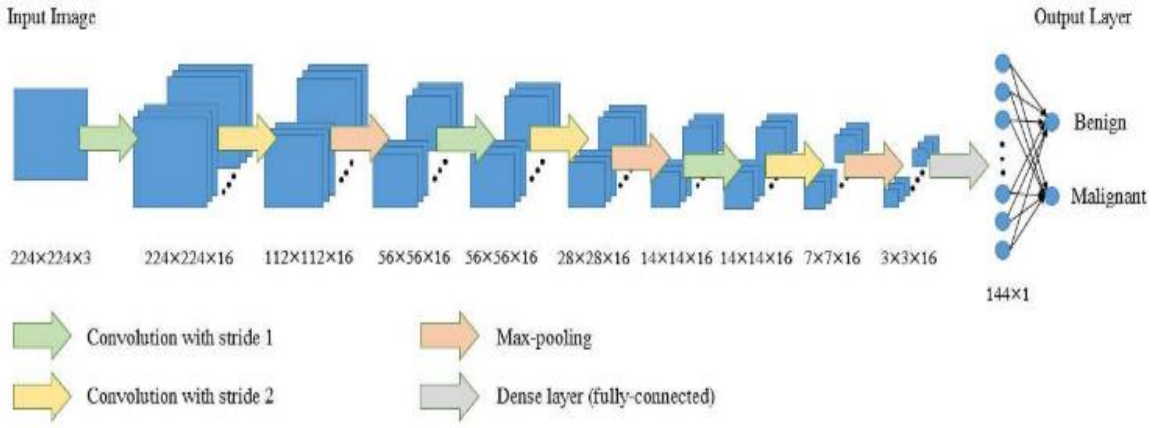


Fig. 3 Architecture of convolutional neural network [8].

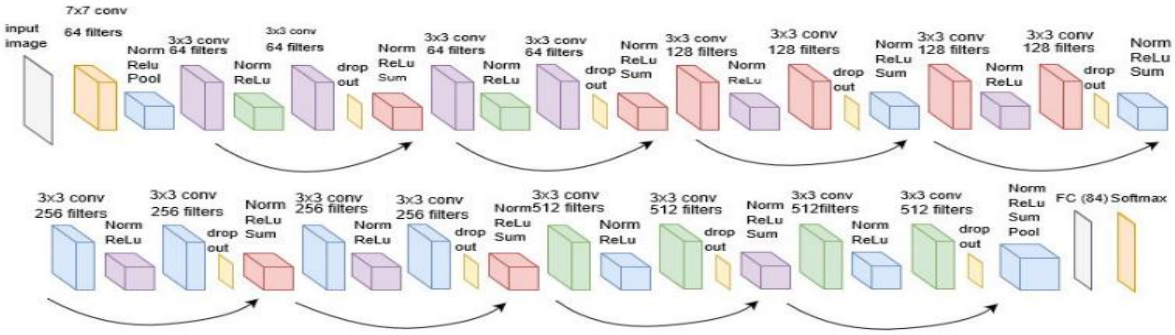
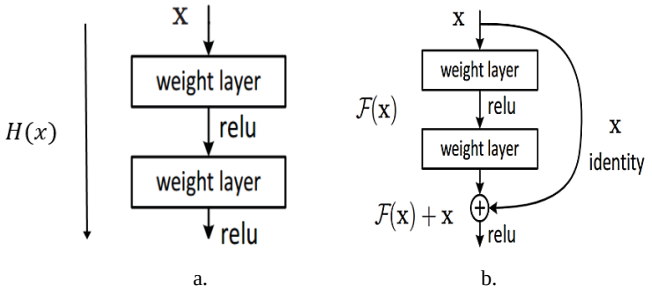


Fig. 4. ResNet18 Layers Architecture [10]



a. b.

Fig. 5 (a). Normal CNN architecture, b) Residual connection CNN [6]

IV. Experimental Results

In this study, 900 images were used for training and 379 images for testing. After training the data set using CNN, the classification and accuracy of skin lesions were calculated by learning with the ResNet18 model. We trained our model with 30 epochs. Accuracy and loss values are compared by using different optimizers to analyze their performance.

The images of our data set are divided into 2 classes. To measure the performance of the network, sensitivity, specificity, precision and F1 values were calculated and compared with other research results.

A. ResNet Classification

In our study, we applied a classification to our data set with the ResNet18 architecture. We resized the images we used in the data set to 224x224x3. ResNet18 architecture consists of 5 Convolution Layers, 2 Max Pooling Layers and 1 Fully Connection Layer (Figure 4). Since we have 2 classes in the data set, the "Output Size" value is arranged as "2" in the Fully Connection Layer. Accordingly, the "Classes and Output Size" values in the classification layer have been updated as "auto".

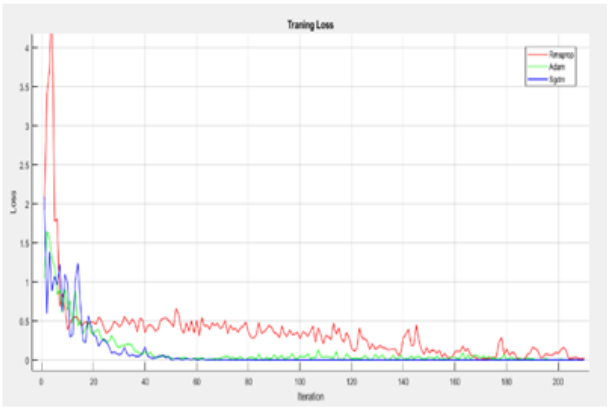


Fig. 6. Training accuracy according to different optimizers

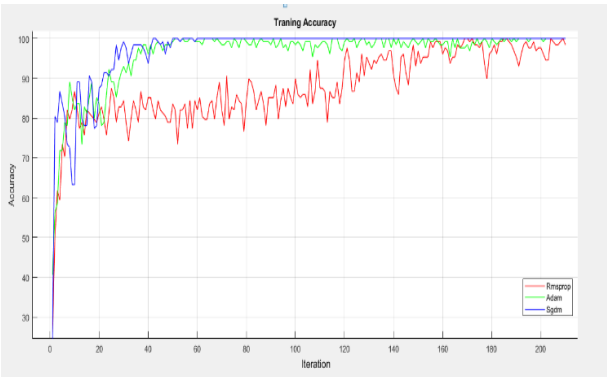


Fig. 7. Loss values according to different optimizer

B. Training

The image data was resized to 224x224x3. ReLU activation function has been selected as the activation function. Training data and test data were fed into classifier by loading into the image data store in the Matlab environment.

Table II summarizes the results obtained for each class using confusion matrix while Table III summarizes the results obtained for each classifier. The results show that Sgdm produced highest classification accuracy with 84,96%, then Adam which was 81,53%, and Rmsprop optimizer was 77,57%. It is also seen that the optimizer with the lowest training and test loss value is again Sgdm. For each optimizer, parameters were empirically obtained during the training. The accuracy and loss obtained for each classifier are shown in Fig. 6. and Fig. 7. Moreover, the quantitative results are presented in Table IV.

C. Evaluation

Confusion matrix is a special table layout that gives us the opportunity to visualize the performance of the algorithm. Each row of the matrix represents instances in a real class, and each column of the matrix represents instances in a predicted class [11].

TABLE II. CONFUSION MATRIX

		Predicted Values		
		Positive (Benign)	**Negative (Malignant)**	**Actual Totals**
Actual Values	**Positive (Benign)**	**290** 76,5%	**43** 11,3%	87,1% 12,9%
	Negative (Malignant)	**14** 3,7%	**32** 8,4%	69,6% 30,4%
	Predicted Totals	95,4% 4,6%	42,7% 57,3%	85,0% 15,0%

True Positive (TP) and False Negative (FN) values correspond to the number of lesions in a particular class that are true and false classified, respectively. True Negative (TN) refers to the number of lesions that do not belong to a particular class that are classified as not belonging to this class. False Positive (FP) values are the number of lesions incorrectly classified as belonging to a particular species [5].

In order to better evaluate the system performance, sensitivity (TPR - true positive rate) and specificity (TNR - true negative rate) measurements were calculated. In this case, sensitivity (3) means the ratio of appropriately classified Malignant cases to the number of Malignant cases, while specificity (4) is the ratio of appropriately classified Benign cases to the number of all Benign cases [9].

Accuracy measures the number of data correctly classified (2). High accuracy, sensitivity and specificity value indicate an efficient classification method [8].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$F1_Score = \frac{(2 * Precision * Sensitivity)}{Precision + Sensitivity} \quad (6)$$

TP = True Positive TN = True Negative

FN = False Negative FP = False Positive

P - number of positive samples (Benign), N - number of negative samples (Malignant), TP-True Positive: Benign is correctly defined as Benign; FP-False Positive: Benign is incorrectly defined as Malignant; TN-True Negative: Malignant is correctly identified as Malignant; FN-False Negative: Represents Malignant image data incorrectly identified as Benign.

TABLE III. CLASSIFICATION ACCURACY (%)

Optimizer	**ACC.**	**SEN.**	**SPE.**	**PRE.**	**F1 Score**
Sgdm	**84,96**	**87,09**	**69,57**	**95,39**	**91,05**
Adam	81,53	84,82	55,81	93,75	89,06
Rmsprop	77,57	85,21	42,65	87,17	86,18

In the training process, optimizing the training with different parameters is important for different performance results. Accuracy, sensitivity, specificity, precision and F1 results that we obtained with different optimizers calculated according to the experiments performed Table III. As shown in '. According to the experimental results, it is seen that he is the optimizer Adam with high accuracy values.

TABLE IV. COMPARISON WITH THE MODELS (%)

Authors	**Method**	**Accuracy**	**Sensitivity**	**Specificity**
Ahmad et al. [4]	ResNet152	87,42	97,04	97,23
Kwasigroch et al. [9]	ResNet50	75,5	90	61
Vesal et al. [12]	SkinNet	93	93	90
Al-Masni et al. [13]	Inception-ResNet-v2	81,79	81,80	71,40
Pham et al. [14]	Inception-v3	87	70	91,2
Pham et al. [14]	ResNet50	79,5	80	96,2
Jayalakshmi et al. [15]	BN_CNN	83,05	83,06	83,23
Cabas et al. [8]	CNN	84,76	91,97	78,71
Wu et al. [16]	ResNet50	82	84	85
Our Model	**ResNet18**	**84,96**	**87,09**	**69,57**

In medical applications, the highest sensitivity coefficient is required because a malignant lesion that is misdiagnosed can seriously affect patient health.

The model Table IV, which we proposed with the methods, Accuracy, Sensitivity and Specificity values of previous studies using different deep learning methods. It has been compared in.

V. CONCLUSION

In this study, we investigated a deep convolution network, ResNet18, for classification of skin lesions from RGB images. The investigated model can automatically classify the given lesion image without need of traditional segmentation and feature extraction process. According to the experimental results, ResNet was effective for skin lesion classification. Moreover, Sgdm, Adam and Rmsprop optimizers were also investigated. Performance results were compared with previous research using various deep learning methods and the proposed model was found to be promising. We hope that the model presented in the study can be further developed into real applications where it can help experts for better diagnosis and treatment.

ACKNOWLEDGMENT

I would like to thank all the authors who contributed to the emergence of this study, the International Skin Imaging Collaboration (ISIC), who published the data set we used in the study, and Istanbul Sabahattin Zaim University.

REFERENCES

[1] O. Yıldız, "Melanoma detection from dermoscopy images with deep learning methods: A comprehensive study," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 34, no. 4, pp. 2241–2260, 2019, doi: 10.17341/gazimmfd.435217.

[2] S. Jain, V. Jagtap, and N. Pise, "Computer aided melanoma skin cancer detection using image processing," *Procedia Comput. Sci.*, vol. 48, no. C, pp. 735–740, 2015, doi: 10.1016/j.procs.2015.04.209.

[3] N. Hameed, A. M. Shabut, and M. A. Hossain, "Multi-Class Skin Diseases Classification Using Deep Convolutional Neural Network and Support Vector Machine," *Int. Conf. Software, Knowl. Information, Ind. Manag. Appl. Ski.*, vol. 2018-Decem, pp. 1–7, 2019, doi: 10.1109/SKIMA.2018.8631525.

[4] B. Ahmad, M. Usama, C. M. Huang, K. Hwang, M. S. Hossain, and G. Muhammad, "Discriminative Feature Learning for Skin Disease Classification Using Deep Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 39025–39033, 2020, doi: 10.1109/ACCESS.2020.2975198.

[5] D. de A. Rodrigues, R. F. Ivo, S. C. Satapathy, S. Wang, J. Hemanth, and P. P. R. Filho, "A new approach for classification skin lesion based on transfer learning, deep learning, and IoT system," *Pattern Recognit. Lett.*, vol. 136, pp. 8–15, 2020, doi: 10.1016/j.patrec.2020.05.019.

[6] K. He and J. Sun, "Deep Residual Learning for Image Recognition," 2016, doi: 10.1109/CVPR.2016.90.

[7] ISIC, "Challenge Isic Archive." https://challenge.isic-archive.com/data#2016.

[8] A. P *et al.*, "Deep Convolutional Neural Network for Melanoma Image Classification," no. September, p. 92027, 2020, [Online]. Available: http://repositorio.unan.edu.ni/2986/1/5624.pdf.

[9] A. Kwasigroch, A. Mikołajczyk, and M. Grochowski, "Deep neural networks approach to skin lesions classification - A comparative analysis," *2017 22nd Int. Conf. Methods Model. Autom. Robot. MMAR 2017*, pp. 1069–1074, 2017, doi: 10.1109/MMAR.2017.8046978.

[10] M. A. R. Alif, S. Ahmed, and M. A. Hasan, "Isolated Bangla handwritten character recognition with convolutional neural network," *20th Int. Conf. Comput. Inf. Technol. ICCIT 2017*, vol. 2018-Janua, no. March 2018, pp. 1–6, 2018, doi: 10.1109/ICCITECHN.2017.8281823.

[11] H. Zhou, F. Xie, Z. Jiang, J. Liu, S. Wang, and C. Zhu, "Multi-classification of skin diseases for dermoscopy images using deep learning," *IST 2017 - IEEE Int. Conf. Imaging Syst. Tech. Proc.*, vol. 2018-Janua, pp. 1–5, 2017, doi: 10.1109/IST.2017.8261543.

[12] S. Vesal, N. Ravikumar, and A. Maier, "SkinNet: A deep learning framework for skin lesion segmentation," *arXiv*, pp. 1–3, 2018.

[13] M. A. Al-masni, D. H. Kim, and T. S. Kim, "Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification," *Comput. Methods Programs Biomed.*, vol. 190, p. 105351, 2020, doi: 10.1016/j.cmpb.2020.105351.

[14] T. C. Pham *et al.*, "Improving binary skin cancer classification based on best model selection method combined with optimizing full connected layers of Deep CNN," *2020 Int. Conf. Multimed. Anal. Pattern Recognition, MAPR 2020*, pp. 1–6, 2020, doi: 10.1109/MAPR49794.2020.9237778.

[15] G. S. Jayalakshmi and V. S. Kumar, "Performance analysis of convolutional neural network (CNN) based cancerous skin lesion detection system," *ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc.*, pp. 1–6, 2019, doi: 10.1109/ICCIDS.2019.8862143.

[16] X. Li, J. Wu, E. Z. Chen, H. Jiang, and C. V Feb, "What evidence does deep learning model use to classify Skin Lesions ?"

Gender Classification Using Deep Learning Techniques

Sahra Tilki
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
sahra.tilki@izu.edu.tr

Hasibe Büşra Doğru
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
hasibe.dogru@izu.edu.tr

Alaa Ali Hameed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
alaa.hameed@izu.edu.tr

Akhtar Jamil
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
akhtar.jamil@izu.edu.tr

Jawad Rasheed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
jawad.rasheed@izu.edu.tr

Erdal Alimovski
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-0909-2047

Abstract*— Gender classification from face images is a challenging task due to presence of complex background, object occlusion, and variations in illumination conditions. Face images can be exploited for various applications such as expression analysis, recognition and tracking. In this paper, two deep learning-based methods are investigated for gender classification using face images. These methods include: convolutional neural network (CNN) and Alex Net. Experiments were performed to evaluate the performance of both models for identification of male and female classes from face images. Results show that both methods were effective for gender classification. Moreover, a comparative analysis was also performed between these two models and some of the popular methods for gender classification.

Keywords*— Gender classification, gender recognition, AlexNet, CNN, deep learning.

I. INTRODUCTION

In people living in communities, gender is an important factor in the interaction between individuals. With the development of technology, the use of smart devices has increased and social media has started to attract attention by everyone. However, day-to-day studies on gender recognition have gained importance and number of applications using such techniques have increased [1]. Facial images are widely used in such applications as they provide useful information that can be exploited for extracting human interaction. Gender classification approach by using facial images generally consists of image processing, feature extraction and classification stages. These stages may vary depending on the purpose of the study and the features of the methods to be used. Therefore, the classifier, method and extracted features have a significant effect on the performance of the study.

Deep learning techniques are now widely used for various tasks such as classification, automatic feature extraction, object recognition etc., due to their high classification accuracy. Motivated from other fields, researchers have also exploited deep learning methods for gender prediction and classification from facial images. The following paragraphs present a summary of some studies.

Janahiraman and Subramaniam [2] aimed to make gender classification using different models of CNN architecture. A dataset was created from facial images from Malaysians and some Caucasians people. Their results achieved accuracy of 88% with the VGG-16 model, 85% with the ResNet-50 model and 49% with the Mobile Net model.

Akbulut et al [4] performed gender recognition from facial images using Local Recipient Areas-Excessive Learning Machine (LRA-ELM) and CNN architecture. The experiments were carried out using approximately 11 thousand images from the Adience dataset for age and gender recognition [3]. The proposed method resulted in 80% and 87.13% accuracy with LRA-ELM and CNN respectively.

In [5], a comparative analysis was perfomed among proposed method, Alex Net and VGG-16 models for gender classification including women, men, old, young, children, and babies. Experimental results show accuracy of 72.20% with the proposed CNN model, 99.41% with the VGG-16 model, and 65.63% with the AlexNet model.

Arora et al [6] proposed a CNN model for gender classification with facial images. In experimental studies, 1500 images were used for training and 1000 images were selected from the CASIA database and verified. As a result of the experiments, 98.5% accuracy was achieved.

In [7], a basic convolutional network architecture was proposed to increase the performance of automatic age and gender classification. This technique also produce optimal results even in presence of limited training data. High classification accuracy was obtained using this model on Adience database [3].

Ranjan et al [8] proposed a method called HyperFace for simultaneous facial recognition, pose prediction, and gender recognition using CNNs. This method combines the middle layers of a deep CNN using a separate CNN and then a multitasking learning algorithm was employed on the fused features. HyperFace-ResNet is based on the ResNet-101 model to increase algorithm speed and Fast HyperFace variants to create zone recommendations have been proposed.

Raza et al. proposed a deep learning method for predicting the gender of pedestrians. A preprocessing step was used to segment the pedestrian from the image. Then, stacked auto encoders with softmax classifier were used for classification. Accuracy rates of 82.9%, 81.8% and 82.4% have achieved in the anterior, posterior and mixed views in

the MIT dataset, respectively, and approximately 91.5% in the PETA dataset [9].

In [10], Abdalrady and Aly presented the exchange of classical CNN models with the PCANet model for gender classification. In addition, the network architecture size was also reduced with PCANet in complex CNN models. this method achieved 89.65% accuracy for gender classification.

Yu et al [11] proposed a low-complexity CNN model with few layers. This method achieved 91.5% accuracy on a dataset consisting of 1496 whole body images.

In this study, a gender classification method is proposed using deep learning techniques. Specifically, CNN and AlexNet model will be employed for gender classification from images. Moreover, a comparative analysis will be performed between our method and state-of-the-art methods.

The rest of the paper is organized as follows. Section II provides a detailed account of the proposed method. Section III presents the obtained results. Finally, the paper will be completed with concluding remarks.

II. METHODOLOGY

A. Deep Learning

Deep learning is an artificial intelligence (AI) technique that aims to mimic human brain by learning from experience. In other words, it is a method that realizes the learning process by aiming to discover the hidden representation of the data [12]. These representations are learnt through a training process. For instance, to learn how to recognize an object, it is necessary to train the program with many object images that we label according to different classes. This training can take hours, days or even weeks. Generally, deep learning-based approaches need large amount of training data and take longer time for training compared to the conventional machine learning methods.

When trying to recognize any object or character on an image, finding unique properties is a time-consuming and difficult process as there are so many properties on the object or character. At this point, unlike classical machine learning, in which features are extracted manually, problems can be overcome with deep learning techniques that automatically extract the relevant features from the data. Deep learning is a neural network with many hidden layers, and these hidden layers can be thousands or millions. After an image is trained over the network, they can form complex concepts from simple concepts. When an image is trained in the network, it can learn objects such as characters, faces, cars by combining simple features such as shape, edges and corners. As the image passes through the layers, each layer learns a simple feature before moving on to the next layer one by one, as the layers increase, the network can learn more and more complex features and finally combine them to predict the image.

Deep Learning methods have proven their importance by achieving remarkable success in Natural Language Processing (NLP), Optical Character Recognition (OCR), Computer Vision (CV), Image Processing, Object Recognition and Classification.

B. Convolutional Neural Networks (CNN)

CNN, one of the deep learning techniques, is a powerful neural network. It is widely used as a solution to problems that may be encountered in areas such as Computer Vision and Image Processing. CNN can replace input data with trainable parameters in each layer and also make accurate assumptions about the nature of the images.

CNN architecture consists of five main types of neural layers; convolution layer, activation layer, pooling layer, fully connected layer and dropout. Each layer type plays a different role. Each layer of CNN converts the input volume into an output volume of neuron activation and eventually delivers it to fully connected layers. While simpler features such as edge information are obtained in the first layers, more complex features representing the image are obtained in deep layers. In the following section, the operations performed on CNN layers are explained in detail.

1)Convolutional Layer: This layer forms the basis of CNN. The transformation process is performed by circulating a filter that can have different sizes such as 3 * 3, 2 * 2 on the image. Filters apply a convolution process on the images coming from the previous layer in order to generate the output data and as a result of this process, the activation map is formed. We can explain the resulting activation map as the regions in which each filter has its own properties. During the training, the coefficients of the filters are updated for each learning in the training set, and thus it is determined which regions of the data are important to determine the features. Simple features of the image used such as edges are usually calculated in the first layers of the CNN model [13].

2)Activation Layer: The network has a linear structure due to the mathematical operations performed in the convolution layer. As a result of the application of the activation functions used in the activation layer, the network becomes a nonlinear structure. Thus, faster learning of the network is provided. It is very important to choose activation functions in a neural network architecture. The most commonly used of these are the sigmoid function, which is usually used in classification problems, Softmax, which is a generalization of the Sigmoid function for multiple categories, and the Rectified Linear Units Layer (ReLu), which is preferred as the activation function in most studies.

3)Pooling Layer: The layer used for size reduction in CNN architectures is the pooling layer. Information loss may occur as a result of the size reduction process, but these losses are beneficial for the network. Because the reduction in size provides less computational load for the upcoming layers of the network, it also works against network overfitting. There are two most commonly used types, average pooling and maximum pooling

4)Fully Connected Layer: Neurons in this layer are fully connected to all activations in the previous layer. As a result of these layers, two-dimensional feature maps are transformed into one-dimensional feature vector. The

derived vector can be included in a certain number of categories for classification or used as a feature vector for further processing.

5) DropOut: It is one of the most used networking techniques in deep learning [14]. When training is done using big data in CNN, the network can overfitting. The basic logic based on removing some nodes in the network prevents memorization from occurring.

The ImageNet Large Scale Visual Recognition Competition (ILSVRC) is one of the largest competitions in the field of object recognition. The winning CNN models can be listed as AlexNet, Le Net, ZF Net, VGG-16, GoogLe Net and Microsoft RestNet. These models play an important role in understanding and enhancing deep learning and neural networks. Therefore, it is preferred in many studies.

Le Net: It is the model that gave the first successful result of the competition and was published in 1998. Postal numbers were created in order to read the numbers on bank checks [15].

Alex Net: It is the winning model of the competition held in 2012. Developed by Krizhevsky, Sutskever and Hinton. Successive convolution layers consist of fully connected layers using activation functions such as maximum pooling and Relu and Sigmoid [16].

ZF Net: This model, which is an improved version of the Alex Net architecture, won the competition in 2013. Unlike the Alex Net architecture, 7x7 size filters were used instead of 11x11 filters in the first layer and a 2-step slip amount in the pooling layer. Thanks to these differences, it is ensured that many original pixel information at the input size can be preserved [17].

VGG-16: It is a model developed for better results than the results obtained in previous years. The model, which has a very smooth architecture, was the second in the competition in 2014. It is similar to Alex Net and has many filters [18].

GoogLe Net: It is the model that won the competition in 2014. It has an architecture similar to Le Net, but differently it is the implementation of a new element called the starter module. Modules linked in parallel were used to reduce the probability of overfitting [19].

ResNet: The winning model of 2015 is the first network structure consisting of Residual blocks with 34 layers. It differs from all other architectures due to its design in a deeper structure [20].

III. Experiments and Result

In this study, a gender classification method is proposed using facial images with CNN models. The accuracy obtained by using the CNN model and Alex Net architecture were compared with the results of similar studies conducted in the literature.

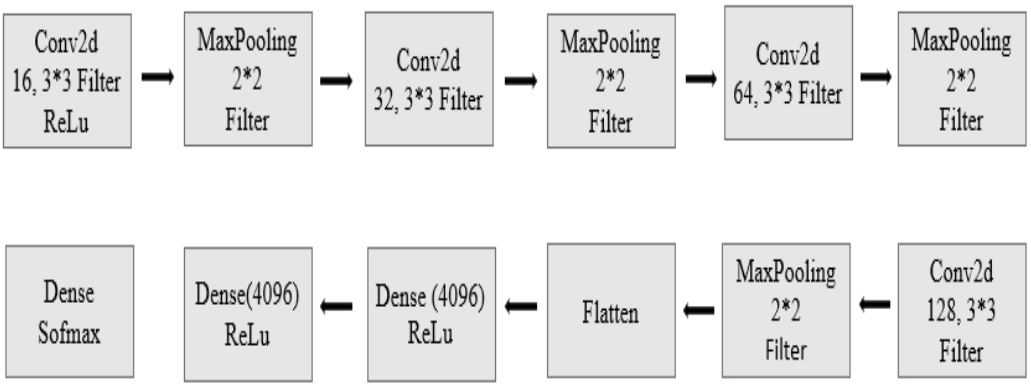


Fig. 1: Layer information of the created CNN model.

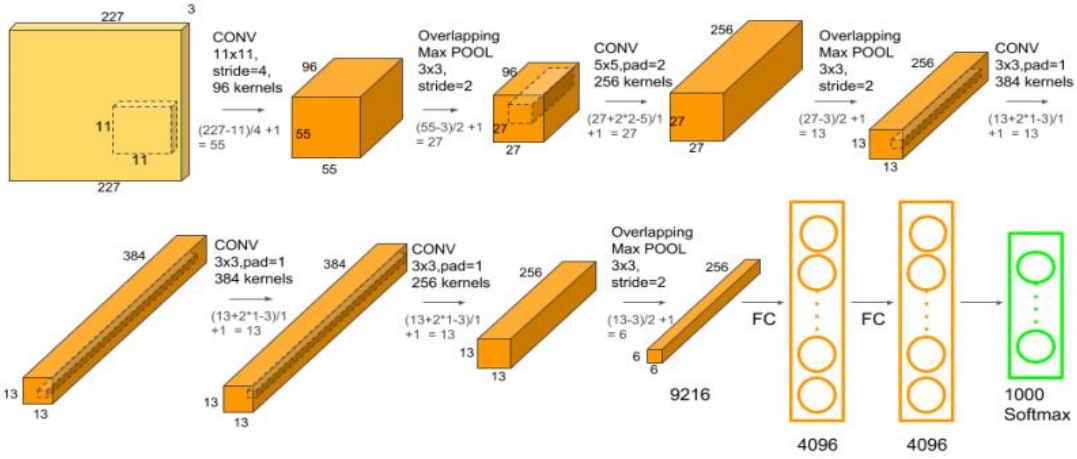


Fig. 2: Alex Net architecture layer information [16].

A. Dataset

Experimental studies have been carried out using a data set consisting of approximately 19,000 images, including female, male and child face images [21]. It is also worth mentioning that the dataset contained images with blurry faces of children and babies, which were not suitable for recognition, as their gender was ambiguous. These images were not included for training and a total of 5000 images were used, including 2500 female and 2500 male face images. Relatively, a smaller dataset has been used as provided in original Adience dataset, which contains 26 thousand face images of approximately 2,284 people obtained by smart phones and were used in previous gender classification studies [3]. Some sample images used in this study are shown in Fig. 3.

Fig. 3: Sample images included in the dataset.

B. CNN Models

In the study, the data set used was trained on two different CNN models which were implemented in Tensorflow and Keras. The first of these is a CNN model consisting of Conv2d, pooling, flatten and dense layers, as well as ReLu and Sigmoid activation functions. Detailed information about the layers of the CNN model created is shown in Fig. 1. The other model was Alex Net. Layer information of Alex Net architecture given in Fig. 2. For both models the dataset was divided into 80% training and 20% testing.

The proposed method was evaluated int terms of accuracy, precision, recall and F1-score. The metrics were obtained using true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Mathematically, these metrics are calculated as:

$$Accuracy = \frac{TN + TP}{TP + TN + FN + FP} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Accuracy: The ratio of correctly predicted data in the model to the total dataset.
Precision: The metric that shows how much of the positively predicted data is actually positive.
Recall: It is the metric that shows how much of the data that should be predicted positively we predicted positively.
F1Score: Harmonic mean of Precision and Recall values. For this reason, it is a good indicator of success.

The overall results obtained for both CNN and AlexNet are summarized in Table I. In terms of F1-score, both models produced same results for classification of males and females. For each class, CNN produced better results (92%) compared to AlexNet (90%). Fig. 4 shows the test and train accuracy obtained for CNN and AlexNet models. Th overall behavior of both models look similar as also indicated by the quantitative results. Similarly, Fig. 5 shows the training and test loss for both models. The visuals indicate that overfitting did not occur in the training, due to the proximity of train and test accuracy values. The loss values express the sum of errors made for each sample in the training and test sets. It starts with high values at the beginning of the training and moves to low values as the model starts learning the data.

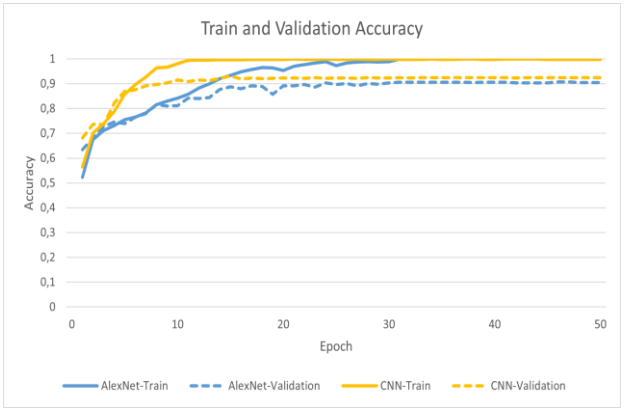


Fig. 4:Accuracy results of CNN and Alex Net models.

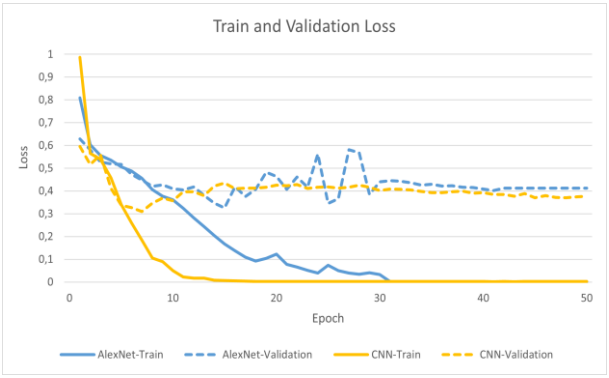


Fig. 5: Loss results of CNN and Alex Net model.

Additional experiments were performed to see the effect of the number of epochs on the validation accuracy of the classifiers. The results obtained for varying number of epochs are summarized in Table II. The results indicate that the test accuracy for both CNN and AlexNet models did not change after a certain number of (probably 20) epochs. However, increasing the number of epochs beyond 20, increase the processing time without any significant increase in the accuracy.

TABLE I. ACCURACY OBTAINED FOR ALEXNET AND CNN (%)

Precision			Recall		F1-Score	
Classes	***AlexNet***	***CNN***	***Alex Net***	***CNN***	***AlexNet***	***CNN***
Male	%89	%91	%92	%93	%90	%92
Female	%92	%92	%89	%91	%90	%92

TABLE II. ACCURACY OBTAINED FOR DIFFERENT EPOCHS (%)

Epoch	***AlexNet***	***CNN***
10	81.10	90.40
20	90.20	92.20
30	89.80	92.30
40	90.50	92.40
50	**90.50**	**92.40**

Finally, we also conducted a comparative analysis of the CNN and AlexNet with some other popular networks for gender classification. The comparative results are presented in Table III. Compared to other methods, both CNN and AlexNet produced relatively higher classification accuracies for the gender classification problem.

TABLE III. COMPARATIVE ANALYSIS OF PROPOSED METHOD WITH OTHER MODELS IN TERMS OF ACCURACY (%)

Previous Works	***CNN***	***AlexNet***
Gündüz and Cedimoğlu [5]	72.20	65.63
Akbulut et al [4]	87.13	-
Levi and Hassner [6]	86.8	-
Yu at al [11]	91.50	-
Proposed	**92.40**	**90.50**

IV. CONCLUSION

In this study, Deep Learning models CNN and Alex Net are proposed for gender classification. Experiments were carried out on a dataset that is less in number than the images found in Adience, which was also used in previous gender recognition and classification studies. Gender classification was aimed by using the specified models together with the images in the dataset. The accuracy rates obtained as a result of experimental studies and the performance of the dataset in models were compared. With these procedures, the accuracy rates achieved were compared with the accuracy rates of similar studies in the literature, and better results were observed. In future studies, it is planned to make comparisons between different CNN models with the dataset containing more images.

REFERENCES

[1] A. Şeker, B. Diri and H. H. Balık, "Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme," *Gazi Mühendislik Bilimleri Dergisi*, *3*(3), 47-64, 2017.

[2] T. V. Janahiraman and P. Subramaniam, "Gender Classification Based on Asian Faces using Deep Learning," In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)* (pp. 84-89), October 2019, IEEE.

[3] E. Eidinger, R. Enbar and T. Hassner, "Age and gender estimation of unfiltered faces", *IEEE Transactions on Information Forensics and Security*, c. 9, sayı 12, ss. 2170– 2179, 2014.

[4] Y. Akbulut, A. Şengür and S. Ekici, "Gender recognition from face images with deep learning," In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-4), September 2017, IEEE.

[5] G. Gündüz and İ. H. Cedimoğlu, "Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini," *Sakarya University Journal of Computer and Information Sciences*, *2*(1), 9-17.

[6] S. Arora and M. P. S. Bhatia, "A Robust Approach for Gender Recognition Using Deep Learning," In *2018 9th International Conference on Computing, Communication and Networking Technologies* (ICCCNT), pp. 1-6, 2018.

[7] G. Levi ve T. Hassner, "Age and gender classification using convolutional neural networks", *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), ss. 34–42, 2015.

[8] R. Ranjan, V. M. Patel and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(1), 121-135, 2017.

[9] M. Raza, M. Sharif, M. Yasmin, M. A. Khan, T. Saba and S. L. Fernandes, "Appearance based pedestrians' gender recognition by employing stacked auto encoders in deep learning," *Future Generation Computer Systems*, *88*, 28-39, 2018.

[10] N. A. Abdalrady, and S. Aly, "Fusion of Multiple Simple Convolutional Neural Networks for Gender Classification," In *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)* (pp. 251-256), 2020 February, IEEE.

[11] Z. Yu, C. Shen and L. Chen "Gender classification of full body images based on the convolutional neural network," In *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)* ,pp. 707-711, IEEE, 2017.

[12] I. Goodfellow, Y. Bengio, A. Courville and Y. Bengio, *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press, 2016.

[13] R. C. Gonzalez, R. E. Woods and S. L. Eddins, " Digital image processing using MATLAB," Pearson Education India, 2004.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, 15(1), 1929-1958, 2014.

[15] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition*," Proceedings of the IEEE 86*(11), 2278–2324, 1998**.**

[16] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, 60(6), 84-90, 2017.

[17] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," In *European conference on computer vision* (pp. 818-833). Springer, Cham, 2014.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition* pp:1-9. IEEE.

[20] K. M. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp: 770-778, 2016.

[21] https://www.kaggle.com/ttungl/adience-benchmark-gender-and-age-classification.

A Comparative Study Of Multi Label Classification With Unlabeled Data

STITINI Oumaima
Computer and Systems Engineering Laboratory
Cadi Ayyad University,FSTG
Marrakech,Morocco
oumaima.stitini@ced.uca.ma

KALOUN Soulaimane
Computer and Systems Engineering Laboratory
Cadi Ayyad University,FSTG
Marrakech,Morocco
so.kaloun@uca.ac.ma

BENCHAREF Omar
Computer and Systems Engineering Laboratory
Cadi Ayyad University,FSTG
Marrakech,Morocco
o.bencharef@uca.ma

***Abstract*—The development of a classifier can be considered one of the foremost basic and complex errands in machine learning and information mining. Conventional classification calculations have a place in the lesson of guided calculations that use knowledge to prepare the classifier as it has been named. Clustering attempts to extract the internal existence of unlabeled data and, due to their similarities, classify data into separate classes. Semi-supervised learning algorithms are the required and efficient technique for machine learning to derive knowledge from both labeled and unlabeled information in order to create effective classifiers. In this work, we compare different approaches the first one about semi-supervised classification approach (Generative methods, Graph-based methods, Margin-based methods, Semi-supervised boosting methods, Self-labeled methods that include two sub-categories self-training and co-training) and the second one is about clustering approach(partitioning methods, hierarchical methods, density-based methods, grid-based methods, model-based methods). Our comparison shows that the problem of multi-classification can be solved by three different ways the first one by binary classification, the second one by using margin- based methods and the last one by boosting methods especially the combination of the clustering method with co-training. All these methods can give good accuracy results, leading to predictive models that are more effective, reliable, and resilient.**

Keywords— ***Unlabeled data, Semi-supervised learning, multiclassification, imbalanced data, clustering***

I. INTRODUCTION

Classification tasks bank intensely on labeled data or tagged information to achieve an efficient classifier. However, because of the technical support from specialist's also as very long time consumption of a manual process for data labeling, it seems troublesome to get enough labeled data to create an efficient classifier. The amount of labelled data is rare, whereas unlabelled data is straightforward.

Because of the extreme data space dimension, unlabeled observations are usually possible in large quantities, whereas labelled samples are uncommonly small. For that, as stated by [1] semi-supervised classification (SSC) is a learning paradigm concerned with improving classification performance by exploiting unlabeled data. Semi-supervised learning (SSL) techniques are considered as an appropriate approach when the amount of labeled data is small and insufficient to build a good classifier and when the number of unlabeled is high.

The aim of this paper is to compare the diverse models used for multi label classification with unlabeled data which are divided into two approach: the semi-supervised learning approach and the clustering approach.

The objective of these comparisons isn't to include new usefulness, however to show the best strategy in examination with these approaches. We organize this paper as follows: Sect. 2 contains the state- of-the art of our work that reviews the relevant approaches to do multi-classification with unlabeled data. We describe the comparison and consolidation of different algorithms in Sect.3. Then in Sect. 4, we will elaborate the discussion of the better algorithms to choose on multi-label classification problem with a sizeable amount of unlabeled data. At the end of the work, we conclude all the work in Sect. 5

II. STATE OF THE ART

A. Semi-supervised approach:

In the absence of labeled data and the adequacy of unlabeled data in the training phase, semi-supervised learning is often the best and most efficient solution. The purpose of proposing a semi-supervised system of learning is to improve the outcomes of learning and resolve the different issues based on data types. For semi-supervised learning, several algorithms have recently been developed such as Self-labeled methods, Semi-supervised boosting methods, Margin-based methods, Graph-based methods and Generative methods.

- Self-labeled methods:

Self-labeled methods can be divided in two sub-categories: Self-training and Co-training.

  - Self-training:

Self-training is the first iterative technique for semi supervised learning, it is among the main models of repetitive strategies for semi-supervised learning. In self-training, with labeled data, a classifier is initially trained. This classifier is then used to assign each of the unlabeled data to a label and apply the most trustworthy unlabeled points together with their anticipated labels to the training set [7].

Traditional self-training strategies focus on heuristics, such as model trust, such as range, which can be costly for manual change. To overcome these challenges,[6] suggest a neural network approach of deep reinforcement learning to dynamically learn the self-training technique and collect the characteristics of training instances.

In self-training, there are two key problems, the first is the determination on which base learner to use for a particular problem and the second is how to locate a collection of unlabeled data high confidence predictions. [4] suggested an algorithm that chooses the best base learner randomly, according to the amount of unlabeled data predictions that are most confident.

[12] Propose a novel and simple approach for classifying semi-supervised texts. First, as a form of model ensemble, the approach produces two sets of classifiers and then initializes the word embeddings differently: one using random, the other using pre-trained word embeddings. The suggested method focuses on various predictions for unlabeled data between the two class-sifiers when observing the self-training.

- Co-training:

Co-training is an algorithm used by machine learning where a limited volume of data is labelled and a multitude of data is unlabeled. Co-training is a method in semi-supervised learning with two viewpoints. Each sample is assumed to be represented using two sets of different characteristics that present different details about the sample.

[1] implements a semi-supervised approach that leverages evidence regarding the co-occurrence of pairwise names. It proposes a solution during the co-training process to resolve the dilemma of class-imbalance and to convey confident labels of multi-label samples.

[9] suggest a new Reinforced Co-Training approach to pick unlabeled high-quality samples in order to properly co-train. More precisely, with a small labelled dataset, the suggested technique uses Q-learning to learn a data collection strategy and then utilizes this policy to automatically train co-training classifiers.

[11] propose an approach to enable co-training for handling multi-label data, two classification models are generated by dichotomizing the feature space with diversity maximization, and then pairwise ranking predictions on unlabeled data is iteratively communicated for model refinement.

- Semi-supervised boosting methods:

Boosting, with many applications, is known as a supervised learning method. The aim of the boost is to minimize marginal costs. For semi-supervised learning, this process has also been developed.

[3] This approach considers semi-supervised learning as a task of clustering and aims to increase the detection rate of clusters using named data as prior information, which in various domains is not necessarily an ideal approach.

- Margin-based methods:

Supervised margin-based strategies are effective classification techniques. In semi-supervised learning, various experiments have been carried out to improve these approaches. The extensions of a support vector machine (SVM) for semi-supervised learning are typically a collection of margin-based techniques.

[14] SVM algorithm supports binary classification. But through the support vector, multi-knowledge based system SMK design proposed it shows four types of classification by using multiple knowledge based system KBS the uses of SVM in multiple knowledge based system handle large amounts of unstructured data and to support multi-class classification.

- Graph-based methods:

Graph-based methods in semi-supervised learning are based on the theory of strings. These methods describe a graph in which nodes (labeled or unlabeled) are samples, and edges represent sample similarities. The label symmetry in the graph is commonly assumed by these techniques.

[15] suggest a new Adaptive Graph Driven Embedding (AG2E) method for semi-supervised multi-label annotation, which uses minimal labeled data associated with large-scale unlabeled data to promote performance of learning.

- Generative methods:

For more precise estimates, this approach employs unlabeled results. Diverse models have been created for semi-supervised learning. The mixed Gaussian distribution, the EM algorithm, the Bayesian distribution, the hidden Markov model.

Cooperative training [16] represent one of the most common routine approaches to semi-supervised learning. Two preferred separate data views are used in Shared Training, all of which are independently appropriate to train a classifier. A classification and a degree of trust for branded data are expected by each classifier. Unlabeled samples that are classified by a highly accurate classifier are used as training data for other samples. Until some of the classifiers shift, this process is repeated. Different learning algorithms should be used in shared testing instead of the different views of data that are uncommon in certain domains. The key problem of repeated strategies is how to pick a high-reliability prediction range. For mutual preparation, which is not always helpful, the agreement between classifiers is chosen and some of the inappropriately labelled examinations are released for subsequent repetitions.

B. Clustering Approach:

The clustering could be used as a means of summarizing the samples' distribution. It is often used before the classification stage to minimize the details. Semi-supervised clustering approaches are classified as clustering methods that can be extended to partly labelled data or data with other kinds of outcome steps (or sometimes as supervised clustering methods).

Several algorithms have recently been developed for semi-supervised clustering such as density-based methods, hierarchical methods, grid-based methods, partitioning methods, and model-based methods.

- partitioning methods:

Based on the attributes and similarities of the data, this clustering approach classifies the information into several classes. It is the data analysts who determine the number of clusters to produce for the methods of clustering.

[17] present the multi-label classification through multi k-means clustering which is utilized for business and user-item reviews.

[8] introduced a system showing good results when there is a very limited number of labeled samples and when there are unlabeled samples available.

- hierarchical methods:

[18] have examined interactive learning applied to hierarchical multi-label classification (HMC).HMC brings new challenges to active learning, as the datasets usually have high number of labels, alongside an underlying hierarchy which defines relationships among the classes.

- density-based methods:

[5] suggests a semi-supervised self-training classification algorithm based on data density peaks and differential evolution.

- grid-based methods:

[19] studies the clustering algorithm based on the combination of grid and density, and proposes the GDStream algorithm.

III. COMPARISON OF EXPERIMENTAL RESULTS

Table 1 shows the comparison of related work discussed on section 2. The table mentions the most important works already done to deal with the problem of multiclassification. The comparison and consolidation of the two main approaches used in multi label classification is shown in Table 2. (see the tables 1 and 2 at the end of the article).

IV. DISCUSSION

Labelled data can significantly help extract patterns correctly in semi-supervised learning. They will also contribute to greater integration by making more impacts on models.

The problem of multi-class classification can be solved in various ways as we already mention on previous section but we can cite the best ways to deal with this problem :

The problem of multi-class grouping can be solved in various ways:

- First suggestion :Starting by binary classification using self training.

Build a binary variable for each class and independently estimate them as a binary classification after averaging the data, but if we have a large number of classes, it is not the best option since it takes good computation time. This multi-class binary classifier can be used with a one-vs-all or all-vs-all reduction technique.

- Second suggestion :Margin based methods.

To solve multiclass problems, we can use algorithms such as Naive Bayes, Neural Networks and SVM.It gives better accuracy than self training.

- Third suggestion :Boosting methods

We can also apply the clustering with co-training algorithm to improve accuracy result for multiclassification.

V. CONCLUSION

Various methods and algorithms for deep learning are being implemented today. The key purpose of schooling, in reality, is to have better outcomes.
Labelled data is very difficult to obtain with respect to machine learning, and unlabeled data is typically readily obtained and accessed. On the other hand, in certain programs of which only such data is labelled, much of the data is unlabeled. Therefore, in addressing much of the challenges, semi-supervised instruction is more realistic. In this work, in order to do multi label classification with unlabeled results, we proposed a comparative analysis of two approaches.

REFERENCES

[1] Xing, Y., Yu, G., Domeniconi, C., Wang, J., & Zhang, Z. (2018). Multi-Label Co-Training. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, {IJCAI-18}, 2882–2888. https://doi.org/10.24963/ijcai.2018/400

[2] Tan, Q., Yu, Y., Yu, G., & Wang, J. (2017). Semi-supervised multi-label classification using incomplete label information. Neurocomputing, 260, 192–202. https://doi.org/https://doi.org/10.1016/j.neucom.2017.04.033

[3] Tanha, J. (2019). A multiclass boosting algorithm to labeled and unlabeled data. International Journal of Machine Learning and Cybernetics, 10(12), 3647–3665. https://doi.org/10.1007/s13042-019-00951-4

[4] Livieris, I., Kanavos, A., Tampakas, V., & Pintelas, P. (2018). An Auto-Adjustable Semi-Supervised Self-Training Algorithm. Algorithms, 11, 139.

[5] Wu, D., Shang, M., Wang, G., & Li, L. (2018). A self-training semi-supervised classification algorithm based on density peaks of data and differential evolution. 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 1–6. https://doi.org/10.1109/ICNSC.2018.8361359

[6] Chen, C., Zhang, Y., & Gao, Y. (2018). Learning How to Self-Learn: Enhancing Self-Training Using Neural Reinforcement Learning. 2018 International Conference on Asian Language Processing (IALP), 25–30. https://doi.org/10.1109/IALP.2018.8629107

[7] Bagherzadeh, J., & Asil, H. (2019). A review of various semi-supervised learning models with a deep learning and memory approach. Iran Journal of Computer Science, 2(2), 65–80. https://doi.org/10.1007/s42044-018-00027-6

[8] Forestier, G., & Wemmert, C. (2016). Semi-supervised learning using multiple clusterings with limited labeled data. Information Sciences, 361–362, 48–65. https://doi.org/https://doi.org/10.1016/j.ins.2016.04.040

[9] Wu, J., Li, L., & Wang, W. Y. (2018). Reinforced Co-Training. ArXiv, abs/1804.06035

[10] Settouti, N., Douibi, K., Bechar, M., Daho, M. E. H., & Saidi, M. (2019). Semi-Supervised learning with Collaborative Bagged Multi-label K-Nearest-Neighbors. Open Computer Science, 9, 226–242.

[11] Zhan, W., & Zhang, M.-L. (2017). Inductive Semi-Supervised Multi-Label Learning with Co-Training. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1305–1314. https://doi.org/10.1145/3097983.3098141

[12] Hwiyeol Jo and Ceyda Cinarel. Delta-training: Simple semi-supervised text classification using pretrained word embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 3456–3461. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1347. URL https://doi.org/10.18653/v1/D19-1347

[13] https://medium.com/analytics-vidhya/a-survey-on-semi-supervised-learning-al gorithms-part-1-ed4eb2ab9501

[14] Ramanathan, T. T., & Sharma, D. (2017). Multiple Classification Using SVM Based Multi Knowledge Based System. Procedia Computer Science, 115, 307–311. https://doi.org/https://doi.org/10.1016/j.procs.2017.09.139

[15] Wang, L., Ding, Z., & Fu, Y. (2018). Adaptive Graph Guided Embedding for Multi-label Annotation. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, {IJCAI-18}, 2798–2804. https://doi.org/10.24963/ijcai.2018/388

[16] Rezende, D.J., Mohamed, S., Danihelka, I., Gregor, K., Wierstra, D.: One-shot generalization in deep generative models. In: Pro- ceedings of 33rd International Conference on Machine Learning (2016).

[17] PREDICTING BUSINESS CATEGORY WITH MULTI-LABEL CLASSIFICATION FROM USER-ITEM REVIEW AND BUSINESS DATA BASED ON K-MEANS.

[18] Nakano, F. K., Cerri, R., & Vens, C. (2020). Active learning for hierarchical multi-label classification. Data Mining and Knowledge Discovery, 34(5), 1496–1530. https://doi.org/10.1007/s10618-020-00704-w

[19] Clustering algorithm based on grid and density for data stream

TABLE I. COMPARISON OF RELATED WORKS:

works	Approach							
	Semi-supervised approach					Clustering Approach		Supervised learning approach
	Co-training	Self-training	Label correlation	Margin based methods	Boosting algorithms	Cluster asumptions	Density based methods	k-nearest neighbor
1	✓							
2			✓					
3					✓	✓		
4		✓						
5		✓					✓	
6		✓						
9	✓							
10	✓							✓
11	✓							
12		✓						
14				✓				

TABLE II. ADVANTAGES AND DISADVANTAGES OF TWO MAJOR APPROACHES:

Approaches	Algorithms	Advantages	Disadvantages
Semi-supervised approach	Co-training	✓ It may be used in various classification systems. ✓ The rate of error is lower than in the self-training process.	❖ It may not be able to separate indices
	Self-training	✓ It is the easiest semi-supervised algorithm for learning. ✓ It is better on binary classification using the vote majority algorithm.	❖ If the error occur on the first iteration it will affect the classifier efficiency. ❖ A lot of details on convergence can not be given.
	Generative models	✓ They will make strong model forecasts that are similar to solutions. ✓ Awareness of data structures or issues may be given	❖ They are not suitable for classification issues. ❖ If there is a limited volume of labeled data, they have challenges matching labeled and unlabeled data. ❖ Unlabeled data will harm the detection of models
	Graph-based methods	✓ It is built on a mathematical structure. ✓ It's going to work better if the graph matches. ✓ It can be extended to graphs targeted.	❖ If the graph doesn't fit, it will generate the worst output. ❖ The production is vulnerable to the structure and edge of the graph.
	Margin based methods	✓ Extremely validated.	❖ The optimization of local optimum can be problematic
Clustering approach	Partitioning methods	✓ Relatively scalable and simple to implement. ✓ If the variables are big, then K-Means is computationally faster most of the time than hierarchical methods of clustering.	❖ Severe effectiveness in high dimensional spaces. ❖ Descriptors of weak clusters. ❖ Predicting the K value is complicated.

			❖ All partitioning methods does not work well with global clusters.
	Hierarchical methods	✓ No need to define the number of cluster in advance. ✓ Good results	❖ Lack of the interpretability regarding the cluster.

Adaptive Neuro Fuzzy Inference System Based Control of a Wind Turbine and Validation of the Real-time Dataset

Meer Abdul Mateen Khan
Center of Research Excellence in Renewable Energy
King Fahd University of Petroleum and Minerals
Dhahran, Kingdom of Saudi Arabia
meerkhan@kfupm.edu.sa (0000-0002-1534-870X)

Mohammed Kamal Hossain
Center of Research Excellence in Renewable Energy
King Fahd University of Petroleum and Minerals
Dhahran, Kingdom of Saudi Arabia
kamalhossain@kfupm.edu.sa (0000-0001-9264-3828)

Md Sarwar M Haque
Dept. of PYP- Computer Science
King Fahd University of Petroleum and Minerals
Dhahran, Kingdom of Saudi Arabia
smhaque@kfupm.edu.sa (0000-0003-4916-0788)

***Abstract*— Permanent magnet synchronous generator (PMSG) based wind power generations has been emerged as a useful technology for wind power harvesting. However, due to the intermittency of the wind speed, there is a need of smart maximum power point tracking (MPPT) controller that estimates the maximum power and tracks reference angular speed at the shaft. This work proposes a novel approach by integrating Adaptive Neuro Fuzzy Inference System (ANFIS)–based wind turbine model with PMSG system. The designed ANFIS model predicted the maximum power at any given wind speed and at the same time generated an optimal reference shaft speed and mechanical torque. The PMSG model was implemented in speed control method where the proportional integral (PI) controller was provided with feedback error signal of the reference shaft speed from ANFIS model and PMSG output rotor speed. The performance of the ANFIS based PMSG model was investigated by subjecting to a range of wind speed from 4 to 13 m/s, where the mechanical power was traced with less than 0.0025% error limit. The PMSG rotor showed prompt response and achieved the reference speed within 10 ms. A test case was conducted by using the field dataset for wind speed recorded at 80 meters height in Hafar Al-Batin, KSA to explore the potential for wind power generation. This approach was found to be robust with less than 0.27% error for a real time data, hence both the mechanical power and reference angular PMSG speed were effectively tacked, and its accuracy was validated theoretically. Such intelligent control systems will facilitate planning on the smooth and sustainable transformation of renewable energy to total energy mix of the Kingdom as per "Vision 2030".**

Keywords*— *Wind turbine, Maximum Power Point (MPP), Adaptive Neuro Fuzzy Inference System, PMSG, Validation

I. INTRODUCTION

One of the abundant RE resource available in GCC region is wind energy [1-3]. The concept of extraction of wind power started in 19th century and had grown immensely to generate several GW of electricity today. Over this period the technology has improved significantly in terms of efficiency, reliability, cost, and performance [4-7]. The fixed speed or variable speed wind turbine are widely available configurations in wind turbine. As the wind is naturally but the intermittent available source, it is extremely important to have a stable generation from the wind turbine when it comes to integration with grid or feeding the energy demand. The wind turbine operates in a specific range of wind speeds bounded cut-in and cut-out speed. The system shutdowns for any other wind speed to protect the generator and the turbine. Power generation by the wind turbine is affected by the wind speed, the turbine rotor size, blade swept area, tip speed ratio, and rotor speed. This leaves a challenge in estimating the maximum power attained by the turbine [8-9]. Many kinds of variable speed wind turbine generator were employed for wind turbines. Doubly fed induction generators (DFIGs) and permanent magnet synchronous generators (PMSG) had emerged as a useful technology [10]. Variable speed wind turbine generators fed by PMSG are preferred due to its better efficiency and good power quality [9]. There are many control strategies for output maximization of a PMSG-based small-scale wind turbine [11]. Different controls were reported in literature, such as, maximum power point tracking (MPPT) controller, pitch angle controller, grid side inverter controller, etc. [10,12]. As wind power harvesting is increasing all over the globe, many control strategies for MPPT had evolved [13]. Many techniques were reported in literature for MPPT algorithms, to name a few are optimal torque (OT) control, power signal feedback (PSF) control, tip speed ratio (TSR) control, etc. [14-15]. TSR is simple, where the optimum tip speed ratio is determined by achieving the maximum power coefficient. However, this method is not suitable for precise measurements and increases the cost of the system. OT and PSF controls are simple and fast, with low efficiency than TSR control method. The hill climb method is also known as P&O, a widely used MPPT technique for wind power estimation as it does not need field test. The algorithm is also independent of turbine characteristics. However, because of the slow response of the wind energy conversion system (WECS) caused by large inertia, it is not suitable for MPPT control [16]. Abo-Khalil and Lee [17] proposed MPPT control of wind energy system by estimating the wind speed based on support vector regression. The technique found to be effective with less than 3.3% error. Han and co-workers demonstrated a method of speed control for wind turbine PMSG driven by a DC motor [18]. However, MPP was estimated only for small range of wind speed between 5-8 m/s. Intelligent techniques are developed with higher accuracy like fuzzy logic and neural networks (NN), that can estimate MPPT in WECS. It was reported that fuzzy logic enhanced the performance of the parameters, although it is more expensive and not that accurate in estimating wind speed [19]. On the other hand, artificial neural network (ANN) was employed to estimate wind speed and power efficiently [19].

This work introduces a novel intelligent ANFIS-based approach for tracking maximum power in wind energy system with PMSG. The key contribution of this research is the implementation of speed control method for PMSG system along with the developed ANFIS-based wind turbine model that achieves maximum mechanical power from the wind turbine and generates a reference angular shaft speed for driving the PMSG rotor at the same time. The PMSG system operates in the speed control loop where the PI controller is fed with the error signal from the difference of the reference speed generated by ANFIS wind turbine model and the PMSG output rotor speed. The performance of the control system has been investigated for various wind speed ranging from 4 to 13 m/s and were found to be very effective in estimating the mechanical power, mechanical torque and angular shaft speed and thereby following variation in wind speed. The mechanical power was achieved with an error as low as 0.0025%. Moreover, fast response of the designed control approach was observed and the PMSG rotor achieved the reference speed in less than 10ms. A case study for a real time dataset for wind speed recorded in the Eastern province (i.e. Hafar Al-Batin, 28.268806° N, 44.203111° E) of KSA was performed to determine the potential for wind energy harvesting. The proposed ANFIS-based control system for wind turbine was found to be robust for a real time field wind data with an acceptable error limit that was less than 0.27%. To the extent, the angular speed and the mechanical power generated from the ANFIS -based wind turbine model (that drives the PMSG rotor) effectively tracked the theoretically estimated angular speed at maximum power. Such ANFIS-based strategy for tracking maximum power in wind energy system has been expected to be very potential in wind energy harvesting particularly in KSA regions.

II. Key Elements and Terminologies

A. Wind energy conversion system

It is well-known that in wind energy harvesting, kinetic energy is converted to useful power. However, the mechanical power of turbine is non-linear in nature due to intermittence wind speed as shown in Fig. 1. The dotted red line therein shows the optimal power line for each individual scenario. This maximum power varies with the operating speed of the rotor. As the rotor speed is sensitive to wind speed, this maximum power point keeps changing [20-21]. Therefore, the MPPT controller is a crucial element to track maximum power from any wind speed connected to such wind energy conversion system.

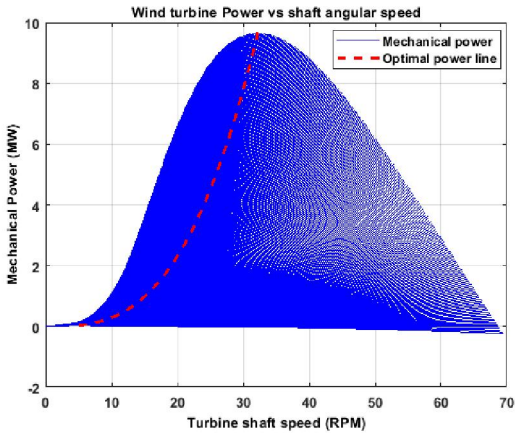


Fig. 1. Relationship between turbine mechanical power and speed at different operating wind speed.

B. ANFIS for wind turbine

In ANFIS, fuzzy rules and membership function are used and hence ANFIS-based controllers convert heuristic and linguistic rules into numbers. Therefore, advantages of neural network and fuzzy logic are achieved in ANFIS-based system. ANFIS-based MPPT controller has been known to

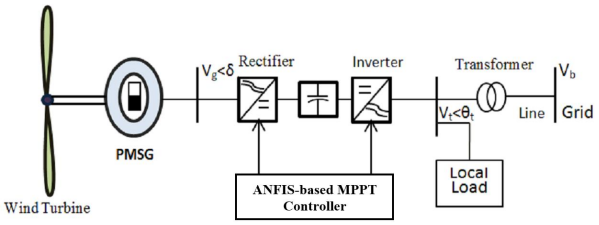


Fig. 2. PMSG-based WECS configuration supported by ANFIS-based MPPT controller.

be efficient and robust [22-24]. ANFIS system is known as a Sugeno type fuzzy model. The model works in such a way that within the framework of an adaptive system, it develops adaption and capability to learning itself. Therefore, the model becomes matured and efficient in time within the box.

C. PMSG model for wind turbine

The fundamental of PMSG has been depicted in Fig. 2 as well. As the turbine shaft is connected to PMSG rotor shaft by means of a gearbox, therefore the torque generated at the shaft near the PMSG rotor need to be specific at specific operating speed [25-27]. The PMSG is driven by the gear shaft, as the entire dynamic model is implemented in dq-frame. As was demonstrated in ref. [28], the PMSG synchronous electrical model can be derives as per Eq. (1), Eq. (2) and Eq. (3).

$$\frac{di_{sd}}{dt} = -\frac{R_{sa}}{L_{sd}} i_{sd} + \omega_s \frac{L_{sq}}{L_{sd}} i_{sq} + \frac{1}{L_{sd}} V_{sd} \quad (1)$$

$$\frac{di_{sq}}{dt} = -\frac{R_{sa}}{L_{sq}} i_{sq} - \omega_s \left(\frac{L_{sd}}{L_{sq}} i_{sd} + \frac{1}{L_{sq}} \psi_p\right) + \frac{1}{L_{sq}} V_{sq} \quad (2)$$

$$T_e = \frac{3}{2} * \frac{P}{2} \left[\psi_p i_{sq} + i_{sd} i_{sq} (L_{sd} - L_{sq})\right] \quad (3)$$

Where V_{sd}, V_{sq}, I_{sd} and I_{sq} are the d-q axis stator voltages and currents, respectively. L_{sd} and L_{sq} are the inductances of the generator. P is the number of poles, ψ_p is the permanent flux, stator resistance is represented as R_{sa} and ω_s is the generator's electrical angular frequency. T_e is the electromagnetic torque. The PMSG is connected to current control pulse width modulation (PWM) inverter.

D. MPPT for wind turbine

Maximum power at any given wind speed will be achieved when the maximum power coefficient is achieved at the optimum tip speed ratio. Maximum power coefficient (C_{pmax}) will be achieved at optimum tip speed ratio (λ_{opt}) and the blade pitch angle (β) is set to zero [12]. The maximum power coefficient (C_{pmax}) is achieved when the blade pith angle is at zero and tip speed ratio is at optimal point (λ_{opt}) [12]. Fig.

3 shows ANFIS-based MPPT control of such PMSG-based wind turbine system.

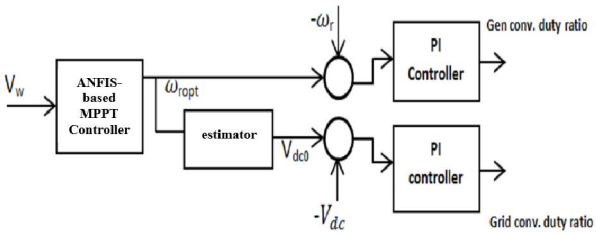


Fig. 3. ANFIS-based MPPT controller for PMSG-based wind system.

Therefore, the maximum power (P_{max}) at any given wind speed is defined in Eq. (4) and the optimum turbine speed (ω_{opt}) at which the maximum power is achieved is also can be defined as Eq. (5).

$$P_{max} = \frac{1}{2}\rho A V_w^3 \, C_{pmax}(\lambda, \beta) \quad (4)$$

$$\omega_{opt} = \frac{\lambda_{opt} V_w}{R} \quad (5)$$

III. Results and Discussion

Based on the control strategy implemented as described in the previous section, simulation was performed to investigate the performance of ANFIS-based PMSG controller for the wind energy system The ANFIS- based MPPT methods was tested under varying input wind speed. This method successfully estimated the maximum power for each sample of wind speed. Fig. 4 and Fig. 5 show ANFIS-based estimated mechanical power that followed exactly the similar pattern of actual calculated mechanical power being affected with respect to wind speed. The error between the ANFIS-based estimated mechanical power and the actual power is less than 0.0025 %.

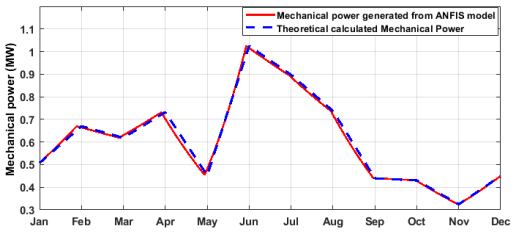


Fig. 4. Mechanical power generated from ANFIS-based Wind turbine model and theoretically calculated mechanical power under real input wind speed for Hafar Al-Batin.

To study and confirm the robustness of the proposed control system, the designed model was subjected to real time dataset of Eastern province (i.e. Hafar Al-Batin, 28.268806° N, 44.203111° E), KSA. Monthly averaged wind speed for a year recorded at 80 meters height was used to quantify the performance of the designed control system under real conditions. The dataset was used as obtained from renewable resource atlas, King Abdullah city for Atomic and Renewable Energy (K.A.CARE). Fig. 4 shows the distribution of monthly averaged wind speed for twelve months in a year at 80 meters height in Hafar Al-Batin. The proposed ANFIS-based control for wind turbine model was simulated with the monthly averaged real wind speed data of twelve (12) months. It was observed that the mechanical power of the ANFIS-based wind turbine model followed and tracked effectively the theoretical and estimated maximum power as shown in Fig. 4 and Fig. 5.

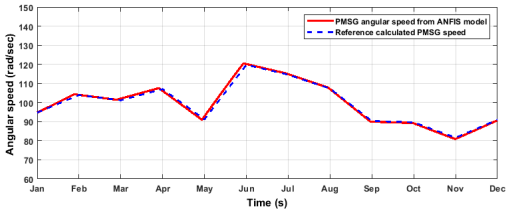


Fig. 5. PMSG angular speed generated from designed ANFIS model tracking the reference PMSG speed to generate maximum power at Hafar Al-Batin.

The theoretical maximum power was calculated by using Eq. (4) and Eq. (5), whereas the optimum tip speed ratio was selected as 6.5 ($\lambda_{opt} = 6.5$) and pitch angle ($\beta = 0$) and maximum power coefficient (C_{pmax}) was calculated to be 0.4818. The table in appendix shows wind turbine parameters used in calculating the maximum power generated from ANFIS model and theoretical and estimated power at given wind speed data of Hafar Al-Batin. Further details will be demonstrated in presentation as well as in revised manuscript.

Table 1: Maximum power generated from ANFIS model and Theoretical calculated power at given wind speed of Hafar Al-Batin.

Month	Wind speed (m/s)	Theoretical calculated Maximum power (MW)	Mechanical power generated from designed ANFIS model (MW)	Error (%)
Jan	7.30	0.5071	0.5084	0.26
Feb	8.01	0.6699	0.6699	0.00
Mar	7.80	0.6186	0.6192	0.10
Apr	8.25	0.7320	0.7312	0.11
May	7.03	0.4529	0.4538	0.20
Jun	9.23	1.0251	1.0249	0.02
Jul	8.83	0.8975	0.8962	0.14
Aug	8.27	0.7373	0.7365	0.11
Sep	6.96	0.4395	0.4403	0.18
Oct	6.91	0.4301	0.4307	0.14
Nov	6.29	0.3244	0.3234	0.31
Dec	7.01	0.4490	0.4499	0.20

IV. Conclusion

This paper presents an intelligent control strategy for wind turbine integrated with PMSG system. The ANFIS-based

wind turbine model was designed and integrated with PMSG system. The ANFIS-based model was trained for a wide range of wind speed ranging from 3 to 19.4 m/s. The model was found to track successfully the maximum power and generate reference shaft angular speed to drive the rotor of PMSG at any given input windspeed. This reference speed drove the PMSG rotor that operated in speed control loop. The PI controller was fed back with the error difference of the PMSG output rotor speed and the reference shaft speed generated from ANFIS model. The approach was tested with fluctuating random wind speed and the mechanical torque, angular speed and power was observed to follow the pattern of the wind sped variations with an error limit of 0.0025%. The robustness of the proposed model was also investigated for a city as a test case in the Eastern province (Hafar Al-Batin, 28.268806° N, 44.203111° E) of KSA with a real time wind speed dataset recoded at 80 meters height. The maximum power and the reference speed were effectively tracked by the ANFIS-based wind turbine PMSG model and the accuracy was validated theoretically with an acceptable error as low as 0.27% and the approach depicts the potential of wind power generation in KSA region. the method was found to be robust as the controllers effectively track the variation in wind speed and the PMSG rotor operates at the reference speed generated by the controller. Such intelligent control systems will facilitate planning on the smooth and sustainable transformation of renewable energy to total energy mix of the Kingdom.

ACKNOWLEDGMENT

Authors thank Center of research excellence in renewable energy (CoRERE), King Fahd University of Petroleum and Minerals (KFUPM), KSA. MKH acknowledges King Abdullah City for Atomic and Renewable Energy (KACARE) for funding support through project KACARE182-RFP-07. KACARE is acknowledged for providing actual onsite dataset of KSA.

APENDIX

Wind Turbine Parameters	
Wind speed range (V_w)	3 to 19.4 m/s
Blade pitch angle (β)	0
Tip speed ratio (λ)	0.1~14
Radius of the wind turbine rotor (R)	37.5 m
Air density (ρ)	1.225 kg/m^3
Maximum power coefficient (C_{pmax})	0.4818
Optimum tip speed ratio (λ_{opt})	6.5
Maximum power (P_{max}) @ 19.4 m/s	9.5 MW
Gear ratio	75
PMSG parameters	
Stator phase resistance (R_{sa})	2.875 Ω
Armature inductance (H)	0.00153
Simulation Parameters	
Operating wind speed	4-13 m/s
Sampling time	2 μs.

REFERENCES

[1] Rehman, S., Baseer, M. A., Meyer, J. P., Alam, M. M., Alhems, L. M., Lashin, A., & Al Arifi, N. (2016). Suitability of utilizing small horizontal axis wind turbines for off grid loads in eastern region of Saudi Arabia. Energy Exploration & Exploitation, 34(3), 449-467.

[2] Baseer, M. A., Meyer, J. P., Rehman, S., & Alam, M. M. (2017). Wind power characteristics of seven data collection sites in Jubail, Saudi Arabia using Weibull parameters. Renewable Energy, 102, 35-49.

[3] Baseer, M. A., Meyer, J. P., Alam, M. M., & Rehman, S. (2015). Wind speed and power characteristics for Jubail industrial city, Saudi Arabia. Renewable and Sustainable Energy Reviews, 52, 1193-1204.

[4] Khan, M. A., Rehman, S., & Al-Sulaiman, F. A. (2018). A hybrid renewable energy system as a potential energy source for water desalination using reverse osmosis: A review. Renewable and Sustainable Energy Reviews, 97, 456-477.

[5] Gupta, R. A., Singh, B., & Jain, B. B. (2015, March). Wind energy conversion system using PMSG. In 2015 International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE) (pp. 199-203). IEEE.

[6] Rehman, S., Alam, M., & Alhems, L. M. (2018). A review of wind-turbine structural stability, failure and alleviation. Advances in Civil, Environmental, & Materials Research (ACEM18), 27-31.

[7] Hossain, M. M., & Ali, M. H. (2015). Future research directions for the wind turbine generator system. Renewable and Sustainable energy reviews, 49, 481-489.

[8] Rehman, S., & Khan, S. A. (2017, July). Application of Fuzzy Goal Programming to Wind Turbine Selection with Multiple Criteria—A Study of Three Potential Sites in Saudi Arabia. In 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) (pp. 853-857). IEEE.

[9] Baseer, M. A. (2017). Wind resource assessment and GIS-based site selection methodology for efficient wind power deployment (Doctoral dissertation, University of Pretoria).

[10] Yin, M., Li, G., Zhou, M., & Zhao, C. (2007, June). Modeling of the wind turbine with a permanent magnet synchronous generator for integration. In 2007 IEEE Power Engineering Society General Meeting (pp. 1-6). IEEE.

[11] Haque, M. E., Negnevitsky, M., & Muttaqi, K. M. (2008, October). A novel control strategy for a variable speed wind turbine with a permanent magnet synchronous generator. In 2008 IEEE industry applications society annual meeting (pp. 1-8). IEEE.

[12] Wang, C. N., Lin, W. C., & Le, X. K. (2014). Modelling of a PMSG wind turbine with autonomous control. Mathematical Problems in Engineering, 2014.

[13] Tan, K., & Islam, S. (2004). Optimum control strategies in energy conversion of PMSG wind turbine system without mechanical sensors. IEEE transactions on energy conversion, 19(2), 392-399.

[14] Kumar, D., & Chatterjee, K. (2016). A review of conventional and advanced MPPT algorithms for wind energy systems. Renewable and sustainable energy reviews, 55, 957-970.

[15] Abdullah, M. A., Yatim, A. H. M., Tan, C. W., & Saidur, R. (2012). A review of maximum power point tracking algorithms for wind energy systems. Renewable and sustainable energy reviews, 16(5), 3220-3227.

[16] Wei, C., Zhang, Z., Qiao, W., & Qu, L. (2016). An adaptive network-based reinforcement learning method for MPPT control of PMSG wind energy conversion systems. IEEE Transactions on Power Electronics, 31(11), 7837-7848.

[17] Abo-Khalil, A. G., & Lee, D. C. (2008). MPPT control of wind generation systems based on estimated wind speed using SVR. IEEE transactions on Industrial Electronics, 55(3), 1489-1490.

[18] Han, K., & Chen, G. Z. (2009, May). A novel control strategy of wind turbine MPPT implementation for direct-drive PMSG wind generation imitation platform. In 2009 IEEE 6th International Power Electronics and Motion Control Conference (pp. 2255-2259). IEEE.

[19] Soetedjo, A., Lomi, A., & Mulayanto, W. P. (2011, July). Modeling of wind energy system with MPPT control. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics (pp. 1-6). IEEE.

[20] Saidi, Y., Mezouar, A., Miloud, Y., Kerrouche, K. D. E., Brahmi, B., & Benmahdjoub, M. A. (2020). Advanced non-linear backstepping control design for variable speed wind turbine power maximization based on tip-speed-ratio approach during partial load operation. International Journal of Dynamics and Control, 8(2), 615-628.

[21] Makhad, M., Zazi, M., Loulijat, A., & Simon, A. O. (2020, April). Robust Integral Backstepping control for Optimal Power Extraction of a PMSG-based Variable Speed Wind Turbines. In 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET) (pp. 1-6). IEEE.

[22] Amara, K., Fekik, A., Hocine, D., Bakir, M. L., Bourennane, E. B., Malek, T. A., & Malek, A. (2018, October). Improved performance of a PV solar panel with adaptive neuro fuzzy inference system ANFIS based MPPT. In 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA) (pp. 1098-1101). IEEE.

[23] Naidu, R. P. K., & Meikandasivam, S. (2020). Performance investigation of grid integrated photovoltaic/wind energy systems using ANFIS based hybrid MPPT controller. Journal of Ambient Intelligence and Humanized Computing, 1-13.

[24] Bin-Halabi, A., Abdennour, A., & Mashaly, H. (2014). An accurate ANFIS-based MPPT for solar PV system. International Journal of Advanced Computer Research, 4(2), 588.

[25] Chen, J., Yao, W., Zhang, C. K., Ren, Y., & Jiang, L. (2019). Design of robust MPPT controller for grid-connected PMSG-Based wind turbine via perturbation observation based nonlinear adaptive control. Renewable energy, 134, 478-495.

[26] Ali, R. B., Schulte, H., & Mami, A. (2017, May). Modeling and simulation of a small wind turbine system based on PMSG generator. In 2017 Evolving and Adaptive Intelligent Systems (EAIS) (pp. 1-6). IEEE.

[27] Tiwari, R., & Babu, N. R. (2017). Comparative analysis of pitch angle controller strategies for PMSG based wind energy conversion system. Int. J. Intell. Syst. Appl, 9(5), 62-73.

[28] Kim, Y. S., Chung, I. Y., & Moon, S. I. (2015). Tuning of the PI controller parameters of a PMSG wind turbine to improve control performance under various wind speeds. Energies, 8(2), 1406-1425.

Determining of Alzheimer from DNA Sequences with One-Dimensional Capsule Networks

Suat Toraman
School of Aviation
Firat University
Elazığ, Turkey
storaman@firat.edu.tr

Bihter Daş
Software Engineering
Firat University
Elazığ, Turkey
bihterdas@gmail.com

***Abstract*— Advances in genomics in recent years have greatly increased our knowledge of the genetic structure of the Alzheimer's disease. The identification of genetic variants affect Alzheimer is quite importance for early diagnosis of this disease. In this study, a method for the recognition of Alzheimer's disease from DNA sequences using capsule networks is proposed. In the proposed approach, DNA sequences are digitized by Entropy-based and EIIP techniques, and then the obtained digital signals are divided into 100-unit sections by sliding window method. Signal fragments belonging to Alzheimer's and healthy individuals were classified by capsule networks. We have achieved classification accuracy rate of 97.47% using Capsule Networks from the data that was digitized by the entropy-based technique.**

Keywords—Alzheimer, DNA sequence, deep learning, capsule networks

I. INTRODUCTION

Alzheimer's disease (AD) is a complex neurological disease caused by the interaction between genes. Although there is no new disease, the number of patients is increasing day by day [1]. In this disease, which is caused by genetic disorders, a severe impairment occurs in memory, learning, speaking, understanding, and reasoning, hence it can negatively affect our daily life [2,3]. AD usually manifests itself initially with deficiencies in short-term memory formation, spatial orientation, problems in additional cognitive functions such as word-finding, reasoning, and problem-solving [4,5]. Early diagnosis is very important to understand the genetic structure of Alzheimer's disease, to find the mechanism of occurrence of this disease, and to determine new drug targets. Wei et al have found that the APP gene is a marker in the diagnosis of Alzheimer's disease, both in the blood and in the brain [6]. Raj et al. have created a database that consists of a comprehensive list of Alzheimer's genes, their associated classes, positive and negative states. The data set called Gold can be used in machine learning algorithms [7]. Chihyun et al. present a new prediction model based on deep learning that predicts Alzheimer's disease. They used 5-fold cross-validation for the performance of the system and achieved an average training accuracy of 88.6% [8]. Rangaswamy et al have developed a machine learning-based model for the prediction of Alzheimer's-associated variants. They achieved 81% and 0.89 AUROC accuracy with 10-fold cross-validation [9]. Shao et al. present an integrated model for the diagnosis of Alzheimer's disease from multidimensional genomic data. They have identified genomic markers for AD disease [10]. Bringas et al. propose an approach that determines the stage of Alzheimer's disease by developing a model based on deep learning. They achieved an accuracy of 90.91% and an F1 score of 0.897 using the proposed convolutional neural networks [11]. In this study, an experiment has conducted to diagnose Alzheimer's disease from DNA sequences using capsule networks. For this study, analog DNA sequences have digitized with Entropy-based and EIIP techniques, and the obtained digital signals were divided into 100 units by the sliding window method. These separated signal fragments and classified as Alzheimer's patients and healthy individuals by capsule networks. A performance of 97.47% has been obtained as a result of the classification by capsule networks from DNA signals digitized with the Entropy-based technique.

II. MATERIAL AND METHOD

In this section, information about the data set, data preprocessing and capsule networks used in the experimental study for the detection of Alzheimer's is presented.

A. Dataset

For this application, the fasta files with accession: AF293341AF293341.1 source: Homosapiens for Alzheimer data group and accession: NM_001300741source: Homosapiens for the healthy data group were used [12].

B. Numerical Mapping Techniques

DNA analog signals must be converted into digital signals to be used in signal processing, image processing or classification applications. There are many numerical mapping techniques in the literature that can be used in this conversion process. In this study, EIIP (Electron-Ion Interaction Pseudo Potential) and Entropy based digital mapping technique are used.

1) Entropy-based Numerical Technique

It has proven that the entropy is quite useful for providing key insights in genetic evolution [13,14]. Shannon entropy gives very good results in measuring regular and irregular DNA sequences. [15]. In this technique, fractional Shannon equation, which is a fractional derivative of Shannon entropy, is used [16]. The formulas used in digitization are shown in (1) and (2) [17,18].

$$Sf = -\sum_i [(-p(x_i))^{\alpha_i} p(x_i) \log(p(x_i))] \quad (1)$$

$$\alpha_i = \frac{1}{\log(p(x_i))} \quad (2)$$

The $p(x_i)$ value shown in Equation 1 represents the repetition frequency of codons (AGG, TAC, TTC, CCG…) in a DNA sequence. Instead of a constant value, the α_i value is calculated adaptively directly from the genome sequence data.

Here, it is aimed to establish a linear relationship between $p(x_i)$ and α_i.

2) EIIP Numerical Technique

This technique is one of the techniques frequently used in the literature. Each base in the DNA sequence is matched to the half-valence number of EIIP. The bases are given values of A = 0.1260, T = 0.1335, G = 0.0806, C = 0.1340, respectively [19,20].

C. Capsule Networks

CNNs are very successful in image / object classification. One of the shortcomings of CNNs is that they cannot reveal the relationship between parts of the object in the image. Capsule networks have been proposed to overcome this deficiency [21,22]. Capsule networks are structures made up of many neurons. CNN's output is a scalar value. The output of the capsule networks is vectorial. CNNs use a scalar input activation function such as ReLU, Sigmoid, and a vector activation function called squashing in capsule networks (Eq. 3) [21,22].

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (3)$$

In the equation, *sj* represents the capsule input and *vj* represents the capsule's output vector [21,23]. In capsule networks, the margin loss in (4) is used.

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1-T_k) \max(0, \|v_k\| - m^-)^2 \quad (4)$$

If there is no k class here, $T_k = 0$, if there is $T_k = 1$ and $m^+ = 0.9$ and $m^- = 0.1$. The length of the calculated vectors shows the probability that the object in the image will be found in that part. [22,23]. The capsule network structure used in the study is shown in Fig. 1.

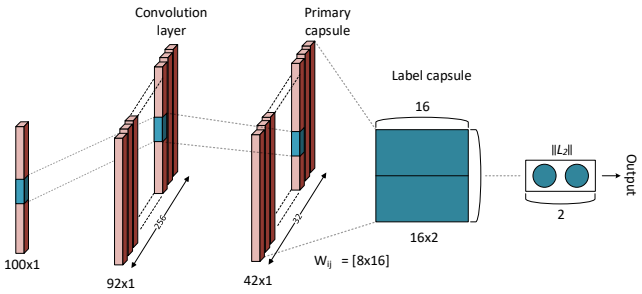


Fig. 1. Capsule network structure for classifying of DNA signals

III. EXPERIMENTAL RESULTS

In the study, DNA gene sequencing data of 1818 bases long Alzheimer's and healthy individuals were used. DNA gene sequences belonging to both groups were digitized by EIIP and Entropy based digital mapping method. Digitized gene sequences were divided into 100 units by sliding window method. 1718 signal fragments were obtained for both groups. The pieces obtained were used to feed the capsule net. In addition, k-fold cross validation method was used to accurately evaluate the performance of the system. The k value was chosen as 10. The results of the classification processes performed with capsule networks are shown in Fig. 2.

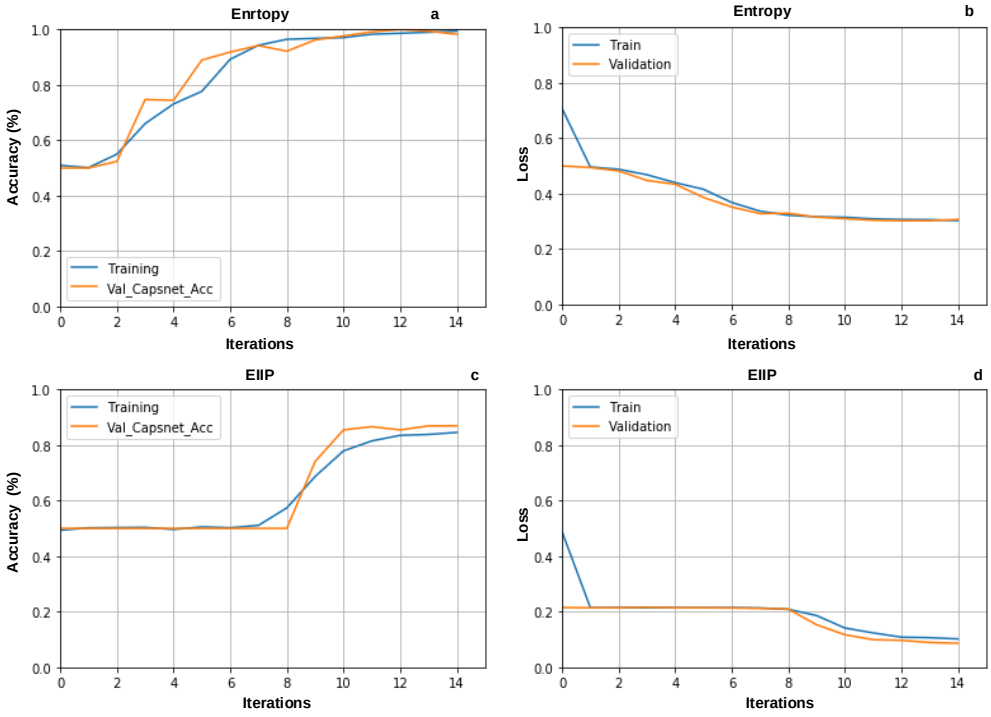


Fig. 2. Accuracy and loss graphs obtained from entropy-based (a, b) and EIIP digital mapping (c, d)

In Table 1 and Table 2, the accuracy, precision, sensitivity, specificity and F1-score values and the averages of these values obtained by 10-fold cross-validation of classification of Alzheimer's and healthy individuals with capsule networks are given.

TABLE I. CLASSIFICATION RESULTS OF ALZHEIMER'S DATA DIGITIZED BY EIIP

Fold	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1-score
1	81.98	74.34	97.67	66.28	84.42
2	85.17	80.40	93.02	77.33	86.25
3	81.40	74.11	96.51	66.28	83.84
4	84.30	87.82	79.65	88.95	83.54
5	88.66	85.19	93.60	83.72	89.20
6	86.92	85.08	89.53	84.30	87.25
7	85.17	81.68	90.70	79.65	85.95
8	81.40	84.18	77.33	85.47	80.61
9	79.24	93.86	62.57	95.91	75.09
10	84.50	78.10	95.91	73.10	86.09
Mean ± SD	83.87±2.71	82.47±5.79	87.65±10.60	80.10±9.09	84.22±3.76

The best classification accuracy was achieved by using entropy based numerical mapping method. While the data digitized with the entropy-based technique was classified with 97.47% accuracy in Capsule Networks, the accuracy remained at 83.87% when the EIIP-based technique was used. According to the results, entropy based digital mapping technique provides a better classification result than EIIP.

TABLE II. CLASSIFICATION RESULTS OF ALZHEIMER'S DATA DIGITIZED WITH ENTROPY TECHNIQUE

Fold	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1-score
1	98.55	97.18	100.00	97.09	98.57
2	97.67	100.00	95.35	100.00	97.62
3	99.42	100.00	98.84	100.00	99.42
4	99.42	99.42	99.42	99.42	99.42
5	98.26	96.63	100.00	96.51	98.29
6	91.57	85.57	100.00	83.14	92.23
7	98.26	96.63	100.00	96.51	98.29
8	98.55	100.00	97.09	100.00	98.53
9	100.00	100.00	100.00	100.00	100.00
10	92.98	87.69	100.00	85.96	93.44
Mean ± SD	97.47±2.69	96.31±5.04	99.07±1.52	95.86±5.85	97.58±2.47

This is because the entropy-based digital mapping technique deepens the rates of discrimination between different species. In addition, this technique reflects the complex structure of DNA sequences much better. Confusion matrix is given in Fig. 3 to see the statistical success of the proposed method.

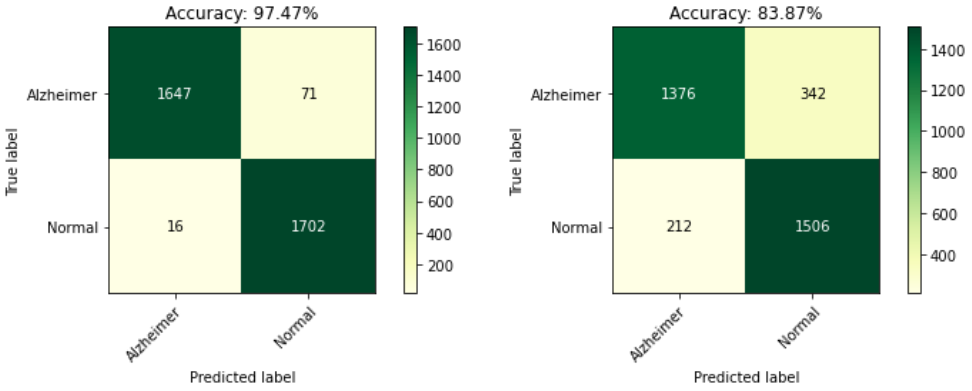


Fig. 3. Confusion matrix. (Right) EIIP (Left) entropy based mapping technique

As seen in Fig. 3, Entropy-based mapping technique correctly identified 1647 of 1718 signal segments, while EIIP defined 1376 segments correctly. In addition, while the entropy technique incorrectly classified only 16 of the gene sequences belonging to normal individuals, the EIIP technique incorrectly classified 212 data pieces.

In this study, an application has been carried out for the classification of DNA sequences, which are digitized by two different techniques, by capsule networks. There are some studies on classification of DNA and RNA sequences with capsule networks and discovery of disease-related genes. Table 3 shows the results of studies using capsule networks in genomic studies. As seen in Table 3, capsule models were applied to different data sets and different performance results were obtained. As seen in Table 3, the proposed method achieves a better result than performances in other studies for the recognition of Alzheimer's disease. The proposed method achieved an accuracy performance of 97.47% by using capsule networks to identify Alzheimer's disease from DNA sequences.

TABLE III. PREVIOUS WORKS ON DNA SEQUENCES USING CAPSULE NETWORKS

Author	Method	Data	Acc (%)
De Jesus et al. [24]	Capsule Net	HRAS, KRAS protein	94.00
Peng et al. [25]	CapsNetMMD	Breast cancer DNA gene	94.60
Wang et al. [26]	scCapsNet	RNA sequences	96.00
Proposed method	1-D Capsule Net	Alzeimer DNA gene	97.47

IV. CONCLUSION

In this study, a method for detecting Alzheimer's disease from DNA signals using capsule networks is proposed. In this proposed approach, Entropy based mapping technique and EIIP technique are used for digitizing DNA sequences. The digitized signals were divided into parts and each section was classified with capsule networks and high classification performance was obtained. With the method used in future studies, a study is planned to detect genetic diseases different from DNA sequences. It is also aimed to use different deep learning architectures for different data sets.

REFERENCES

[1] Alzheimer's Association Report 2019 Alzheimer's disease facts and figures, Alzheimer's Dementia, vol. 15 (3), pp. 321-387, 2019.

[2] R. Cacace, K. Sleegers, C. Van Broeckhoven, Molecular genetics of early-onset Alzheimer's disease revisited Alzheimer's Dement.. 12 (6), pp. 733-748, 2016.

[3] S. M. Neuner, J. Tcw, and A. M. Goate, "Genetic architecture of Alzheimer's disease". Neurobiology of Disease. vol. 143. pp. 104976. Sept. 2020. doi: 10.1016/j.nbd.2020.104976.

[4] S. Karantzoulis,. J.E. Galvin, Distinguishing Alzheimer's disease from other major forms of dementia Expert. Rev. Neurother.. 11 (11), pp. 1579-1591, 2011.

[5] G.M. McKhann, D.S. Knopman, H. Chertkow, B.T. Hyman, R. Jack Jr., C.H. Kawas, C.H. Phelps, The diagnosis of dementia due to Alzheimer's disease: recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease Alzheimer's Dement.. vol. 7 (3), pp. 263-269, 2011.

[6] S. J. Andrews, B. Fulton-Howard, and A. Goate, "Interpretation of risk loci from genome-wide association studies of Alzheimer's disease". The Lancet Neurology. Vol. 19(4). pp. 326-335. Apr. 2020. doi: 10.1016/S1474-4422(19)30435-1.

[7] S. Raj, A. Vishnoi, and A. Srivastava, "GOLD standard dataset for Alzheimer genes", Data in Brief, vol. 30, pp. 105439, Jun. 2020, doi: 10.1016/j.dib.2020.105439.

[8] C. Park, J. Ha, S. Park "Prediction of Alzheimers disease based on deep neural network by integrating gene expression and DNA methylation dataset", Expert Systems With Applications, vol. 140, 2020. https://doi.org/10.1016/j.eswa.2019.112873

[9] U. Rangaswamy, S. A. P. Dharshini, D. Yesudhas, and M. M. Gromiha, "VEPAD - Predicting the effect of variants associated with Alzheimer's disease using machine learning", Computers in Biology and Medicine, vol. 124, pp. 103933, Sept. 2020. doi: 10.1016/j.compbiomed.2020.103933.

[10] W. Shao, S. Xiang, Z. Zhang, K. Huang, and J. Zhang, "Hyper-graph based sparse canonical correlation analysis for the diagnosis of Alzheimer's disease from multi-dimensional genomic data", Methods, Apr. 2020. doi: 10.1016/j.ymeth.2020.04.008.

[11] S. Bringas, S. Salomón, R. Duque, C. Lage, and J. L. Montaña, "Alzheimer's Disease stage identification using deep learning models", Journal of Biomedical Informatics, vol. 109, pp. 103514, Sept 2020. doi: 10.1016/j.jbi.2020.103514.

[12] NCBI Genbank. https://www.ncbi.nlm.nih.gov (accessed August 15, 2020).

[13] B. Kozarzewski,"A method for nucleotide sequence analysis", Computatıonal Methods in Science and Technology, vol. 18(1): pp.5-10, 2012.

[14] Zhang, J. Su, D. Yu, Q. Wu, ve H. Yan, "EpiDiff: Entropy-based quantitative identification of differential epigenetic modification regions from epigenomes", In 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 655-658, 2013.

[15] R. E. Monge, J. L. Crespo, "Comparison of complexity measures for DNA sequence analysis", 3rd IEEE International Work-Conference on Bioinspired Intelligence, pp. 71-75, 2014.

[16] A. Karcı, "New kinds of entropy: fractional entropy", International Conference on Natural Science and Engineering (ICNASE'16), 2016.

[17] B. Das, I. Turkoglu, A novel numerical mapping method based on entropy for digitizing DNA sequences, Neural Comput. Appl. 29, pp. 207-215, 2018.

[18] B. Daş, Development of New Approaches Based On Signal Processing For Disease Diagnosis From Dna Sequences, Fırat University, PhD Thesis, 2018.

[19] S. N. Achuthsankar, S. Sreenadhan, A. Pillai, "Coding measure scheme employing electron-ion interaction pseudo potential (EIIP)", Bio-information, vol.1 pp. 197-202, 2006.

[20] I. Cosic, "Macromolecular Bioactivity: Is it resonant interaction between macromolecules? Theory and Applications", IEEE Transactions on Biomedical Eng., vol. 41, pp. 1101-1114, 1994.

[21] R. Mukhometzianov, J. Carrillo, CapsNet comparative performance evaluation for image classification, arXiv:1805.11195. arXiv.org, 2018.

[22] S. Sabour, N. Frosst, GE. Hinton, Dynamic routing between capsules. In: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 3859–3869, Dec 2017.

[23] S Toraman, TB Alakus, I Turkoglu. Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. Chaos, Solitons & Fractals, vol. 140, pp.110122, 2020.

[24] D.A. Rosa De Jesus, J. Cuevas, W. Rivera, S. Crivelli, Capsule Networks for Protein Structure Classification and Prediction, ArXiv:1808.07475v1. pp. 1–12, 2018.

[25] C. Peng, Y. Zheng, D. Huang, Capsule Network based Modeling of Multi-omics Data for Discovery of Breast Cancer-related Genes, IEEE/ACM Trans. Comput. Biol. Bioinforma. 2019. https://doi.org/10.1109/tcbb.2019.2909905.

[26] L. Wang, R. Nie, R. Xin, J. Zhang, J. Cai, scCapsNet: A deep learning classifier with the capability of interpretable feature extraction, applicable for single cell RNA data analysis, BioRxiv. 2019. https://doi.org/10.1101/506642.

Artificial Neural Networks Based Survival Prediction of Heart Failure Using Only Serum Creatinine and Ejection Fraction

Zehra KARAPINAR SENTURK
Computer Engineering Department
Duzce University
Duzce, Turkey
zehrakarapinar@duzce.edu.tr

***Abstract*—Cardiovascular diseases are the first cause of death all over the world. Once the heart is unable to pump sufficient blood that the body needs, heart failure arises. In this paper, an early prediction system for survival of patients was proposed. Distinctive feature of the proposed method is that only two attributes were used for the prediction. Artificial neural networks (ANN) were used to predict death event. The experimental results show that only two attributes are enough to almost accurately predict whether a patient will die. This achievement will give a chance to make the patients live. The doctors will be able to take proper precautions based on the decision of the proposed system. Also, since only two attributes were used in this system, only two tests will be performed and the cost and exertion of heavy laboratory tests will be reduced.**

Keywords—artificial neural networks, heart failure, machine learning, death event prediction

I. INTRODUCTION

According to National Health Service of United Kingdom, heart failure means that the insufficiency of the blood pumped through the body [1]. Heart failure (HF) does not mean stopping of heart, but the heart need some support to work properly. Mostly older people are under the risk of HF. HF mostly cannot be treated but can be controlled for many years [1]. WHO reports that cardiovascular diseases are the first cause of death around the world and approximately 18 million people die each year [2]. This number makes the early prediction of death events crucial.

Machine learning (ML) has been frequently used for the diagnosis of different diseases for decades. ML has been exploited especially for the deadliest diseases like cancer. Senturk and Kara [3] tried to diagnose breast cancer through seven machine learning methods in 2014 and they proved the usability of ML for breast cancer diagnosis. After this date, the researches on ML based disease diagnosis were increased. ML based breast cancer diagnosis has still been studied in many papers ([4], [5], [6], [7]). Endometrial cancer [8], tongue cancer [9], lung cancer [10], and skin cancer [11] are the example researches that uses ML for cancer diagnosis.

Apart from cancer diagnosis, ML has also been used for cardiovascular diseases. Jiang et. al [12] used ML based decision support system for emergency department triage for the patients with possible cardiovascular diseases. Four ML methods were applied to decide on the level of triage. Multinomial logistic regression, extreme gradient boosting, random forest and gradient-boosted decision tree were compared and extreme gradient boosting was shown to be better for the problem. Hossain, Uddin, and Khan [13] established a cardiovascular disease (CVD) risk modelling for type 2 diabetics T2D). Features of the patients with both CDV and T2D and patient patients with only T2D were analyzed. Six ML models were used to assess the risk of CVD in T2D patients. Classification accuracies were measured between 79% and 88%. Cardiovascular disorder severity was detected by extreme learning machine (ELM) method in [14]. 86 myocardial anatomic features were used to classify ejection fraction as normal, mild, moderate, and severe. High multi-class classification accuracies were obtained by optimized ELM. Sanchez-Cabo et al. [15] showed that ML cardiovascular risk definition for young and asymptomatic people. An ML model was exploited to predict the presence and extent of subclinical atherosclerosis (SA). Estimating SA with the proposed model was used to refine risk estimation.

There are also several papers related to heart failure (HF). Sax et al. [16] developed an ML based risk classification tool for emergency department patients with acute HF. Jing et al. [17] aimed to develop a strategy to manage the populations with HF using ML based methodology. Angraal et al. [18] predicted mortality and hospitalization in HF. 86 variables were used in modelling of ML. Logistic regression, random forest, gradient boosted trees, and support vector machines were evaluated. They also evaluated the dataset for several statistical measures. Mean area under curve was measured best with random forests as 0.72 for mortality.

Most of the ML-based studies uses too much features for diagnosis. However, Chicco and Jurman [19] showed that ML can predict survival of patients with HF using only two features. They showed that only serum creatinine and ejection fraction can be used to predict whether an HF patient will die or survive. But their Random Forest based classification method lacks high recall value. That is, they predict many of the patients who will die wrongly. In this paper, we aimed to improve recall value of the prediction of death event for HF patients. An Artificial Neural Networks (ANN) based method was proposed for the problem. ANN is one of the most popular machine learning methods. Wide range of problems were solved using ANN-based approaches. They are easily adapted to any type of problems and mostly produce good classification performance. Therefore, death event prediction for HF patients was performed using ANN. Contributions of this study are as follows:

- Death event prediction was performed with a lightweight neural network (one hidden layer and 25 neurons)
- Normalization was performed to improve the performance of ANN for HF survival prediction before the features were inputted to the ANN model.

- Survival prediction was realized only two attributes, serum creatinine and ejection fraction, with high performance.
- Very high recall, specificity, accuracy and f1-score were achieved when compared to the state-of-the-art.

This paper is organized as follows: Section 2 describes the dataset used in the research and the details of the method, the experimental results are given in Section 3 and lastly Section 4 draws some conclusions.

II. Materials and Methods

In this part of the paper, the dataset and the methodology will be elaborated.

A. Heart Failure Dataset

We analyzed a dataset constructed by Ahmad et al. [20] and last updated in 2020. The dataset contains data about 299 patients with heart failure between April and December 2015 at the Faisalabad Institute of Cardiology and at the Allied Hospital in Faisalabad (Punjab, Pakistan). There are 13 features that define each patient as given in Table I.

TABLE I. Features in the Dataset

No	Feature	Meaning	Range
1	Time	Follow up period	[4,285]
2	Event	Patient died or not during follow up period	0,1
3	Gender	Sexes of patients	0,1
4	Smoking	Smoker or not	0,1
5	Diabetes	Has diabetes or not	0,1
6	BP	Blood Pressure: Has hypertension or not	0,1
7	Anaemia	Has anemia or not	0,1
8	Age	Age of patient	[40,95]
9	Ejection Fraction	Percentage of blood leaving the heart at each contraction	[14,80]
10	Sodium	Sodium level in blood	[113,148]
11	Creatinine	Creatinine level in blood	[0.5,9.4]
12	Platelets	Platelets in blood	[25100,850000]
13	CPK	Creatinine PhosphoKinase enzyme level in blood	[23,7861]

All of the patients in this dataset has left ventricular systolic dysfunction and they are in class III and class IV according to NYHA [21] functional classification. Event attribute which indicates whether the patient died or not during the follow up period is the target variable. Some attributes in the dataset are binary. Gender, smoking, diabetes, BP, and anemia are binary-valued attributes. In addition, some attributes are continuous including age, sodium, and CPK. There are also categorical attributes. For the details about the attributes [20] can be read. The aim of this dataset was to determine death event using other 12 attributes. In this study, e focus on increasing the diagnosis performance using only serum creatinine and ejection fraction. We intended to prove the hypotheses claimed by Chicco and Jurman [19] and improve their diagnosis performance.

B. Artificial Neural Networks (ANN) Based Death Event Prediction

Inspired by the human neurological system, ANN adapts the learning ability of human and applied to many nonlinear real life problems. Its success in nonlinear problems and easy adaptation made it widespread. It is used in broad spectrum of problems of both life sciences and engineering. ANN has been used for disease diagnosis for decades ([3], [22], [23], [24], [25]). ANN has also preferred for the diagnosis of heart diseases ([26], [27], [28], [29]).

ANN has three or more layers including one input layer, one output layer and one or more hidden layers between input and output layers. Small units in layers are called as neurons. Input neurons take the signals from the environment and transfer them through the output layers. All the neurons are fully connected to the neurons in consecutive layers. Neurons in the hidden and output layers sum up the weighted signals from the previous layers and inputs the weighted sums to activation function. The results returned from the activation functions become the outputs of the neurons. Once the output is obtained in the output layer, it is compared by the target output and if it is not same, then the error is computed and this error is back propagated through previous layers. The weights of the connections are updated during back propagation. This process is repeated for all input samples in the dataset many times till the stopping criterion is satisfied and the network learns the weights of the connections in order to decide about any input correctly. This process is called as training. A typical neural network is shown in Fig. 1. x_1, x_2, and x_3 indicate the input signals and Y denote the output of ANN. *W*s on the connections stand for the weights, *b*s are the biases. Weights indicate the slope of the decision lines, and biases are used to shift decision lines through the axes.

Fig. 2 shows the architecture of the proposed heart failure survival prediction system. First, heart failure dataset was imported, and then its two attributes, namely creatinine and ejection fraction were selected in “Select Attributes” module. After that, all the attributes were normalized into the range [0,1]. Normalization affects the classification performance considerably [30]. After normalization, the dataset was divided into two sets randomly as training and test sets. Training data was given as the input of ANN and the model whose details are shown in Fig. 3 were trained. Test data was not given to the network during training. The performance of the network for this previously unseen samples in the test set was measured in order to make a proper evaluation. The performance was evaluated by accuracy, specificity, recall, and f1-score metrics which are frequently used for the performance evaluations in many classification problems. The formulas of the metrics are given in Table II.

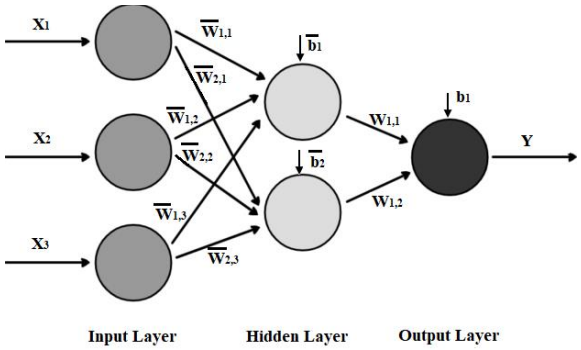


Fig. 1. Example ANN structure

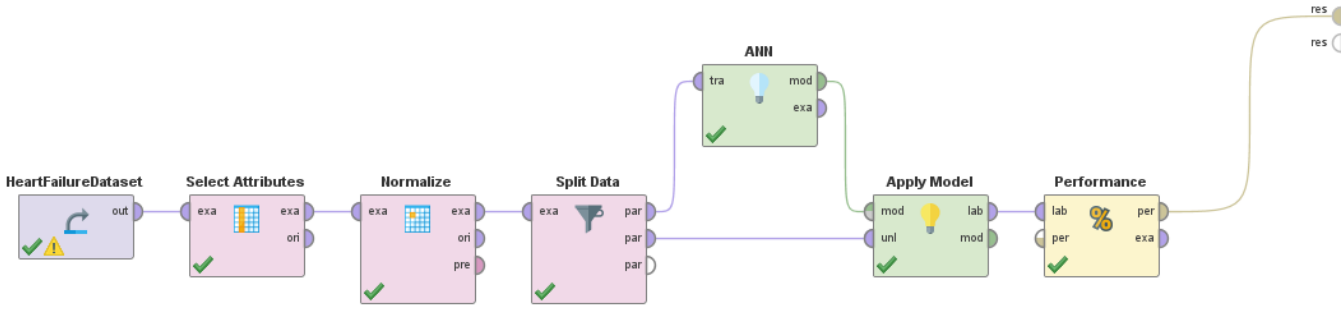


Fig. 2. Architecture of the proposed ANN based survival prediction system for heart failure patients

TABLE II. PERFORMANCE METRICS

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Recall	$\frac{TP}{TP+FN}$
f1-score	$\frac{TP}{TP+\frac{1}{2}(FP+FN)}$
TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative	

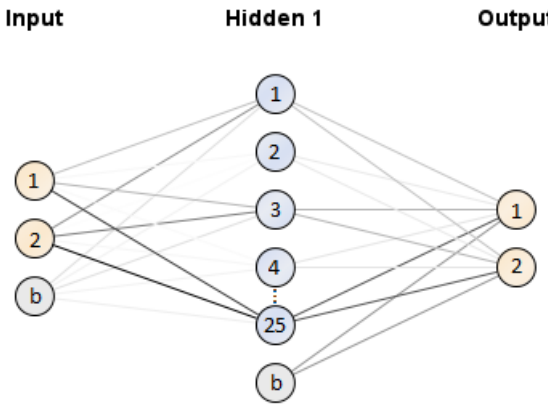


Fig. 3. Proposed ANN topology

III. EXPERIMENTAL RESULTS

In this study, we used an ANN with one hidden layer. Since we aimed to predict survival of heart disease patient with only two variables, there will be two input neurons. We have two output neurons, '10' indicates that the patient dies and '01' indicates that the patient will survive. We designed the network with 25 neurons in hidden layer as shown in Fig. 3. Different network architectures were tried to achieve the best performance. Table III shows the performances of different network topologies. Ideal number of neurons was determined through a series of experiments. The performance was measured for different number of neurons in hidden layer. The tradeoff between the specificity and recall were considered. Previous solution of this problem in [19], i.e. survival determination of heart failure using only two attributes namely creatinine and ejection fraction, lacks high recall value. Recall states the rate of correctly classified positive samples and total number of positive samples. That is, it shows the prediction success of death event. This is more important than the prediction success of survival event. In this research, we aimed to improve recall value of the mentioned classification problem rather than accuracy. We carried out the experiments in this direction. In addition, f1-score is the harmonic mean of precision and recall. Alibi et al. [31]states that f1 score must be included in the performance metrics in medical applications. Therefore, we also measured f1-score for the performance evaluation.

TABLE III. PERFORMANCE EVALUATION OF DIFFERENT MODELS

Number of neurons	Performance (%)			
	Accuracy	*Specificity*	*Recall*	*f1-score*
3	**90**	93.62	76.92	76.92
5	86.67	93.62	61.54	66.67
7	88.33	**95.74**	61.54	69.57
10	88.33	**95.74**	61.54	69.57
15	85.00	**95.74**	46.15	57.14
20	86.67	93.62	61.54	66.67
25	86.67	85.11	**92.31**	**75.00**
30	86.67	**95.74**	53.85	63.64
35	86.67	93.62	61.54	66.67

As seen from Table III, highest recall and f1-score are obtained by using 25 neurons in the hidden layer. Therefore, the best model was chosen so. Specificity and accuracy of this model were also satisfactory enough for survival prediction of patients with heart failure.

We also compared the results of the proposed approach by the most recent study related to the topic. Table IV shows the comparison with the state-of-the-art. ANN achieved remarkable performance for previously unseen samples in terms of all metrics.

All the experiments were performed using Rapid Miner studio, an open machine learning tool [32].

TABLE IV. COMPARISON WITH THE STATE-OF-THE-ART

Reference-Year	Performance (%)			
	Accuracy	*Specificity*	*Recall*	*f1-score*
[19]-2020	74.00	83.10	53.20	54.70
Proposed	**86.67**	**85.11**	**92.31**	**75.00**

IV. Conclusions

Heart diseases cause the most deaths in the world. Therefore, determination of death event for the patients with heart failure (HF) is crucial. Determining whether a patient will die or not gives a big chance to take adequate precautions. In this paper, different from most of the heart failure studies, we predicted if a patient with HF will survive or not using only two features, serum creatinine and ejection fraction. Serum creatinine is measured by a blood test and ejection fraction is mostly measured by echocardiogram. This paper shows that only these two tests are enough to determine whether a patient will die or survive. Since only two attributes were used in this system, only two tests will be performed on the patients and the cost and exertion of heavy laboratory tests will be reduced. The proposed methodology outperforms the state-of-the-art in terms of accuracy, specificity, recall, and f1-score.

ANN-based classification can also be adapted to other diseases. Normalization improved the performance and in future studies, the architecture of the proposed method will be applied to different diseases using a few attributes.

References

[1] "Heart failure - NHS." https://www.nhs.uk/conditions/heart-failure/ (accessed Jan. 17, 2021).

[2] WHO, "Cardiovascular diseases." https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 (accessed Jan. 17, 2021).

[3] Z. K. Senturk and R. Kara, "Breast Cancer Diagnosis Via Data Mining: Performance Analysis of Seven Different Algorithms," *Comput. Sci. Eng. An Int. J.*, vol. 4, no. 1, pp. 35–46, Feb. 2014, doi: 10.5121/cseij.2014.4104.

[4] N. Goyal and M. Chandra Trivedi, "Breast cancer classification and identification using machine learning approaches," *Mater. Today Proc.*, Dec. 2020, doi: 10.1016/j.matpr.2020.10.666.

[5] E. H. Houssein, M. M. Emam, A. A. Ali, and P. N. Suganthan, "Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review," *Expert Systems with Applications*. Elsevier Ltd, p. 114161, Oct. 27, 2020, doi: 10.1016/j.eswa.2020.114161.

[6] N. Al-Azzam and I. Shatnawi, "Comparing supervised and semi-supervised Machine Learning Models on Diagnosing Breast Cancer," *Ann. Med. Surg.*, vol. 62, pp. 53–64, Feb. 2021, doi: 10.1016/j.amsu.2020.12.043.

[7] A. G. V. Bitencourt *et al.*, "MRI-based machine learning radiomics can predict HER2 expression level and pathologic response after neoadjuvant therapy in HER2 overexpressing breast cancer," *EBioMedicine*, vol. 61, p. 103042, Nov. 2020, doi: 10.1016/j.ebiom.2020.103042.

[8] A. M. Praiss *et al.*, "Using machine learning to create prognostic systems for endometrial cancer," *Gynecol. Oncol.*, vol. 159, no. 3, pp. 744–750, Dec. 2020, doi: 10.1016/j.ygyno.2020.09.047.

[9] R. O. Alabi, A. A. Mäkitie, M. Pirinen, M. Elmusrati, I. Leivo, and A. Almangush, "Comparison of nomogram with machine learning techniques for prediction of overall survival in patients with tongue cancer," *Int. J. Med. Inform.*, vol. 145, p. 104313, Jan. 2021, doi: 10.1016/j.ijmedinf.2020.104313.

[10] Y. Xie *et al.*, "Early lung cancer diagnostic biomarker discovery by machine learning methods," *Transl. Oncol.*, vol. 14, no. 1, p. 100907, Jan. 2021, doi: 10.1016/j.tranon.2020.100907.

[11] S. Naresh Kumar and B. Mohammed Ismail, "Systematic investigation on Multi-Class skin cancer categorization using machine learning approach," *Mater. Today Proc.*, Dec. 2020, doi: 10.1016/j.matpr.2020.11.484.

[12] H. Jiang *et al.*, "Machine learning-based models to support decision-making in emergency department triage for patients with suspected cardiovascular disease," *Int. J. Med. Inform.*, vol. 145, p. 104326, Jan. 2021, doi: 10.1016/j.ijmedinf.2020.104326.

[13] M. E. Hossain, S. Uddin, and A. Khan, "Network analytics and machine learning for predictive risk modelling of cardiovascular disease in patients with type 2 diabetes," *Expert Syst. Appl.*, vol. 164, p. 113918, Feb. 2021, doi: 10.1016/j.eswa.2020.113918.

[14] M. Muthunayagam and K. Ganesan, "Cardiovascular Disorder Severity Detection Using Myocardial Anatomic Features Based Optimized Extreme Learning Machine Approach," *IRBM*, Jun. 2020, doi: 10.1016/j.irbm.2020.06.004.

[15] F. Sánchez-Cabo *et al.*, "Machine Learning Improves Cardiovascular Risk Definition for Young, Asymptomatic Individuals," *J. Am. Coll. Cardiol.*, vol. 76, no. 14, pp. 1674–1685, Oct. 2020, doi: 10.1016/j.jacc.2020.08.017.

[16] D. R. Sax *et al.*, "Use of Machine Learning to Develop a Risk-Stratification Tool for Emergency Department Patients With Acute Heart Failure," *Ann. Emerg. Med.*, Dec. 2020, doi: 10.1016/j.annemergmed.2020.09.436.

[17] L. Jing *et al.*, "A Machine Learning Approach to Management of Heart Failure Populations," *JACC Hear. Fail.*, vol. 8, no. 7, pp. 578–587, Jul. 2020, doi: 10.1016/j.jchf.2020.01.012.

[18] S. Angraal *et al.*, "Machine Learning Prediction of Mortality and Hospitalization in Heart Failure With Preserved Ejection Fraction," *JACC Hear. Fail.*, vol. 8, no. 1, pp. 12–21, Jan. 2020, doi: 10.1016/j.jchf.2019.06.013.

[19] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," *BMC Med. Inform. Decis. Mak.*, vol. 20, no. 1, p. 16, Feb. 2020, doi: 10.1186/s12911-020-1023-5.

[20] T. Ahmad, A. Munir, S. H. Bhatti, M. Aftab, and M. A. Raza, "Survival analysis of heart failure patients: A case study.Dataset," *PLoS One*, Jul. , doi: 10.1371/journal.pone.0181001.

[21] C. Bredy *et al.*, "New York Heart Association (NYHA) classification in adults with congenital heart disease: Relation to objective measures of exercise and outcome," *Eur. Hear. J. - Qual. Care Clin. Outcomes*, vol. 4, no. 1, pp. 51–58, Jan. 2018, doi: 10.1093/ehjqcco/qcx031.

[22] Z. H. Zhou, Y. Jiang, Y. Bin Yang, and S. F. Chen, "Lung cancer cell identification based on artificial neural network ensembles," *Artif. Intell. Med.*, vol. 24, no. 1, pp. 25–36, Jan. 2002, doi: 10.1016/S0933-3657(01)00094-X.

[23] Z. Karapinar Senturk, "Early diagnosis of Parkinson's disease using machine learning algorithms," *Med. Hypotheses*, vol. 138, p. 109603, May 2020, doi: 10.1016/j.mehy.2020.109603.

[24] A. Nithya, A. Appathurai, N. Venkatadri, D. R. Ramji, and C. Anna Palagan, "Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images," *Meas. J. Int. Meas. Confed.*, vol. 149, p. 106952, Jan. 2020, doi: 10.1016/j.measurement.2019.106952.

[25] B. Jaber and D. Bernal-Casas, "Artificial neural networks and the path to diagnosing Parkinson's Disease," *Parkinsonism Relat. Disord.*, vol. 79, p. e9, Oct. 2020, doi: 10.1016/j.parkreldis.2020.06.059.

[26] B. A. Mobley, E. Schechter, W. E. Moore, P. A. McKee, and J. E. Eichner, "Predictions of coronary artery stenosis by artificial neural network," *Artif. Intell. Med.*, vol. 18, no. 3, pp. 187–203, Mar. 2000, doi: 10.1016/S0933-3657(99)00040-8.

[27] W. Zeng, J. Yuan, C. Yuan, Q. Wang, F. Liu, and Y. Wang, "Classification of myocardial infarction based on hybrid feature extraction and artificial intelligence tools by adopting tunable-Q wavelet transform (TQWT), variational mode decomposition (VMD) and neural networks," *Artif. Intell. Med.*, vol. 106, p. 101848, Jun. 2020, doi: 10.1016/j.artmed.2020.101848.

[28] I. D. Mienye, Y. Sun, and Z. Wang, "Improved sparse autoencoder based artificial neural network approach for prediction of heart disease," *Informatics Med. Unlocked*, vol. 18, p. 100307, Jan. 2020, doi: 10.1016/j.imu.2020.100307.

[29] G. Eslamizadeh and R. Barati, "Heart murmur detection based on wavelet transformation and a synergy between artificial neural network and modified neighbor annealing methods," *Artif. Intell. Med.*, vol. 78, pp. 23–40, May 2017, doi: 10.1016/j.artmed.2017.05.005.

[30] E. Öztemel, *Yapay Sinir Ağları*, 4th ed. İstanbul: Papatya Bilim, 2016.

[31] R. O. Alabi *et al.*, "Comparison of supervised machine learning classification techniques in prediction of locoregional recurrences in early oral tongue cancer," *Int. J. Med. Inform.*, vol. 136, p. 104068, Apr. 2020, doi: 10.1016/j.ijmedinf.2019.104068.

[32] "RapidMiner | Best Data Science & Machine Learning Platform." https://rapidminer.com/ (accessed Jan. 17, 2021).

Using data science to detect software aging

Fatma Bozyiğit
Department of Computer Engineering
İzmir Bakırçay University
İzmir, TURKEY
fatma.bozyigit@bakircay.edu.tr

Kadir Sert
Department of Information Systems
Yurtiçi Kargo
İstanbul, TURKEY
kadir.sert@yurticikargo.com

Murat Şahin
Department of Data Science
Kalybe.AI
Manisa, TURKEY
murat.sahin@kalybeai.com

Dursun Dinçer
Department of Information Systems
Yurtiçi Kargo
İstanbul, TURKEY
dursun.dincer@yurticikargo.com

Deniz Kılınç
Department of Computer Engineering
İzmir Bakırçay University
İzmir, TURKEY
deniz.kilinc@bakircay.edu.tr

***Abstract*— The software aging phenomenon is related to the decrease in performance of long-running systems like business-critical servers. In this study, we build automated software rejuvenation design to prevent software aging. We analysed resource usage data over a period benefiting with the help of state-of-the-art data science libraries. Consequently, we detected unplanned restarts through the monitoring of resource usage and performability metrics of the application server. In case that some unusual behaviour is identified, the system creates notification and initiates an automatic rejuvenation action. This rejuvenation design interrupts the running system for the minimum time and achieves zero downtime.**

Keywords—business-critical servers, software aging, software rejuvenation, unplanned restarts

I. INTRODUCTION

The business-critical servers are intended to run continuously except software upgrades and maintenances. System failures or crashes in these servers cause data losses and some inconsistency in enterprise applications. Therefore, making server activities sustainable is an important issue that receives attention from enterprises with the emergence of big data concepts.

The term software aging refers to the system's degradation because of unexpectedly terminating a running process, cleaning its internal state, and restarting it [1]. It mostly occurs in long-running applications such as resource managers in server machines. Consequently, the crashes in business-critical servers can be covered in software aging phenomena. To solve this problem, mostly traditional recovery methods such as server redundancy, load balancers, and fail-over methods are preferred by the enterprises. Although these methods perform well enough for system recovery, it is seen that self-healing techniques which is also called as software rejuvenation [2] have become a popular research area recently.

The main idea behind software rejuvenation removing the collected error states and releasing system resources utilizing garbage collection, flushing operating system kernel tables, and reinitializing internal data structures [3]. This method is realized with planned restart operations that enable the system to return to maximum performance and so avoid the system lockout/crash. Since manual data analysis is time-consuming, there is a need to scale up human analysis capabilities to deal with a large of performability metrics of the application server. Therefore, business organizations focus on automated software rejuvenation in recent times.

In this study, we proposed automated software rejuvenation on real life data provided by Yurtiçi Kargo company [4]. This model detects and reports unplanned restarts and so losing any request or session data at the time of a restart. In this context, we perform data science methods to estimate the optimal time for restart and notify the system users before the system restart time.

The rest of the paper is organized as follows. The second section briefly outlines related studies. The third section gives a brief overview of the software aging and software rejuvenation concepts. The next section presents the results of our experimental study. The fifth section concludes the paper and give information about the future works.

II. LITERATURE REVIEW

In recent years, studies about software rejuvenation have become more popular to provide feasibility in business applications. In one of these studies, Candea et al. [5] introduced the concept of micro-rebooting to reduce the rejuvenation cost. Researchers claimed that rebooting a few numbers of components instead of applying restarts at certain times. The experimental results showed that the proposed approach decreases the applications' “mean time to repair (MTTR)” and provides the expected output. In [6], Fox and Patterson suggest decreasing the MTTR instead of dealing with the “mean time between failures (MTBF)”. They explained that the end-user does not see any interruption impact by reducing the value of MTTR. Avritzer and Weyuker [7] addressed research on software aging in a telecommunication system considering a progressive restart mechanism. They used the basis of the Markov Chain model to determine the proper times for the restart. It is aimed to recover the systems' internal state providing an uninterrupted execution period. In another study, Huang et al. [8] came up with a software rejuvenation approach regarding downtime costs. Trivedi et al. [9] utilized a probabilistic model considering the execution time and the workload parameters. Vaidyanathan and Trivedi [10] proposed a study to detect and identify the aging problem in the Apache webserver. In [11], Autoregressive Moving Average (ARMA) time series model is performed to predict random restart times, causing webserver aging. Machida et al. [12] introduced a novel method which enables to delay the aging process by allocating extra available resources, unlike traditional restart-based rejuvenation techniques.

TABLE 1 GENERAL INFORMATION ABOUT REVIEWED STUDIES

Study	Used parameters	Proposed model
[5]	MTTR	Markov reward model
[6]	MTBF, MTTR	Markov chain model
[7]	workload parameters	Markov chain model
[8]	downtime cost	Semi Markov model
[9]	execution time and the workload parameters	Probabilistic model
[10]	cpu usage parameters	Deterministic models
[11]	-	ARMA
[12]	system availability and mean job completion time	Semi Markov model

III. SOFTWARE AGING AND REJUVENATION

Software aging is a progressive performance degradation of the software's internal state during its operational life [13]. It generally occurs in continuously running processes and seen as unplanned restarts which result in request losses and therefore intermittent server activities. To solve software aging, software rejuvenation or self-healing techniques are widely used. Figure 1 illustrates the generic systems with rejuvenation policy.

Software rejuvenation is reviewed in two approaches such as open loop and closed loop [14]. For the open-loop approach, there is no need any feedback from the system to trigger rejuvenation action. In this case, rejuvenation depends on periodic time and the total number of operations on the network. On the other hand, closed-loop rejuvenation is accomplished by monitoring data on the resource usage and system activities at certain time intervals. Accordingly, obtained data is analysed to predict resource exhaustion, leading to system degradation. This evaluation can be based on time, system workload, or uncontrolled resource consumption. The data analysis in the closed-loop approach can be off-line or on-line. Off-line data analysis is based on system data collected over a specific period while the on-line data analysis performs with continuously collected data. Off-line approach is suitable for systems having reasonably deterministic behaviours. Online approach is performed after every new set of data is collected, and it is applied to systems with non-deterministic behaviours.

In this study, we performed closed loop rejuvenation based on off-line data analysis. We collected resource usage data over a period and perform data science techniques to detect extraordinary conditions which cause to unplanned restarts.

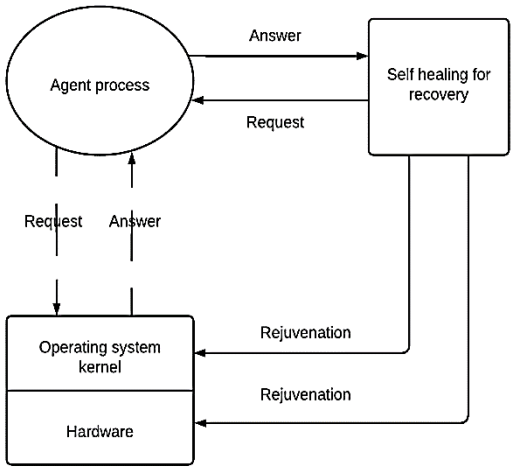


Figure 1 Software systems with rejuvenation policy.

IV. MATERIALS AND METHODS

Unplanned restarts in the business-critical servers results with serious costs since it causes losses in the system states to be rebuilt. In this study, we performed closed loop rejuvenation based on off-line data analysis to solve this problem. We collected resource usage data over a period and perform data science and analytics methods to detect extraordinary conditions which cause to unplanned restarts. Overall, we organized the rejuvenation actions to be invisible from the user's perspective.

A. Dataset

To prepare the experimental dataset, we collected resource usage data by using a shell program to run httperf at specific time intervals. In our dataset, there are fifteen attributes (explained in Table 2) and 587.520 observations for seven servers. The data frame starts from the time stamp 2020-06-01 00:00:10 and ends at 2020-08-11 00:00:00.

TABLE II FEATURES IN THE DATASET (OS: OPERATING SYSTEM)

Feature	Definition	Unit
timestamp	It is the timestamp of the data sampled.	Time stamp
cpu-user	It indicates the percentage of the total processor time the user processes running in the os.	Percentage
cpu-system	It displays percentage of the total processor time the system processes running in the os.	Percentage
cpu-wait	It indicates percentage of the total processor time which is used by processes waiting for input and output data in the os.	Percentage
disk-read	It shows the amount of data read over the disk per unit time.	Bytes per second (Bps)
disk-write	It indicates the amount of data writing on disk per unit of time.	Bps
mem-used	It displays the amount of memory (RAM) used by the operating system.	Kilo bytes (KB)
swap-used	It displays the amount of swap space used by the operating system.	KB
net-eth0-rx	It shows the amount of data in the input direction passed from the first network adapter per unit time.	Bits per second (bps)
net-eth0-tx	It shows the amount of data in the downstream direction passed from the first network adapter per unit of time.	bps
net-eth1-rx	It shows the amount of data in the input direction which passed from the second network adapter per unit of time.	bps
net-eth1-tx	It shows the amount of data in the downstream direction which passed from the second network adapter per unit of time.	bps
sch-threads	The number of threads that the Scheduler/Batch java process has.	Integer
sch-hbtime	It is the time elapsed since the last notification sent by the Scheduler / Batch java process.	Milliseconds
app-status	It shows the application status with respect to threshold value of 660.000 ms using the value of the sch-hbtime attribute.	responding or nonresponding
build-status	It indicates whether Scheduler/Batch java application is in the unplanned restart state or not.	running or not running

As original data have noisy records and null values, attributes with zero variances are removed in the data pre-processing step. The correlation of missing data can be observed in the heat map in Figure 2.

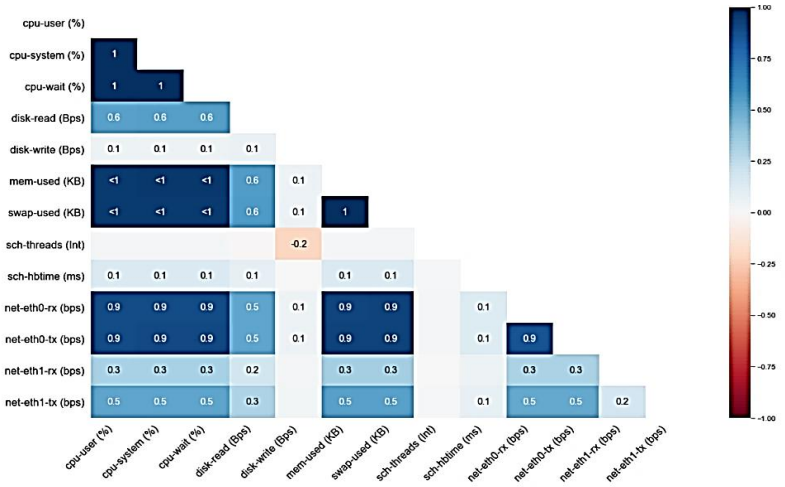


Figure 2 Heatmap analysis of the features.

B. *Experimental Study*

In this paper, we configured resources monitors to be deployed IT infrastructures of Yurtiçi Kargo, a well-known courier company in Turkey. Data collected from server applications for seventy-one days was analyzed to track server activities. Consequently, we detected the parameters which are the most informative about overload conditions.

We analyzed the collected data to determine thresholds indicating software aging, and the rejuvenation is triggered when the values of monitored parameters exceed such thresholds. We used Python libraries are Scikit-learn [15], Pandas [16], Numpy [17], and timeseries_functions [18] to identify of the best indicators and threshold values for them. Scikit-learn is one of the most useful and powerful libraries to effectively use machine learning designs and statistical modeling. It is built on NumPy, SciPy, and Matplotlib. Pandas library provides useful and expressive data structures designed to be working with structured and time-series data. It is an ideal tool for cleaning, modeling, and organizing the analysis results into a suitable form. Numpy is a numeric python, widely used for its fast-mathematical computation on arrays and matrices. We practiced time_series_functions to analyze a sequence of observations recorded at regular time intervals. Then, we applied the Shapiro-Wilk test [19] to detect outliers in the features which cause unplanned restart. The Shapiro-Wilk test employs the null hypothesis that the data was drawn from a normal distribution. Considering experimental results, the most informative features are determined as cpu-system, cpu-wait, disk-read, and swap-used. The scatter plots of the swap-used and cpu-system parameters is visualized in Figure 3 and 4, respectively. It is seen that the value of lower limit and upper limit are -40780.0 and 171732.0, respectively. Also, it is observed that there is no outlier under the lower limit and 68282 outliers above the upper limit for swap-used parameter (see Figure 3).

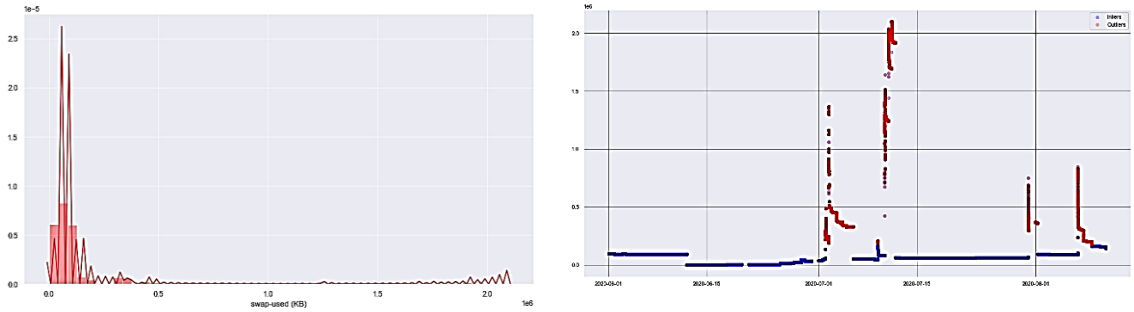


Figure 3 Scatter plot of swap-used parameter

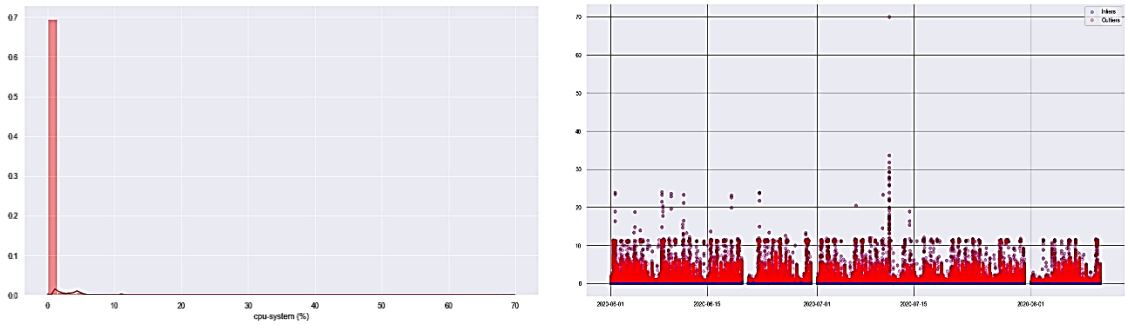


Figure 4 Scatter plot of cpu system parameter

V. Conclusion and Future Works

This study proposed automated software rejuvenation on real-life data provided by Yurtiçi Kargo company. We first detected and reported unplanned restarts causing request or session data losses through practicing data science methods. Then, we predict the optimal time for the restart and notify the system users before the system restart time. The experimental results of our study are promising and motivates us for further studies. We plan a new research to develop self-healing approach based on online analysis.

Acknowledgment

Funding for this work was partially supported by the Research and Development Centre of Yurtiçi Kargo.

References

[1] K. Vaidyanathan and K. S. Trivedi, "A comprehensive model for software rejuvenation," in IEEE Transactions on Dependable and Secure Computing, vol. 2, no. 2, pp. 124-137, April-June 2005, doi: 10.1109/TDSC.2005.15.

[2] Trivedi K S, Vaidyanaman K. Software Rejuvenation-Modeling and Analysis[C]. IFIP Congress Tutorials, 2008: 151-182.

[3] Lindeberg, T., & Wah, B. (2009). Encyclopedia of Computer Science and Engineering. Hoboken, New Jersey: John Wiley and Sons, 4, 2495-2504.

[4] YurtiçiKargo, 17.01.2020. [Online]. Available: https://www.yurticikargo.com/

[5] G. Candea, A. Brown, A. Fox, and D. Patterson, "Recovery Oriented Computing: Building Multi-Tier Dependability," Computer, vol. 37, no. 11, pp. 60-67, Nov. 2004.

[6] A. Fox and D. Patterson, "When Does Fast Recovery Trump High Reliability?" Proc. Second Workshop Evaluating and Architecting System Dependability, 2002.

[7] A. Avritzer and E. J. Weyuker, "A Monitoring Smoothly Degradating Systems for Increased Dependability", Empirical Software Engineering Journal, v. 2, no1, 1997, pp. 59-77.

[8] Y. Huang, C. Kintala, N. Koletis, and N.D. Fulton, "Software Rejuvenation: Analysis, Module and Applications", in Proc. 25th Symposium on Fault Tolerant Computer Systems, 1995, pp. 381-390.

[9] K. Vaidyanathan and K.S. Trivedi, "A Measurement-based Model for Estimation of Resource Exhaustion in Operational Software Systems", in Proc. 10th IEEE Int. Symposium on Software Reliability Engineering, 1998, pp. 84-93.

[10] L. Li, K. Vaidyanathan, and K.S. Trivedi, "An Approach for Estimation of Software Aging in a Web Server", Int. Symposium on Empirical Software Engineering, 2002, pp. 91-100.

[11] K.S. Trivedi, K. Vaidyanathan, and K. Goseva-Popstojanova, "Modeling and Analysis of Software Aging and Rejuvenation", in Proc. 33rd Annual Simulation Symposium, 2000, pp. 270-279.

[12] F. Machida, J. Xiang, K. Tadano and Y. Maeno, "Lifetime Extension of Software Execution Subject to Aging", IEEE Transitions on Reliability, vol. 66, pp. 123-134, 2017.

[13] S. Garg, A. van Moorsel, K. Vaidyanathan, and K. Trivedi, "A Methodology for Detection and Estimation of Software Aging," Proc. Ninth Int'l Symp. Software Reliability Eng., pp. 282-292, 1998.

[14] A. Andrzejak and L.M. Silva, "Deterministic Models of Software Aging and Optimal Rejuvenation Schedules," Proc. 10th IFIP/IEEE Int'l Symp. Integrated Network Management (IM '07), May 2007.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel,405J. Vanderplas, Scikit-learn: Machine learning in python, Journal of Ma-chine Learning Research 12 (2011) 2825–2830.

[16] W. McKinney, pandas: a foundational python library for data analysis andstatistics, Python for High Performance and Scientific Computing 14 (9)(2011)

[17] S. van der Walt, S. Colbert, G. Varoquaux, The numpy array: A structurefor efficient numerical computation, Computing in Science & Engineering13 (2) (2011) 22–30.

[18] McKinney, W., Perktold, J., & Seabold, S. (2011). Time series analysis in Python with statsmodels. Jarrodmillman Com, 96-102.

[19] Royston, P. (1992). Approximating the Shapiro-Wilk W-test for non-normality. Statistics and computing, 2(3), 117-119.

Hypermeters Optimization in Recurrent Neural Networks-LSTM Approach for Human Activity Recognition

Ayşenur Topbaş
Department of Computer Engineering
Sabahattin Zaim University
Istanbul, Turkey
aysenur.gencdogmus@izu.edu.tr

Alaa Ali Hameed
Department of Computer Engineering
Sabahattin Zaim University
Istanbul, Turkey
alaa.hameed@izu.edu.tr

Akhtar Jamil
Department of Computer Engineering
Sabahattin Zaim University
Istanbul, Turkey
akhtar.jamil@izu.edu.tr

***Abstract*— Human activity recognition (HAR) has become more popular with the increase of applications involving human-computer interaction. The problem of recognizing and classifying people's daily life activities is a very important and challenging issue in the field of artificial intelligence. In recent years, deep learning based techniques have achieved high accuracy for HAR. However, these methods require obtaining a optimal number parameters. In this study, we investigated Long Short-Term Memory (LSTM) for HAR from videos. In addition, hyperparameters of the model were obtained thourh a thorough search to optimize ther performance of the model. Specifically, we obtained optimal values for number of layers, batch size, epochs to obtained best accuracy for the model. Experiments were performed to evaluate the performance of the proposed model on WISDM dataset. The proposed model produced an overall 97.18% overall accuracy indicates that LSTM is an effective technique for HAR.**

Keywords*— *Human Activity Recognition, RNN, LSTM Neural Networks, Deep Learning

I. INTRODUCTION

Human Activity Recognition problem is an active research area as it can be applied in assorted applications such as human machine interactions and robotics. In particular, the widespread use of human-computer interactive applications has made the HAR subject even more interesting. The daily movements of people can be tracked and recorded, and activities can be classified for this activity data. Especially in the public health field, HAR is very useful. Monitoring daily activities will help predict many diseases. One of the most common sensors used to monitor human activities are accelerometers in smart phones. It is possible to record activity through the sensors in smart phones, which have become an integral part of our day. Human activities can be captured in three axes (x,y,z axes) through these accelerometers. This tri-axial data is used to classify people's daily activities like sitting, jogging, walking, downstairs, upstairs, standing.

Nowadays, recognition of human activities is a very common research problem. A lot of work has been done on this subject in recent years. In particular, artificial neural networks and deep learning methods are highly preferred methods. Researchers have also proposed a neural network, a network that combines convolutional layers with long short-term memory (LSTM). [1] Through this model, human activities can be automatically extracted and classified through a few model parameters. The performance of the proposed model has been evaluated on 3 public datasets. (UCI, WISDM, and OPPORTUNITY datasets.) Similarly, we trained the model we proposed in our study using WISDM dataset. Others have proposed a model named EnsemConvNet for human activietis recognition [2]. This model used three classifiers: Encoded-Net, CNN, and CNN with LSTM. Three benchmark data sets are used to evaluate their proposed model: WISDM, MobiAct and UniMiB SHAR. Finally, they have also compared their method with some well know deep learning technique including hybrid of CNN, Multi Headed CNN, and LSTM models.

Although deep learning algorithms show great success in HAR, but these are expensive in terms of processing and it become difficulty to carry out on edge devices. [3] proposes a novel model making it suitable on edge devices by requiring less computational power for HAR. This models name is Lightweight Deep Learning Model. And they developed their models on the RNN-LSTM model similar to our study.

In this paper, we propose the RNN-LSTM model, which is one of the deep learning methods to recognize human activities. We trained our model on WISDM dataset, which is a public dataset. We tested it to find the optimum values of the hyperparameters to improve the performance of the model. We have shown that with optimum hyperparameters, our model achieves higher accuracy than previous studies.

The rest of this paper can be summarized as follows. Section 2 will examine previous studies using deep learning methods of human activity recognition. In section 3, we will give detailed information about the dataset used in our study. In the 4th section, the model and the hyperparameters of the proposed model will be examined. Section 5 will show the experiment results. And we will summarize this research with a brief summary in the final section.

II. RELATED WORK

HAR has been an important and challengin research problems for years. Many researchers are working to reach the best classification model on this subject. There has been a lot of work on HAR before. This section mainly examines the studies that used deep learning methods before.

The advent of deep learning era has lead to development of innovative methods to solve various problems in HAR. For instance, in [4], authors proposed deep neural network using bi-directional residual cells with LSTM. The technique has several advantages, such as it can be combined in the two-way connection with negative time direction (reverse state) and positive time direction (forward state). Considering that the Deep LSTM neural network can

automatically learn the feature representations and model long-term transient dependencies, the experimental results showed that the method was effective for HAR [5]. In [6], authors proposed mobile and wearable sensor-based human activity as a compilation that provides an in-depth summary of deep learning methods.

Nowadays, recent advances in deep learning methods have made automatic high-end feature extraction possible and performed promising in many areas. This study, which summarizes the literature in terms of deep learning model, deep learning applications and sensor method, also examines the latest developments in sensor-based deep learning activity recognition. [7]. In another study [8] where deep convolutional neural network is proposed to realize effective and efficient human activity recognition application, an almost perfect classification was obtained, especially for activities very similar to those already considered to be very difficult to classify.

In another study [9] in which the deep learning model was applied independently of the user, Convolutional Neural Networks were proposed for feature extraction. Similarly, in another study, the Long Term Short Term Memory (LSTM) architecture is designed to apply human activity recognition. Over 94% accuracy and less than 30% loss were achieved in the first 500 training epochs [10]. Some researchers have proposed a hierarchical deep learning method for the LSTM neural network. They named this model H-LSTM. After the smoothing and denoising preprocesses, it will be featured with the time-frequency domain method [11].

A single layer stacked LSTM model was used in a study [12] where the LSTM model was proposed for the recognition of data including six different human movements obtained via smart phones. The proposed network have tried on network UCI data. The performance of the model is calculated in terms of precision-call and average accuracy.

Recognizing human activities has become a very important problem in terms of human health as it gives information about the movement intensity of people. LSTM model was used in this study [13], which deals with both the intensity of movement and the classification of movements problem. In another study, many methods used in human activity recognition were examined and these methods were tested on 10 publicy datasets [14]. In particular, CNN, gated recurrent unit networks (GRU), LSTM, bidirectional LSTM (biLSTM), and deep belief networks (DBN) models were evaluated on their dataset.

The data obtained from the sensors in smart phones are used especially by machine learning methods to recognize human activities. A bidirectional long short-term memory (LSTM) network has been proposed to classify this data collected by gyroscopes and accelerometers in smartphones [15].

In our study, we optimized the hyperparameters of the LSTM model to improve the accuracy performance obtained from the above methods. We evaluated it on the WISDM data set, one of the most widely used public data sets. We demonstrated the effect of parameters in the model on accuracy and determined the optimum parameters.

III. DATASET DESCRIPTION

The WISDM dataset is used in this study which consists of a total of 1,098,209 samples. The percentage of total samples present in the data set for each activity is summarized in Table 1. There are six different human activities which include walking (Walk), jogging (jog), upstairs (Up), downstairs (Down), sitting (Sit), standing (Std).

It is worth noting that data distribution is not balanced. For instance, walking and jogging accounts for more than 30% of all activities while rest of the classes account for less than 12% each. This can adversly effect the classification accuracy of the classifiers as model can be biased towards the class with donimant samples.

The data in the dataset were collected from 36 different subjects. Each subject carried out daily activities by putting a smartphone in their front leg pockets. This sensor used while creating the data set is an accelerometer. This accelerometer has a sampling frequency of 20 Hz and is also a motion sensor built-in smartphones.

TABLE I: SUMMARY OF THE WISDM DATASET

Activities	*No. of Samples*	*Percentage*
Walking	424400	38.6%
Jogging	342177	31.2%
Upstairs	122689	11.2%
Downstairs	100427	9.1%
Sitting	59939	5.5%
Standing	48397	4.4%

IV. DEEP LEARNING ARCHITECTURE

A. Deep Learning Methodology

The deep learning model consists of artificial neural networks. ANNs are composed of neuron cells just like the human brain structure. Neurons are all interconnected in the network, and the way they connect affects the output. There are basically 3 layers in a neuron. These layers; input layer, hidden layer (s) and output layer.

- **Input Layer:** It is the layer that contains the input data. It transmits input data from the input layer to the first hidden layer.
- **Hidden Layer(s):** It is the layer on which mathematical calculations are made on the input data. The expression "deep" in the concept of deep learning indicates that the neural network has multiple hidden layers. One of the biggest challenges in creating artificial neural networks is determining the number of hidden layers as well as determining the number of neurons for each layer.
- **Output Layer:** It is the layer from which the output data is obtained.

One of the most difficult tasks in deep learning methods is training the artificial neural network. The reason for this is that a lot of computational power and a very large data set are needed in order to apply the deep learning methods. In our study, we conducted LSTM (Long Short-Term Memory) neural network training for human activity data obtained from sensors.

B. Creating the LSTM Neural Network

We created the LSTM model, which is a specialized method of RNN, which is one of the deep learning methods commonly used in HAR. A neuron; LSTM consists of point processes and layers. These are several layers, data input and output layers that act as gates to forget to feed the cell state. What holds long-term memory and context for the inputs and the network is called the cell state.

An LSTM consists of several neurons arranged in layers. It consists of several gates which are terms as input, output, and forget. It also maintain a cell state. This cell state resembles the long-term memory which is maintanied across the network.

LSTM is a type of RNN. The main feature of such networks is that these architectures have feedback links in their architecture. This makes them more suitable for processing sequential data. In addition, these model have also been used to process image data.

The input layer of the network must have same number of neurons as the dimension of features in the data set, the size of each segment of the segmented data, and batch size properties. The weights and biases of for each layer in the network were initialized with normal distribution in the LSTM. In this neural network, a single output is reached from multiple inputs.

The tensors of the input and output in our model are defined separately. In the model, the L2 regulator will be used and this regulator we use should also specify the lost. While creating the model, Adam optimizer was used.

AdamOptimizer: This optimizer has become more popular and has been widely used for it high efficiency. The weight update is done iteratively in the network during the training dataset. It calculates learning rates from the first and second estimates of the gradients for the different parameters.

The activation function we use in our model is the softmax activation function.

Softmax Activation Function: it transforms the input vectors into probabilities. These probabilities are directly proportional to the scale of each element in the input vector [17]. The softmax activation function used in machine learning algorithms of applications is that it commonly acts as a neural network model activation. In cases where more than two classifications are required, this function has produced highly optimal results for neural networks. It ensures that the probability of the input belonging to a certain class is determined by producing values in the range of 0-1.

The parameters used in the proposed LSTM neural network and whose effects on accuracy are measured are as follows.

Epoch: To train the models, it is necessary to repeat the training not just once, but over and over again. Weights are updated for each training round. In order to find the most suitable weight value for the model, weight calculation is made for each new training data. The most suitable weight values for the model are thus calculated. Each of these steps is called an epoch.

Batch Size: It is the memory used by the model while it is being trained. It can be in the form of 2 and its multiples (4, 8, 16, 32, 64, ..., 512).

Number of Layers: Especially in complex problem examples, the most important feature that makes deep learning methods different from other artificial neural networks is the number of layers.

V. EXPERIMENTAL RESULTS

In this section, the results of the tests performed to examine the effect of the values of the hyperparameters (epoch, batch size, numbers of LSTM layers) on the accuracy of the proposed model are presented.

A. Effects of Number of LSTM Layers on Accuracy

Our aim in this section is to determine the effect of the number of LSTM layers in the model on accuracy. Layer number is very important especially in complex problems. In order to determine the appropriate number of layers for the dataset we use in our model, the number of epochs was kept constant as 200 and batch size as 32 in the tests in this section. Tests were carried out as LSTM layer numbers 2, 4 and 6. The accuracy and loss values of the tests are shown in Table 2.

TABLE 2: EFFECT OF LSTM LAYER SIZE ON ACCURACY AND LOSS (EPOCHS=200, BATCH SİZE=32)

LSTM Layers	*Accuracy*	*Loss*
2	95.99%	27.33%
4	**97.18%**	**27.33%**
6	95.66%	30.29%

The number of layers was initially set to 2 and the behavior of the model was observed by increasing it (4 and 6 layers). The dataset was divided into train (70%) and test (30%) data subset. As a result of our tests, when the effect of LSTM layer number on accuracy is examined, increasing the number of layers from 2 to 4 increased the accuracy and decreased the loss. However, when we built the model with 6 layers, the accuracy decreased and the loss increased. Fig. 1 shows the visualization of results for (a) train loss, (b) train accuracy, (c) test loss and (d) test accuracy.

B. Effects of Number of Epochs on Accuracy

Our aim in this section is to determine the effect of the number of epochs in the model on accuracy. In order to determine the appropriate number of epochs for the dataset we use in our model, the number of LSTM layers was kept constant as 2 and batch size as 32 in the tests in this section. Tests were carried out as number of epochs 300, 350 and 400. The accuracy and loss values of the tests are shown in Table 3.

TABLE 3: THE EFFECT OF EPOCHS ON ACCURACY AND LOSS

LSTM Layer=2, Batch size=32		
Epochs	*Accuracy*	*Loss*
300	96.30%	26.67%
350	96.54%	25.93%
400	**96.57%**	**25.53%**

The number of epochs was initially set to 300 and the behavior of the model was observed by increasing it (350 and 400 epochs). The train and test data in the LSTM neural network constituted 70% training data and 30% test data. As a result of our tests, when the effect of the number of epochs on accuracy is examined, with the increase of the number of epochs, the accuracy increased and the loss decreased. Fig. 2 shows the optimal results obtained for epochs (epochs number=400). It shows (a) train loss, (b) train accuracy, (c) test loss and (d) test accuracy.

C. *Effects of Batch Size on Accuracy*

Our aim in this section is to determine the effect of the batch size in the model on accuracy. In order to determine the appropriate batch size for the dataset we use in our model, the number of LSTM layers was kept constant as 2 and number of epochs as 200 in the tests in this section. Tests were carried out as batch size 32, 64 and 128. The accuracy and loss values of the tests are shown in Table 4.

TABLE 4: THE EFFECT OF NUMBER OF EPOCHS ON ACCURACY AND LOSS

LSTM Layer=2, Number of Epochs=200		
Epochs	***Accuracy***	***Loss***
32	96.30%	26.67%
64	96.54%	25.93%
128	**96.57%**	**25.53%**

The batch size was initially set to 32 and the behavior of the model was observed by increasing it (64 and 128). The train and test data in the LSTM neural network constituted 70% training data and 30% test data. As a result of our tests, when the effect of the batch size on accuracy is examined, with the increase of batch size, the accuracy increased and the loss decreased. Fig. 3 shows highest accuracy obtained with batch size (batch size = 128). It shows (a) train loss graph, (b) train accuracy graph, (c) test loss graph and (d) test accuracy graph.

D. *Optimized Hyperparameters Results*

We evaluated different values for the hyperparameters in the model and tried to find the optimum values. These the number of epochs, number of LSTM layers and the batch size. We empirically obtained the best performance of the model using 4 LSTM layers, batch size = 128 and number of epochs = 400.

With the optimum hyperparameters, we achieved the highest accuracy as 97.18% and the loss value as 27.33%. Fig. 4 shows the visual results obtained for these optimal paperaemters: (LSTM layer number=4, epochs number=400, batch size = 128). It shows (a) train loss, (b) train accuracy, (c) test loss and (d) test accuracy.

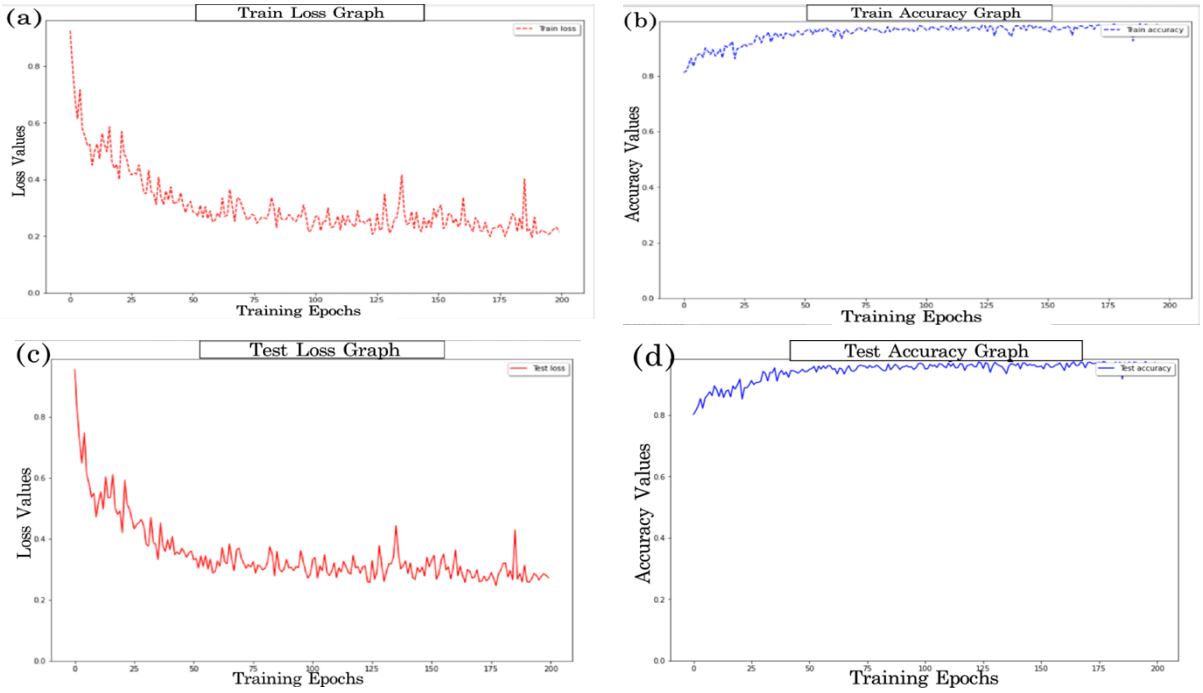


Fig 1. Train, test loss and train, test accuracy for LSTM with 4 layers.

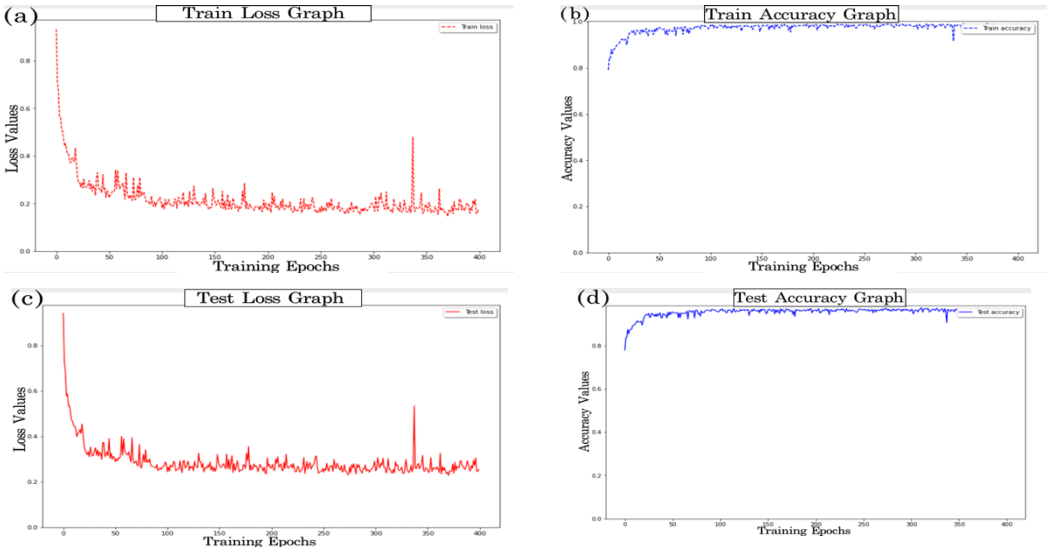


Fig 2. Train, test loss and train, test accuracy for LSTM with number of 400 epochs.

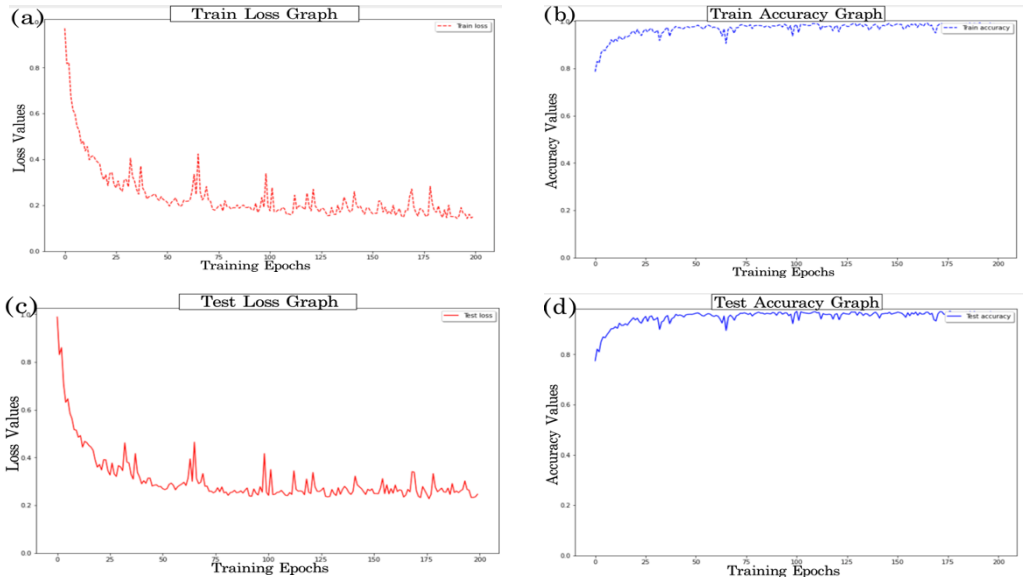


Fig 3. Train loss, train accuracy, test loss and test accuracy graphs of the model with batch size 32.

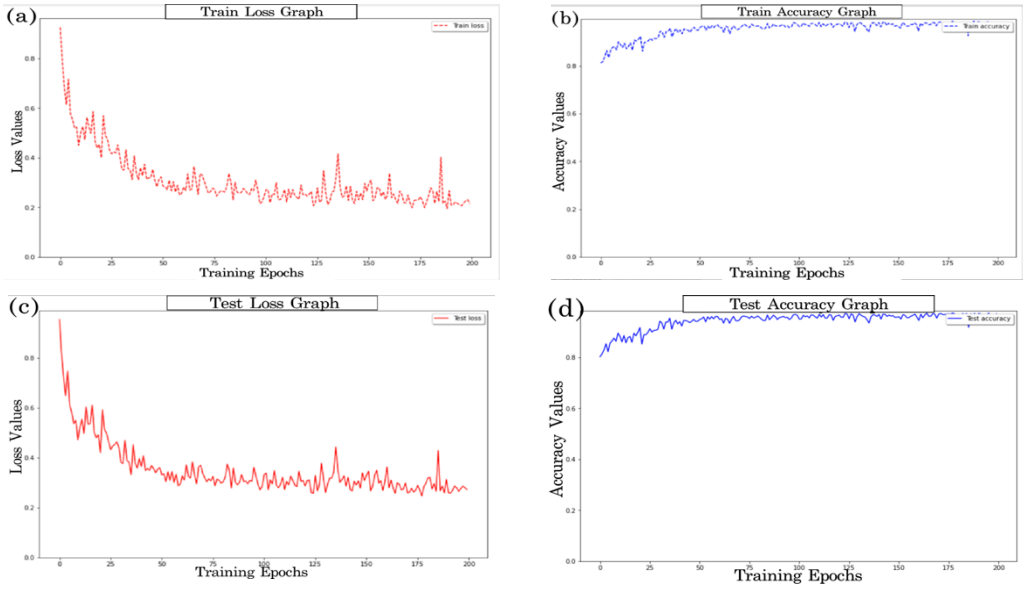


Fig 4. Train loss, train accuracy, test loss and test accuracy graphs of the model with optimum hyperparameters (number of LSTM layers=4, number of epochs=400, batch size=32)

VI. CONCLUSIONS

This study investigated the application of LSTM for HAR. Moreover, the hyperparameters were fine-tuned to obtain optimal results for the model as these parameters affect the accuracy and loss. The hyperparameters which were evaluated include: number of epochs, number of LSTM layers and batch size. LSTM layers were varied between 2-6, batch size in the range 32 – 128 with increments of 32 and the epoch values were tested for 300, 350 and 400. As a result of all hyperparameter tests, the optimum hyperparameter values were determined as LSTM layer number 4, epoch number 400 and batch size 128. And the highest accuracy achieved with these hyperparameters was 97.18%. In future, we will integrate LSTM with Parameter-Less Self-Organizing Map for hyperparameter fine-tuning.

REFERENCES

[1] Xia, K., Huang, J., & Wang, H. (2020). LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access*, *8*, 56855-56866.

[2] Mukherjee, D., Mondal, R., Singh, P. K., Sarkar, R., & Bhattacharjee, D. (2020). EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. *Multimedia Tools and Applications*, *79*(41), 31663-31690.

[3] Agarwal, P., & Alam, M. (2020). A lightweight deep learning model for human activity recognition on edge devices. *Procedia Computer Science*, *167*, 2364-2373.

[4] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[5] Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., & Xie, X. (2016). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. *arXiv preprint arXiv:1603.07772*.

[6] Nweke, H. F., Teh, Y. W., Al-Garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, *105*, 233-261.

[7] Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3-11.

[8] Ronao, C. A., & Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, *59*, 235-244.

[9] Ignatov, A. (2018). Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. *Applied Soft Computing*, *62*, 915-922.

[10] Pienaar, S. W., & Malekian, R. (2019, August). Human activity recognition using LSTM-RNN deep neural network architecture. In *2019 IEEE 2nd Wireless Africa Conference (WAC)* (pp. 1-5). IEEE.

[11] Wang, L., & Liu, R. (2020). Human activity recognition based on wearable sensor using hierarchical deep LSTM networks. *Circuits, Systems, and Signal Processing*, *39*(2), 837-856.

[12] Ullah, M., Ullah, H., Khan, S. D., & Cheikh, F. A. (2019, October). Stacked Lstm Network for Human Activity Recognition Using Smartphone Data. In *2019 8th European Workshop on Visual Information Processing (EUVIP)* (pp. 175-180). IEEE.

[13] Barut, O., Zhou, L., & Luo, Y. (2020). Multi-task LSTM Model For Human Activity Recognition and Intensity Estimation using Wearable Sensor Data. *IEEE Internet of Things Journal*.

[14] Sansano, E., Montoliu, R., & Belmonte Fernández, Ó. (2020). A study of deep neural networks for human activity recognition. *Computational Intelligence*.

[15] Hernández, F., Suárez, L. F., Villamizar, J., & Altuve, M. (2019, April). Human activity recognition on smartphones using a bidirectional lstm network. In *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)* (pp. 1-5). IEEE.

[16] Time series prediction using lstm deep neural networks." [Online]. Available: https://www.altumintelligence.com/articles/a/Time-Series-Prediction-Using-LSTM-Deep-Neural-Networks

[17] "Softmax Activation Function with Python." [Online]. Available: https://machinelearningmastery.com/softmax-activation-function-with-python/

COVID-19 Detection from Chest X-ray Images using CNN

Elif Aşıcı
Department of Computer Engineering
Istanbul Sabahattin Zaim University
İstanbul, Turkey
elif.asici@std.izu.edu.tr

Alaa Ali Hameed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
İstanbul, Turkey
alaa.hameed@izu.edu.tr

Akhtar Jamil
Department of Computer Engineering
Istanbul Sabahattin Zaim University
İstanbul, Turkey
akhtar.jamil@izu.edu.tr

Jawad Rasheed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
İstanbul, Turkey
jawad.rasheed@izu.edu.tr

***Abstract*—After Covid-19 was detected in China in December 2019, it spread rapidly and affected the whole world. The similarity of COVID-19 disease with other lung infections makes it difficult to make an accuracy prediction and diagnosis. In addition, with the high spread rate of COVID-19, the need for a fast system and method for diagnosing cases has increased over time. For this reason, studies in the field of health have increased rapidly to prevent this pandemic disease, and various methods base on artificial intelligence (AI) have been developed to support health practitioners in quick decision making. This study focuses on COVID-19 identification with CNN using X-ray images. Moreover, the proposed method was compared with some recent studies for COVID-19 classification. The result presented are in line with the start of the art method as the proposed method provides a good recognition rate for the detection of Covid-19 .**

Keywords – Covid-19, CNN, Deep Learning, Image Processing, artificial intelligence

I. INTRODUCTION

The new type of coronavirus, which was first detected in China on December 12 and turned into a pandemic in a short time, is a contagious virus that causes respiratory tract infection and can be passed from person to person. It has been confirmed by studies that it is transmitted by contact or airborne respiratory tract [1]. This virus spread rapidly among people and affected the whole world. The disease causes various problems in the human body. According to research, the disease has been found to damage the lungs, brain, liver and many other organs [2]. Biomedical studies are among the computer sciences such as artificial intelligence, deep learning, image processing, and these fields, in this period, apply various methods in the stage of detecting the disease on the bodies of people and investigating the damage caused by the disease on the organs with studies on treatment and diagnosis in medicine [3]. For example, it is possible to see the damage caused by lung disease on radiological images such as x-rays.

Various methods commonly used in computer science, such as deep learning and image processing, have helped make successful biomedical analysis. These methods include many mehods such as SVM [4], CNN and Random Forest [5] and are widely used. In this article, a classification was made according to the features extracted using the deep learning model CNN on Covid-19 patients with chest radiographic images. The aim of the study is to determine the model that gives the best accuracy, recall, precision and f1-score, as well as compare it with the work of other researchers. In this article, CNN was used as the classification method and the accuracy, recall, precision and f1-score values were found to be 95.9, 96, 96 and 96 respectively. Similary, in [6] authors proposed two commonly used classifiers were selected: logistic regression (LR) and CNN for COVID-19 identification from x-ray images. Moreover, principal component analysis (PCA) was applied to reduce the dimensionality and increase the processing speed. A detailed survey of the recent developments in COVID-19 identification using AI techniques can be found in [7].

In this study, it was aimed to make early and accurate diagnosis upon the increasing prevalence of the Covid-19 virus, and for this, a study was carried out on the CNN method, a deep learning model, in order to facilitate studies in the field of health. Examples of current studies using Covid-19 X-ray images have been given, and a situation analysis has been made in relevant areas.

II. RELATED WORK

In this study, a deep learning approach using chest x-ray was developed to be an alternative and supportive to traditional diagnostic tools. The purpose of this deep learning approach with convolutional neural networks (CNN) architecture is to diagnose Covid-19 disease from X-ray images and to make a classification accordingly.

In the first study reviewed [8], a new artificial neural network, Convolutional CapsNet, was proposed for the detection of COVID-19 disease on chest X-ray images with capsule networks. The proposed approach is designed to provide rapid and accurate diagnosis of COVID-19 diseases with dual classification and multi-class classification. The proposed method obtained an accuracy of 97.24% and 84.22% for dual class and multi class, respectively. It has been observed that the study on dual class gives better results. In the study, it was concluded that capsule networks can classify effectively even in a limited data set.

In another study discussed, [9] proposed CoroNet, a Deep Convolutional Neural Network model, to automatically detect COVID-19 infection from chest X-ray images. The proposed model was seen to be based on the pre-trained Xception architecture in the ImageNet dataset, and a new dataset was created by collecting COVID-19 and other chest pneumonia X-ray images from two different public databases, and then end-to-end training was carried out with this dataset. As a result, CoroNet was trained and tested on the prepared dataset and the experimental results showed that the proposed model gave an overall accuracy of 89.6%. When the study was repeated for 3-class classification, it was seen that the accuracy value changed to 95%.

In the last study we reviewed [10], deep learning supported automated detection schemes for COVID-19 and other pneumonia are recommended using a small amount of COVID-19 chest X-rays. CNN-based architecture, called CovXNet, proposes an architecture that uses deep convolution. Because of the high similarity of the X-ray images, many chest X-rays were first used for curving the recommended ones.

In the proposed method, different forms of CovXNets are designed and trained with X-ray images of various resolutions. A stacking algorithm is used for further optimization of the predictions. As a result, different results were obtained in two and multiple classification studies. Accuracy values for two, three and four classes, respectively, were specified as 98.1%, 95.1% and 91.7%.

In these three related studies [8-10], it is clearly seen in Table 2 that each study conducted a classification study on the number of different classes such as two, three, and our study with classification studies on three classes in the sample studies from Table 2. If compared, we have obtained the highest accuracy result with the architecture of the proposed CNN model.

III. METHODOLOGY

A. Deep Learning

Deep learning is an exciting branch of machine learning that is a subfield of artificial intelligence. Deep learning is used to teach machines and systems things that people can do with large amounts of data used in machine learning methods. Some of the most exciting and challenging topics such as the detection of certain objects in the image [11], exploration of the environment in autonomous vehicle systems, natural language processing [12] are among the subjects of interest of deep learning [13] [14].

B. CNN Metod

Convolutional neural networks (CNN), a deep learning technique, is a powerful neural network. Deep Convolutional Neural Network is a special type of Neural Networks that performs exemplarily in various competitions related to computer vision and image processing. Some of CNN's exciting areas of application and study include Image Classification and Segmentation, Object Detection, Video Processing and Speech Recognition. Deep CNN's strong learning ability is primarily due to the use of multiple feature extraction stages that can automatically learn representations from the data. The availability of large amounts of data and advances in hardware technology have accelerated the search in CNN's.

A typical CNN architecture usually includes alternate convolution layers. The arrangement of CNN components plays a fundamental role in designing new architectures and thus achieving improved performance [15].

C) CNN Layer Structure

- ***Convolutional Layer:*** In this layer, which forms the basis of CNN, the transformation process is performed by circulating a certain filter created on all images in the data set. Dividing an image into small blocks helps to extract feature motifs [15].

- ***Pooling Layer:*** It is the layer used to reduce the creep size of the representation, parameters within the network and the number of calculations. In this way, incompatibility in the network is checked. Summarizes similar information in the neighborhood of the recipient area and gives the dominant response in this local area [15].

- ***Activation Function:*** This function serves as a decision function and helps to learn complex patterns. With the selection of an appropriate activation function, the deep web is turned into a nonlinear structure, thus accelerating the learning process [15].

- ***Batch normalization:*** Batch normalization is used to address issues with internal covariance shift in feature maps. Internal covariance shift is a change in the distribution of values of hidden units and slows convergence [15].

- ***Dropout:*** It is one of the most used networking techniques in deep learning. When working with large data sets, memorization may often occur in the model. In order to prevent this, the main goal is to increase the learning of the network by removing some connections or units in the network and to prevent memorization [15].

- ***Fully connected layer:*** The fully connected layer is mostly used at the end of the network for classification. Unlike pooling and convolution, this is a global process. It takes input from the feature extraction stages and analyzes the output of all previous layers globally. As a result, it creates a nonlinear combination of selected features used in data classification [15].

IV. DATA SET

In our experimental analysis, a data set that is accessible to everyone was used and the classification process was applied on three different classes. The entire dataset is X-ray images and each image has been converted to PNG format. Since Covid-19 is a new disease, the number of images related to this virus is limited. This study was conducted using the data set shared by a researcher named Tawsifur Rahman via the Kaggle website [16].

There are three classes in the dataset, namely the Covid class with X-ray images of Covid-19 patients, the Normal class with normal X-ray images, and the Viral class with the Viral Pneumonia X-ray images. In the data set, Covid-19, Normal, Viral classes are 1143, 1341 and 1345 data, respectively, and the total number of data is 3829. In the experimental analysis, 80% of the data set was used as training data and 20% as test data. Some samples for each class are shown in Fig. 1.

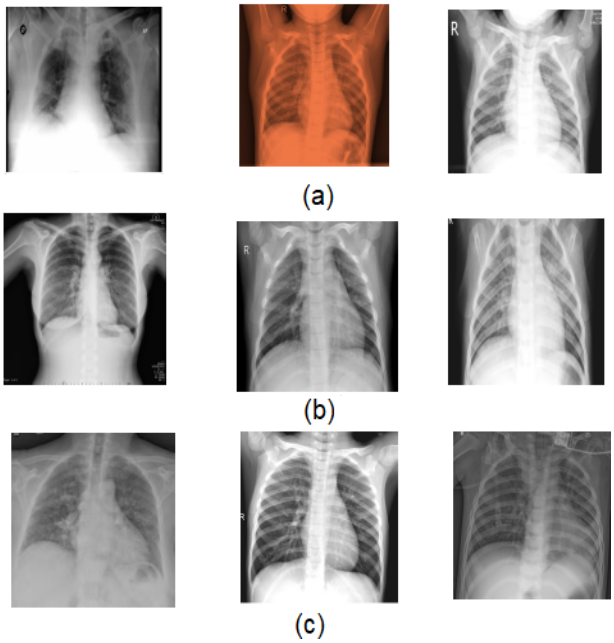


Fig. 1 Sample images used in this study. (a) Covid-19 cases (b) Normal cases (c) Viral Pneumonia cases.

V. Proposed Method

The CNN is used to classify the input images into three different classes as mentioned above. The CNN architecture used in the study is show in Fig. 2. As a processing step, we first resized all images to a fixez size 64x64 so that the model can input the images of same dimension. It also helped process every image with less time.

The layer structure of the CNN model is as follows and the operations started with the filter created in the Convolutional layer with a size of 11x11. Relu activation function is used in for each layer. For the last layer, Softmax activation function was used in the dense fully connected layer.

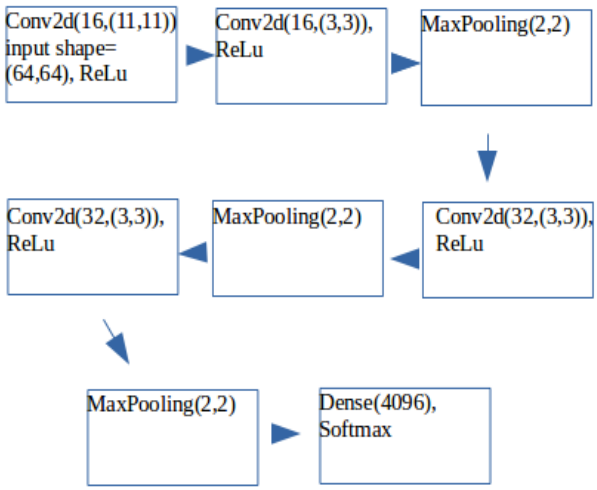


Figure 2 :Architecture of the CNN model used in the study

VI. Experimental Results

A. Experimental setup

For the experimental purposes, the data set was divided into training (80%) and testing (20%) subsets. During the training, hyperparameters were also obtained that produced optimal results for CNN architecture such number of layers and batch size. Moreover, the experiments were run in Matlab © Environment with a processor of 2.7 GHz speed, RAM 16 GB and CPU.

B. Metric Calculation

The proposed method was evaluated using four different metrics: Accuracy (Acc), Precision (P), Recall (R) and F1-score. These metrics are defined as:

Accuracy: It is the value calculated by the ratio of correctly predicted areas in the model to the total data set. Recall: Positive is a metric value that shows how much of the transactions that need to be predicted as Positive. Precision: It is the value showing how many of the values predicted as Positive are actually Positive. F1-Score: It is the value showing the harmonic average of the Precision and Recall values.

$$Accuracy: TN+TP/TP+TN+FN+FP \quad (1)$$

$$Recall: TP/TP+FN \quad (2)$$

$$Precision: TP/TP+FP \quad (3)$$

$$F1\text{-}Score: 2*Precision*Recall / Precision+ Recall \quad (4)$$

C. Evaluation

Three different optimizers tested to see which optimizer performs best on the data set used in this study for three different class identification. The purpose of this is to obtain the result with the best accuracy value. Optimizers used are Adam, Adadelta and SGD optimizers. The overall results obtained for our experiments are summarized in Table I. The best optimizer in the study was Adam optimizer with 95.9% accuracy. The accuracy values of Adadelta and SGD optimizers are 95.4% and 92.1%, respectively.

Accuracy and error graphs of the results obtained from three different optimizers used in this study are shown in Fig. 4 - 8. The accuracy and error charts of Adam optimizer, which is the first optimizer used in the study, are given in Fig. 3 and Fig. 4 respectively. For the Adam optimizer, Accuracy: 95.9, Precision: 96, Recall: 96 and F1-Score: 96.

The accuracy and error graphs of Adadelta optimizer, which is the second optimizer used in the study, are given in Fig. 5 and Fig. 6 respectively. For Adadelta optimizer, Accuracy: 95.4, Precision: 95, Recall: 96 and F1-Score: 95.

The accuracy and error graphs of SGD optimizer, which is the last optimizer used in the study, are given in Fig. 7 and Fig 8, respectively. For SGD optimizer, Accuracy: 92.1, Precision: 92, Recall: 92 and F1-Score: 92.

TABLE I: THE OVERALL RESULTS OBTAINED FOR CNN (%)

Optimizer	Data	Classes	Acc	P	R	F1
Adam	X-ray images	3	**95.9**	96	96	96
Adadelta			95.4	95	96	95
SGD			92.1	92	92	92

*Acc = Accuracy, P = Precision, R = Recall, f1 = F1-score

Figure 3: Adam optimizer accuracy graphs

Figure 4: Adam optimizer error graphs

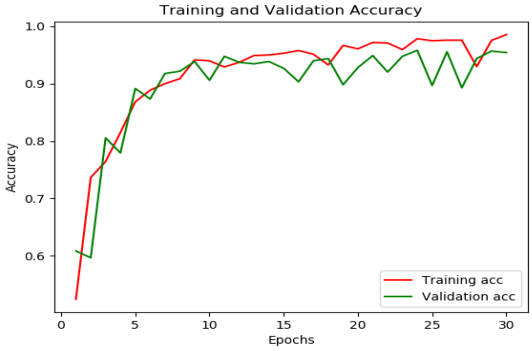


Figure 5: Adadelta optimizer accuracy graphs

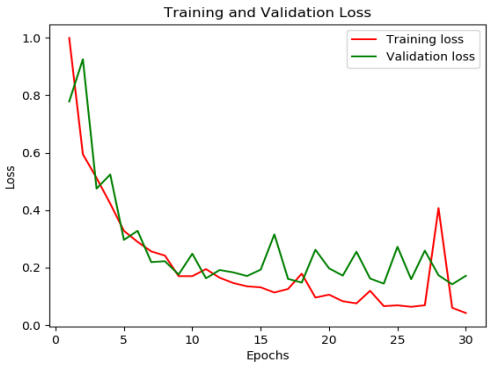


Figure 6: Adadelta optimizer error graphs

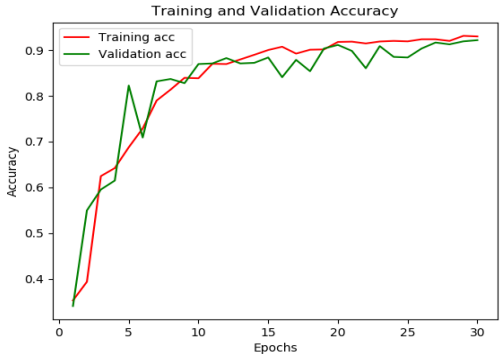


Figure 7: SGD optimizer accuracy graphs

Figure 8: SGD optimizer error graphs

TABLE II :COMPARATIVE ANALYSIS OF PROPOSED METHOD WITH STATE OF THE ART TECHNIQUES FOR COVID-19 CLASSIFICATION FROM X-RAY IMAGES

Ref	Model/Method Type		Data Type	Class	Acc	P	R	F1-score
[8]	CNN-based Deep Convolutional CAPSNET		X-ray images	2	97.23	97.08	-	97.24
				3	84.22	84.61	-	84.21
[9]	Pretrained Xception-based Deep CNN CoroNet		X-ray images	2	99	98.3	99.3	98.5
				3	95	95	96.9	95.6
				4	89.6	90	89.92	89.8
[10]	Transferable multi-receptive features optimizer with Deep CNN-based CovXNet		X-ray images	2	98.1	98	92.83	98.5
				3	95.1	94.9	90.3	95.5
				4	91.7	92.9	89.9	92.6
Recommended	CNN	Adam	X-ray images	3	95.9	96	96	96
		Adadelta			95.4	95	96	95
		SGD			92.1	92	92	92

VII. Conclusion

In this study, we propose the CNN model, which is an important deep learning model for classifying the diagnosis of Covid-19 disease using chest x-rays. Classification processes were carried out on X-ray images of Covid, Normal and Viral classes with the proposed CNN model. The data set used in the study was divided into two as 80% training and 20% test data. During the training, the epoch value was determined as 30. The results obtained in the study have been tested for different optimizers of the CNN method. These are Adam, Adadelta and SGD optimizers and their accuracy values were found to be 95.9, 95.4 and 92.1, respectively. As a result of the comparison of the classification studies on three classes in the reference studies and our study, it is seen that the highest accuracy value was obtained with the architecture of the CNN model, which we recommend. Considering these results, it has been concluded that the subject and the study are worth improving.

REFERENCES

[1] Zaki, A.M., van Boheemen, S., Bestebroer, T.M., Osterhaus, A.D.M.E., Fouchier, R.A.M.: Isolation of a Novel Coronavirus from a Man with Pneumonia in Saudi Arabia. New England Journal of Medicine. 367, 1814–1820 (2012). https://doi.org/10.1056/NEJMoa1211721

[2] Cui, J., Li, F., Shi, Z.L.: Origin and evolution of pathogenic coronaviruses. Nature Reviews Microbiology. 17, 181–192 (2019). https://doi.org/10.1038/s41579-018-0118-9.

[3] Toğaçar, M., Ergen, B., Cömert, Z.: COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X- ray images using fuzzy color and stacking approaches. Computers in Biology and Medicine. 121, (2020). https://doi.org/10.1016/j.compbiomed.2020.103805.

[4] Mei, X., Lee, H.C., Diao, K. yue, Huang, M., Lin, B., Liu, C., Xie, Z., Ma, Y., Robson, P.M., Chung, M., Bernheim, A., Mani, V., Calcagno, C., Li,K., Li, S., Shan, H., Lv, J., Zhao, T., Xia, J., Long, Q., Steinberger, S., Jacobi, A., Deyer, T., Luksza, M., Liu, F., Little, B.P., Fayad, Z.A., Yang, Y.: Artificial intelligence–enabled rapid diagnosis of patients with COVID-19. Nature Medicine. (2020). https://doi.org/10.1038/s41591-020-0931- 3.

[5] Malki, Z., Atlam, E.S., Hassanien, A.E., Dagnew, G., Elhosseini, M.A., Gad, I.: Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches. Chaos, Solitons and Fractals. 138, 110137 (2020). https://doi.org/10.1016/j.chaos.2020.110137.

[6] Rasheed, J., Hameed, A.A., Djeddi, C. et al. A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images. Interdiscip Sci Comput Life Sci (2021).

[7] Jawad Rasheed, Akhtar Jamil, Alaa Ali Hameed, Usman Aftab, Javaria Aftab, Syed Attique Shah, Dirk Draheim, A survey on artificial intelligence approaches in supporting frontline workers and decision makers for the COVID-19 pandemic, Chaos, Solitons & Fractals, Volume 141, 2020.

[8] Toraman, S., Alakus, T.B., Turkoglu, I.: Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. Chaos, Solitons & Fractals. 140, 110122 (2020). *https://doi.org/10.1016/j.chaos.2020.110122*

[9] *Khan, A.I., Shah, J.L., Bhat, M.M. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. Computer Methods and Programs in Biomedicine. 196, 105581 (2020). https://doi.org/10.1016/j.cmpb.2020.105581.*

[10] *Mahmud, T., Rahman, M.A., Fattah, S.A.: CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. Computers in Biology and Medicine. 122, 103869 (2020). https://doi.org/10.1016/j.compbiomed.2020.103869.*

[11] Han, J., Zhang, D., Cheng, G., Liu, N., Xu, D.: Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey. IEEE Signal Processing Magazine, Volume 35, Issue 1, Pages 84-100 (2018). https://doi.org/10.1109/MSP.2017.2749125

[12] Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent Trends in Deep Learning Based Natural Language Processing. IEEE Computational Intelligence Magazine, Volume 13, Issue 3, Pages 55-75 (2018). https://doi.org/10.1109/MCI.2018.2840738

[13] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV), Volume 115, Issue 3, Pages 211-252 (2015). https://doi.org/10.1007/s11263-015-0816-y

[14] Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Confer- ence on Neural Information Processing Systems, Volume 1, Pages 1097-1105 (2012). https://doi.org/10.1145/3065386

[15] Khan, A., Sohail, A., Zahoora, U. and Qureshi, A., 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), pp.5455-5516.

[16] Kaggle. COVID-19 Radiography Database Kaggle 2020 https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

Electricity Loss and Fraud Prediction with Deep Learning

Orçun Kitapcı
Department of Computer Engineering
İstanbul Sebahattin Zaim University
Istanbul, Turkey
orcun.kitapci@dedas.com.tr

Alaa Ali Hameed
Department of Computer Engineering
İstanbul Sebahattin Zaim University
Istanbul, Turkey
alaa.hameed@izu.edu.tr

Akhtar Jamil
Department of Computer Engineering
İstanbul Sebahattin Zaim University
Istanbul, Turkey
akhtar.jamil@izu.edu.tr

***Abstract*—In developing countries, energy usage has increased with remaining population, industry, widespread technology and the increasing trend of economy. The main energy source of this increase is electricity. From this perspective, the forecasting of electricity fraud has an important role in the control of this trend to support, run, plan of distribution network's investment. High percentage of fraud in this region damages both the region economy growth and also electricity distribution network. The main source of Fraud usage comes from industry so fraud detection is very hard. So with the correct analysis of daily usage, the usage before theft and last usage of electricity which retrieved from Automatic Meter Reading System (AMRS), we can forecast future theft with Deep Learning. If we use more than one method so we can decide to use one that gives us the best proven.**

Keywords—Electricity Distribution, Lost and Fraud Prediction, Deep Learning, Artificial Intelligence

I. INTRODUCTION

Both electricity loss and fraud may be technical or non-technical. Electricity fraud is generally a non-technical type of fraud, which can be performed in several ways such as intervention in the meter, external intervention to the electricity line, non-payment of invoices, and invoice irregularities [1]. This study will focus on illegal electricity, intervention to the meters or external interventions to the electricity line. With field operation, it is very hard to identify such frauds. So data of the AMRS's which consists of indexes of usage are analyzed with the models of deep learning to predict high possibility theft usage on the power line.

Electricity distribution consists of an interconnected network that distributes electricity at large scale. In some cases, the distribution network spans over many countries. Even a small interruption or outage in the distribution network can cause a disruption at large scale across countries [2]. A simple disruption in a region with high electricity loss and fraud can actually affect the whole system in country and its neighbours. Therefore, it is important to identify the electrical loss and fraud as high outage of electricity effects both production industries and public institutions such as hospitals, schools, sewage treatment, plants.

The literature indicates that some researchers have already work war prediction and identification of electricity theft and loss. Some of them based on the electricity usage data on low tension electric installation [3]. Irregularity detection on low tension electric installations uses ensemble model of neural networks to increase the level of accuracy to identify irregularities. The data retrieved from usage of Light S.A. Company, the Rio de Janeiro in Brazil. The proposal consists of two modules part; one of data mining to obtain the correct subscribers and the other for classification of the subscriber. Each module has 5 neural networks which each of classifies output that indicates whether there is fraud or not. Low tension electricity usages don't decisive to reduce the rate of fraud regionally, however it reduces individual fraud.

On the other hand, the other proposal implements a neural network as previous one but additionally suggest hierarchical model for classification of customers with library of support vector machines (SVM) which is required properly parameters and a training function [4]. When evaluated in terms of feasibility, normally selection of the parameters is very hard because it takes a lot of time for evaluation of instant consumption. With the NN it could be facilitated to predict classification.

Some researchers have proposed methods based on supervised machine learning. For instance, in [5] both gas and electricity consumption data was used. Learning is improved controlling the accuracy with campaign feedback and newly detected technics of fraud usage. The data has no regional characteristics which are not localized and at the same geographical climates also retrieved both gas and electricity consumption. So the learning model could not be used for local consumption and fraud detection.

Since obtaining data on large scale is difficult, therefore the data analysis were performed on a regional basis. Table 1 includes the top 7 distribution companies that are exposed to the most illegal use by distribution region and their fraud rates [6]. Dicle Electricity Distribution Region consists of Diyarbakır, Şanlıurfa, Mardin, Siirt and Şırnak provinces. The region covers most of the Southeastern Anatolian provinces that are developing compared to the west of the country. It is a gateway to countries such as Iraq, Iran, Saudi Arabia and Egypt. Therefore, it has an important position in terms of energy and information transfer to these regions. The Loss and Fraud rate in the region is 51% as of 2020. With this rate, it is seen that there is a long way to go [7].

In addition, the region's other important parameters of energy usage are constantly increasing population and GDP. Population growth has reached 6.5 million from 5 million in 2010 with an annual average increase of 2.5%, and the GDP growth has approached from 36 million TL to 120 million TL with an annual average increase of 19% in 2010 [7].

The development of the region, being a gateway to developing countries, includes important opportunities with its constantly increasing population and GDP. In this respect, loss and fraud prediction is important both to support this development and to provide projection to infrastructure and superstructure works, and to increase the profitability of the company. This will ultimately help the industry to grow and improve the overall economic stability of the country by providing input to economic activities of country.

TABLE 1 DISTRIBUTION REGION LOSS AND FRAUD RATES (%) **[6]**

Distribution Company	Target year 2013	Realized year 2013	Target period (2015)	Realized period (2015)
Dicle Elektrik	71,07	75,41	49,03	-26,38
Vangölü	52,1	65,84	35,94	-29,9
Aras	25,7	27,58	17,73	-9,85
Toroslar	11,8	15,24	10,72	-4,52
Yeşilırmak	9,41	11,47	8,78	-2,69
Akdeniz	8,05	11,32	8,02	-3,3
Boğaziçi	10,76	9,89	9,78	-0,11

In this paper, we performed a detailed comparison of deep learning and traditional machine learning methods for electricity theft detection. With ANN, Random Trees, Logistic Regression, and Linear Support Vector Machines to predict most accurately after classification using training data which is real and on a regional basis. The methodology was validated and its result are detailed in the following section.

The rest of paper consists of five more sections. Section-I explains the methodology, section II explains describes the dataset. The following section IV and V present proposal method experimental results respectively. Finally, the paper is completed with conclusion.

II. METHODOLOGY

There are generally two types of predictions: qualitative and quantitative. Qualitative estimation is mostly based on the interpretation of experts who have experience in the field. Quantitative estimation is based on accepted and proven mathematical functions. Therefore, in this article, a comparison has been made using the quantitative estimation method such as ANN (Artificial Neural Network, Artificial Neural Networks), Random Forest, Random Trees, Logistic Regression, Linear Support Vector Machines. Following sections describe the algorithms used in this study.

A. Artificial Neural Network

Artificial neural networks are a computing technology inspired by the information processing technique of the human brain. The operation of the simple biological nervous system is imitated with ANN. In other words, it is digital modeling of biological neuron cells and the synaptic connection between these cells. Neural Network is a structure established in layers. The first layer is called input and the last layer is called output. The layers in the middle are called "Hidden Layers". Each layer contains a certain number of "Neurons". These neurons are linked to each other by "Synapse". Synapses contain a weigh. These weighs indicate how important the information in the neuron to which they are attached.

Consider that x_0 is an input value and the weight in dendrite (w_0) are multiplied, ($x_0 . w_0$) is transmitted to the nerve cell and this multiplication is done in the nerve cell. After all input and weigh multiplied, all these results are summed. In other words, weighted addition is done. Then, after being summed with a bias (b), the activation function is then transferred to the output. This output can be the final output or the input of another cell. Mathematically, weights and inputs are multiplied finally bias is added. Thus, a simple mathematical model is obtained.

B. Random Forest

Random Forest is a highly effective algorithm developed by Breiman (2001). Random Forest is a collective learning algorithm consisting of many individual training data. Random selection is used with generate random sets to create Random Forest Setup. While in standard trees each node is branched using the best split among all variables, in the Random Forest each node is branched using the best among randomly selected subsets of prediction at that node [8]

C. Random Trees

Random Tree is a classification algorithm that generates a tree by taking randomly selected features on a certain number of nodes in each node. There is no pruning and there is an option that allows prediction of class probabilities based on the data set held [9].

D. Linear Support Vector Machine (LSVM)

Support vector machines are mostly used to separate binary classification data, for example separating each data in a data set into female or male. On the other hand, the data can sometimes belong to more than two classes. In such cases, the basic SVM algorithm becomes dysfunctional. For example, the classification of a data set where certain characteristics of dogs of different breeds are kept based on these characteristics [10]. SVM is based on the principle of inherent risk minimization. SVM can be analysed theoretically using concepts in computational learning theory and can achieve good performance in real world problems. Support vector machines are supervised learning models that select a small number of critical boundary samples called support vectors from each class and create a linear discriminant function that separates them as much as possible [11].

III. DATASET

A. Training Data

As a requirement of deep learning, we will first try to obtain training data. Firstly, training data was obtained by using the data obtained from Automatic Meter Reading

System (AMRS) and enriching the data obtained in SYS (Field Management System). The subscriber information and the date when the fraud was detected was obtained from the SYS system. On the other hand, from the AMRS system, the usage before the date of fraud detection (BF), the usage after the date of fraud detection (AF), the date of the prediction (SYS), the indication that shows it is fraud or not are retrieved for four weeks. Special information such as the Subscriber Number in the data provided here has been changed because it is private as summarized in Table 1. In order to understand the increase between these usages, the rates of this information on the basis of rows are used as input parameters. There is a total of 2,993 rows of training data. Since the data here are training data, there is no subscriber information. The second important column here is the Target column. The training

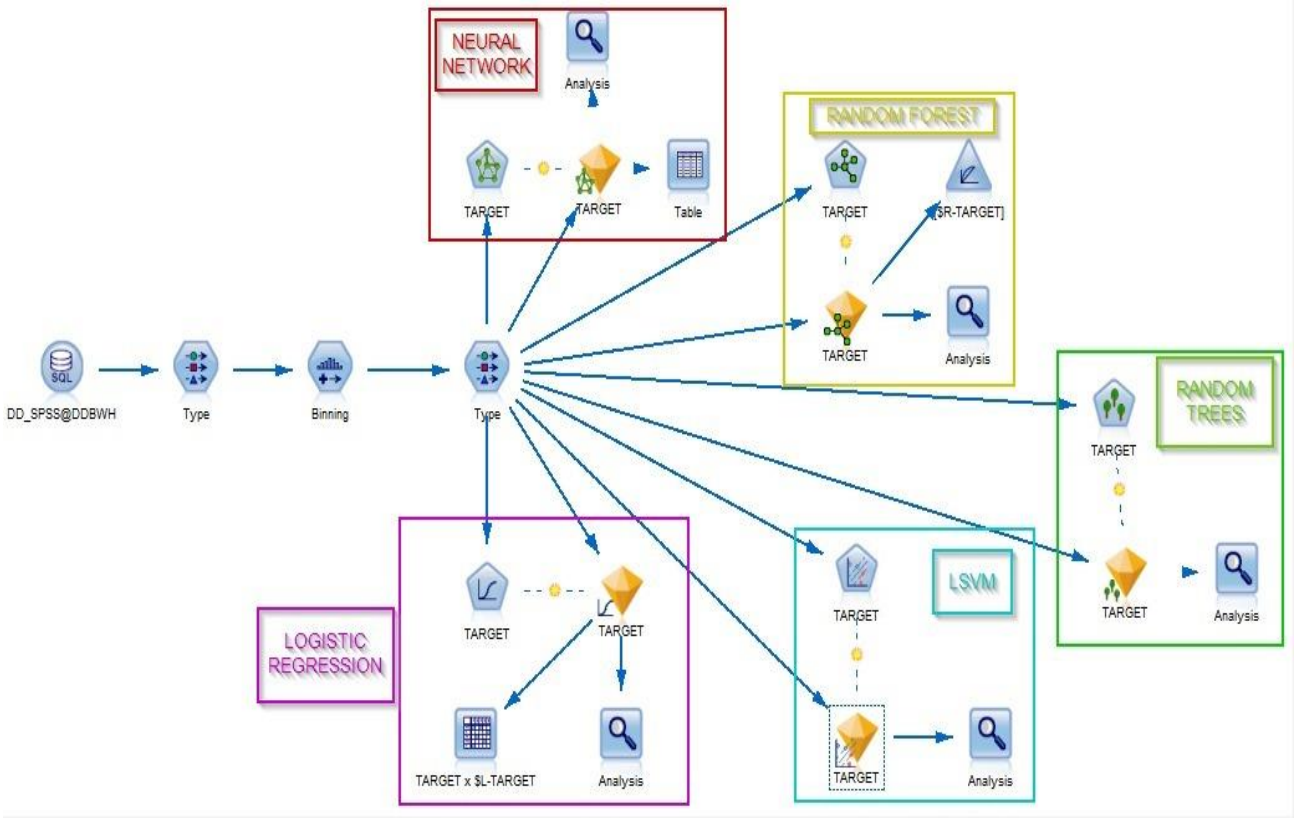


FIGURE 1 PREDICTION MODEL

TABLE 2 SAMPLE CONSUMPTION DATA

No	BF_1	BF_2	BF_3	BF_4	AF_1	AF_2	AF_3	AF_4	SYS_1	SYS_2	SYS_3	SYS_4
1	20,32	20,66	28,07	29,04	21,53	30,53	30,62	38,88	42,07	40,56	37,93	42,93
2	60,9	55,87	58,5	42,62	259,1	170	275,4	269	40,87	25,25	23,65	16,77
3	34,76	34,95	35,29	37,26	36,1	35,73	40,09	45,42	91,74	78,53	77,47	67,41
4	163	165,2	144,7	176,8	72,91	30,28	34,33	33,94	30,03	41,71	29,97	32,2
5	721,3	700,9	962,4	802,9	1322	769,9	620,4	699,3	0	1709	1720	1326

TABLE 3 SAMPLE TRAINING DATA

No	BF_AF_RATE_1	SYS_BF_RATE_1	BF_AF_RATE_2	SYS_BF_RATE_2	BF_AF_RATE_3	SYS_BF_RATE_3	BF_AF_RATE_4	SYS_BF_RATE_4	TARGET
1	0,94	2,07	0,68	1,96	0,92	1,35	0,75	1,48	1
2	0,24	0,67	0,33	0,45	0,21	0,4	0,16	0,39	1
3	0,96	2,64	0,98	2,25	0,88	2,2	0,82	1,81	0
4	2,24	0,18	5,45	0,25	4,22	0,21	5,21	0,18	1
5	0,55	0	0,91	2,44	1,55	1,79	1,15	1,65	0

TABLE 4 SAMPLE TEST DATA

SUBSCRIBER_ID	BF_AF_RATE_1	SYS_BF_RATE_1	BF_AF_RATE_2	SYS_BF_RATE_2	BF_AF_RATE_3	SYS_BF_RATE_3	BF_AF_RATE_4	SYS_BF_RATE_4
10000000	1,16	1,10	2,03	0,94	1,30	0,79	1,54	0,70
11111111	0,37	0,75	0,55	0,85	0,46	0,78	0,89	0,88
22222222	1,00	4,92	0,48	2,96	0,29	5,75	0,26	4,14
33333333	1,15	2,76	0,20	1,78	0,20	2,38	0,19	1,77
44444444	1,93	3,73	0,27	5,43	0,17	5,67	0,20	5,39
55555555	0,92	1,55	0,68	2,53	0,58	1,33	0,65	1,03
66666666	0,82	1,05	0,33	1,69	0,53	1,51	2,37	0,89
77777777	1,20	0,69	0,17	6,26	0,35	3,91	0,83	1,10
77777788	1,00	5,11	0,90	3,21	0,96	2,07	1,18	1,50
99999999	0,00		0,00		4,11	0,01	0,78	0,00

data summarized in Table 3 retrieved also from the AMRS system, including the Target Value information, training data also doesn't contain the subscriber value. The ratio is obtained by dividing each column into one in weekly basis, the columns are summarized in Table 2. If any rate divergent to 1, it indicates that there is probably fraud usage. Because if there are common usages, the consumptions are close to each other so dividing convergent to 1.

TARGET as a boolean field;

$$TARGET = \begin{cases} 1, & There\ is\ Fraud \\ 0, & There\ is\ No\ Fraud \end{cases} \quad (1)$$

B. Prediction Test Data

The data retrieved from the AMRS system, including the subscriber information summarized in Table 4. In this data there is not target column. Each column has its own rate of before usage of fraud, after usage of fraud or consumption on the date of test into one in weekly basis. Each column summarized in Table 4 have similarities with the training data of the table summarized in Table 3 except Target Value and Subscriber ID. Using this dataset in Table 4 prediction will be made and thus the illegal usage of the subscriber will be predicted. The total number of test data is 3.715.

BF/AS	:Before/After Fraud (Pre-Fraud/Post-Fraud)
SYS	:Fraud on System Date
BF_AF	:Ratio of Pre-Fraud to Post-Fraud
SYS_AF	:Ratio of System Date Fraud to Post-Fraud
SYS_BF	:Ratio of System Date Fraud to Pre-Fraud

IV. Proposed Model

In order to detect fraud, we need weekly usage data obtained from AMRS system and fraud data obtained from SYS system to determine whether there is fraud or not. We took pre-fraud and post-fraud data, as well as the ratios between the current usage, and we will get how the usage pattern has changed in these time periods proportionally and linearly. If the use is not fraud, it is expected that the rates before or after the fraud will be convergent to 1, that means, the usage will not change much *(1)*. If there is a fraud, it is expected that there will be no big changes in these ratios and the linear distribution of these rates. The target (2) and (3) equations in which the fraud occurred from the SYS System will enable us to evaluate the realization of these rates and to turn this into a training data. We will be able to obtain the success rate by giving the training data in Table 2 as the input parameter to the Deep Learning methodologies, and then comparing the actual value with the predicted value after the prediction. We will make our prediction by developing 2 projects in SPSS Software. One of them will be our modelling, the other will be the project where our prediction is made with this modelling and the results are obtained.

$$BF_n_AF_n = \frac{BF_n}{AF_n} \quad (2)$$

$$SYS_n_AF_n = \frac{SYS_n}{AF_n} \quad (3)$$

A. Modelling

The training data in Table 2 is used as input for our modelling. We classify the rates in our input we use as. We will divide each of our proportions into 36 parts. What is important here is that if our success rate is low, we will increase our success rate by taking our classification lower or higher. Also in here some SPSS Models are used to create prediction model and binning values. After the model are created, it is compiled retrieving data which is already prepared with ETL process on the database preparing a cube and create some binning values to create prediction model.

B. Prediction

After model is ready we can make our prediction on the test data in Table 3 with the classification algorithm we have obtained through modelling. Classified binning values are used to create prediction mode. Each of prediction model of ANN, Random Forest etc. which are created before on the stage of Modelling are used individually to calculate its prediction value of fraud. Also predicted values are viewed, analyzed and confusion matrix calculated by SPSS tools.

V. Experimental Results

Experiments were performed with five different models on our dataset to evaluate their performance. The detailed result of sample data obtained from ANN Model are summarized in Table 5. Because the test data for the other models are similar to Table 5, the test data aren't given for all. In the Test Data, as given in training table, there are rate of the pre-fraud to post-fraud usages, rate of pre-fraud to system date usages, rate of the post-fraud to system date usages for 4 weeks for each. All rows (which means all subscriber in the table) has high probability of fraud usage because the column of Target divergent to 1. This subscriber should be controlled with the field operation to be sure whether there is fraud usage or not. With this operation the test data can be examined. Also comparative table of results that indicates the accuracies are summarized in Table 6 As it can be seen from table, almost all model has the same accuracy except the Random Trees. Accuracy rate are mostly %83. The important amount here is fraud usage prediction. Most of all are very small amount of fraud usage. The main target is to find the fraud usage. So the amount of fraud should be increased. The percentage of accuracy for Random Tree is very low. Because it is a classification algorithm that creates a tree by taking randomly selected features in a certain number of nodes in each node. Our learning and test data are not categorical data, they do not have a tree-pattern structure. Therefore, the accuracy percentage is very low for the Random Tree.

As it can be seen from Table 6, we can evaluate the probability of illegal usage of the relevant subscriber on the basis of Subscriber ID as a percentage in the Target column on the data we predict, and concentrate on the fraud controls of these subscribers. When the weekly consumption chart of these subscribers is examined, the rates of BF_AF and SYS_BF are far away from 1. If these rates can be more less than 1 or greater than 1. This means usage of electricity are not regular. So the irregularity of subscribers' usage can be understood looking at the rate.

VI. Conclusion

In this paper, we exploited different types of machine learning techniques for identification of theft/loss of electricity in various regions in Turkey. The obtained results indicate that machine learning methods were effective. In future, we would like to extend our method on more regions and apply deep learning methods for higher accuracy.

TABLE 5 PREDICTION RESULTS OBTAINED FROM TEST DATA

SUBSCRIBER_ID	BF_AF_RATE_1	SYS_BF_RATE_1	BF_AF_RATE_2	SYS_BF_RATE_2	BF_AF_RATE_3	SYS_BF_RATE_3	BF_AF_RATE_4	SYS_BF_RATE_4	TARGET
222222	1,56	0,00	70,41	0,00	931,80	0,00	1,87	0,00	0,80
333333	3,79	0,04	103,37	0,00	10,08	0,00	2,56	0,00	0,80
444444	1,26	2,33	89,69	0,03	6,30	0,35	2,01	1,00	0,80
555555	0,67	0,02	91,33	0,02	0,93	0,02	197,45	0,01	0,80
666666	0,02	37,38	104,62	0,63	0,52	0,05	0,03	1,48	0,80

TABLE 6 CONFUSION MATRIX OF PREDICTION FOR EACH MODEL (%)

	ANN	
	Predicted No Fraud	Predicted Fraud Exists
Actual No Fraud	2484	3
Actual Fraud Exists	502	3
$Accuracy = \frac{(TP + TN)}{Total}$	83%	

	Random Forest	
	Predicted No Fraud	Predicted Fraud Exists
Actual No Fraud	2484	3
Actual Fraud Exists	497	8
$Accuracy = \frac{(TP + TN)}{Total}$	83%	

	Random Trees	
	Predicted No Fraud	Predicted Fraud Exists
Actual No Fraud	1707	780
Actual Fraud Exists	149	356
$Accuracy = \frac{(TP + TN)}{Total}$	69%	

	Logistic Regression	
	Predicted No Fraud	Predicted Fraud Exists
Actual No Fraud	2484	3
Actual Fraud Exists	497	8
$Accuracy = \frac{(TP + TN)}{Total}$	83%	

	LSVM	
	Predicted No Fraud	Predicted Fraud Exists
Actual No Fraud	2487	0
Actual Fraud Exists	503	2
$Accuracy = \frac{(TP + TN)}{Total}$	83%	

VII. REFERENCES

[1] B. S. Thomas, "Electricity theft: a comparative analysis," *ELSEVIER - ENERGY POLICY*, p. 1, 2013.

[2] Ö. TUTTOKMAĞI and A. KAYGUSUZ, "Büyük Ölçekli Elektrik Kesintilerinin İncelenmesi," *BEU Journal of Science*, no. İnönü Üniversitesi, Elektrik-Elektronik Mühendisliği, Malatya, 2019.

[3] C. Muniz, K. Figueiredo, M. Vellasco, G. Chavez and M. Pacheco, "Irregularity Detection on Low Tension Electric Installations by Neural Network Ensembles," Vols. June 14-19, no. Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, 2009.

[4] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni and P. Nelapati, "A hybrid neural network model and encoding technique for enhanced classification of energy consumption data," no. IEEE Power and Energy Society General Meeting, 2011.

[5] B. Coma-Puig, J. Carmona, R. Gavald`a, S. Alcoverro and V. Martin, "Fraud Detection in Energy Consumption: A Supervised Approach," *IEEE International Conference on Data Science and Advanced Analytics*, 2016.

[6] "Kaçak Elektrik ile Mücadele Üzerine Bir Değerlendirme," 2013. [Online]. Available: https://www.pwc.com.tr/tr/sektorler/enerji-altyapi-madencilik/enerji-spotlights/kacak-elektrik-ile-mucadele-uzerine.html.

[7] Türkiye İstatistik Kurumu, "İstatistik Göstergeler," 15 Mayıs 2019. [Online].

[8] S. Kalmegh, "Analysis of WEKA Data Mining Algorithm REPTree, Simple Cart and RandomTree for Classification of Indian News," *International Journal of Innovative Science*, pp. 438-446, 2015.

[9] E. Akçetin and U. Çelik, "İstenmeyen Elektronik Posta (Spam) Tespitinde Karar Ağacı Algoritmalarının Performans Kıyaslaması," *Internet Applications & Management*, 2014.

[10] M. R. Ogiela and L. C. Jain, Computational intelligence paradigms in advanced pattern classification, Berlin: Springer, 2012.

[11] I. H. Witten, E. Frank and M. A. Hall, "Data Mining: Practical," *Machine Learning Tools and Techniques*, 2011.

RADIOLOGICAL MEDICAL REPORTS CLASSIFICATION

Alda Kika
Department of Informatics
Faculty of Natural Sciences, University of Tirana
Tirana, Albania
alda.kika@fshn.edu.al

Suela Maxhelaku
Department of Informatics
Faculty of Natural Sciences, University of Tirana
Tirana, Albania
suela.maxhelaku@fshn.edu.al

***Abstract*— The data gathered from the information system of Neuroradiology, Department of Neurosciences in "Mother Teresa" University Hospital Center in Tirana, Albania is composed of text and scan images. The textual information is very important as it not only represents the data about the patient but also the radiological interpretation of the images can help the physicians in medical diagnosis and patient treatment. Text classification would help further in analyzing the data and automatic classification into classes of interest. A dataset of 1313 radiology reports classified by radiologists in 6 categories is used in this study to analyze and compare two types of classifiers: the Naïve Bayes and character level n-gram classifier trained using sequences of three, four, five and six characters. Experimental results show that the most successful classifier is the Naïve Bayes classifier with best efficiency and accuracy reaching 98%.**

Keywords—text classification, Naïve Bayes, character n-gram.

INTRODUCTION

Healthcare Industry produces a vast amount of data. The data that comes from different resources needs to be organized, transformed and processed in order to get information and help in decision making process. To better identify diagnosis, treatment and characteristics of the patient it is important to analyze the healthcare data.

Neuroradiology Service at Department of Neurosciences, University Hospital Center "Mother Teresa" in Tirana, Albania gathers data about patient, radiologist reports and scan images in DICOM (Digital Imaging and Communications in Medicine) format from CT Scanner or Magnetic Resonance. In [1] we proposed a Single Page Application (SPA) that will integrate the patient's data, report from radiologists and the DICOM scan images into a Radiology Information System. Although this system fulfills one of the main challenges that faces medical systems such as integration of many different data into one system a lot of work should be done towards analysing, searching and finding hidden knowledge from the data.

Precision medicine that takes individual variability into account encourages computational tools for analysing large sets of data [2]. Automatic text classification of the medical documents, known as text categorization is one of the fundamental tasks with broad applications. It offers the possibility to assign a text into a set of predefined classes or categories [3]. It has a crucial role in healthcare, in predicting diseases and also in analyzing the content of the radiology reports.

Electronic health records are growing tremendously in medical centers, hospital, etc. Processing clinical free-text reports have become a considerable challenge in order to improve clinical research and patient care [4]. The data that is found in medical documents presents several challenges for classification such as it is unstructured and may be produced by several radiologists which have different way of writing. These reports integrate the patient data and interpretation of the radiology scan images. Although the documents in each category have differences they have a lots of keywords and phrases in common which will help in their classification.

Finding the model that will be used for text classification includes different stages from data gathering, data selection, data transformation, choosing the classifier, training of the model and its evaluation. There are different classifiers for classification of text into categories such as Naïve Bayes, Decision Trees, K-Nearest Neighbors, Support Vector Machine, etc.

In this paper, a dataset that is labeled into 6 categories by radiologists is used to develop a classification model. This model can be used to classify a larger dataset of unclassified medical reports in the future. The classifiers that will be trained and evaluated from these data will be Naïve Bayes and character level n-gram with three, four, five and six grams.

The rest of this paper is organized as follows: in section 2, the related work is presented, in section 3 the algorithms that are used to build the classification models are described followed by section 4 which will present the experiments and results. and concluded by section 5, in which the conclusions of this research are drawn, and the future work is presented.

RELATED WORK

The most important part of the information of the patient at the radiology reports is free text and most of them remain as unstructured text format [5, 6] so automated extracting and analyzing this information is very crucial in healthcare. Reviewing and collecting this data is a very time-consuming task. In this section, we have concluded some research work that has been done using the text classifiers in discovering medical knowledge from electronic medical records (EMR) and especially from different types of radiology reports.

Y. Shen et al. proposed a method for extracting data and analyzing 30,000 electronic medical record (EMR) information. This method was efficient in calculating the possibility that a given patient will suffer from a specific disease and was developed using ontology Naïve Bayes [7].

Naïve Bayes classifier is used also from M. Benndorf et al. [8] in developing a decision support tool on mammographic mass lesions focusing on two predictive variables that are the mass lesion morphology and the age of the patient. The classifier is used to relate combinations of BI-RADS (Breast Imaging, Reporting and Data System) descriptors and patients' age to categories of risk. This probabilistic classifier shows similar performance related to the clinical performance. The figure 1 represent Naïve Bayes classifier where "cancer" serve as the disease status and his relation to the predictive variables of the BI-RADS descriptors and age [8].

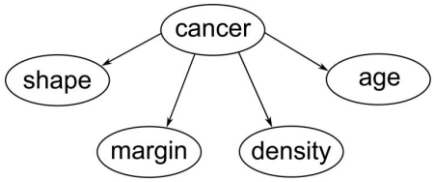


Fig. 1 Naïve Bayes Classifier [8]

In [9] the researchers have evaluated radiological reports using supervised and unsupervised language modelling in Chest X-Ray and they have compared performance of testing sets using different classifiers like Naïve Bayes, Logistic Regression KNN, SVM, Random Forest, BERT etc.

I. Krsnik et al.[10] consider the probability of extracting information automatically from 1295 narrative radiology reports. They have compared two methods of text classification: traditional classification models (Naïve Bayes, Logistic Regression, Support Vector Machine and Random Forests with Bag-of-Words features) with Convolutional Neural Network method together with dense word vectors as input features [10].

M. Pandey et. al. studied collected data from 2008-2018 of 122,025 Computed Tomography (CT) reports with a total of 11,808 patients suffering from heart failure. The approach demonstrated excellent accuracy. First of all, they trained the classifier by converting the output of the filter module into word embedding using TF-IDF (Term frequency - inverse document frequency). After that, the researchers trained the classifier with Naïve Bayes and SVM classifiers [11].

The researchers in [12] have created a machine learning model that automates the process of predicting and finding the most important sentences in clinical reports. This model has a very good performance in medical case reports.

Bucher et al.[13] have studied 4716 patients. They compared the performance of the machine learning n-gram method and structured data that were collected from the e-health records and found that Natural Language Processing engine had excellent performance on document-level and also on patient-level [13].

The platform that is developed by Wansing et al. [14] offers the possibility to upload bio datasets, share them and analyze different data using different methods like Naïve Bayes classifier, Multiclass Support Vector Machine, Support Vector Machine (SVM), k-nearest neighbor classifier, Decision trees, and Discriminant analysis classifier [14].

As we can see in the healthcare industry, the text classification into categories is the most widely used method for detection, forecasting and optimization [15].

RESEARCH METHODS

Naïve Bayes is chosen as one of the classifiers that will be used on our documents to train a model because it is very simple and offers a good performance in classification of texts and documents [16].

Naïve Bayes classification is used in classifying various models according to their features. It is based on a supervised learning algorithm that can do simultaneous multi-class predictions. The used data set classification is already known. It will be used to train and evaluate the model. Naïve Bayes classifier is one of the most efficient and simplest that can be useful for predicting different diseases and sometimes having a better performance than other methods in order to help radiologists, the doctors etc. in the process of decision-making [17].

This model allows each attribute to contribute equally and independently.

Let d_i denote a feature in the vector of document D<i>, i= 1,2, ,n and C{c1,c2,…,ck} is the set of predefined classes. The classifier will assign a label c_j from C to the document.
Given the above data, the vector of attributes of the document and classes, by using Bayes' theorem we state our problem with the following formula:

$$P(cj|D) = \frac{(P(cj)P(D|cj))}{P(D)} \quad (1)$$

where cj from the set C is the class variable and D is dependent feature vector with dimension n. P(D|cj) is the distribution probability of document D in classes.
Naïve Bayes classifier will appoint the document to the class with highest probability:

$$NB(D)= \text{argmax}(j,P(cj|D)) \quad (2)$$

where argmax is a function that finds the argument j that gives the maximum value from the function P(cj|D).

Assuming the naive conditional independence assumption that all features of D are mutually independent, conditional on category c:

$$P(D|cj)=\prod_{i=1}^{n} P(di|cj) \quad (3)$$

Equation (1) when we substitute (3) becomes:

$$P(cj|D) = \frac{(P(cj)\prod_{i=1}^{n} P(di|cj)}{P(D)} \quad (4)$$

P(D) is identical to each class cj, therefore equation 2 becomes:

$$NB(D) = \text{argmax}(j, P(cj) \prod_{i=1}^{n} P(di|cj)) \quad (5)$$

The features of the documents are character sequences converted to bags of tokens with counts.
The character sequences will be converted into the sequence of tokens and counted forming a bag of words representation. A set of categories will be set at construction time.
The probability of each category independent of the tokens as well as the probability that each token is in any category will be estimated. These estimates will provide zero probability for the cases that were not seen in the training process. For this reason we will use the additive technique that adds a prior count to every count in the training data.

The next classifier that will be used will be based on character level n-gram categorization. An n-gram is a chain of n consecutive characters of a string. The classifier, proposed by Kešeljet al. [18], compares the frequencies of the most common groups of n-characters in the documents that must be classified.

This model has several advantages when compared to other models of text classification: it is language syntax and language semantics independent and it is not necessary to remove the stop words or to find the stems of the words [19]. Using the language independence feature of this classifier it can be applied to a new language discovering knowledge about the context and content [20].

EXPERIMENTS

The used dataset is constructed from 1313 patient data written in Albanian language and filled in an information system by the radiologists of Neuroradiology Service at Department of Neurosciences of Hospital Center "Mother Teresa" in Tirana, Albania. Albanian language is an Indo-European language that is complex because it has different forms for the same lemma. For example, nouns have different forms in singular and plural and verbs also.

The data in the clinical report is composed of two main sessions: the data about the patient and the data of the examination. We will use data from the second session to construct the data set.

We have used one field of the information system that is primary used for searching the documents. This field is plotted by the radiologists and we will use it to classify our data into 6 classes: head, spinal column, superior abdomen, inferior abdomen, left knee, right knee. These classes describe the part of the body that has been scanned.
The data that has the medical report from the radiologist for the scanned images will be used to classify the document into one of these six classes.

The data used for the training of the classifiers is extracted from the database of the information system. We are interested in two fields: the classification field and medical report. The data of these fields for each document is placed in one separate row in the resulted excel file. This file, used to store the data, will be used for training and testing. In figure 2 two examples from the data from the categories: head and spinal column are shown.

Classification	Report Data
Kokë	Lezion temporal sin me shtrirje drejt insulës së kësaj ane që përforcon në mënyrë heterogjene pas injektimit të kiv me aspekt glioblastoma. Fossa krani posteriore pa lezione evidente. Cisternat bazale të lira. Ventrikujt me aspekt normal. Regjioni selar dhe paraselar me aspekt normal. Tonsilat cerebelare me pozicion normal.
Kollonë	Medula spinale me dimensione dhe strukturë normale. Prolabim diskal C3-4 median. Pa lezione evidente paraspinale. Në kolonën lumbare ndryshime degjenerative diskale. Pa imazh për hernie diskale.

Fig. 2 An example from the dataset of the data

The code to train the classifiers and run the experiments was written in Java. The performance of each classifier will be measured in term of accuracy and time complexity. The time that takes a classifier to be trained is measured in milliseconds. These results are presented in the table 1.

TABLE I. TRAINING TIME OF CLASSIFIERS

Classifier	Naïve Bayes	3-grams	4-grams	5-grams	6-grams
Training time (ms)	194	336	335	359	379

To test the classifiers in terms of classifying efficiency a dataset composed of 100 documents chosen at random from the set of all records was prepared. The time that takes a classifier to classify the prepared dataset was measured in milliseconds. These results are presented in the table 2.

TABLE II. EFFICIENCY OF CLASSIFIERS

Classifier	Naïve Bayes	3-grams	4-grams	5-grams	6-grams
Efficiency (ms)	172	302	206	208	236

We conduct 10-fold cross-validation using Naïve Bayes and character level n-gram categorization with 3, 4, 5, 6 grams. The results of these experiments are presented in the table 3. Accuracy is used to compare the classifiers.

TABLE III. ACCURACY OF CLASSIFIERS

Classifier	Naïve Bayes	3-grams	4-grams	5-grams	6-grams
Accuracy	0.981	0.977	0.973	0.974	0.973

As we can see from these results the best efficiency and accuracy is reached by Naïve Bayes classifier.

CONCLUSIONS AND FUTURE WORK

In this paper we compared two classifiers that are used for the text classification of unstructured radiology reports written in Albanian language: Naïve Bayes and character level n-gram. The analyzed dataset is constructed from 1313 radiology reports gathered at Neuroradiology at Department of

Neurosciences, “Mother Teresa” University Hospital Center in Tirana, Albania.

The experiments show that Naïve Bayes algorithm performs better than the other algorithm applied with different n-grams in terms of efficiency and accuracy. This classifier reaches a very good performance as well as the other classifiers, but further analysis should be done in other types of medical documents written by a larger number of physicians.

In the future, we plan to use the relations of the known concepts to construct an ontology that will be used in the classification of documents.
The ontology will present the semantic relationships about the radiological tests and possible diseases related to the body parts that are usually referred by the physicians to be scanned.

ACKNOWLEDGMENT

The authors thanks Neuroradiology at Department of Neurosciences of “Mother Teresa” University Hospital Center for their support.

REFERENCES

[1] S. Maxhelaku and A. Kika, "Implementation of SPA in Radiology Information System," *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, Bari, Italy, 2020, pp. 1-5, doi: 10.1109/EAIS48028.2020.9122761.

[2] F. S. Collins and H. Varmus, “A new initiative on precision medicine,” New England Journal of Medicine, vol. 372, no. 9, pp. 793–795, 2015.

[3] Sebastiani, F. Machine learning in automated text categorization. ACM Computing Surveys 34, pp. 1-47, 2002.

[4] A. D. Pham, A. Névéol, T. Lavergne, et al., “Natural language processing of radiology reports for the detection of thromboembolic diseases and clinically relevant incidental findings,” BMC Bioinformatics, vol.15, pp. 8, 2014.

[5] W. W. Yim, T. Denman, S. W. Kwan and M. Yetisgen, “Tumor information extraction in radiology reports for hepatocellular carcinoma patients,” AMIA Jt Summits Transl Sci Proc., pp. 455-464, July 2016.

[6] S. Hassanpour, G. Bay and C. P. Langlotz, “Characterization of Change and Significance for Clinical Findings in Radiology Reports Through Natural Language Processing,” J Digit Imaging, vol. 30, pp.314-322, 2017.

[7] Y. Shen, Y. Li, H. T. Zheng, B. Tang and M. Yang, “Enhancing ontology-driven diagnostic reasoning with a symptom-dependency-aware Naïve Bayes classifier,” BMC Bioinformatics, vol. 20, 330, June 2019.

[8] M. Benndorf, E. Kotter, M. Langer, C. Herda, Y. Wu and E. S. Burnside, “Development of an online, publicly accessible naive Bayesian decision support tool for mammographic mass lesions based on the American College of Radiology (ACR) BI-RADS lexicon,” Eur Radiol, vol. 25, pp. 1768-1775, 2015.

[9] I. Drozdov, D. Forbes, B. Szubert, M. Hall, C. Carlin and D. J. Lowe, “Supervised and unsupervised language modelling in Chest X-Ray radiological reports,” PLoS One, vol. 15, March 2020.

[10] I. Krsnik I, G. Glavaš, M. Krsnik, D. Miletić and I. Štajduhar, “Automatic Annotation of Narrative Radiology Reports,” Diagnostics (Basel), vol. 10, April 2020.

[11] M. Pandey et al., “Extraction of radiographic findings from unstructured thoracoabdominal computed tomography reports using convolutional neural network based natural language processing,” PLoS One, vol.15, July 2020 .

[12] M. Luo, A. M. Cohen, S. Addepalli and N. R. Smalheiser, “Identifying main finding sentences in clinical case reports,” Database (Oxford), June 2020.

[13] B. T Bucher, J. Shi, R. J. Pettit, J. Ferraro, W. W. Chapman and A. Gundlapalli, “Determination of Marital Status of Patients from Structured and Unstructured Electronic Healthcare Data,” AMIA Annu Symp Proc., pp. 267-274, March 2020.

[14] Ch. Wansing , O. Banos , P. Gloesekoetter , H. Pomares and I. Rojas, “Development of a platform for the exchange of biodatasets with integrated opportunities for artificial intelligence using MatLab,” Third World Conference on Complex Systems (WCCS), Marrakech, pp. 1-6, 2015.

[15] I. O. Ogundele, O.L Popoola, O. O. Oyesola and K.T. Orija, “A Review on Data Mining in Healthcare,” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol . 7, pp. 698-704, September 2018.

[16] S. Chakrabarti, S. Roy, and M.V. Soundalgekar, “Fast and accurate text classification via multiple linear discriminant projection”, The VLDB Journal The International Journal on Very Large Data Bases, 2003, pp. 170–185.

[17] M. Langarizadeh and F. Moghbeli, “Applying Naive Bayesian Networks to Disease Prediction: a Systematic Review,” Acta. Inform. Med., vol. 24, pp. 364-369, 2016.

[18] V. Kešelj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING’03, pages 255–264, Dalhousie University, Halifax, Nova Scotia, Canada, August 2003.

[19] M. Jankowska, V. Kešelj and E. Milios, "Relative N-gram signatures: Document visualization at the level of character N-grams," 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), Seattle, WA, 2012, pp. 103-112.

[20] J. Kruczek, P. Kruczek and M. Kuta. “ Are n-gram Categories Helpful in Text Classification?,” Computational Science – ICCS 2020, pp. 524-537, June 2020.

A Survey of Artificial Intelligence Driven Blockchain Technology: Blockchain Intelligence

Naim Ajlouni
Department of Software Engineering
İstanbul Aydın University
İstanbul, Turkey
naimajlouni@aydin.edu.tr

Adem Özyavaş
Department of Computer Engineering
Istanbul Aydın University
İstanbul, Turkey
ademozyavas@aydin.edu.tr

Mustafa Takaoğlu
Department of Computer Engineering
Istanbul Aydın University
İstanbul, Turkey
mustafatakaoglu@aydin.edu.tr

***Abstract*— This work looks at the literature on the interaction and possible collaboration areas between the blockchain technology and artificial intelligence (AI) that can benefit both. Blockchain has gained considerable popularity because of cryptocurrencies and has been one of the centers of interest and research recently. Like any other technology, it brings its own buzzwords such as smart contracts, proof of work, and consensus algorithm together with their shortcomings which AI can help with. Similarly, blockchain technology can provide the reliable and quality data that AI algorithms need for more accurate results.**

Keywords—blokchain technology, artificial intelligence, smart contracts, distributed ledger technology, blockchain intelligence.

I. INTRODUCTION

Technological developments affecting humanity have always been the result of an accumulation of knowledge that transformed the way we carry out our daily activities. Many innovations, from the evolution of computing to the use of the internet have taken a similar path of evolution. Artificial intelligence which started in 1950s has now been mingled with blockchain and cryptocurrencies [1], which are widely discussed today, have a similar history. Merkle trees, started in the 1970s [5] and widely accepted as the foundation of blockchain, form the digest of all the transactions in a block. The impact of new technologies in our lives may be hard to predict. To some the idea of blockchain and the way it has the potential to transform the way we go about daily activities is revolutionary. Now artificial intelligence can help the blockchain technology to take its accomplishments even further by providing the tools where blockchain technology falls. Short. It is probable that in the next few decades, we will witness many innovations that both technologies can bring to our lives. It is of great importance that study subjects with such great potential are maintained under the same roof.

Blockchain technology in the simplest terms is a data center [4]. The most important feature of blockchain systems from the perspective of artificial intelligence is that it can provide the reliable and high quality data that artificial intelligence algorithms crucially depend on for accurate predictions. At the same time, artificial intelligence can help the blockchain technology with the structural problems that it faces. Therefore, the purpose of this study is to focus on the symbiosis of both technologies and look into the effects of such a work together. The fact that blockchain technology has the potential and been used outside of the financial applications today and the fact that artificial intelligence algorithms are used behind the scenes in many applications could be an indication that blockchain and artificial intelligence hybrid solutions will appear more in the near future.

The concept of Blockchain Intelligence [71] has already been mentioned in some studies. Blockchain systems have been added as a layer in software systems to benefit from the advantages brought by the blockchain technology. As opposed to blockchain's integration into the system as a layer after the system is developed, new systems will have the blockchain intelligence integrated in the system while it is being conceptualized.

The flow of our study is planned as follows: First, a detailed blockchain technology introduction is made in the second part. In the third part, artificial intelligence is summarized with the emphasis of both technologies' support for each other. In the fourth chapter, blockchain intelligence is detailed and introduced through an example.

II. BLOCKCHAIN TECHNOLOGY

Bitcoin, which is based on blockchain technology, was introduced in 2008 by Satoshi Nakamoto [1]. After bitcoin, blockchain technology has become a popular subject that has been widely studied both by private sector companies and academicians [2][3].

Blockchain architecture was proposed by Stuart Haber and W. Scott Stornetta in 1991 [4]. From a historical perspective, the roots of technology appear when Ralph C. Merkle proposed the Merkle tree in the late 1970s [5]. In 1990, the first encryption money for electronic payments was used and names as e-Cash. Further evolution and improvements of the mixed chain concept were introduced in the 1994 in an article by Neil Haller with S / KEY, a mixed chain for Unix login [6]. In 2002, Adam Back proposed a hashcash [7][8], a blockchain-based electronic currency, with a proof-of-compromise algorithm with many of the features of Bitcoin, and was referred to by Nakamoto as Bitcoin's reference study [9].

Blockchain is a distributed digital ledger technology that every node contains a copy of all transaction records that make up on the blockchain system [10][11]. The blockchain that gains a distributed structure thanks to the sharing of transaction records with all nodes avoids the problem of being the target of cyber-attacks faced by centralized systems [12][13]. Besides, the blockchain systems enable peer-to-peer transactions, which eliminates the possibility of a third party being an intermediary [14][15]. System' nodes see all transactions making blockchain-based systems attain transparency [16].

Cryptology [17] is the basis of the blockchain [18]. The system consists of blocks [19]. Each block has a cryptological hash function [20]. These hash functions connect all end-to-end blocks as a chain [21]. The first block created is called the genesis block. The characteristics of the chain are prescribed in the genesis block, and all operations that are performed are written to the ledger in one direction [22]. A transaction approved by the nodes cannot be modified. When a block changes, the system throws that block out of the chain. This way, blockchain systems are tamper-proof [23]. Blockchain technology is a supportive technology meaning that it can be used to support other existing technologies. For this reason,

blockchain technology is currently being adapted in many areas [24].

A. Characteristics of Blockchain Technology

Each blockchain system has the following common characteristics.

Distributed and sustainable: As the nodes forming the system are spread all over the world, a distributed system is obtained, unlike central data servers [25][26]. Also, there is no vital information or piece of code in the nodes to maintain the system. Thus, blockchain systems are sustainable systems.

Secure: Cryptological hash functions [27] for example SHA256 and consensus protocols, explained in next sections, Merkle hash three and digital signatures [28] are all there to increase the security of the data which stored in the blocks [29].

Private: The information of users who make transactions in blockchain systems as a node is not shared with the public. Furthermore, in private and hybrid blockchain systems [30], transaction records are not visible to everyone and are only accessible by permissioned nodes ensuring blockchain systems' privacy [31].

Indelible: In order for a transaction to register in the blockchain system, all nodes must approve. Information that is approved and written to the blockchain becomes definite and cannot be modified [32]. In other words, it is not possible to modify or delete a transaction that has been written to the block [33].

Transparent and auditable: Nodes forming the blockchain have access to the system and could see all the transactions performed [34]. This makes the blockchain systems transparent and auditable [35].

Consensus-based: A consensus algorithm can be defined as the mechanism used by a blockchain network to reach an agreement. There are many consensus algorithms and each has some advantages and disadvantages based on the needs of the system in which they are used [36]. Algorithms such as PBFT-based, Stellar, Ripple, Proof-of-Work (PoW), Proof-of-Stake (PoS), Threshold Relay, Proof-of-Authority (PoA), Proof-of-Burn (PoB), Proof-of-Elapsed Time (PoET) are consensus algorithm examples [37].

B. Public and Private Blockchains

Public (Permissionless) Blockchains: It is a blockchain type owned by popular crypto coins such as Bitcoin [38] and Ethereum [39]. There is no limit to access and transaction. The participant joins the system and becomes part of it [40]. Because of this, the system has a distributed structure. In public blockchains, all nodes are synchronized with each other. When the established chain grows in time, the processing speed decreases with the increasing number of blocks, and hence the increase in the amount of energy consumed. As opposed to Private Blockchain, public blockchain systems allow users to monitor system events [41].

Private (Permissioned or Consortium) Blockchain: In private blockchains, access must authenticated [42] hence allowing only registered nodes [43]. Decision-making nodes are determined among the nodes granted access to the system, and the transactions are approved accordingly by registered nodes [44].

C. Blockchain Structure

The blockchain structure consists of nodes of distributed architecture [45] and which are granted access to the structure that has records of all the transactions that took place in the blockchain system [46]. The data of the transactions performed in blockchain systems are stored in blocks of the chain [47].

The first generated block is known as the genesis block [48]. The characteristics of the blockchain system to be created are prescribed in the first block and rules in this first block do not change during the lifetime of the blockchain. All the blocks after the genesis block are connected securely by cryptological methods. Also, each block has a timestamp, data, hash and previous hash value, nonce and public, and private keys [49]. Timestamp holds the time of the block created. Transactions comprise the data. The nonce is an arbitrary value and makes the blocks unique. Public and private keys are used to send data securely [50].

Blocks and Block Time: The information of transactions approved by the majority of nodes are stored in blocks. Data encoded with hash algorithms are organized as a Merkle (Hash) Tree as a fast retrieval method. All blocks are linked to each other by cryptologic hash algorithms. To link all blocks with each other hash pointers are used. Hash pointer and Merkle Tree data structure are placed in the block header [48].

In blockchain networks, the block time varies. The shorter the newly created block time, the faster the system is. For example, the Ethereum system has a block time of 15 seconds and a Bitcoin process of 10 minutes [51].

Decentralization: Today, the data centers of many advanced applications are centralized. This puts big and valuable data at the target of hackers [52]. For this reason, data security is always a significant problem. Blockchain technology enables peer-to-peer communication, public and private key usage, and hash algorithms for secure connection of the chain. Blockchain technology has a distributed architecture that deviates from being a central structure. This way, each node has become a part of this distributed system and increases the system's security [53].

Openness: In most blockchain systems, the source code is public, i.e., it is open source. Anyone can download this source code and access the system. Also, the data of the created blockchain system is open to everyone. In this way, the system becomes open and is monitored by the nodes with access rights [54].

D. Advantages and Disadvantages of Blockchain

Advantages of Blockchain: Due to the distributed structure of the blockchain, the data is not stored on a central server, thus protecting against technical malfunctions and cyber-attacks. Preventing cyber-attacks leads to the reduction of direct fraud attempts [55].

Stability is achieved in the system because the blockchain is unidirectional, a transaction approved by all nodes cannot be changed, and all changes can be monitored transparently by the access nodes [56].

The increase in the use of cryptocurrencies has enabled the possibility of reducing costs by removing third institutions or organizations through the peer-to-peer communication feature. Also, fake information sharing by financial institutions and the hidden charges applied to their customers will make customers have distrust in these companies. Use of blockchain by institutions will increase the transparency hence causing customers to trust in them [57].

Disadvantages of a Blockchain: Scalability is one of problems with the blockchain systems. Take Bitcoin for example. Bitcoin cryptocurrency is one of the applications of blockchain. As the system size, that is, the number of blocks increases, scalability has become an issue. The Bitcoin blockchain platform can handle four transactions per second, while the Ethereum blockchain platform can handle twelve transactions per second. This processing power is unacceptable compared to the Visa card system or merely Facebook's processing capacity per second [58].

Although consensus algorithms are widely studied, the weaknesses of consensus algorithms still remains unsolved.

For example, the PoW algorithm is inefficient in mining operations, and the PoS algorithm does not perform sufficiently in the process validation stage [59].

As blockchain systems' data grow with time, alternative solutions should be found in terms of data storage. Today, a blockchain system requires an average of 200 GB of storage space. The high demand for data storage that the blockchain systems need over time can become a problem [60].

Another problem is that the keys that are used as private keys used by the blockchains loss of these keys will result in the failure to reach the system. If the password is forgotten in the central data structures, there are steps to obtain a new password. Loss of private keys in distributed systems can lead to the loss of the valuable asset stored as well as the security it provides [61].

The 51% attack mentioned above poses a risk in theory. No successful 51% attack has been seen to date. However, an attack by quantum computers is likely to succeed. Therefore, it is worth noting that the 51% attack is a disadvantage. In the future, 51% attacks will no longer be a risk with the implementation of cryptology applications created with quantum calculation methods [62].

Finally, the strict transparency provided by blockchain architecture brings along the problem of lack of privacy. Credentials hidden by cryptology can be detected when open transaction records can be revealed. Therefore, hybrid solutions can be studied to solve the privacy problem [63].

E. *Blockchain Application Areas*

Blockchain technology can be used on banking applications, internet security, supply chain, internet of objects, insurance, personal and public transportation, online data storage, foundations and endowments, voting processes, public applications, health applications, energy management, intellectual property and copyright applications, real estate and title deeds, digital identity, smart cities, smart contracts and legal compliance examination, applications in the field of education [64].

After the introduction of the Bitcoin cryptocurrency and its attention with its success, many crypto coins are introduced and used. After bitcoin, which aims to remove central authority, cryptocurrencies have attracted attention by many central authorities, and they are trying to make their systems compatible with the blockchain. Today, many central banks are working on blockchain-based cryptocurrency and financial instruments. Besides, the applications of blockchain technology on cryptocurrency and finance are called blockchain 1.0 [4].

Smart contracts [65] are one of the most critical developments that blockchain technology brings to our lives. Smart contracts can be thought of as part of the blockchains where adds dynamicity to the system. As the name implies, smart contracts are short snippets of code. These contracts, which are developed as the solution for a unique problem, are a piece of code in the blockchain system [66]. These systems, which only get executed once the condition arises, need to be carefully optimized since they cannot be corrected after they are coded. The use of smart contracts in Blockchain technology is called Blockchain 2.0 [4].

Every day blockchain technology has been adapted to new areas of application. All non-financial application areas are called blockchain 3.0. Also, blockchain technology and artificial intelligence applications are being tested. Although not explicitly accepted by the Community, it is considered to be called artificial intelligence adaptations for Blockchain 4.0 [4].

III. Artificial Intelligence

Artificial intelligence is building computer systems that can simulate human intelligence. All learning can only be achieved when enough and proper data exist. The data set should be of high quality, up-to-date and relevant in order to obtain the most effective results from the data to be used, rather than having a very large size. Along with the quality data obtained in this direction, the algorithm and models used must be selected correctly. In addition, the proposed artificial intelligence solution should be scalable according to new requirements.

The concept of artificial intelligence (AI) was first mentioned in the 50s. In 1956, the term of artificial intelligence was firstly announced by John McCarthy [67]. The first artificial intelligence studies were on neural networks and the idea was of a machine which can think the way a human does. The first wave of studies took place between the 1950s and the 1970s. Between 1980 and 2010, artificial intelligence studies were mostly on machine learning which is a sub field of AI. Studies after 2010 focus on another subfield of AI, the deep learning.

An examination of the literature will reveal research in General AI, Machine perception (speech recognition and computer vision), Natural Language Processing (NLP), and Robotics, Knowledge Representation and Reasoning (KRR), Pattern Recognition (PR), Machine Learning (ML), Artificial Neural Networks (ANN) and Social intelligence [68]. These research areas of artificial intelligence have many applications such as Automotive and Logistics, Retail, Healthcare, Military, Space Studies, Industry, Telecommunication, and Finance to name just a few.

A. *Possible Contributions of Blockchain Technology to Artificial Intelligence Applications*

The use of blockchain technology in artificial intelligence applications strengthens artificial intelligence applications in terms of reliability, security, transparency and trust.

Quality problems encountered in data used in artificial intelligence applications lead to low interpretability success. For this reason, the data stored in blockchain systems and collected within the framework of certain standards and whose accuracy is beyond doubt will enable artificial intelligence applications to give clearer results. In addition, blockchain technology can be a solution to privacy and trust problems, which are shown as weaknesses in artificial intelligence studies in the literature. Thanks to the fact that the data used is taken from distributed data structures and the data cannot be tampered with, the results obtained can be more reliable. In addition, privacy concerns will be eliminated in artificial intelligence applications developed with data obtained from private blockchain systems.

For example, in the study published in 2019, researchers proposed a self-testing and tracking systems for COVID19 [69]. In this study, it is aimed to process the data obtained with a blockchain-based application from health institutions using a machine learning algorithm. The point that should not be overlooked in this study is the machine learning study on the data obtained from the blockchain system. In other words, while the data is obtained through smart contracts in the blockchain system, there is no artificial intelligence applied in the data extraction.

Many study ideas are suggested in global warming, smart agriculture and other similar areas. The use of data obtained from blockchain-based distributed data structures in studies with artificial intelligence applications at the center is of great importance in this respect.

B. *Possible Contributions of Artificial Intelligence to Blockchain Technology Applications*

Blockchain is a relatively new topic of study and is on its way to become more mature and stable. Like any other new technology, it has its urgent shortcomings that need better solutions. The main issues that need to addressed in blockchain technology are high energy consumption,

scalability, security problems arising from improper development of smart contracts [70].

Especially the high energy consumption seen in the bitcoin blockchain application is a huge problem in terms of sustainability. Although the use of new alternatives proposed in consensus algorithms that can solve high energy consumption is promising. The success of artificial intelligence applications in energy consumption optimization is well known. For this reason, the contribution of artificial intelligence applications in the most optimal determination of the energy spent in the mining process, which is needed not only in the bitcoin blockchain system, but also in all similar blockchain systems, will undoubtedly be very high.

Blockchain systems are systems that cannot be intervened after they are installed and increase in size as their use increases. When the idea of bitcoin crypto money was proposed by Satoshi, solution suggestions were also shared for the problem to be encountered in scalability. Pruning of old transaction records in blockchain was proposed by Satoshi [1]. However, the same problem is encountered in many similar blockchain systems, as in the example of bitcoin, which is moving towards 350 GB. Federated Learning may be a very convenient solution method to this problem. Federated learning is a machine learning technique that trains an algorithm between multiple decentralized nodes without exchanging data. Because of this feature, it is a machine learning algorithm that is very suitable for blockchain architecture [4].

In blockchain systems, the smart contracts that we encounter in the Blockchain 2.0 process have made the blockchain systems more functional. Smart contracts are small pieces of code prepared for previously anticipated situations and automatically complete the transaction in case the condition it states is met. An overlooked flaw in smart contracts developed in blockchain systems can cause serious problems in the future. For this reason, it is of great importance that the smart contract codes are checked by an artificial intelligence algorithm and their deficiencies are detected. The platform named Oyente offers a service for Ethereum-based blockchain smart contracts. However, this service needs to be provided in all blockchain systems and the need for a developed one using artificial intelligence algorithms is undoubtedly very high.

In cases where mining is required in blockchain systems, efficiency is an important issue. Artificial intelligence applications that track and organize the working conditions of miners are needed to increase efficiency. Thanks to the solution suggestions that ensure that miners do not work in vain, energy consumption will decrease as well as faster transactions on the network.

C. *Projects on Artificial Intelligence and Blockchain Technology in the World and Turkey*

In the literature reviews we have done, there are no institutions working on or projects about artificial intelligence and blockchain technology in Turkey. For this reason, adding blockchain technology to the work of Turkish organizations working on artificial intelligence is of considerable importance.

However, there are initiatives such as DeepBrain Chain, Synapse AI, Endor, AiX, Peculium, Autonio, burs iQ, Indorse, Matrix, Neureal, BotChain, Singularity Net, Numerai, Dopamine working in this field in the world. If shared initiatives are examined, the potential of blockchain and artificial intelligence studies carried out todays will be better understood.

IV. BLOKCHAIN INTELLIGENCE

The concept of blockchain intelligence, as its name suggests, is one that involves and integrates the use of both technologies. It is aimed to obtain a more efficient system by using artificial intelligence tools for various improvements in a blockchain system while designing a new one. As briefly touched before, developing a blockchain intelligence system will help to solve many structural problems of “regular” blockchain systems. When the distributed, decentralized, anonymous, secure, transparent structure provided by blockchain technology is combined with the learning, detection, diagnosis, improvement and automation opportunities provided by artificial intelligence algorithms, the mingling process will result in more efficient blockchain systems.

There will be significant improvements in three aspects in the systems we call blockchain intelligence. These can be expressed as smart operational maintenance of the blockchain, smart quality assurance of smart contracts, and automatic malicious behavior detection of blockchain systems [71]. Explanations on these issues are included in the titles A, B and C.

A. *Smart Operational Maintenance*

A blockchain system can be monitored using artificial intelligence and the errors that the system may encounter can be detected and maintained in advance. Data is constantly produced in all blockchain systems that are widely used today. If the system is not monitored regularly, the problems that may arise will cause very destructive results. For this reason, it is possible to keep the performance of the system in the most optimal way by constantly monitoring it with artificial intelligence and making necessary interventions.

B. *Smart Quality Assurance of Smart Contracts*

The verification of the structural accuracy of smart contracts using artificial intelligence algorithms was a requirement that we mentioned in the previous sections. However, the determination and development of smart contracts required in blockchain intelligence by the system will emerge as one of the issues we will encounter under the roof of blockchain intelligence in the future. In this way, a blockchain system will detect new smart contracts it needs and include it in the system in the most appropriate way. As it can be understood from here, changes will be possible in the structures of smart contracts used in blockchain systems. Because there is a need for alternative solutions to overcome the almost impossible problems arising from smart contracts that cannot be changed after they are introduced in the blockchain system. The most obvious solution method will be to bypass the smart contract used and define a new smart contract instead. In an environment where studies on blockchain technology are increasing, it is possible for us to encounter innovations in the above-mentioned direction.

C. *Automatic Malicious Behavior Detection*

One of the most used products of blockchain systems today is crypto money. Since the first use of Bitcoin crypto money in 2009, there have been many discussions about who used cryptocurrencies and for what purpose. Undoubtedly, it is very natural to use cryptocurrencies to finance many illegal activities due to its anonymous structure. Turkey also experienced as previously "Çiftlik Bank" fraud attempt which is prepared with the logic of the Ponzi scheme for fraudulent purposes. Unfortunately such fraud attempts are going on around the world. And finally, it is known that serious amounts of theft have been experienced using systemic errors found in some blockchain systems. Automatic malicious behavior detection systems are needed to prevent all these negativities. It is aimed to establish a system that will alert in suspicious situations encountered with an artificial intelligence that examines blockchain systems, and will detect illegal purposes, various manipulations, and attempts like Ponzi scheme [73].

D. *Blockchain Intelligence Case Study*

The use of blockchain technology in the field of health is a very interesting subject of study. The multi-stakeholder structure of the healthcare industry is in great need of technological innovations in order to facilitate the management of institutions. For this reason, many studies are carried out, including ourselves, on the application possibilities of blockchain technology in the health sector. However, the studies carried out jointly include the suggested solutions to increase data security and anonymity. Considering the possibilities of blockchain implementation in health, which blockchain intelligence will suggest, a much more complex structure emerges.

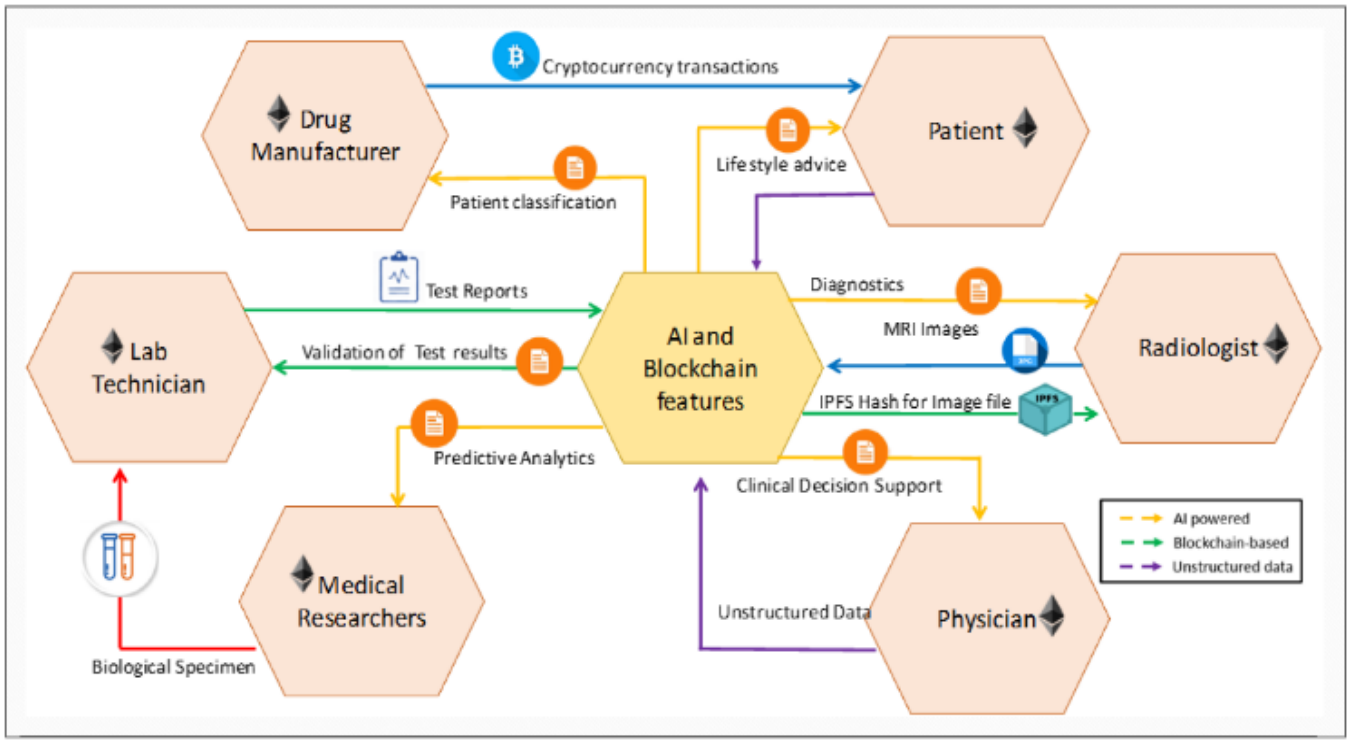


Figure 1 Blockchain Intelligence Example in Healthcare

As it can be understood from Figure 1, it is seen that the blockchain systems to be developed will have a much more dynamic and intelligent structure if the concept of blockchain intelligence begins to be used in real applications.

Also the study conducted in 2018 (It is the study where the visual shown in Figure 1 was used to our study), in a health management system proposal in which all stakeholders consist of solution suggestions developed with blockchain technology, it is recommended to use artificial intelligence with blockchain technology and add features such as analysis, diagnosis, advice, support and classification [72].

Solutions such as education, retail, government and management can be developed with the same logic as the example we shared in the field of health. As it can be understood, if the use of artificial intelligence and blockchain systems together is widespread and easy to implement, it is no exaggeration to say that an era in which blockchain intelligence will prevail is about to begin.

V. CONCLUSION

In our study, the concept of blockchain intelligence is introduced. It was stated that what can be achieved in closing the structural deficiencies of blockchain technology with artificial intelligence. Solutions that can be provided by blockchain intelligence are shared to detect structural problems encountered in smart contracts in advance. The necessity of blockchain intelligence in tracking suspicious transactions and taking necessary precautions has been shared. The necessity of blockchain intelligence has been stated to monitor blockchain systems and to detect performance problems anticipating them and taking precautions before they happen. In addition, as can be seen in the shared health example, it has been shown that the innovations provided by artificial intelligence are a very suitable method in eliminating the deficiencies needed with the distributed, secure and anonymous structure of blockchain technology.

ACKNOWLEDGMENT

We would like to express our gratitude to Istanbul Aydın University for all its support in publishing this conference paper, which is a result of the work we are conducting under the roof of Istanbul Aydın University Blockchain Application and Research Center.

REFERENCES

[1] S. Nakamoto, “Bitcoin: A Peer to Peer Electronic Cash System.” (Web Sourse, link: https://bitcoin.org/bitcoin.pdf), 2008.

[2] R. Latypov and E. Stolov, “A new watermarking method to protect Blockchain records comprising small graphic files.” 42nd International Conference on Telecommunications and Signal Processing (TSP), pp.668-671, 2019.

[3] C. Ehmke, F. Wessling and C. M. Frederich, “Proof-of-Property – A Lightweight and Scalable Blockchain Protocol.“ ACM\IEEE 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, WETSEB’18, pp. 48-51, 2018.

[4] M. Takaoğlu, Ç. Özer and E. Parlak, “Blockchain Technology and Possible Implementation Areas in Turkey.” International Journal of East Anatolian Science Engineering and Design, ISSN: 2667-8764, 1 (2), 260-295, 2019.

[5] R. C. Merkle, "Protocols for Public Key Cryptosystems." Proceedings of the 1980 {IEEE} Symposium on Security and Privacy, pp.122-134, 1980.

[6] N. M. Haller, "The S/KEYTM One-Time Password System." In Proceedings of the Internet Society Symposium on Network and Distributed Systems, pp.151-157, 1994.

[7] A. Back, “Hashcash - A Denial of Service Counter-Measure.” Web Source. Link: http://www.hashcash.org/papers/hashcash.pdf, 2002.

[8] A. Back, “Hashcash - Amortizable Publicly Auditable Cost-Functions.” Web Source. Link: http://www.hashcash.org/papers/amortizable.pdf, 2002.

[9] T. Aste, P. Tasca and T. Di Matteo, “Blockchain Technologies: The Foreseeable Impact on Society and Industry.” Computer, 50(9), 18-28, 2017.

[10] L. Bahri and S. Girdzijauskas, “Blockchain Technology: Practical P2P Computing (Tutorial).” IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), pp.249-250, 2019.

[11] Y. W. Chang, ”Blockchain Technology for e-Marketplace.” PhD Forum on Pervasive Computing and Communications, pp.429-430, 2019.

[12] M. Takaoğlu and Ç. Özer, "The Effect of Machine Learning on Intrusion Detection Systems." International Journal of Management Information Systems and Computer Science, 3(1), 11-22, 2019.

[13] S. Cai, M. Xu and L. Zhang, "Automatic Information Disclosure with Value Chains Based on Blockchain Technology." IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp.1534-1538, 2019.

[14] Z. Ma, W. Huang, W. Bi, H. Gao and Z. Wang, "A Master-Slave Blockchain Paradigm and Application in Digital Rights Management." in China Communications, 15(8), pp.174-188, 2018.

[15] H. Magrahi, N. Omrane, O. Senot and R. Jaziri, "NFB: A Protocol for Notarizing Files over the Blockchain." 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1-4, 2018.

[16] C. Molina-Jimenez, I. Sfyrakis, E. Solaiman, I. Ng, N. W. Wong, A. Chun and J. Crowcroft, "Implementation of Smart Contracts Using Hybrid Architectures with On and Off–Blockchain Components." 8th International Symposium on Cloud and Service Computing (SC2), pp. 83-90, 2018.

[17] F. Takaoğlu, F. Sönmez and O. Kaynar, "Ideal Steganography Scenario: Calculation of Capacities of Carrier Images, OPA Method in Frequency-Based Steganography." ACTA INFOLOGICA, 2(1): 12-21, 2018.

[18] K. Qiao, H. Tang, W. You and Y. Zhao, "Blockchain Privacy Protection Scheme Based on Aggregate Signature." 4th International Conference on Cloud Computing and Big Data Analytics, pp.492-497, 2019.

[19] S. Morishima and H. Matsutani, "Accelerating Blockchain Search of Full Nodes Using GPUs." 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, pp.244-248, 2018.

[20] B. Sakız and E. Kutlugün, "Bitcoin Price Forecast Via Blockchain Technology And Artificial Intelligence Algorithms." 26th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2018.

[21] R. A. Sarıtekin, E. Karabacak, Z. Durğay, and E. Karaarslan, "Blockchain Based Secure Communication Application Proposal: Cryptouch." 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-4, 2018.

[22] T. Faisal, N. Courtois and A. Serguieva, "The Evolution of Embedding Metadata in Blockchain Transactions." International Joint Conference on Neural Metworks (IJCNN), 2018..

[23] Y. Lin, Z. Qi, H. Wu, Z. Yang, J. Zhang and L. Wenyin, "CoderChain: A BlockChain Community for Coders." 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), pp.246-247, 2018.

[24] S. Al-Megren, S. Alsalamah, L. Altoaimy, H. Alsalamah, L. Soltanisehat, E. Almutairi and A. S. Pentland, "Blockchain Use Cases in Digital Sectors: A Review of the Literature." IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, pp.1417-1424, 2018.

[25] B. Ekbote, V. Hire, P. Mahajan and J. Sisodia, "Blockchain based Remittances and Mining using CUDA." International Conference On Smart Technologies For Smart Nation (SmartTechCon), pp.908-911, 2017.

[26] H. Duan, Y. Zheng, Y. Du, A. Zhou, C. Wang and M. H. Au, "Aggregating Crowd Wisdom via Blockchain: A Private, Correct and Robust Realization." IEEE International Conference on Pervasive Computing and Communications (PerCom), pp.1-10, 2019.

[27] D. Semenovich, "Blockchain, smart contracts and their potential insurance applications." Actuaries Institute General Insurance Seminar, pp.1-9, 2016.

[28] L. Y. Yeh, P. J. Lu and J. W. Hu, "NCHC Blockchain Construction Platform (NBCP): Rapidly Constructing Blockchain Nodes around Taiwan." ACM/IEEE Joint Conference on Digital Libraries (JCDL), pp.1-2, 2017.

[29] C. Worley and A. Skjellum, "Blockchain Tradeoffs and Challenges for Current and Emerging Applications: Generalization, Fragmentation, Sidechains, and Scalability." IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp.1582-1587, 2018.

[30] G. J. Ra and I. Y. Lee, "A Study on Hiybrid Blockchain-based XGS (XOR Global State) Injection Technology for Efficient Contents Modification and Deletion." Sixth International Conference on Software Defined Systems (SDS), pp.300-305, 2019.

[31] S. Salah, M. H. U. Rehman, N. Nizamuddin and A. Al-Fuqaha, "Blockchain for AI: Review and Open Research Challenges." in IEEE Access, 7, 10127-10149, 2018.

[32] H. Orman, "Blockchain: the Emperor's New PKI?" IEEE Internet Computing, pp.23-28, 2018.

[33] J. Zhong, H. Xie, D. Zou and D. K. W. Chui, "A Blockchain Model for Word-Learning Systems." 5th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC), pp.130-131, 2018.

[34] J. Golosova and A. Romanovs, "The Advanteges and Disadvanteges of the Blockchain Technology." IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), pp.1-6, 2018.

[35] A. V. Kumar, A. Prasad and R. S. Murthy, "Application of blockchain in Usage Based Unsurance." International Journal of Advance Research, Ideas and Innovations in Technology, 5(2), 1574-1577, 2019.

[36] M. Belotti, N. Bozic, G. Pujolle and S. Secci, "A Vademecum on Blockchain Technologies: When, Which and How." in IEEE Communications Surveys & Tutorials, pp.1-47, 2019.

[37] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi and J. Wang "Untangling Blockchain: A Data Processing View of Blockchain Systems." IEEE Transactions on Knowledge and Data Engineering, 30(7), 1366-1385, 2018.

[38] M. H. Ziegler, M. GroBmann and U. R. Krieger, "Integration of Fog Computing and Blockchain Technology Using the Plasma Framework." International Conference on Blockchain and Cryptocurrency (ICBC), pp. 120-123, 2019.

[39] P. Urien, "Introducing Innovative Bare Metal Crypto Terminal for Blockchains and BigBang Paradigm." 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1-4, 2019.

[40] S. Rouhani and R. Deters, "Performance Analysis of Ethereum Transactions in Private Blockchain." 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 70-74, 2017.

[41] G. M. Arantes, J. N. D'Almeida, M. T. Onodera, S. M. B. M. Moreno and V. R. S. Almeida, "Improving the Process of Lending, Monitoring and Evaluating Through Blockchain Technologies: An Application of Blockchain in the Brazilian Development Bank (BNDES)." IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1181-1188, 2018.

[42] P. E. Sedgewick and R. Lemos, "Self-adaptation made easy with Blockchains." 13th International Symposium on Software Engineering for Adaptive and Self-Managing Sytems, pp.192-193, 2018.

[43] H. Hellani, A. E. Samhat, M. Chamoun, H. Ghor and A. Serhrouchni, "On BlockChain Technology: Overview of Bitcoin and Future Insights." IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), 2018.

[44] R. Norvill, M. Steichen, W. M. Shbair and R. State, "Demo: Blockchain for the Simplification and Automation of KYC Result Sharing." IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 9-10, 2019.

[45] M. Zhang, S. Wang, P. Zhang, L. He, X. Li and S. Zhou, "Protecting Data Privacy for Permissioned Blockchains using Identity-Based Encryption." 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp.602-605, 2019.

[46] Z. Moezkarimi, F. Abdollahei and A. Arabsorkhi, "Proposing a Framework for Evaluating the Blockchain Platform." 5th International Conference on Web Research (ICWR), pp.152-160, 2019.

[47] T. Salman, R. Jain and L. Gupta, "Probabilistic Blockchains: A Blockchain Paradigm for Collaboratie Decision-Marking." 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp.457-465, 2018.

[48] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y. C. Liang and D. Kim, "A Survey on Blockchain: A Game Theoretical Perspective." in IEEE Access, 7, 47615-47643, 2019.

[49] B. L. Radhakrishnan, A. S. Joseph and S. Sudhakar, "Securing Blockchain based Electronic Health Record using Multilevel Authentication." 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp.699-703, 2019.

[50] D. Ding, K. Li, L. Jia, J. Li and Y. Sun, "Privacy Protection for Blockchains with Account and Multi-Asset Model." in China Communications, 16(6), 69-79, 2019.

[51] R. Kuvvarapu and B. Kuvvarapu, "Research on Application of Blockchain in Cloud ERP Systems." Thesis for: Masters in Management Of Information Systems, Advisor: Mihails Savrasovs, 2018.

[52] C. Cai, H. Duan and C. Wang, "Tutorial: Building Secure and Trustworthy Blockchain Applications" IEEE Secure Development Conference, pp.120-121, 2018.

[53] A. Ambegaonker, U. Gautam and R. K. Rambola, "Efficient approach for Tendering by introducing Blockchain to maintain Security and Reliability." 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1-4, 2018.

[54] M. Z. Masoud, Y. Jaradat, I. Jannoud and D. Zaidan, "CarChain: A Novel Public Blockchain-based Used Motor Vehicle History Reporting System," IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 683-688, 2019.

[55] P. Zhong, Q. Zhong, H. Mi, S. Zhang and Y. Xiang, “Privacy-Protected Blockchain System.” 20th IEEE International Conference on Mobile Data Management (MDM), pp.457-461, 2019.

[56] A. Balaskas and V. N. L. Franqueira, "Analytical Tools for Blockchain: Review, Taxonomy and Open Challenges," International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1-8, 2018.

[57] H. Watanabe, S. Ohashi, S. Fujimura, A. Nakadaira, K. Hidaka and J. Kishigami, "Niji: Autonomous Payment Bridge Between Bitcoin and Consortium Blockchain." IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1448-1455, 2018.

[58] I. Weber, Q. Lu, B. Tran, A. Deshmukh, M. Gorski and M. Strazds, “A Platform Architecture for Multi-Tenant Blockchain-Based Systems.” IEEE International Conference on Software Architecture (ICSA), pp. 101-110, 2019.

[59] N. Al-Zaben, M. M. Hassan Onik, J. Yang, N. Lee and C. Kim, "General Data Protection Regulation Complied Blockchain Architecture for Personally Identifiable Information Management," International Conference on Computing, Electronics & Communications Engineering (iCCECE), pp. 77-82, 2018.

[60] U. Nadiya, K. Mutijarsa and R. Y. Cahyo, “Block Summarization and Compression in Bitcoin Blockchain.” International Symposium on Electronics and Smart Devices (ISESD), pp. 1-4, 2018.

[61] W. Wang, P. Hu, D. Niyato and Y. Wen, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," in IEEE Access, vol. 7, pp. 22328-22370, 2019.

[62] C. Killer, B. Rodrigues and B. Stiller, “Security Management and Visualization in a Blockchain-based Collaborative Defense.” IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 108-111, 2019.

[63] G. Sagirlar, B. Carminati, E. Ferrari, J. D. Sheehan and E. Ragnoli, "Hybrid-IoT: Hybrid Blockchain Architecture for Internet of Things - PoW Sub-Blockchains," IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1007-1016, 2018.

[64] M. Takaoğlu and A. Güneş, “Blockchain Technology in Higher Education.” 5th International Congress on Education, Distance Education and Educational Technology (ICDET), pp. 104-116, 2019.

[65] N. Szabo, “Smart contracts.” URL: http:/szabo.best.vwh.net, 1994.

[66] R. Tonelli, M. I. Lunesu, A. Pinna, D. Taibi and M. Marchesi, "Implementing a Microservices System with Blockchain Smart Contracts," IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE), pp. 22-31, 2019.

[67] S. Preethi, “A Survey on Artificial Intelligence”, International Journal of Intelligent Computing and Technology (IJICT), 3(2), pp.39-42 , 2020.

[68] K. F. Bharati, “A Survey on Artificial Intelligence and its Application”, International Journal of Innovative Research in Computer and Communication Engineering, 5(60), pp.11614-11619, 2017.

[69] T.P. Mashamba-Thompson and E.D. Crayton, "Blockchain and Artificial Intelligence Technology for Novel Coronavirus Disease 2019 Self-Testing." Diagnostics. 10(4):198, 2020.

[70] M. Tshilidzi and X. Bo, "Blockchain and Artificial Intelligence." SSRN Electronic Journal. 10.2139/ssrn.3225357, 2018.

[71] Z. Zheng, and H. Dai, “Blockchain Intelligence: When Blockchain Meets Artificial Intelligence.” ArXiv, abs/1912.06485, 2019.

[72] K. Salah, M. H. U. Rehman, N. Nizamuddin and A. Al-Fuqaha, "Blockchain for AI: Review and Open Research Challenges," in IEEE Access, vol. 7, pp. 10127-10149, 2019.

[73] B. İşler, M. Takaoğlu and U. Küçükali, "BLOKZİNCİRİ VE KRİPTO PARALARIN İNSANLIĞA ETKİLERİ." Yeni Medya Elektronik Dergisi , 3 (2) , pp. 71-83, 2019.

Developing a Clinical Decision Support System by Classification of Melanoma using Machine Learning Techniques

Fatma Betül Kara
Department of Computer Engineering
Duzce University
Duzce,Turkey
fbetulkara@gmail.com

Pakize Erdoğmuş
Department of Computer Engineering
Duzce Univercity
Duzce,Turkey
pakizeerdogmus@duzce.edu.tr

***Abstract*— Correct detection of cancer in patient is critical, and changing the next treatment, can increase the survival. Machine learning (ML) techniques are effective in disease detection and are used in various classification problems with accurate predictive performance. Diverse algorithms can supply different accuracies. Therefore, it is necessary to use the most appropriate method that provides the best results. This study aims to measure the success of various ML algorithms in melanoma classification problem in different development environments. With this aim, the ML techniques including Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, and Support Vector Machine (SVM) in Matlab© and Python environments, have been used for the classification of Melanoma and the best successful one is selected for diagnosing for CDSS. As a result of the trainings, the highest classification success was obtained with the KNN classifier in the Python environment with 93.3%.**

Keywords— melanoma classification, machine learning, KNN, SVM, logistic regression, decision tree

I. INTRODUCTION

Recently skin diseases have become common to everybody. Many factors affect the beginning of these diseases and each age group has different symptoms. Exposure to excessive amounts of ultraviolet radiation in sunlight will make the skin sensitive, make it easier to become infected, and possibly cause skin problems [1]. Melanoma causes 75% of deaths due to skin cancer, as its incidence is 4% among skin cancer types [2]. Melanoma is a type of skin cancer that occurs as a result of mutations in the melanocytes in the skin and occurs on the skin surface, hair follicles, and eye. The rate of correct diagnosis by expert dermatologists is estimated at 75–84% [3]. For this reason, computer-based solutions are needed to increase diagnostic accuracy in medicine, and CDSSs have emerged [4]. CDSS is being used because of its potential to reduce medical errors and improve healthcare quality and efficiency [5].

There are many studies conducted for skin cancer classification. One of the first studies conducted in this area, using a neural network model had a higher success with 92.62% accuracy [6]. Kolkur, Kalbande, and Kharkar's in this study were used artificial neural network (ANN), KNN, SVM, Decision Tree, and Random Forest classifier models. 100% success has been achieved with ANN and Random Forest models [7]. Finally, six machine learning classification techniques: Linear discriminant analysis (LDA), Passive-aggressive classifier (PAC), Gaussian Naïve Bayesian (NB)Radius Neighbors Classifier (RNC), Bernoulli Naïve Bayesian (BNB) , and Extra tree classifier (ETC) are used to classify the skin disease and three ensemble techniques: Bagging, AdaBoost and Gradient Boosting classifiers are applied to improve the accuracy obtained by machine learning algorithms. A feature selection method is also applied on the skin disease dataset to obtain an accuracy of 99.68% in the case of the Gradient Boosting ensemble method applied on RNC [8].

This study aims to classify skin lesions in 2 groups as nevus and melanoma with machine learning techniques to develop helper CDSSs for supporting the expert decision. The CDSS will help increase both the speed and the accuracy of the diagnosis. In this study, we made the classification of skin lesions in Python and Matlab development environments with KNN, SVM, Logistic Regression and Decision Tree ML algorithms and comparison of their classification performances. For the classification of skin lesions with ML techniques, the melanoma data set created by Duzce University Pathology Department pathologists was used.

As a result of the training, ML techniques have shown different success rates in different techniques. When ML models were compared, the model created with skin lesions KNN was the most successful algorithm in both Python and MATLAB environment. It also achieved the highest success in the Python environment with 93.3% in the KNN test data set. The model created with the Desicion Tree algorithm was the lowest successful model in both environments. In line with the results obtained, the KNN model was chosen because it would give the most accurate result to the expert in diagnosing CDDS.

This paper consists of the following sections: Chapter 2 gives information on the data set, explains the theorical background of ML algorithms, ML environments and gives information about the evaluation of classification performance. Then, the methods used in the study are explained in chapter 3. The experimental results are given in chapter 4. The results of the study are explained in chapter 5.

II. BACKGROUND STUDY

A. Machine Learning Algorithms

There have been numerous ML techniques used for examining the classification of cancer problems. In this section, it gives explanations about SVM, KNN, decision trees and logistic regression classification algorithms.

1) Decision Tree: Decision tree is an ML algorithm in tree structure that learns with supervised learning. It designs a model to estimate the class as per the input data. The algorithm helps in making decisions through tree nodes. The best selected data set is placed at the top and then further integrated to achieve the best possible result [9].

2) K-Nearest Neighbor: KNN is a supervised learning ML technique that doesn't make assumptions on the given data. On the contrary, it grows with the training set, so each training set can be considered as a new parameter [10].

3) Support Vector Machine: Support Vector Machine (SVM) is a technique based on the statistical learning theory. It has been used for solving classification and regression problems [11]. SVM divides the data set into two classes by specifying the linear classifier that will maximize accuracy and minimize error. This linear classifier is called hyper plane [12].

4) Logistic Regression: Logistic regression is a supervised machine learning model and is often preferred in algorithms that require binary classification [13]. Classification in logistic regression is done by dividing the data set into two with the help of a straight line. Separating parts of the data set are considered as two separate classes. In this study, logistic regression classifier algorithm was preferred because a binary classification algorithm (melanoma and nevus) was required.

B. Machine Learning Environments

There are many available open-source code tools and libraries that are frequently being used in machine learning especially in the case of health care applications. Using this study tools are as follows.

1) MATLAB: MATLAB©, which means "Matrix Laboratory", is a development environment designed for scientists and engineers to create functions, create models, draw data, develop applications, and more [14]. In Machine Learning and Data Analysis problems. It is used for high-level mathematical analysis.

2) Python: Python is a high-level language. It is used in data science and ML because of its flexibility, easy, open-source, and compatibility with many tools.

C. Data Pre-Processing

Data preprocessing is adapt the data to the requirements posed by each ML algorithm, and enable the user process complex data [15]. Data preprocessing includes data transformation, cleaning, normalization, and aims to reduce the complexity of the data by feature selection. After the application of a data preprocessing process, the final data set obtained can use for any ML algorithm applied afterwards [16].

D. Dataset

This study was used the Duzce University Faculty of Medicine Melanoma data set. In this data set, 150 different lesions were classified by experts in 2 groups as nevus and melanoma. input data of the study are age, gender, location, and tumor size criteria obtained as a result of the dermoscopic examination of the lesions.

TABLE I. DEFINING DERMOSCOPIC CRITERIA

Input Name	Value
Age	9-85
Location	Non = 0 Head = 1 Body = 2 Foot = 3 Hand = 4
Tumor size	1 – 40 mm
Sex	Women = 0 Man = 1

E. Evaluation Classification Performance

Evaluation of classification results is an important process in classification problems. There are three criteria for measuring the performance of a classification model: accuracy, specificity and sensitivity [17]. Table 2 gives the necessary parameters for the calculation of classification performance.

TABLE II. MEASUREMENT PARAMETERS TO EVALUATE THE PERFORMANCE OF A CLASSIFIER

Parameters	Description
True Positive (TP)	Cases of correct classification of diseases
True Negative (TN)	Cases of correct classification of no disease
False Positive (FP)	Cases of incorrect classification of disease
False Negative (FN)	Cases of incorrect classification of no disease.

Accuracy is the measure of the success of classification algorithms [17]. It is defined as the ratio of the correct number of states to the total number of states:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\% \quad (1)$$

Specificity is defined as the proportion of actual negatives, which got predicted as the true negative. Specificity can be calculated as the following:

$$\text{Specifity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\% \quad (2)$$

Sensitivity is a measure of the proportion of actual positive cases that got predicted as true positive. It is calculated as the following:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (3)$$

F. *Machine Learning Based Clinical Decision Support System*

CDSS, are smart software systems to support the expert decision, increasing the accuracy of the diagnose and fasting the diagnosing process. A CDSS estimates case-specific diagnosis by storing data on the patient's disease. Various ML algorithms and programming tools are required to develop intelligent CDSS software. Numerous ML and prediction algorithms are available and produce different results. Therefore, the appropriate learning and prediction algorithm should be selected for the decision-making system to give the most accurate result to the expert [18]. Advanced CDDSs such as QC PathFinder, TheraDoc, Sentri7, Safety Surveillor, and MedMined is widely used in hospitals and clinics around the world [19].

III. METHODOLOGY

In this study, 1440 clinic data were used. The classification has been made in MATLAB and Python development environments using SVM, KNN, Decision Tree, and Logistic Regression machine learning algorithms. Many factors affect the success of ML. The quality of the data set is crucial to achieving a high classification success. Therefore, a quality data set is obtained by applying pre-processing techniques such as data cleaning, data normalization, data transformation and feature extraction on the data set [20].

In this study, the data processing steps were applied on the melanoma data. Thus, incomplete and meaningless data has been eliminated. The quality of the data set was increased by removing the duplicate data and turning it into a clean data set. 80% of the data set was used for training all algorithms. The remaining data are reserved for testing the classification models.

In the first classifications, in MATLAB environment in the online Matlab Classification Learner tool using SVM, KNN, Decision Tree, and Logistic Regression algorithms were performed by applying 20 fold cross-validation. In the second classifications, in Python 3.7 version was applied by using the Scikit Learn library that includes machine learning algorithms SVM, KNN, Decision Tree, and Logistic Regression algorithms.

As a result of the comparisons, the highest success was achieved with the KNN algorithm in the Python environment. A model has been created with the KNN algorithm to be used in decision making in CDDS. A database has been created for use in recording patient data received for diagnosis. The personal information of the patient and the dermoscopic data to be used by the specialist in diagnosis are recorded in the created database. The recorded data are sent to the KNN model for classification. Estimation of the KNN model is presented to the specialist physician.

IV. EXPERIMENTAL RESULT

Classification performances of KNN, SVM, Logistic Regression and Decision Tree algorithms in Python and Matlab environments were compared in melanoma data set. In all algorithms, 80% of the data set was used as training, 20% as a test, a random selection value of 10 and cross-validation value 20. In the KNN algorithm, the value of the n_neighbors parameter is determined as 5 optimum value. In the SVM algorithm, kernel value is sigmoid, degree value is 3, and gamma value is auto. In Decision Tree and Logistic Regression, algorithms are used default values.

TABLE III. ACCURACY SCORES OF MACHINE LEARNING ALGORITHMS

	Accuracy	
Algorithm	***Matlab***	***Python***
KNN	81.2%	93.3%
SVM	78.5%	81.6%
Logistic Regression	75.8%	80%
Decision Tree	66.8%	69.5%

Table 3 shows a comparative analysis for four different machine learning algorithms. According to the results in Table 3, the algorithm with the highest success in Python and Matlab environments is KNN. The Decision Tree algorithm showed the lowest performance in both environments. The Python environment has achieved higher results than Matlab. As a result, it has been revealed that the KNN algorithm can make accurate predictions in the melanoma classification problem with high classification success. In this way, it can help the specialist physician make the correct diagnosis in the case of melanoma.

V. CONCLUSIONS

In this study, the detection of melanoma from clinic evaluations was achieved at a high rate with ML classifiers. Four different ML algorithms have been compared with each other and in two different development environments and their effect on performance has been examined. The highest classification success was obtained with the KNN classifier in Python program with 93.3%. The success of the classification shows that CDSS can assist the expert in the diagnosis of melanoma with high accuracy.

ACKNOWLEDGMENT

The authors would like to thank the Duzce University Department of Pathology for contributions to this work.

REFERENCES

[1] Barati, E., Saraee, M. H., Mohammadi, A., Adibi, N., & Ahmadzadeh, M. R. (2011). A survey on utilization of data mining approaches for dermatological (skin) diseases prediction. *Journal of Selected Areas in Health Informatics (JSHI)*, *2*(3), 1-11. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Karakök Güngör H, Akay BN. İlerlemiş Melanomada Kullanılan Güncel Tedavi Yöntemleri. Turk J Dermatol 2016; 10:137-144 K. Elissa, “Title of paper if known,” unpublished.

[3] Kizilbey, Kadriye, and Z. Akdeste. "Melanoma cancer." Sigma: Journal Of Engineering & Natural Sciences

[4] Baig, M. M., Hosseini, H. G., & Lindén, M. (2016, November). Machine learning-based clinical decision support system for early diagnosis from real-time physiological data. In *2016 IEEE region 10 conference (TENCON)* (pp. 2943-2946). IEEE.

[5] Sim, I., Gorman, P., Greenes, R. A., Haynes, R. B., Kaplan, B., Lehmann, H., & Tang, P. C. (2001). Clinical decision support systems

for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, *8*(6), 527-534.

[6] Chang, C. L., & Chen, C. H. (2009). Applying decision trees and neural networks to increase the quality of dermatologic diagnosis. *Expert Systems with Applications*, *36*(2), 4035-4041.

[7] Kolkur, S., Kalbande, D. R., & Kharkar, V. (2018). Machine learning approaches to multi-class human skin disease detection. *International Journal of Computational Intelligence Research*, *14*(1), 1-12.

[8] Verma, A. K., Pal, S., & Kumar, S. (2019). Comparison of skin disease prediction by feature selection using ensemble data mining techniques. *Informatics in Medicine Unlocked*, *16*, 100202.

[9] Siddiqui, Z. U., & Ali, I. (2019, December). Comparative Study Of Classification Techniques On Occurrence Of Breast Cancer Disease. In 2019 4th International Conference on Emerging Trends in Engineering, Sciences, and Technology (ICEEST) (pp. 1-4). IEEE.

[10] You, H., & Rumbe, G. (2010). Comparative study of classification techniques on breast cancer FNA biopsy data.

[11] Prakash, R., Tharun, V. P., & Devi, S. R. (2018, April). A Comparative Study of Various Classification Techniques to Determine Water Quality. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1501-1506). IEEE.

[12] Hüdaverdi BİRCAN.(2004).Lojistik Regresyon Analizi: Tıp Verileri Üzerine Bir Uygulama.Kocaeli Üniversitesi Sosyal Bilimler Dergisi

[13] Nadler, D. W. (2019). Decision support: using machine learning through MATLAB to analyze environmental data. Journal of Environmental Studies and Sciences, 9(4), 419-428.

[14] Masood, A., & Ali Al-Jumaily, A. (2013). Computer-aided diagnostic support system for skin cancer: a review of techniques and algorithms. *International journal of biomedical imaging*, *2013*.

[15] Pyle, D. (1999). *Data preparation for data mining*. morgan kaufmann.

[16] García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*, *1*(1), 1-22.

[17] Amirjahan, M., & Sujatha, D. N. (2016). Comparative analysis of various classification algorithms for skin Cancer detection. *PG & Research Department of Computer Science, Raja Doraisingam Govt. Art College, Sivagangai, Tamil Nadu, India*, 199-205.

[18] e Costa, C. A. B., Ensslin, L., Cornêa, É. C., & Vansnick, J. C. (1999). Decision support systems in action: integrated application in a multicriteria decision aid process. *European Journal of Operational Research*, *113*(2), 315-335.

[19] Anakal, S., & Sandhya, P. (2017, December). Clinical decision support system for chronic obstructive pulmonary disease using machine learning techniques. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)* (pp. 1-5). IEEE.

[20] Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science*, *1*(2), 111-117.

Implementation of One Stop Shop model for Government Services

Ina Hyseni
Faculty of Natural Sciences
Department of Computer Science
Tirana, Albania
ina.hyseni@fshn.edu.al

Endri Xhina
Faculty of Natural Sciences
Department of Computer Science
Tirana, Albania
endri.xhina@fshn.edu.al

***Abstract*— The development of a platform that would enable local government units to serve the citizens has always been a challenge in Albania. Generating and developing this model has consisted in analyzing the current processes, figuring out all the services that local government unit (LGU) are offering, designing, implementing and testing. This paper analyses the platform created during the process of defining a redesign of a large information system: the information plan, the data conversion, the interaction and the release strategy. The IOSS model for local government unit implements a service delivery model based on the One Stop Shop service delivery philosophy.**

Keywords*— *one-stop-shop, process, government, software, service

I. INTRODUCTION

The Integrated One Stop Shop (IOSS) model for local government implements a service delivery model based on the One Stop Shop service delivery philosophy. The importance of online one-stop government in public service delivery is recently pointed out in a study of the European Commission [1].

In this model, Administrative Units serve as a service desk, responsible for accepting applications for administrative services and providing the final response to the citizen. Many countries worldwide have set the realization of e-Government as one of their primary targets [2]. The European Union, through the Europe initiative, recognizes that e-Government - and also one-stop government could provide significant advantages to citizens, business and the public sector itself [3]. The integration is created through a data network that connects the Administrative Units with the Municipality, while process automation software is used to provide the necessary functionalities to initiate service requests at desks (front office) of the administrative units and review them at the administration offices (back office) in the Municipality. The one-stop-shop idea enables people and clients to have a single area to access to information and service transactions [4]. The integrated one-stop shop model will enable citizens access to information 24 hours a day, 7 days a week from public, private and government sectors to increase the quality of those services [5]. IOSS model has three main components which provide the basis for the functionality of this model, administrative procedures, software system and training curricula. This Software System enables the implementation of IOSS model and has a number of features that guarantee a cost effective, flexible and self-sustainable system. The conceptual model of the IOSS is illustrated on fig. 1.

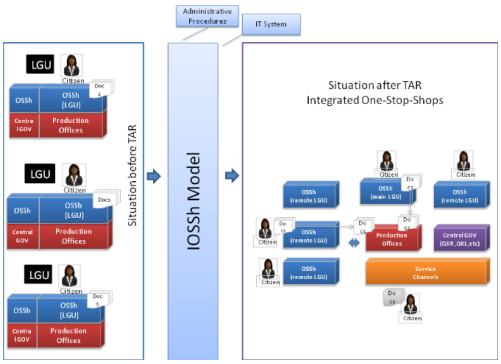


Fig. 1. Conceptual Model of IOSS.

II. OSSH SOFTWARE

A. *Smart Processes software*

A good understanding of administrative processes is crucial for adapting public services to one-stop government. Proper comprehension is no small problem, as in the field of operational administrative action, a huge variety of different processes can be encountered [2]. Most processes are rather complex due to several causes (cf. [6]).

The complexity of many public services grew during the twentieth century, and the functional specialization of the traditional bureaucracy ran counter to the nature of many citizen errands, which involve interactions with multiple authorities [7]. The software created for the implementation of IOSS model provides a straight forward method of entering the prepared extended service templates into it, and start running cases for that service requests. The software has a key feature which guaranties a long life of the IOSS system, which is “the capability to accommodate changes”, since this is the biggest challenge to be faced by IOSS model at municipalities.

The IOSS model is implemented in process automation software called Smart Processes . The software is web based and provides features which enable the IOSS to be:

- Flexible to changes
- Customized Data Entry Interface for each individual service (Case Management instead of file tracing)
- Data Privacy Control for Scanned Documents
- Digital Signing Capability
- Able to provide multi-channel services
- Able to integrate with Back Office Systems or other Gov Systems

- Able to serve as a platform to provide services on behalf of other institutions

B. Ability to accommodate changes

Processes are designed with a friendly use concept, web based drag and drop interface in all the components of a process:

- Data entry forms
- Workflow (cases statuses, steps and rules)
- User Roles
- Generated Printable Forms
- Attached Scanned Documents
- Communication with other systems
- Communication with other channels (email or SMS notification on events)

After being created, the processes are sent for execution in the "Cases Portal", which is the main workspace of IOSS users. A similar environment with limited functionality is also provided in the web portal to the citizens. This method provides a level of functionality which guaranties that the

IOSS system is able to adapt to legal changes which might affect the administrative services. This is an important aspect for the sustainability of the IOSS in the conditions when several reforms are on the process in Albania. The following scenario aims to illustrate the ability of the IOSS system to accommodate changes.

III. Sample Scenario: Add a new service in the IOSS system

This scenario explains how a new service is added into the IOSS. The chosen service is "Request for Financial Assistance in case of Natural Disaster". This service is added to the IOSS system in Shkodra and Lezha, the first week of January, 2016 due to the fact that water floods have caused damages to the citizens in these municipalities. The new service is added by the local IT Administrator, which in this case have executed the "Process Designer" role, indicated in the Fig. 2.

The "Process Designer" should not be necessary an IT professional; it could also be an administrative service domain expert, which has the necessary training to use Smart Processes. This is possible because creating new processes with the most common features is an easy task done with a drag and drop interface. Configuring advanced features such

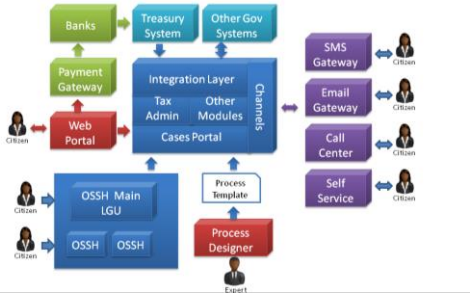


Fig. 2. Role of the Process Designer.

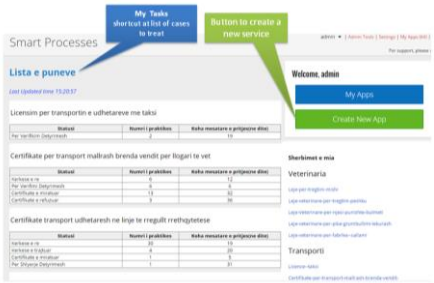


Fig. 3. Adding a new service.

as child processes, web service integration or notification events is also done through a drag and drop interface, however IT knowledge about web services and events is required.

A. Creating the process

The print screens that are edited demonstrate some of the steps from the application of a new service creator(illustrated on Fig. 3). The "New App" button creates a new service. Afterwards the process configuration is opened.

B. Configure the process

Process Designer then configures the process in terms of:

- General parameters (process name, a short description, a process category, and a process icon),
- Process Capacity (Maximum Number of Cases per Process, Maximum Number of Cases per Case Initiator ,Maximum Number of open Cases per Case Initiator),
- Process Availability (Time when it is possible to start creating cases in this process,
- Deadline up to which is possible to create new cases in this process,
- Deadline up to which is possible to modify cases in this process). Afterward the Data entry form(s) have to be designed through the Form Designer. (illustrated on Fig. 4)

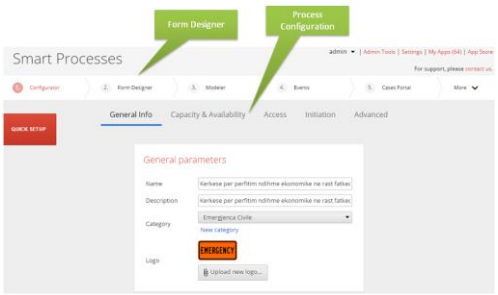


Fig. 4. Configure the process.

C. *Creating data entry forms*

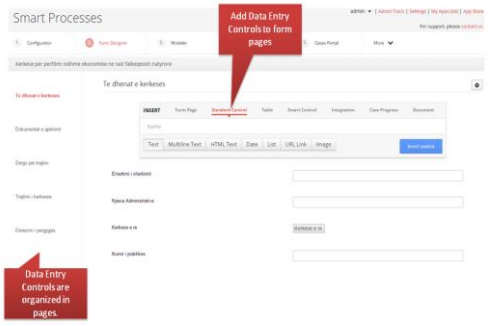


Fig. 5. Creating data entry form.

Each process has a data-entry form. Data-entry controls are organized into pages. Pages are a container object which is refereed in the access rights configuration, thus providing different views of the data entry form for different user roles depending on the status of the individual case of the processes. All input controls are added through form designer including attachments and printed out forms (called SmartDocs in Smart Processes). (illustrated on Fig. 5)

D. *Design the service workflow*

Each process has a data-entry form. Data-entry controls are organized into pages. Pages are a container object which is referenced in the access rights configuration, thus providing different views of the data entry form for different user roles depending on the status of the individual case of the processes. All input controls are added through form designer including attachments and printed out forms (called SmartDocs in Smart Processes) (illustrated on Fig. 6).

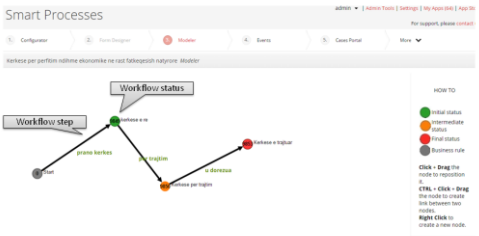


Fig. 6. Design the workflow.

E. *Prepare the Printout Forms (SmartDocs)*

The process designer prepares a Microsoft Word document with content and named bookmarks. This template will be used to generate the final report. The document is attached in the SmartDocs configurator. After this, mapping should take place, it is an easy process where you have to map the bookmark field and find the appropriate bookmark, previously added in the Microsoft Word document (illustrated on Fig. 7).

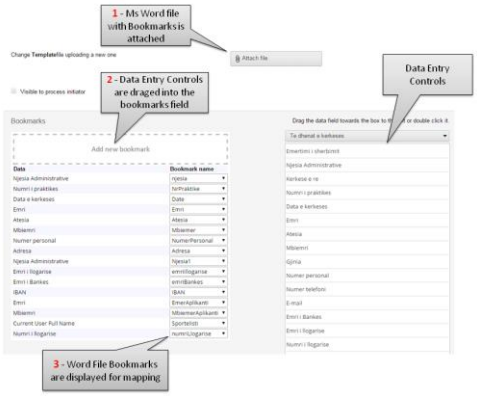


Fig. 7. Prepare the Printout Forms (SmartDocs).

F. *Managing the cases*

The main view of the web portal is used to create and manage cases in this process. All the data are structured and set under their specific category. There are two different menus, the right side one and the top one. The right menu contains information related to the selected case, while the top menu shows the different types of services that the web portal is offering (illustrated on Fig. 8).

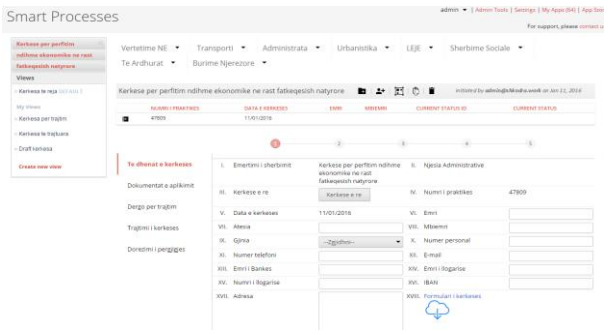


Fig. 8. Cases Portal.

IV. BENEFITS

A. *Case Management instead of file tracing*

As demonstrated in the sample scenario of creating a new service, each component of a service is created through a drag and drop interface. This allows the creation of customized data-entry forms for each service, depending on the data that the service requires. This approach provides several advantages compared to an approach where only data that has to do with the request handling is captured, and get information required in the attached documents:

- Data capture at the application eases the application/request review through filtering, dynamic routing of cases using data based rules etc.;
- A database with browse able data is created as a normal working procedure;
- A detailed analysis of the service delivery performance could be done based on the data generated by created cases;
- Back Office enters data for the application/request review in the case practice created in the application, herewith providing a transparent and easily auditable case management practice.
- Integration of the cases with other systems such as back office software or other Gov Systems could be implemented to read and write data from the cases.
- Integration with other systems allows operating as an interface for services provided by other institutions.

B. *Data Privacy Control for scanned documents*

There are regulatory guidelines that obligate LGUs for data privacy protection of the scanned documents, provided but the citizen at the application for a service. The web-based scanning tool part of the Case Management triggers the scanning device directly from the browser, thus scans the documents without leaving a copy of the scanned document in the local desk computer.

C. *Digital Signing Capability*

Smart Processes has implemented the secure authentication and digital signing services provided by Aleat, the responsible company for Secure Authentication and Digital Signing. Given that, a commercial agreement between the municipality and the service provider is established, it would be easy to activate the authentication and signing of digital signing of the documents.

D. *Multichannel services*

Beyond the service desks, IOSS system has:

- A web portal which would allow the citizen to get information about the status of his service case/application. In addition, the web portal provides the option of application online for an administrative service.
- Integration with SMS Gateway to trigger messages based on Case/Application Events.
- Self-service kiosks could also be connected to IOSS system to provide status update of the cases/application.

E. *Ability to integrate with Back Office Systems or other Gov Systems*

Despite the research available, data integration remains a major challenge for governments [8]. For instance, when data are integrated, it becomes more difficult to keep them secure and to ensure their use does not violate citizens' rights to privacy [9]. Because of this, the number of governments that have achieved high levels of data integration is small.

The software, on which the IOSS system is based, allows through the feature “Web Service Event” the configuring of an event which calls a web service through an easy to use “drag and drop” interface.

- Process Designer enters the web service address (URL).
- Smart Processes will query the metadata of the web service and list its published methods.
- Process Designer chooses the desired method of the web service to be called.
- Smart Processes will query the metadata of the web service and list the arguments of the chosen method.
- Process Designer maps the data-entry fields of the process with the arguments of the web method.

After this configuration steps when the triggering case event will fire the web service method will be called and the data from the case will be synchronized with the mapped arguments of the web service method.

For example, when a tax-id data-entry field is updated a web method is called which takes the tax-id values as an input argument and provides the name of the company as an output argument. The output argument will be written to the field value of the data-entry form to which it is mapped in the configuration.

This feature enables IOSS system to integrate very easily with other systems through standard web service interface. This integration can be configured by the IT Administrator of the municipality and does not require any additional software programming.

F. *Ability to serve as a platform to provide services on behalf of other institutions*

The web service interface allows the IOSS system to operate as an interface for other systems. The above feature enables scenarios where citizens make request for services offered by other institutions through the IOSS system. IOSS system will store the request, send it to the back end system of the institution through web service, query the status of the case from the back end system and provide the answer to the citizen. This feature allows IOSS to serve as a platform to provide services on behalf of other institutions.

G. *IOSS system base technology*

Smart Processes are developed on top of Microsoft .NET framework and use Microsoft SQL Server as a database system. Smart Processes is fully operational when running on top of SQL Server Express Edition. The limit of 10GB per database imposed by SQL Server Express Edition is far

beyond the expected database size increase of IOSS database. The estimated yearly database size increase is less than 500 MB per year, while the current database size of both IOSS systems implemented is less than 600 MB. The projected database size after 10 years would be less than 6GB. This simple calculation demonstrates that a fee-based database license is not required for IOSS system.
Due to the fact that is based on widely known programming languages and environments in Albania (Microsoft .NET programming languages and SQL Server) IOSS system gives the necessary level of independence to the Municipality to develop other functionalities connected to the IOSS system using its internal resources or external contracts.

V. Conclusions

The ultimate goal of these reforms is to solve problems with a malfunctioning government bureaucracy for frontline service delivery, ensuring that all citizens have access to the basic personal documents needed in a variety of life situations, as well as access to basic public services [10].

The IOSS system is under implementation phase in the Municipalities of Shkodra and Lezha, including Administrative Units of Dajc, "Bregu i Bunes", Velipoje, Kallmet and Shengjin. The implementation of IOSS system first started in Lezha and after 6 months of success started the implementation in Shkodra. Up to now, there are 86 digitalized services configured and active in both municipalities.

The implementation projects in both municipalities aims to consolidate the digitalized services as well as to enter new services which might emerge due new functions delegated to the Municipalities.

The so far implementation of the IOSS in the municipalities of Shkodra and Lezha generated interesting conclusions which would help other LGUs to plan and execute the projects for IOSS implementation. The tables below provide an overview of the configured services in the two municipalities.

Table 1. Overview of Services Configured in Lezha Municipality

Services Configured	
Number of Services	78
Average Forms per Service	3.62
Average Workflow Steps per Services	5.55

Table II. Overview of Services Configured in Shkodra Municipality

Services Configured	
Number of Services	67
Average Forms per Service	6.66
Average Workflow Steps per Services	6.54

The tables above are an illustration of the fact that it is not possible to have a unified IOSS system for all municipalities:

1. While legal framework assigns functions to the Municipalities, the way how those functions are translated into services is subject to the decision of the municipality

2. Different Organizational Structures and different management decisions will create different workflows which would provide different printed forms to accompany the workflow of a service case.

Based on the situation and how the real implementation is currently going, this software is able to accommodate changes. The IOSS system must allow to reconfigure the workflow, the forms printed user roles in a fast way without additional software programming. The IOSS system must allow creating new services without additional software programming. This IOSS system feature must be available in a simple user interface in order to be used by the Municipality's personnel, without having to contract an external company for this task.

"Fig. 9" below provides evidence on the number of services digitalized in both municipalities of Lezha and Shkoder, based on the reference list of 67 services unified during the study phase of the project.

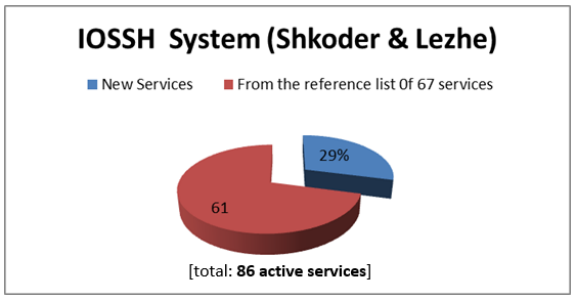


Fig. 9. IOSS System.

References

[1] eEurope. Web-based Survey on Electronic Public Services, available at http://europa.eu.int/information_society/eeurope/egovconf/documents/pdf/eeurope.pdf [10/01/2002].

[2] Wimmer, Maria A. & Tambouris, Efthimios. (2002). Online One-Stop Government: A working framework and requirements. 10.1007/978-0-387-35604-4_9.

[3] eEurope2002. An Information Society For All, Action Plan of the European Commission, available at http://europa.eu.int/information_society/international/candidate_countries/doc/eEurope_june2001.pdf [10/01/2002].

[4] Gashi, K., & Krasniqi, A. D. I. (2019). The One-Stop Shop Approach: New Public Management Model in Transition Countries.

[5] Ndou, V. (2004). E–Government for developing countries: opportunities and challenges. The electronic journal of information systems in developing countries, 18(1), 1-24.

[6] Lenk, K., Traunmüller, R., Wimmer, M. The Significance of Law and Knowledge for Electronic Government, in Grönlund (ed.), "Electronic

Government - Design, Applications and Management", Idea Group Publishing, 2002, pp. 61-77.

[7] Kubicek, H., & Hagen, M. (2000). One stop government in Europe: An overview. In M. Hagen & H. Kubicek (Eds.), One stop government in Europe. Results from 11 national surveys (pp. 1– 36). Bremen, Germany: University of Bremen.

[8] Kim et al., 2014 G.-H. Kim, S. Trimi, J.-H. Chung Big-data applications in the government sector Communications of the ACM, 57 (3) (2014), pp. 78-85

[9] Scholta, H., Mertens, W., Kowalkiewicz, M., & Becker, J. (2019). From one-stop shop to no-stop shop: An e-government stage model. Government Information Quarterly, 36(1), 11-26. doi:10.1016/j.giq.2018.11.010

[10] Fredriksson, A. (2020). One Stop Shops for Public Services: Evidence from Citizen Service Centers in Brazil. Journal of Policy Analysis and Management, 39(4), 1133-1165. doi:10.1002/pam.22255

Identifying Pneumonia in SARS-CoV-2 Disease from Images using Deep Learning

Abdulkadir Şahiner
[1] Department of Computer Engineering,
Istanbul Sabahattin Zaim University
[2] Yildiz Technical University,
Dept. of Mathematical Engineering
Istanbul, Turkey
0000-0002-5528-2733

Alaa Ali Hameed
Department of Computer Engineering,
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-8514-9255

Akhtar Jamil
Department of Computer Engineering,
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0002-2592-1039

***Abstract*— Deep learning methods are commonly used in various applications now a days. It has also shown its effectiveness in the field of medicine and is actively used as an auxiliary technology. Although the COVID-19 epidemic is an unprepared process in the world, it has been one of the times when alternatives were most needed for diagnosis and treatment. Considering the cost of tests such as PCR in the diagnosis of the COVID-19 epidemic, it has become a need to create economical alternatives. In this context, the main objective of this study was to diagnose and distinguish between COVID-19 and pneumonia cases with CNN-based multiple models using images of chest X-rays. Specifically, three different CNN-based models were used, namely: InceptionV3, ResNet50, and InceptionResNetV2. Moreover, different optimizers were also investigated to identify the best performing one. The models were trained on chest X-ray of 100 patients with COVID-19 and pneumonia. The experimental results showed that the SGD optimizer with the highest accuracy came to the fore with a value of 98.8%.**

Keywords*— *pneumonia, COVID-19, SARS-CoV-2, Deep Learning

I. INTRODUCTION

The epidemic caused by the SARS-CoV-2 virus and commonly known as COVID-19 was first seen in Wuhan, China, and affected the whole world. The COVID-19 epidemic declared as a pandemic by the World Health Organization (WHO), has affected 82,579,768 people worldwide and caused the deaths of 1,818,849 people today (03.01.2020) [1, 2].

The mortality rate is very high as there is no specific drug or treatment method is available to treat the disease. Therefore, it continues to be an epidemic that people are very worried about around the world [3].

The similarity between pneumonia disease and COVID-19 epidemic disease makes it inevitable to confuse COVID-19 with pneumonia disease in this process. It can be stated that technology can be a helpful factor in distinguishing two diseases since the number of experts trained in this field is insufficient [4].

When pneumonia disease, also known as pneumonia, is not diagnosed early and correctly, it can lead to the death of patients, although not as high as the high mortality rate of COVID-19 disease today [5]. Pneumonia can be expressed as a disease that causes severe respiratory distress by affecting the lungs with the effect of viruses and bacteria [5]. Pneumonia causes damage to the person by making breathing difficult as a result of the damage to the air sacs that must be active in the lungs during breathing. This disease usually affects patients with weakened immune systems and can cause death [6].

COVID-19 epidemic disease is caused by the SARS-CoV-2 virus and in humans; It presents with symptoms of coughing, high fever, sore throat, and difficulty breathing. Along with these symptoms, different symptoms such as taste difficulty and muscle pain can be seen. The virus transmitted from person to person through channels such as coughing shows its effects during the 14-day incubation period [7].

In this study, it was aimed to compare the results with other studies in the literature by applying the Convolutional Neural Network (CNN), which is a deep learning model for the determination of X-rays of COVID-19 and pneumonia patients using data sets consisting of chest X-rays. The results obtained within the scope of the research are presented in tables and graphics. The results obtained within the scope of the research; It has been compared with other studies in the literature within the scope of values such as accuracy, loss, and sensitivity.

II. LITERATURE REVİEW

As the importance and effect of pneumonia disease after the COVID-19 epidemic is repeated more frequently, understanding the difference between these diseases directly affects the treatment processes. In this context, the number of studies in which models are developed to distinguish COVID-19 and pneumonia from chest X-rays using deep learning techniques is increasing. Details of some of the studies in the literature are as follows:

Within the scope of the study, it is seen that the model created with deep learning techniques using two open data sets of COVID-19, pneumonia, and normal patients was tested and the results were shared. In the study, a model created by the combination of Xception and ResNet50V2 networks is proposed. The average accuracy of the proposed model in detecting COVID-19 cases was 99.50%, and the overall mean accuracy for all classes was 91.4% [10].

In a different study, it was provided to train with datasets on COVID-19 and normal (healthy) human chest X-rays using CNN models. Within the scope of the study, CNN models ResNet18, ResNet50, ResNet101, VGG16, and VGG19 were used. During the experimental process of the study, measurements of classification accuracy were made using a data set containing 180 COVID-19 and 200 normal (healthy) chest X-rays. Within the scope of the created model, the highest result obtained as a result of the deep features extracted from the ResNet50 model and SVM classifier was observed as an accuracy score of 94.7%. It has been emphasized that CNN models and SVM classifiers are highly efficient in the detection of COVID-19 compared to local tissue descriptors [11].

By using chest X-rays and tomography films, the researchers used CNN, which is one of the deep learning

models, in the process of identifying COVID-19 by training the data. Within the scope of the research, a new model named CoroDet based on CNN was proposed. In the proposed model 2 (COVID-19, Normal), 3 (COVID-19, Normal, Non-COVID-19 Pneumonia) and 4 (COVID-19, Normal, non-COVID-19 viral pneumonia and non-COVID-19 bacterial pneumonia) classifier is used . The accuracy values of the proposed model are expressed as 99.1% in a 2-class classifier, 94.2% in a triple classifier, and 91.2% in a 4-class classifier. As a result of the research, it was stated that the CoroDet model is superior to existing technologies and can be helpful in decision-making in clinical processes [12].

One of the healthiest models recommended for diagnosis during the COVID-19 pandemic is PCR tests. However, in the study, which stated that alternatives could be created in diagnostic processes with technological different methods due to the limited number of test kits, [13], it was aimed to investigate a new model to be created with a pre-trained multiple CNN model for COVID-19 chest X-ray films. Open data sets were used in the research. Within the scope of the research, in the first data set consisting of 453 COVID-19 chest X-rays and 497 non-COVID chest X-rays, an accuracy of 0.963 AUC and 91.16% was achieved with the model created. In addition, in the second data set consisting of 71 COVID-19 chest X-ray images and 7 non-COVID chest X-rays, an accuracy of 0.911 AUC and 97.44% was achieved with the model created. As a result of the research, it was stated that pre-trained multiple CNNs were more effective than single CNN in the diagnosis of COVID-19.

F In a different study, a model using chest X-rays was proposed because of the higher cost of obtaining tomography images. In addition, the fact that the images that are easier to access in hospitals are chest X-rays is also presented as a factor. Within the scope of the research, a new model created from multiple CNN models was proposed and compared with different CNN models. COVID-19 is used as the classifier, and a 3-classifier is used as the other. As a result of the research, accuracy of 99.5246% was obtained. It has been stated that the proposed model gives more effective results compared to other CNN models (VGGNet, ResNet50, Alexnet, Googlenet InceptionnetV3, etc.) [7].

The effect of a new CNN model-based model was investigated using data sets consisting of images of chest X-rays. The name of the new model proposed within the scope of the research was determined as CovXNet and normal, pneumonia (viral and bacterial), and COVID-19 chest x-ray images obtained from open data sets were used. For the COVID / Normal classifier the accuracy was 97.4%, for the COVID / Viral pneumonia classifier 96.9%, for the COVID / Bacterial pneumonia classifier 94.7%, and for multi-class COVID / normal / Viral / Bacterial pneumonia% An accuracy of 90.2 has been achieved. It can be stated that the model proposed as a result of the research can serve as a diagnostic tool in the current situation of the COVID-19 pandemic [4].

In the study where a new model based on the CNN model was proposed, a data set consisting of images of chest X-rays was used. Within the scope of the model, images of chest X-rays were trained with a 12-class structure and it was stated that an accuracy value of 86% was reached as a result of testing the model [14].

In a different study [15], it was aimed to identify tuberculosis patients with a CNN-based model using images of chest X-rays. The accuracy value obtained as a result of testing the developed model was specified as 85.68%.

Information on other studies conducted is summarized in Table I. The results presented in these studies indicate that deep learning-based techniques are very effective for identification of COVID-19 positive cases.

TABLE I. INFORMATION ON OTHER STUDIES IN THE LITERATURE

No	Name of Study	Purpose of Study	Result
1	Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images [16]	It is aimed to identify the COVID-19 patient with the CNN-based model, using images of chest X-rays.	As a result of the research, the model using the data set using the images of 13,975 chest X-rays reached 98.9% accuracy.
2	Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images [17]	Automatic diagnosis of COVID-19 disease is aimed with a newly developed model named COVIDX-Net.	Images of 50 normal and 25 COVID-19 patients were used within the scope of the study, and the accuracy rate of the proposed model named COVIDX-Net was expressed as 91%.
3	Deep learning based detection and analysis of COVID-19 on chest X-ray images [18]	Images of chest X-rays of healthy and COVID-19 patients were processed and analyzed comparatively using CNN models.	As a result of the research, in the study in which Inception V3, Xception and ResNeXt models were used, 5467 of the open data sets were used for training models and 965 for verification. When the comparison results are examined, the Xception model has achieved 97.97% results compared to other models.
4	Cascaded deep learning classifiers for computer-aided diagnosis of COVID-19 and pneumonia diseases in X-ray scans [19]	Within the scope of the study, it is aimed to train the data set consisting of images of COVID-19, non-COVID-19 viral pneumonia and non-COVID-19 bacterial pneumonia chest X-ray films with VGG16, ResNet50V2, and Dense Neural Network (DenseNet169) models.	Within the scope of the study, the data set consisting of images of COVID-19, non-COVID-19 viral pneumonia and non-COVID-19 bacterial pneumonia chest x-ray films were trained with VGG16, ResNet50V2, and Dense Neural Network (DenseNet169) models, and as a result, 99.9% accuracy was achieved.

III. MATERIAL AND METHODS

The sample architecture of the proposed method used in this study is shown in Fig. 2. Although the model to be used in the research is the CNN model, the images were initially resized as 224 × 224 pixels in order to be suitable for this model.

Brief information about the CNN model as the deep learning model suggested in the study is as follows:

Deep learning models are used quite effectively in processes such as examining medical data, classification, and segmentation. As a result of processing these data with deep learning models, it can be used to facilitate the diagnosis of epidemic diseases such as pneumonia and some types of cancer as well as epidemic diseases such as COVID-19 [20, 21].

The Convolutional Neural Network (CNN) model can be defined as a class of deep neural networks used in image recognition problems [22].

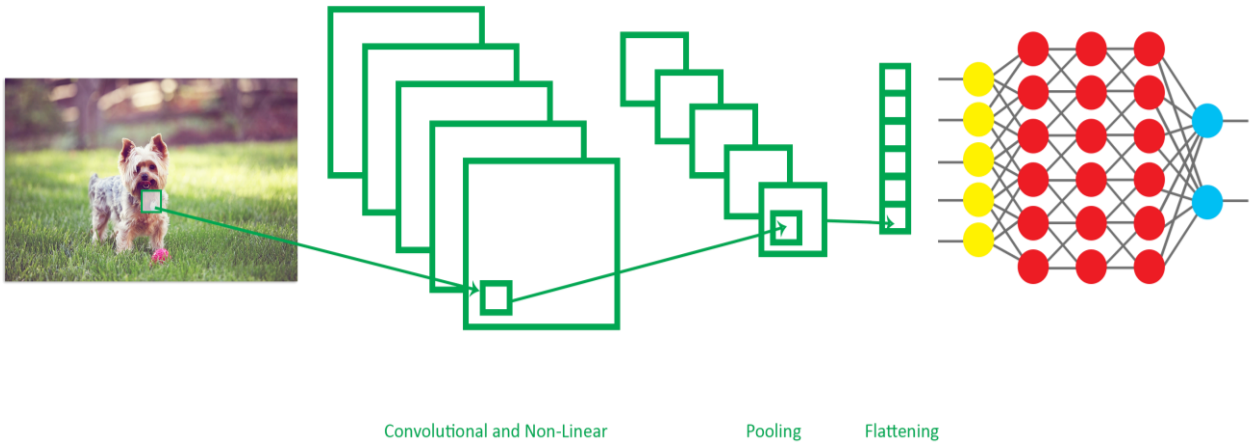


Fig. 2. CNN Model Layers Diagram [23]

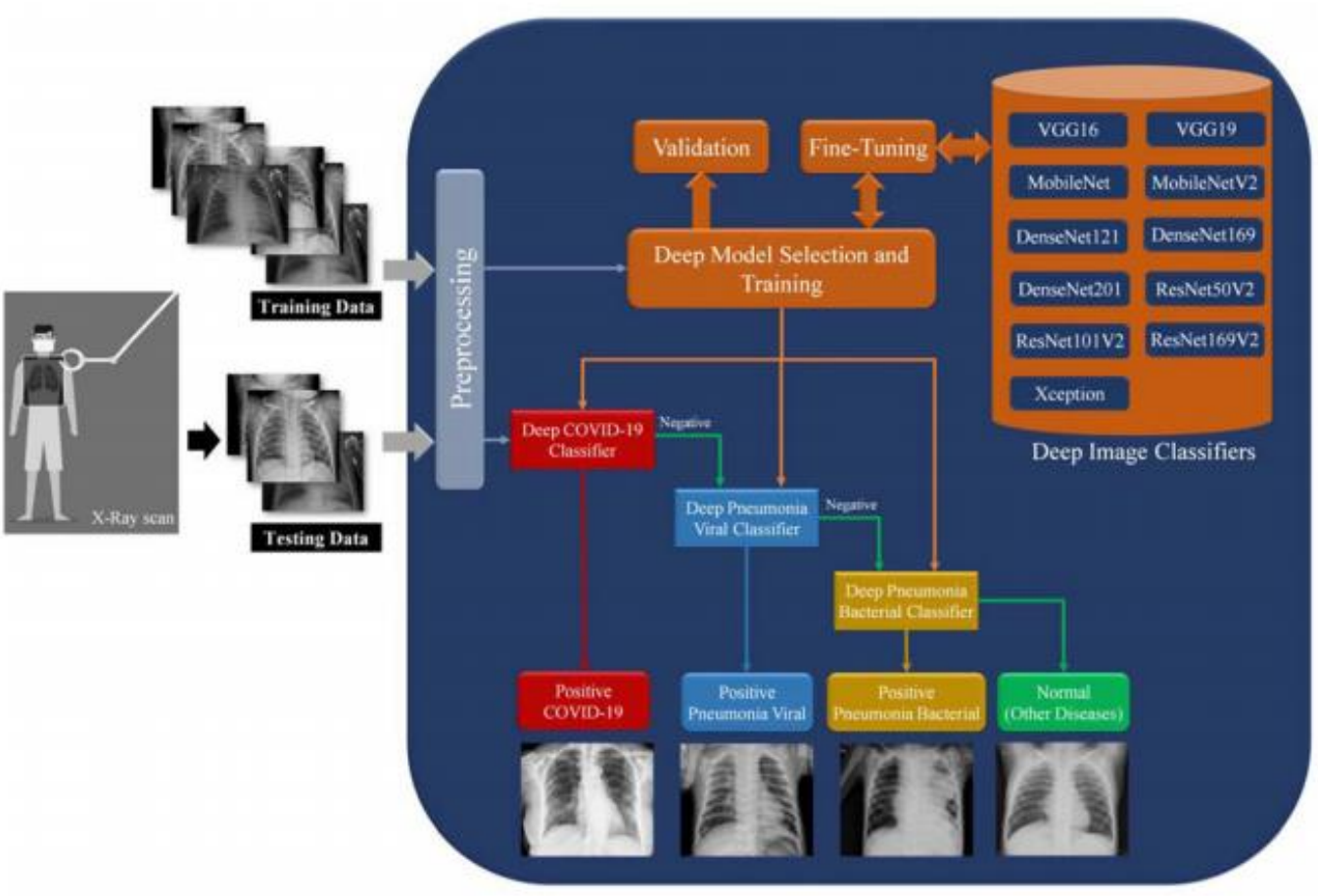


Figure 3. Work Flow of the CNN-Based Proposed Model within the Scope of Identifying COVID-19 and Pneumonia Patients Using Chest X-rays [19]

The working principle of the CNN model can be expressed as converting the images taken as input into a format that can be processed by the computer. Accordingly, the images are converted into matrix format and they try to determine which label the new images belong to by learning the effects on the label according to the image differences, namely the matrix changes. The learning process of this model consists of different layers. These layers are named as a convolutional layer, pooling layer, and fully connected layer [22].

TABLE II DATA SET USED IN THIS STUDY

	Train	Train	Test	Test
Pneumonia	3875	40	390	10
SARS-CoV-2	930	40	930	10
Total	**4805**	**80**	**1320**	**20**

3.1. Data Set

Two open data sets were used within the scope of the study. The first of these data sets is Dr. Joseph Cohen et al. It was obtained from chest X-ray images of 930 COVID-19 patients created and shared on the GitHub platform [9]. The second one is the data set shared by Paul Mooney under the name of "Chest X-Ray Images (Pneumonia)" on the Kaggle platform [8]. The second data set contains 5856 images in total. All images in the data set are adjusted to 224 x 224 size. Information on the data set used in this study is summarized in Table 2.

The CNN models used within the scope of the study are InceptionV3, ResNet50, and InceptionResNet V2. These models can be briefly explained as follows:

3.2. Deep Learning Models Used in the Study

Explanations regarding the model used in the study are as follows:

InceptionV3: It is a kind of CNN-based model. In this model, the pooling steps are maximum and form a connected neural network structure at the last stage [24].

ResNet50: It is an improved version of the CNN model. By creating shortcuts between layers to solve the problem, provides convenience in the complex structure and prevents deterioration [25].

InceptionResNetV2: It is a kind of CNN-based model. In this model, it can be expressed as an estimated class probability list with 299 * 299 images to be trained in the ImageNet 2012 data set [26].

3.3. Experimental Setup

Python programming language was used in the training of the model proposed in the study [27]. The experiments carried out within the scope of the study were carried out using the Central Processing Unit (CPU), Graphics Processing Unit (GPU), or Tensor Processing Unit (TPU) hardware and the online cloud service. CNN models (ResNet50, InceptionV3, and Inception-ResNetV2) are pre-trained with random start weights by optimizing the cross-entropy function with the optimizer (1 = 0.9 and β2 = 0.999) called adaptive moment estimation (ADAM). Group size, learning speed, and a number of periods were experimentally set to 2, 1e-5, and 30 for all experiments, respectively. The data set used was randomly divided into two independent data sets, 800 and 40, respectively, for training and testing. The parameters used in the experimental process can be summarized in Table III:

TABLE III PARAMETER SETTINGS

Parameter	**Value**
Activation function	ReLU
Batch size	2
Learning rate	0.00001
Loss function	Binary
Optimizer	Adam, SGD, RMSProp

3.4. Performance Criteria

Deep learning criteria were used to examine the performance criteria of the model proposed in the study are:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Sensitivity = \frac{TP}{TP + FP} \quad (3)$$

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

TP, FP, TN and FN given in Equations (1) - (3) represent the numbers of True Positive, False Positive, True Negative and False Negative, respectively.

IV. RESULTS

This section describes the results obtained in this study. As mentioned earlier, three CNN models were used: InceptionV3, ResNet50, and InceptionResNetV2. More experiments were also performed to investigate different optimizers. In this study, three optimizers were tested: ADAM, SGD, and RMSProp optimizers.

The classification results obtained are summarized in Table IV. The highest classification accuracy was obtained for SGD optimizer with InceptionV3, ResNet50, and InceptionResNetV2 model. It produced 98.8% accuracy with just 20 epochs. This was followed by the RMSProp optimizer which resulted in 95.0% accuracy while ADAM optimizer produced 86.3% accuracy. In terms of the estimation level of the model applied in the test data set, in the case of using the SGD optimizer, the sensitivity, recall, and f1 score have an accuracy value of 1.00, while in the RMSProp optimizer, the sensitivity, recall, and f1 score are seen as 0.95. In ADAM optimizer, sensitivity is 0.88, recall and f1 score is 0.85.

The highest classification accuracy was obtained for SGD and RMSProp optimizer with InceptionV3, ResNet50, and InceptionResNetV2 model. It produced 98.8% accuracy with just 30 epochs. This was followed by the ADAM optimizer produced 82.5% accuracy. In terms of the estimation level of the model applied in the test data set, in the case of using the SGD and RMSProp optimizer, the sensitivity, recall, and f1 score have an accuracy value of 1.00, while in the ADAM optimizer, is 0.81, recall 0.70, and f1 score 0.67.

TABLE IV HIGHEST ACCURACY (%) OBTAINED FOR EACH OPTIMIZERS

Optimizer	**Epoch**	**Accuracy**	**Sensivity**	**Recall**	**F1Score**
SGD	20	98.8	100	100	100
RMSProp	20	95.0	95.0	95.0	95.0
ADAM	20	86.3	88.0	85.0	85.0
SGD	30	98.8	100	100	100
RMSProp	30	98.8	100	100	100
ADAM	30	82.5	81	70	67

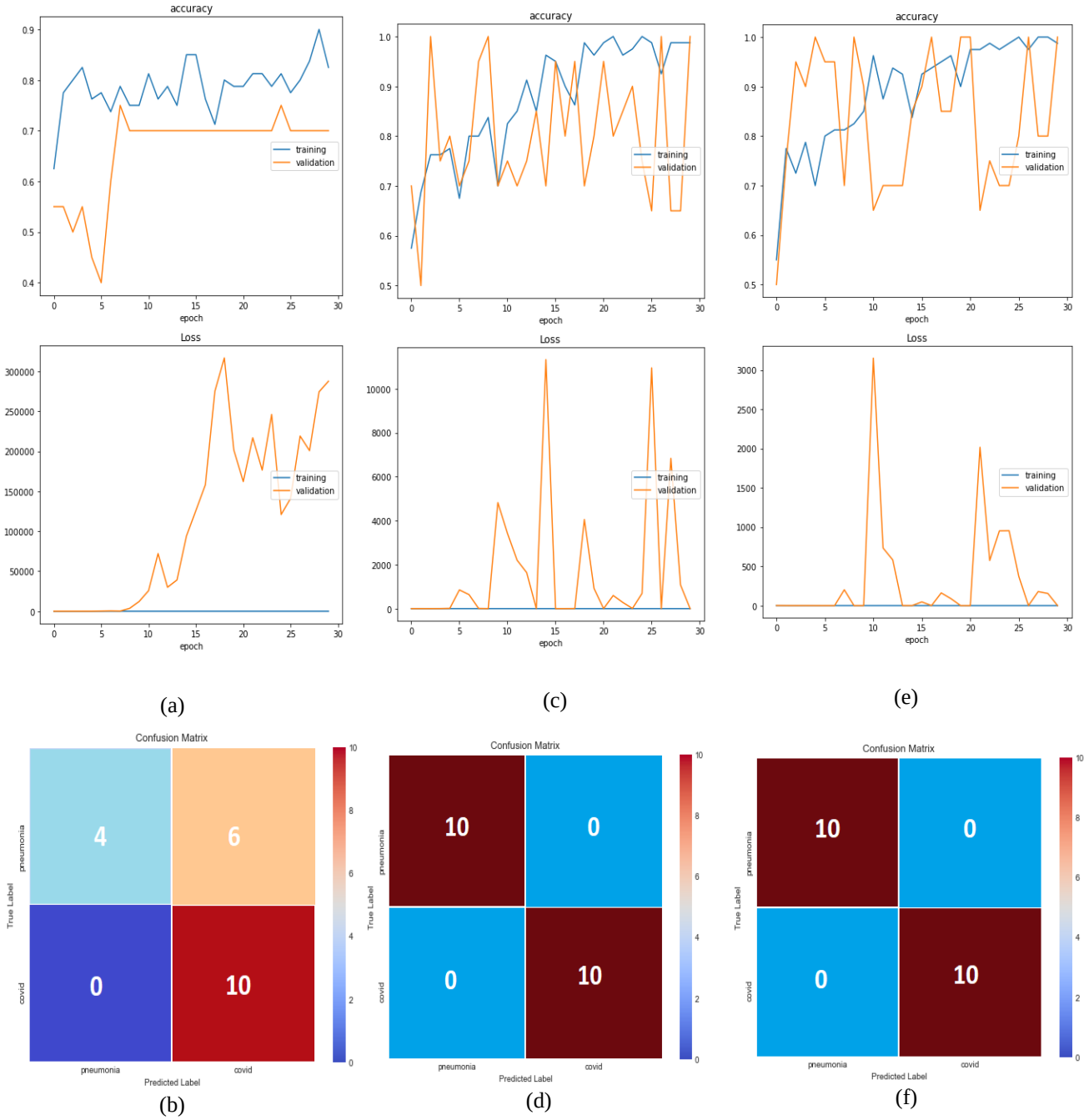


Figure 4. Comparison of results according to different optimizers (a) Accuracy and Loss Plots Using ADAM Optimizer (Epoch=30) (b) Correct and Prediction Matrix Using ADAM Optimizer (c) Accuracy and Loss Plots Using SGD Optimizer (Epoch=30) (d) Correct and Prediction Matrix Using SGD Optimizer (e) Accuracy and Loss Plots Using RMSProp Optimizer (Epoch=30) (f) Correct and Prediction Matrix Using RMSProp Optimizer

The overall summary of the results obtained for this study using CNN models with different optimizers can be seen from Fig. 4. a) shows the accuracy and loss obtained for the Adam optimizer for 30 epochs while b) shows the confusion matrix for the same optimizer. Similarly, c) and d) shows the accuracy/loss and confusion matrix for SGD optimizer. e) show accuracy and loss for RMSProp optimizer and f) shows the confusion matrix obtained using it. As it is clear that both SGD and RMSprop produced optimal results on our test data while the accuracy obtained for Adam optimizer was relatively lower than its two counterparts.

The models applied within the scope of the research have been compared some other well known models and the details regarding the comparison are summarized in Table V. In previous studies, data type, method, number of classes, and accuracy were compared.

In the study, it was aimed to reach the best result of the model applied to the data set of chest X-rays with the use of different optimizers. As a result of the findings obtained, it can be stated that SGD optimism stands out in the results obtained with three different optimizers for the model applied in the study. However, in the prediction process on the test data set, the RMSProp optimizer gave the best result and all predictions were obtained correctly.

TABLE V. COMPARISON WITH DIFFERENT STUDIES IN TERMS OF METHOD AND ACCURACY (%)

Study	Data Type	Method	No. of Classes	Accuracy (%)
Sahinbas and Catak [28]	X-Ray Images	VGG16, VGG19, ResNet, DenseNet, InceptionV3	2	80
Khan et al. [29]	X-Ray Images	CoroNet	4	89.60
Singh et al. [30]	X-Ray Images	MADE based CNN	2	92.55
Narin et al [27]	X-Ray Images	InceptionV3, ResNet50, ResNet101, ResNet152, Inception-ResNetV2	2	96.1
Proposed	X-Ray Images	InceptionV3, ResNet50, Inception-ResNetV2 (SGD Optimezer)	2	98.8 (epoch 20) 98.8 (epoch 30)
	X-Ray Images	InceptionV3, ResNet50, Inception-ResNetV2 (ADAM Optimezer)	2	86.3 (epoch 20) 82.5 (epoch 30)
	X-Ray Images	InceptionV3, ResNet50, Inception-ResNetV2 (RMSProp Optimezer)	2	95.0 (epoch 20) 98.8 (epoch 30)

These studies show that although there are different degrees of accuracy, different values of parameters such as epochs, the number of classes, and the model directly affect the classification accuracy. The literature showed that deep learning based approaches produced satisfactory results for identification of COVID-19 bases. However, less studies were conducted to distinguish between Covid-19 and Pneumonia cases.

CONCLUSION

The main purpose of this study is to facilitate the diagnosis of COVID-19 and pneumonia patients by training the images taken from open data sets of chest X-rays with the CNN-based model (InceptionV3, ResNet50, and InceptionResNetV2). In addition, every technological method that facilitates economic conditions has become very important during the current COVID-19 outbreak. It is aimed to reveal the difference between these two diseases with deep learning models within the scope of the study that the PCR tests required for the diagnosis of the COVID-19 epidemic are similar to pneumonia in terms of the burden of the economy and the lung effects of this disease.

We investigated different optimizers with the CNN models and evaluated their accuracy. The results indicate that the SGD optimizer with 20 epochs produced highest accuracy with 98.8%, followed by RMSProp and ADAM optimizers with 95% and 86.3%, respectively. However, when the epoch value was 30, SGD and RMSProp optimizers gave the best degree of accuracy with 98.8%, followed by ADAM optima with 82.5%.

In future we would like to extend the CNN models on relatively larger data sets and obtain high classification accuracy. In this context, the creation of regional data sets together with the contribution to open data sets may also allow deep learning to examine the effect of the epidemic disease such as COVID-19 according to the conditions of the region.

REFERENCES

[1] World Health Organization (WHO). Coronavirus Disease (COVID-19) Dashboard. https://covid19.who.int/ (03.01.2020).

[2] Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A. & Agha, R. (2020). World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *International Journal of Surgery*, 71-76.

[3] Rothan, H. A., & Byrareddy, S. N. (2020). The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak. *Journal of autoimmunity*, https://doi.org/10.1016/j.jaut.2020.102433.

[4] Mahmud, T., Rahman, M. A., & Fattah, S. A. (2020). CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Computers in biology and medicine*, *122*, https://doi.org/10.1016/j.compbiomed.2020.103869.

[5] Militante, S. V., Dionisio, N. V., & Sibbaluca, B. G. (2020, October). Pneumonia and COVID-19 Detection using Convolutional Neural Networks. In *2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE)* (pp. 1-6). IEEE.

[6] Militante, S. V., & Sibbaluca, B. G. (2020). Pneumonia Detection Using Convolutional Neural Networks. *International Journal of Scientific & Technology Research*, *9*(04), 1332-1337.

[7] Das, N. N., Kumar, N., Kaur, M., Kumar, V., & Singh, D. (2020). Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays. *IRBM*, . https://doi.org/10.1016/j.irbm.2020.07.001.

[8] Mooney P. Chest X-ray Images (Pneumonia). Kaggle Repository: https://www.kaggle.com/paultimothymooney/chest-xray-pnemonia, 2018.

[9] Cohen JP, Morrison P, and Dao L. COVID-19 Image Data Collection. arXiv:2003.11597,2020.

[10] Rahimzadeh, M., & Attar, A. (2020). A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Informatics in Medicine Unlocked*, https://doi.org/10.1016/j.imu.2020.100360.

[11] Ismael, A.M., Şengür, A. (2020). Deep Learning Approaches for COVID-19 Detection Based on Chest X-ray Images. *Expert Systems with Applications*, doi: https://doi.org/10.1016/j.eswa.2020.114054

[12] Hussain, E., Hasan, M., Rahman, M. A., Lee, I., Tamanna, T., & Parvez, M. Z. (2020). CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. *Chaos, Solitons & Fractals*, https://doi.org/10.1016/j.chaos.2020.110495

[13] Abraham, B., & Nair, M. S. (2020). Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier. *Biocybernetics and biomedical engineering*, *40*(4), 1436-1445.

[14] Kesim, E., Dokur, Z., & Olmez, T. (2019). X-Ray Chest Image Classification by A Small-Sized Convolutional Neural Network. In

2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT). IEEE. https://doi.org/10.1109/EBBT.2019.8742050

[15] Liu, C., Cao, Y., Alcantara, M., Liu, B., Brunette, M., Peinado, J., & Curioso, W. (2017). TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 2314-2318). IEEE.

[16] Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, *10*(1), 1-12.

[17] Hemdan, E. E. D., Shouman, M. A., & Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint arXiv:2003.11055*.

[18] Jain, R., Gupta, M., Taneja, S., & Hemanth, D. J. (2020). Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Applied Intelligence*, 1-11.

[19] Karar, M. E., Hemdan, E. E. D., & Shouman, M. A. (2020). Cascaded deep learning classifiers for computer-aided diagnosis of COVID-19 and pneumonia diseases in X-ray scans. *Complex & Intelligent Systems*, 1-13.

[20] Yildirim, O., Talo, M., Ay, B., Baloglu, U. B., Aydin, G., & Acharya, U. R. (2019). Automated detection of diabetic subject using pre-trained 2D-CNN models with frequency spectrum images extracted from heart rate signals. *Computers in biology and medicine*, *113*, 103387.

[21] Celik, Y., Talo, M., Yildirim, O., Karabatak, M., & Acharya, U. R. (2020). Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images. *Pattern Recognition Letters*.

[22] Jmour N, Zayen S, and Abdelkrim A. (2018). Convolutional neural networks for image classification. International Conference on Advanced Systems and Electric Technologies (IC_ASET), Hammamet, Tunisia, pp. 397-402.

[23] Ergin, T. (2018). Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?. https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad (Erişim Tarihi: 01.01.2021)

[24] Ahn, J. M., Kim, S., Ahn, K. S., Cho, S. H., Lee, K. B., & Kim, U. S. (2018). A deep learning model for the detection of both advanced and early glaucoma using fundus photography. *PloS one*, *13*(11), e0207982.

[25] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211-252.

[26] Byra, M., Styczynski, G., Szmigielski, C., Kalinowski, P., Michałowski, Ł., Paluszkiewicz, R., ... & Nowicki, A. (2018). Transfer learning with deep convolutional neural network for liver steatosis assessment in ultrasound images. *International journal of computer assisted radiology and surgery*, *13*(12), 1895-1903.

[27] Narin, A., Kaya, C., & Pamuk, Z. (2020). Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*.

[28] Sahinbas K, and Catak FO. Transfer Learning Based Convolutional Neural Network for COVID-19 Detection with X-Ray Images. https://www.ozgurcatak.org/files/papers/covid19-deep-learning.pdf, (Erişim Tarihi: 29.12.2020).

[29] Khan, A. I., Shah, J. L., & Bhat, M. M. (2020). Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, 105581.

[30] Singh, D., Kumar, V., Yadav, V., & Kaur, M. (2020). Deep Neural Network-Based Screening Model for COVID-19-Infected Patients Using Chest X-Ray Images. *International Journal of Pattern Recognition and Artificial Intelligence*, 2151004.

Weather Forecasting Using Back Propagation Feed Forward Neural Network and Multiple Linear Regression

Doğancan Ulutaş
Department of Computer Engineering
Istanbul Sabahattin Zaim University
0000-0002-5025-1161

Alaa Ali Hameed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
alaa.hameed@izu.edu.tr

Erdal Alimovski
Department of Computer Engineering
Istanbul Sabahattin Zaim University
erdal.alimovski@izu.edu.tr

***Abstract*—Weather forecast is one of the most important research areas in world problems such as meteorology, human civilization, drought, agriculture and dams. We propose Back Propagation Feed Forward Neural Network and Multiple Linear Regression method and models for predicting air precipitation in the project. Proposed Neural Network model were trained with 8 optimization algorithms in order to find proper accuracy. The performance of the models in the project is evaluated with the most appropriate statistical methods. Coefficient of correlation (r), Root Means Square Error (RMSE), Mean Percent Error (MPE), Mean Absolute Percent Error (MAPE), Mean Square Error (MSE) and R-square statistics were used to measure the accuracy of the model proposed in the project. The data sets used in the project were taken from the Istanbul Provincial Meteorology System. Obtained results demonstrated that, It is seen that the model proposed in the project gives better results than the algorithms in other studies. It is seen that the model proposed in the project gives better results than the algorithms, models and techniques in other studies.**

***Keywords*— Neural Network, Multiple Linear Regression, Rainfall, Machine Learning**

I INTRODUCTION

Rainwater, which is the most important part of the world, is a very important issue in hydrology, meteorology and the future of humanity. Rain is the result of the interaction of the weather system and is affected by many environmental factors such as the ecosystem, tree density and terrain. [1]. The impact of rainfall on human civilization and agriculture from past to present is enormous. Rainfall is a difficult and important natural climate event that can be predicted days in advance for human civilization. The most important part of precipitation forecast is required for the advance planning and management of water resources and water use. It is very important that water channels and dams do not overflow. Additionally, rainfall has a strong impact on systems such as traffic, drought, sewage, landslides, floods, tsunamis and other human activities in urban areas. However, due to factors such as the complexity, disorder, continuous change of the ecosystem, the incredible diversity of atmospheric processes that make up both space and time, precipitation is one of the most complex and difficult to predict issues to understand, analyze and model. For these reasons, despite advances in many technological fields such as the increase in weather forecast models and algorithms in recent years, accurate weather forecast is becoming very difficult. Precipitation means agriculture and crops; agriculture and crop mean life. Weather forecast is very important for the agricultural sector, which contributes significantly to the country's economy. Numerous research and projects have been conducted by different researchers around the world to accurately predict weather precipitation using a variety of techniques and models. However, the prediction accuracy obtained with these studies, algorithms and techniques is also below the desired level because the precipitation does not continue in the same way.

The artificial neural network algorithm succeeds in precipitation prediction thanks to its highly nonlinear, flexibility, data-driven learning in modeling without prior knowledge of collection behavior and flow processes. [2]. Artificial neural networks have been highly preferred and successfully used in various aspects of science and engineering in recent years due to their ability to model and analyze both linear and nonlinear systems without making assumptions, as they are more successful in most traditional statistical approaches. ANN technique is frequently preferred and successfully used in all countries of the world for the prediction, recognition and classification of many weather events [3].

In this study, in order to predict the rainfall in a very effective way we propose back propagation feed forward neural network and multiple linear regression models. Proposed Neural Network model were trained with 8 optimization algorithms in order to find proper accuracy. The performance of all models to be applied is evaluated and analyzed using the most appropriate statistical methods for the project. Coefficient of correlation (r), Root Means Square Error (RMSE), Mean Percent Error (MPE), Mean Absolute Percent Error (MAPE), Mean Square Error (MSE) and R-square statistics were used to measure the accuracy of the model proposed in the project.

The remainder of the article is as follows: Part 2 explains relevant work and provides information on topics. Chapter 3 explains the data set used and details of the proposed models and algorithms. Section 4 presents all the data and results obtained in the project. Chapter 5 contains descriptions of data, estimates, results.

II RELATED WORK

There are many studies and sources for precipitation prediction in past studies. This section is where these studies are mentioned.

In study [4], researchers proposed a precipitation forecast project using artificial neural networks. The proposed model predicts air precipitation with artificial neural networks of the Udupi region in Karnataka state of India. BPNN with feed forward network structure and repeating layer architecture was tested in the project. When we look at the results, it seems that the recurrent network gives more accurate results than BPNN.

In study [5], researches in order to predict the rainfall precipitation proposed LSTM and Convolution Neural Network (CNN) models. Proposed models, Estimated monthly average rainfall data for 10368 Geo Locations worldwide for 39 Months. Obtained results shows that LSTM gets RMSE of 2.25, whereas the RMSE of CNN was 2.44.

In study [6], researches perform C4.5 algorithm in order to predict the rainfall in city Bandung regency in Indonesia. Furthermore, the final sifting method is used to optimize sifting on the model. From the results, it can be seen that the

analysis positively affects the model and gives correct results. The average accuracy test result without pruning is 60% and the use of pruning is 93.33%.

In study [7], researches in order to predict the weather in New Delhi, India and Australia applied five different classification algorithms such as Extra Trees, Random Forests, Logistic Regression, Stochastic Gradient Descent (SGD) and Support Vector Machines (SVM). Obtained results shows that Random forest and Extra Trees reached the highest accuracy of 85%. Followed by SVM and Logistic Regression with an accuracy of 83%. SGD reached the lowest accuracy 82%.

In study [8], researcher proposes a hybrid DL approach, a combination of one-dimensional Convolution Neural Network (Conv1D) and Multi-Layer Perception (MLP), All step forward daily precipitation forecast from day 1 to day 5. Next, proposed hybrid model were compared with MLP and SVM. When this model is examined, it is seen that the hybrid model gives more accurate results. Looking at the models in general, the predictive data from the recommended hybrid DL model can be helpful in agricultural irrigation planning and even flooding due to heavy rainfall.

To improve and demonstrate the average prediction accuracy for short-term precipitation, the study [9] proposed a Dynamic Regional Combined short-term precipitation Prediction model (DRCF) using a Multi-Layer Sensor. Experiments were conducted on data sets taken from 56 real space meteorology sites in China. As a result of the experiments, it shows that the proposed model performs better and more successfully than the existing approaches in terms of both threat score (TS) and root mean square error (RMSE).

III Materials and Methods

In this section, first about the data set, then about the reconstruction of the missing data and important factors such as data transformation, model selection and accuracy are explained. Secondly it is described the proposed Artificial Neural Network model. Lastly it is described the Linear Regression model.

A. Data set

Data sets used in the project were used from the Istanbul provincial meteorological system and data falling to Istanbul rainfall stations from the FreeMeteo website. In total, 365 data were taken by taking the average values of the years between 2015-2019. Data is divided into 5 branches: temperature, Max. temperature, min. the temperature is classified as water temperature and precipitation. Note that, 70% of the data set were used for training, 15% for Validation and remaining 15% for testing. The reason for separating data in this way is because these values produce values that are the most accurate and the least error value.

B. Neural Network Model

Artificial neural networks (ANN) have the ability to represent nonlinear processes and many collaborative articles have been published in this area. [10], [11], [12]. An artificial neural network consists of a simple and synchronized substance called neurons, which looks like biological neurons in the human brain. Feed forward ANN can have multiple layers. These neurons are lined up in layers in the network. Neurons in one layer are connected to the next layer. The strength of the connection between two neurons in adjacent layers is called "weight". There are weights between the links, and these weights are changed during the training process in order to produce a result close to the desired value from the input. The training rule is used to adjust the weights for this change. [13]. ANN consists of input, hidden and output layers. In a feed forward network, the weighted links feed forward from only one input layer to the output layer. Each node in a tier receives the weighted input from the previous tier, and jobs then forward its output to nodes in the next tier through links. In forward and backward feed, activation is fed backward. From output layer to hidden or input layer. Our proposed Feed forward back propagation neural network model consists of four layers; one input, two hidden and one output layer with 4,10,10,1 neurons respectively. As activation functions, in both input and hidden layers ReLu, while in output layer soft max (2) were used. Learning rate was set to 0.07. In order to find a proper accuracy, we trained our model with nine different optimization algorithms such as: Lavenberg-Marquardt (1), Bayesian regularization (2), Scaled conjugate gradient (3), Batch training, BFGS(4), Powell-Beale (5), Fletcher-Powell (6), One step scan (7), Resilient (8). Note that, all the mentioned hyper parameters above are constant for each experiment.

$$\Delta w = (J^T J + \mu I)^{-1} J^T e \tag{1}$$

$$\frac{1}{\sigma^2 D}\left[\frac{\sigma^2 D}{2\sigma^2 W}\sum_i w_i^2 + E\right] \tag{2}$$

$$\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T . g_k} \tag{3}$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{4}$$

$$|g_{k-1}^T g_k| \geq 0.2\|g_k\|^2 \tag{5}$$

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \tag{6}$$

$$\tau_n = \tau_{n-1} - f(\tau_{n-1})\frac{\tau_{n-1} - \tau_{n-2}}{f(\tau_{n-1}) - f(\tau_{n-2})} \tag{7}$$

$$dX = \Delta X . sign(gX) \tag{8}$$

C. Linear Regression

The method proposed in the project is based on multiple linear regression. For estimation, data have been collected from publicly available sources. 70 percent of this data is for education and 30 percent of the data is for testing. The multiple regression method is used to estimate values in the project, and this method is both a mathematical and statistical method. There is a linear relationship between variables and output values. The multiple linear regression equation is as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \tag{9}$$

The number of observations is denoted by "n". The dependent variable is y_i and the descriptive variable is x_i. β_0 and β_p are the constant y intercept and slop of descriptive variable respectively. Model error is indicated by $\in$. In the proposed model within the project, more than one meteorological parameter is required to predict the weather. For this estimate to be more accurate, Using multiple linear regression instead of linear regression gives much more

successful results. The assumptions which are made by the multiple linear regression are: linear relationship between the both the descriptive and independent variables, the highly correlated variables are independent variables, y_i is calculated randomly and the mean and variance are *0* and σ .

IV Experimental Results

This section shows the results of the ANN and linear regression model proposed in the project. The precipitation data used from 2009 to 2018 consists of 10959 data sets. The data is divided into two parts: 70% for training data and 30% for test data.

In order to evaluate how close are the predictions of proposed neural network model to the eventual outcome and which optimization algorithm effects more positively to the proposed model we perform error metrics. The major error parameters we use are coefficient of correlation (r).

Root Means Square Error (RMSE), Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE) and R2.

The correlation coefficient is a mathematical, statistical measure of the linear relationship between two variables. The value of R is calculated as follows:

$$r = \frac{\sum_{i=1}^{n}(y_i - \overline{y})(y_i' - \overline{y}')}{\sum_{i=1}^{n}(y_i - \overline{y})\sum_{i=1}^{n}(y_i' - \overline{y}')\,(1)} \tag{10}$$

Where y_i (mm) and y_i' (mm) represent the observed and predicted amount of weather precipitation "t", (mm) and $\overline{y}$ (mm) are the estimated and observed air precipitation, respectively, and "n" indicates the total number of data points. The "R" value ranges from [−1, 1], where -1 represents a perfect negative linear relationship. 0 indicates no linear relationship, and 1 represents a perfect positive linear relationship. The higher the "R" value is in the applied project, the higher the model performance.

RMSE is often used to measure the difference between observed, measured and predicted values. Its value is Always positive and a lower RMSE means a better model performance. RMSE is expressed as:

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{n}} \tag{11}$$

MPE the calculated average of percentage errors when the estimates differ from the actual estimated amount. MPE can be defined as:

$$MPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{a_t - f_t}{a_t} \tag{12}$$

Where a_t "n" dictates that the estimated amount is the true value. f_t is the number of times the variable was predicted and n was the number of different times the variable was predicted.

MAPE is used to measure the efficiency, consistency and accuracy of the proposed model and algorithm in the project. Its value is calculated by eq. (13)

$$M = \frac{1}{n}\sum_{i=1}^{n}\frac{a_t - f_t}{a_t} \tag{13}$$

Where a_t is the actual value and f_t is the forecast value. MAPE is also presented as a percentage, which is the above equation multiplied by 100, according to different models. The difference between a_t and f_t is divided by the actual value a_t again. The absolute value in this solution is summed up for each point predicted and divided by the number " n". Multiplying by 100% gives the percent error.

The mean square error between observed, predicted, and computed outputs can be defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

R-Square measures how much of the variability in the dependent variable will be regulated and explained by the model. It is expressed as:

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \hat{y})^2} \tag{15}$$

Figure 1 illustrates the Multi linear regression analysis for rainfall prediction. Blue circles represent the distribution of the normalized data. It is seen that the deviation and distribution of the values in the regression analysis progresses in the plane and the precipitation values diverge according to the months.

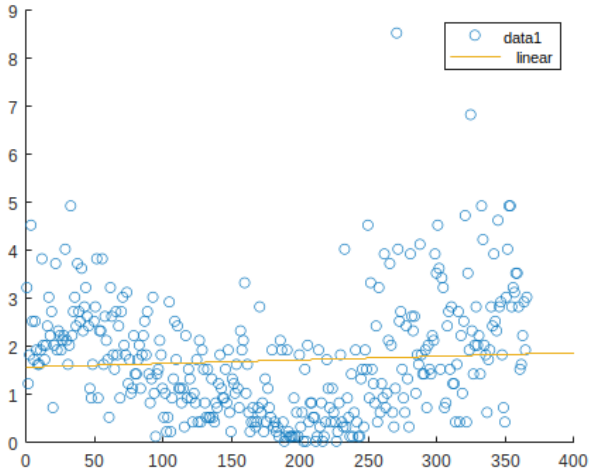


Figure 1: Multi Linear Regression plot.

A. Model Performance

The performance of proposed model with different optimization algorithm in terms of different evaluation metrics is described in Table 1.

First in Table 1. Proposed model is compared among different optimization algorithms. In terms of MPE, FFNN with Scaled Conjugate Gradient performed worst with 2.11, whereas FFNN with Powell-Beale performed the best with value of 1.69. In terms of MAPE, FFNN with BFGS achieved the best value 0.49, whereas FFNN with Powel Fletcher the worst value 0.61. In terms of MSE, FFNN with Powell-Beale performs well with value 0.10, whereas FFNN with One step scan achieved the worst result. In terms of RMSE, FFNN with Powel Fletcher performed the best with 0.18, whereas Scaled Conjugate gradient performed worst with 0.34. In terms of R-square evaluation metric, Lavenberg-Marquardt performed best, whereas Fletcher Powell performed worst. From the Table 1, it can be concluded that Lavenberg-Marquardt and Powell-Bale algorithms achieved the best results.

Secondly, the performance of proposed model with different optimization algorithms in testing phase is demonstrated in Figure 2. In terms of R, Lavenberg-Marquard

algorithm gets the best results with 0.38, following by Bayesian Regularization with 0.43.

Next, we select three optimizations to evaluate the error rate in training, validation and testing tasks of the proposed model in the project: Lavenberg-Marquardt, Bayesian Regularization and Scaled Conjugate Gradient. This experiment was performed in Matlab's NF-tool. Note that, In the project, 70% of the data set was used for training, 15% for verification and 15% for testing. It was observed that these rates gave more accurate results. Obtained results in Figure 3 demonstrates that FFNN with Lavenberg-Marquardt (Lavenberg-Marquardt based FFNN) achieved the best result.

Following, Figure 3 and 4 depict how different optimization algorithms effects to the proposed model in testing phase in terms of MSE.

Table 1: Comparison of optimization algorithms in terms of different evaluation metrics.

Optimization Algorithms	**Evaluation metrics**				
	MPE	**MAPE**	**MSE**	**RMSE**	**R2**
Lavenberg-Marquardt	1.95	0.56	1.14	0.28	0.3
Bayesian Regularization	1.96	0.52	0.68	0.29	-0.2
Scaled Conjugate Gradient	2.11	0.58	1.15	0.34	-0.1
Batch training	1.99	0.58	0.12	0.31	-0.2
BFGS	1.87	0.49	0.08	0.23	-0.3
Powell–Beale	1.90	0.54	0.10	0.25	0.3
Fletcher-Powell	1.69	0.61	0.11	0.18	-0.7
One step scan	1.81	0.55	0.87	0.23	-0.6
Resilient	1.76	0.57	0.19	0.22	-0.6

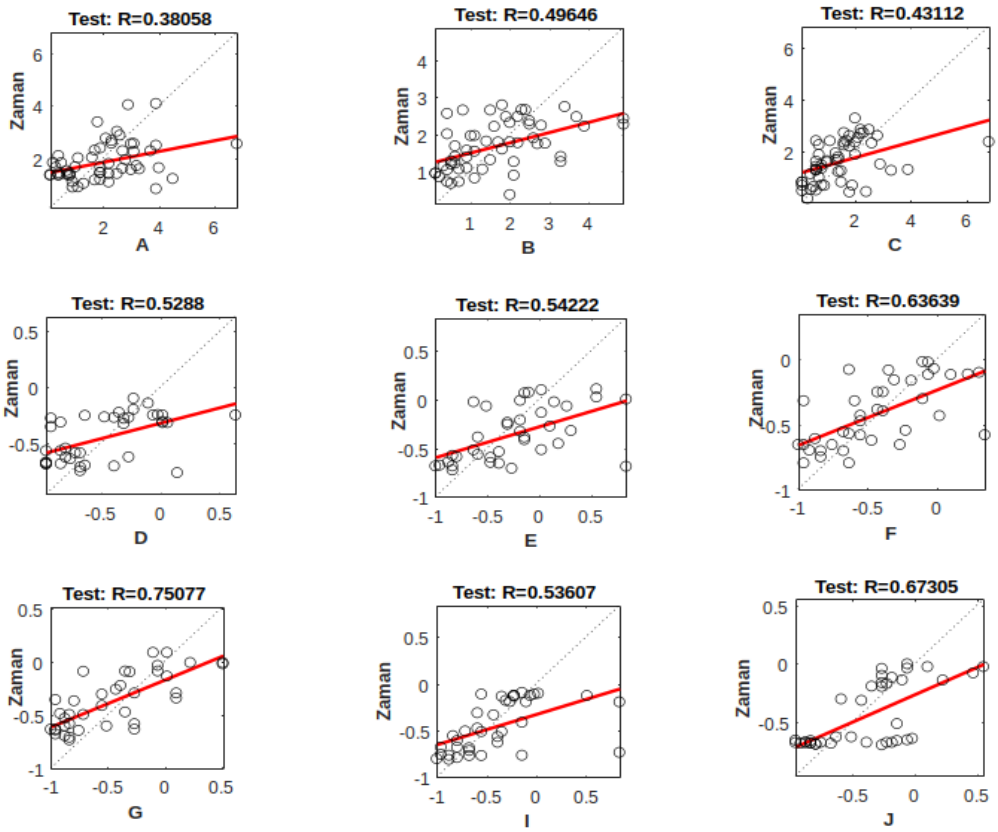


Figure 2: Proposed model - test performances with different optimization algorithms.

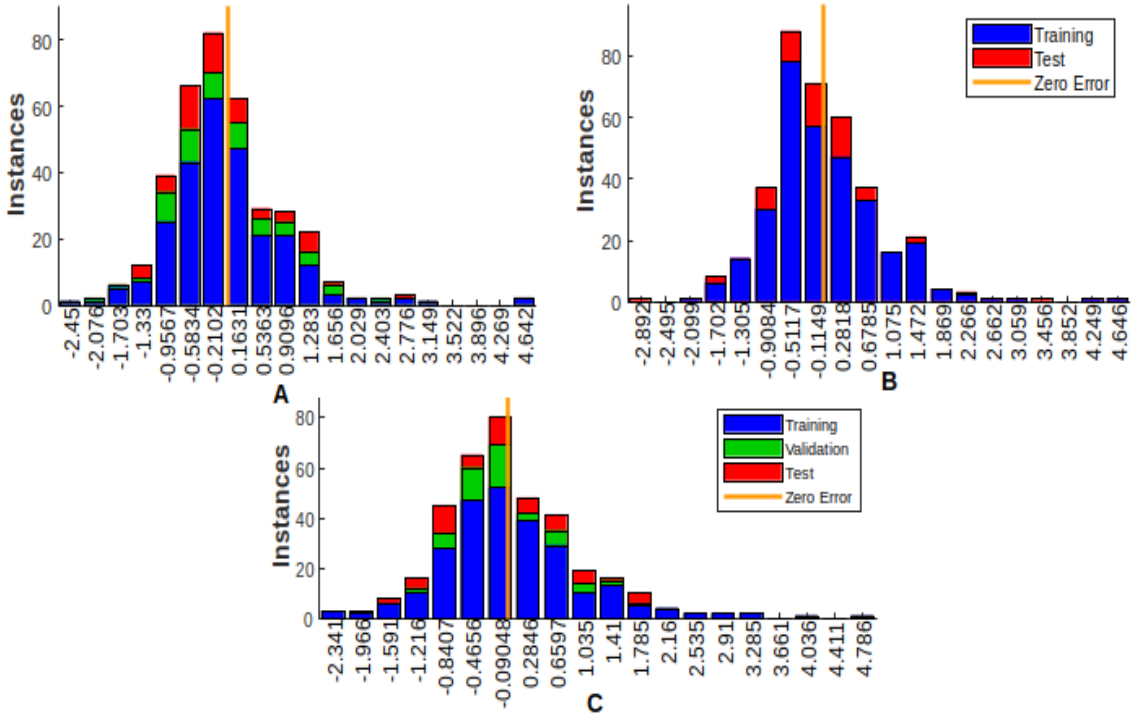


Figure 3: Comparison error performance of proposed model with different optimizer.

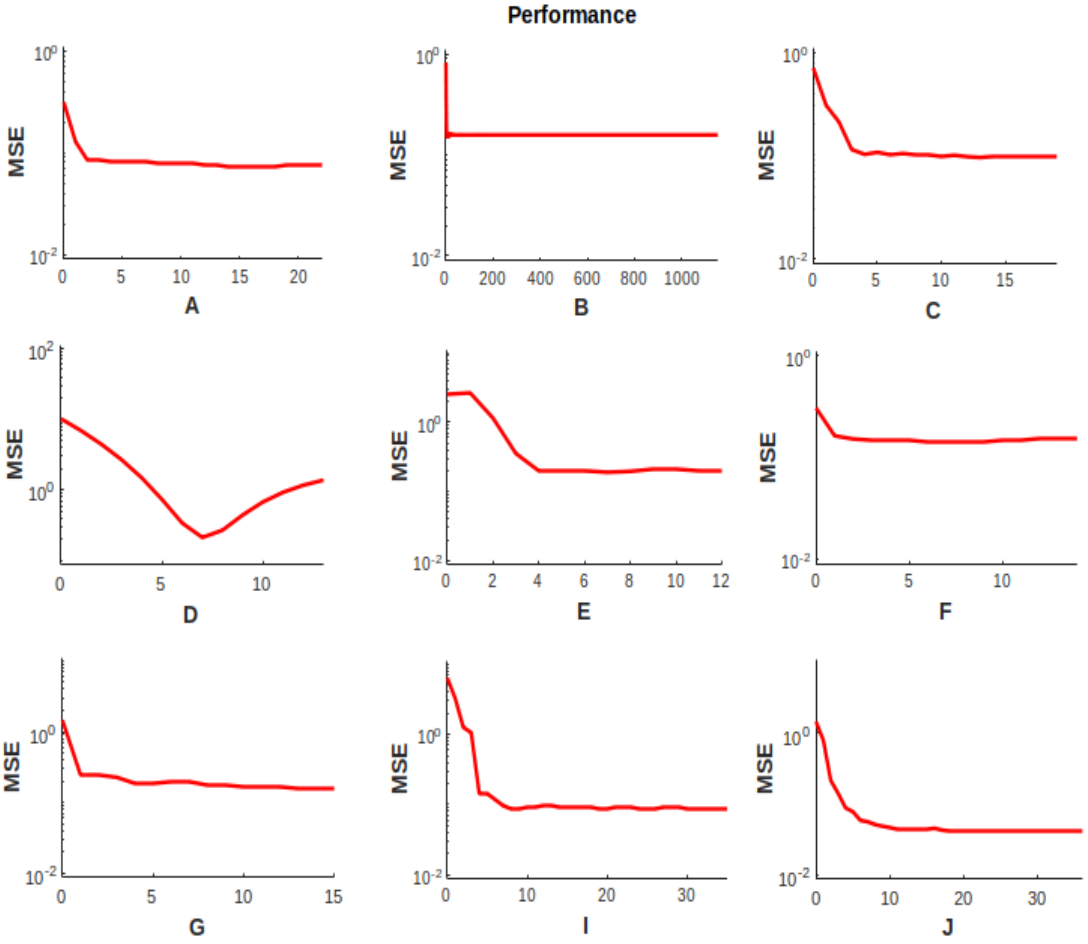


Figure 4: Performance of proposed model in terms of MSE.

A detailed comparison of the method proposed in the project with state-of-the-art techniques is presented in Table 2., in terms of, RMSE, R^2 and. Our proposed model overwhelmed the existing techniques and achieved the highest performance, which are the best results achieved so far over rainfall prediction. In addition, unlike the values in the tables, the MPE value is 1.90 for us. The Correlation value is 0.65 and the MAPE value is 0.54. Other studies: [4] MSE value 0.42. [5] MSE value 2.55, MAPE value 1.68. [7] The MSE value is 0.42. [8] 5.85 RMSE value. [14] RMSE value of 2.55. [17] MSE value 0.25. The R^2 value is 0.9. [19] RMSE value of 9.04. [20] The R^2 value is 0.94. [21] RMSE value 3.94.

Table 2: Comparison of the proposed model with state of the art techniques.

STUDIES	EVALUATION METRICS	
	RMSE	**R2**
Cavazos [3]	2.36	0.59
Nastos et. [15]	16.4	0.48
Hung et. [16]	1.84	0.41
Moust. et. [18]	12.5	0.24
Azadi & S.[22]	0.28	0.67
Proposed	**0.25**	**0.3**

V CONCLUSION

This article describes a back propagation feed forward neural network and multiple linear regression models to predict air precipitation in Istanbul. Turkey. Proposed neural network based model, were trained with eight different optimization algorithms in order to find best accuracy. The error measurement metrics which we use are: client of correlation (r), Root Means Square Error (RMSE), Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE) and R-square. Generally, proposed model achieved best accuracy with Lavenberg-Marquardt optimizer. From the obtained results, it can be concluded that proposed models in terms of error metrics achieved better performance compared with the other algorithms in the literature.

REFERENCES

[1] M. D. Crown, “Validation of the NOAA space weather prediction center’s solar flare forecasting look-up table and forecaster-issued probabilities,” Space Weather, vol. 10, no. 6, pp. 1–4, 2012

[2] A. A. Imran Maqsood Muhammad Riaz Khan, "An ensemble of neural networks for weather forecasting," Neural Computing & Applications, vol. 13, no. 2, pp. 112-122,2004.

[3] Cavazos, Tereza. "Downscaling large-scale circulation to local winter rainfall in north–eastern Mexico." International Journal of Climatology: A Journal of the Royal Meteorological Society 17.10 (1997): 1069-1082.

[4] Kumar Abhishek, Abhay Kumar, Rajeev Ranjan, Sarthak Kumar,” A Rainfall Prediction Model using Artificial Neural Network”, 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012), pp. 82-87, 2012.

[5] Aswin, S., P. Geetha, and R. Vinayakumar. "Deep learning models for the prediction of rainfall." 2018 (ICCSP). IEEE, 2018.

[6] Suyatno, Joko Azhari, Fhira Nhita, and Aniq Atiqi Rohmawati. "Rainfall Forecasting in Bandung Regency Using C4. 5 Algorithm." 2018 6th International Conference on Information and Communication Technology (ICoICT). IEEE, 2018.

[7] Quinn, Brandan, and Eman Abdelfattah. "Machine Learning Meteorologist Can Predict Rain." 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, 2019.

[8] Khan, Mohd Imran, and Rajib Maity. "Hybrid Deep Learning Approach for Multi-Step-Ahead Daily Rainfall Prediction Using GCM Simulations." IEEE Access 8 (2020): 52774-52784.

[9] Zhang, Pengcheng, et al. "Short-term rainfall forecasting using multi-layer perceptron." IEEE Transactions on Big Data (2018).

[10] Minns AW, Hall MJ (1996) Artificial neural networks as rainfall runoff models. Hydrol Sci J 41(3):399–418

[11] Sudheer KP, Gosain AK, Ramasastri KS (2002) A data-driven algorithm for constructing artificial neural network rainfall-runoff models. Hydrol Process 16:1325–1330

[12] Senthil Kumar AR, Sudheer KP, Jain SK, Agarwal PK (2005) Rainfallrunoff modelling using artificial neural networks: comparison of network types. Hydrol Process 19:1277–1291

[13] Ajmera TK, Goyal MK (2012) Development of stage discharge rating curve using model tree and neural networks: an application to peachtree creek in Atlanta. Expert Systems With Applications, Elsevier Ltd. 39(5):5702–5710

[14] Shah, Urmay, et al. "Rainfall Prediction: Accuracy Enhancement Using Machine Learning and Forecasting Techniques." 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC). IEEE, 2018.

[15] Nastos, P.T., Paliatsos, A.G., Koukouletsos, K.V., Larissi, I.K. & Moustris, K.P. Artificial neural networks modeling for forecasting the maximum daily total precipitation at Athens, Greece. Atmospheric Research 144, 141 - 150 (2014).

[16] Hung, Nguyen Q., et al. "An artificial neural network model for rainfall forecasting in Bangkok, Thailand." Hydrology & Earth System Sciences 13.8 (2009).

[17] Ilaboya, I . "Performance of Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) for the Prediction of Monthly Maximum Rainfall in Benin City, Nigeria" . International Journal of Engineering Science and Application 3 (2019): 21-37

[18] Moustris, K.P., Larissi, I.K., Nastos, P.T. & Paliatsos, A.G. Precipitation Forecast Using Artificial Neural Networks in Specific Regions of Greece. Water Resources Management 25, 1979 - 1993 (2011).

[19] Rajurkar, M. P., U. C. Kothyari, and U. C. Chaube. "Artificial neural networks for daily rainfall—runoff modelling." Hydrological Sciences Journal 47.6 (2002): 865-877.

[20] Riad, Souad, et al. "Rainfall-runoff model usingan artificial neural network approach." Mathematical and Computer Modelling 40.7-8 (2004): 839-846.

[21] Sahai, A. K., M. K. Soman, and V. Satyan. "All India summer monsoon rainfall prediction using an artificial neural network." Climate dynamics 16.4 (2000): 291-302.

[22] Azadi, S., Sepaskhah, A.R. Annual precipitation forecast for west, southwest, and south provinces of Iran using artificial neural networks. Theor Appl Climatol 109, 175–189 (2012).

Augmented Fake News Detection using LSTM and Particle Swarm Optimization

1st Ghulam Mustafa
Computer Science Department
University of Engineering and Technology
Taxila, Pakistan
g.mustafa@uettaxila.edu.pk

2nd Shahid Alam
Computer Engineering Department
Adana Alparslan Türkeş Science and Technology University
Adana, Turkey
salam@atu.edu.tr

3rd Muhammad Asif Khan
Computer Engineering Department
University of Engineering and Technology
Taxila, Pakistan
masif.khan@uettaxila.edu.pk

4th Basit Shahzad
Computer Engineering Department
National University of Modern Languages
Islamabad, Pakistan
bshahzad@numl.edu.pk

***Abstract*—Fake news detection using machine learning algorithms has matured over the years due to its importance in journalism and cultural perspectives. The financial sector, stock market, policy matters of conglomerates, and various elements of culture directly affect due to the dissemination of fake news over public platforms. In this paper, a neural network model, LSTM (Long short-term memory), is trained using a large dataset of fake news. An evolutionary optimization algorithm, PSO (Particle swarm optimization), is utilized, leading to more accurate, efficient, and scalable detection of fake news. This combination of PSO along with LSTM is the state-of-the-artwork for fake news detection.**

***Index Terms*—word-embedding, particle swarm optimization, LSTM, CNN, Neural Network, backpropagation, gates, Natural Language Processing**

I. Introduction

We live in a world of misinformation, deception, and fake news. Humans tend to evaluate arguments based on empirical evidence. For them, it is possible to not recognize deception if inclined to the topic at hand. The three elements involved in disseminating fake news are originators, facilitators, and the readers who do not verify (or do not want to) the news or source of information. The latest characteristics identified of fake news is that it either spreads online or on social media and without factual evidence. That includes news reports, editorials, exposes and more that are intentionally and knowingly deceptive, with the purpose of either political or monetary gain [1] [2]. False but concrete information does not fall into the category of fake news because information can be verified easily. Disaster fake news Twitter dataset [3] that this paper use represents those that cannot directly be verified or that are abstract in nature. So the problem also involves sentiment analysis concerning the source of information.

Natural Language Processing (NLP) is a domain of artificial intelligence (AI), enables computers to learn without being explicitly programmed. Hence comes under the umbrella of machine learning. Machine learning equipped computer programs learns from training data and can change their function based on new data. Types of how machine learning is:

- Supervised learning makes use of a pre-classified dataset trained by the neural network model and predicts the input data.
- Unsupervised learning is the type of machine learning algorithm that learns from an unclassified dataset.
- Reinforcement Learning is a trial and error sequence of decisions made by the algorithm to maximize cumulative reward based on rewards and penalties.

The fake news detection problem falls into the category of supervised learning. Neural nets are the backbone of machine learning algorithms and are divided into layers of nodes. Often called a feed-forward network because data moves from nodes in a single direction. Artificial Neural Network (ANN), also called vanilla network, works as the brain processes information and contains a hidden layer. It maps a fixed-size input to a fixed size output. It can only process immediately available data that limits its potential for more advanced applications. A Recurrent Neural network (RNN) is an updated version of a vanilla network and contains a feedback loop that gives the power to retain previous neural layer information. But the problem of training Recurrent Neural Network (RNN) is the Vanishing Gradient Problem.

Artificial neural network gradients are calculated during backpropagation. Derivatives of the network are calculated in backpropagation by moving from the outermost layer (close to output) back to the initial layers (close to the inputs). The chain rule is used during the calculation in which derivatives from the final layers are multiplied by the derivatives from the early layer. The gradients keep diminishing exponentially and therefore the weights are reduced and no longer being updated as shown in figure1.

With the increase in the learning rate, the search space increases, and the global minimum value is achieved faster.

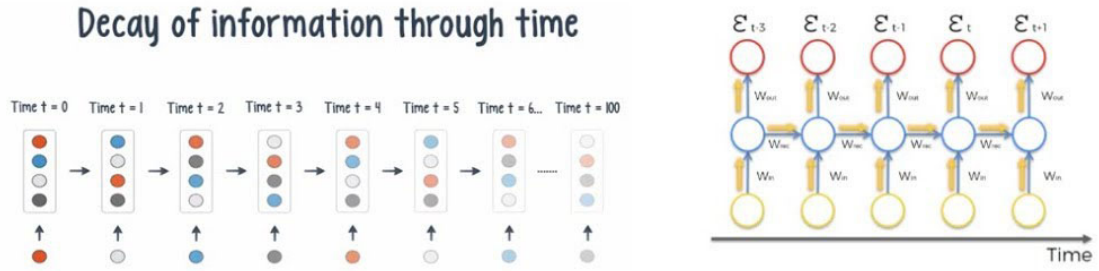


Fig. 1. How Gradients Diminish
[4]

But it can overfit the target value as performance is good on training data but inaccurate generalization on test data. Whereas, with a small learning rate, time to achieve optimal value increases and results in poor performance for training and testing data.

A. Hyperparameter Tuning

Tuning hyperparameters is a complex task and manual tuning of parameters to come up with an accurate model takes much time. Search for the proper parameters initially involves iterating between trying out different models, hyperparameters, and features [5].

In this paper, we automate the solution to the problem of predicting binary classification as true or fake news and aim to answer questions

1) How to detect fake news using an AI model and NLP?
2) How can this process be more automated than traditional LSTM solutions by tuning hyperparameters?

Contributions of the work are summarized below:

- To learn tweets for truth detection is difficult with natural means. Neural network layers learn the textual representation. Data cleaned, and stop words deleted before that. Then corpus is tokenized for the embedding layer of LSTM.
- A supervised dimensionality reduction using NLP learns word representation. Bi-Directional RNN (Updated as LSTM) is used to model tweet data through encoding words. It learns the binary classification for news tweets. This approach learns the tweet structure in the model architecture and predicts true or false news [6] [7]. The dataset Disaster Tweets [3] containing classified 11,000 tweets labeled as true or false used for the purpose.
- An automated hyperparameter search method is adopted to find the global maximum position vector where the model accuracy is maximum. Particle Swarm Optimization is used to search the three-dimensional space for best fit hyperparameter vectors.

The next section II will discuss various methodologies that are applied to detect fake news from a dataset [3]. Section III will discuss the process this work has adopted to detect fake news and how it is a state-of-the-art method in terms of automation in hyperparameter fine-tuning for the purpose.

II. Literature Review

Hamid et al. [8] used Convolutional Neural Network (CNN) along with LSTM for automatic textual feature extraction to determine the degree of the fakeness of news. CNN extracts local patterns similar to n-gram features. Then LSTM is applied to determine the temporal dependencies. This work does not consider fake news as a binary classification but as multi-facets fake news characteristics identification.

According to Borges [6], it is vital to know what other news sources have a stance on the news. They built a Fake News Challenge 1 (FNC-1) system to serve the purpose. It checks the news stance to the headline for the agrees, disagree, discuss, or unrelated classifications. The work solves the stance detection using similarity features using LSTM. The stance detection system labels the news as true or false is based on the credibility weightage of the sources.

Bahad et al., [9] has applied traditional LSTM, Bi-directional LSTM, CNN, and GloVe word embeddings to solve the binary classification problem of fake and true news detection articles. Results combined to get the average of the cascading output. The approach is reasonable but a repeat cycle to come up with a global maximum position vector where the model is accurate lacks time efficiency in the calculation since more time is required to search for optimal parameters for which the model is most accurate.

[10] has evaluated algorithms on a cascading basis. The algorithms include Support Vector Machine (SVM), Decision Tree, Random Forest, and Gradient descent optimization for a dataset containing 11,000 news articles.

The text embeddings are the deep representation of vectors. It enables to measure correlations between text by calculating the distance among vectors. Neural networking has proved its performance in Natural Language Processing with the pre-trained word embeddings [9] [11]. A CNN is feature engineering applied to generate n-grams or BagOfWords (BoW) to extract text features [12] [9].

1) Accuracy of the Model: Alam [13] has summarized different methods that used for fake news detection in table I. This work is augmented to automate the discovery of global maximum position vectors from three-dimensional space. That increases the time, space, and accuracy of the trained model as a result of augmentation.

TABLE I
ACCURACY OF THE MODELS

Sr.	Feature	ML Method	Dataset	Accuracy
1.	Tri-relationship among publishers, news pieces and users. User is assigned credibility score based on online behavior	Linear Classifier	FakeNewsNet	87.8%
2.	User characteristics based on local and global news propagation	Neural Network	Weibo and Twitter	92%
3.	Linguistic	Neural Network	Twitter	95%
4.	Stylometry	SVM Classifier	Kickstarter	96.6%

III. AUGMENTED LSTM-PSO MODEL

Conroy et al. [14] categorize deception into linguistic and network approach. The linguistic way deals with semantic analysis of the text, whereas the network approach uses a knowledge network for fact-checking using metadata and behavioral questions. Machine learning tools require data to be fit for the model to learn. Textual data cannot be fit directly into the model without having it fine-tuned for the machine learning model. Denotational Semantics is representing an idea as a one-hot vector. It is a generalized form and cannot be relied upon for the true meaning. Hence, it is called localist representation, for each neuron represents a single concept. Context of the word is a set of *m* surrounding words and based on the idea that similar words have similar context and are called similarity. There are two methods:

- one-hot encoding of tokens
- Integer index converted to a binary vector of size N

Bag-of-words is a feature engineering tool consisting of an unordered structure of words used in shallow languages processing models such as logistic models and random forests. BoW categorizes similar context words in a single vector. The Embedding layer uses Word2vec for data representation in vector form. Natural Language Processing (NLP) converts text into numbers and then feeds those into artificial intelligence (AI) models or trains them to make predictions for us. Companies and media must check automatically whether the news is fake or not. Text data includes news, social media platforms, and news articles. Twitter API key is required to collect Twitter data using search string features. Use the dataset for comparison time required to tune hyperparameters, search space, and accuracy of the augmented artificial intelligence model with the table I.

A. NLP Process

Steps to build a fake news detection model are:

Step 1: Cleaning Data

Tweets are transformed from social media excluding jargon, special symbols, usernames, and single-letter words to discern meaningful words by processing text data from Twitter. The task performed by using python libraries like *gensim*.

objective function of LSTM = binary cross-entropy + regularization term

Step 2: Tokenize

Text data transform to a list of dictionary words, refer to the python nltk library for detail. The result is a structured list of words called n-gram or BagofWords (BoW) an internal vocabulary. It creates word indexes based on frequency. The method *texts_on_sequences* transforms texts into a sequence of integers with a corresponding integer value from word index from the method *fit_on_texts*. The deep learning model uses the statistical structure of the language to learn the pattern of text using numeric tensors that is *vectorization*. The token is breaking down the text into units with semantic meaning and is called tokenization. Word level tokenization creates a huge vocabulary size that results in memory problems. Character level token is more effective than word level in terms of effectiveness in getting semantics of the text to citecharacterleveltokenization. The training of the ML model is made possible by converting tokens to numeric vectors. This task of vectorization achieved using an embedding layer in LSTM.

Step 3: RNN

Now the data is viewed for word frequency and sampling distribution. The data can resize to achieve the desired level of accuracy for the model to train by the LSTM network later. The Gradient Descent minimize the cost function.

Step 4: Classification There are three types of NLP classification.

- Rule-based systems
- Machine-learning based
- Hybrid systems

Natural Language Processing (NLP) converts text into numbers and is given to the machine learning model as input to train and predict. Companies and media need to generate a model backed by automatically tuning hyperparameters since it is a hectic task.

B. Neural Networks

Long Short-Term Memory (LSTM) is an extended version of recurrent neural network (RNN) architecture. Textual data represented in vector form after processing language features. These include dimensional reduction and cleaning of data. Data in vector form is an input to the neural network model[1]. Disaster Tweets version 3 [3] of twitter dataset is used to detect fake news using the method as discussed in III-A.

C. Particle Swarm Optimization (PSO) and LSTM

The advantage of using PSO is the utilization of diversity of search space before convergence to the optimal solution. PSO consists of particles position and velocity in the three

[1]http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/

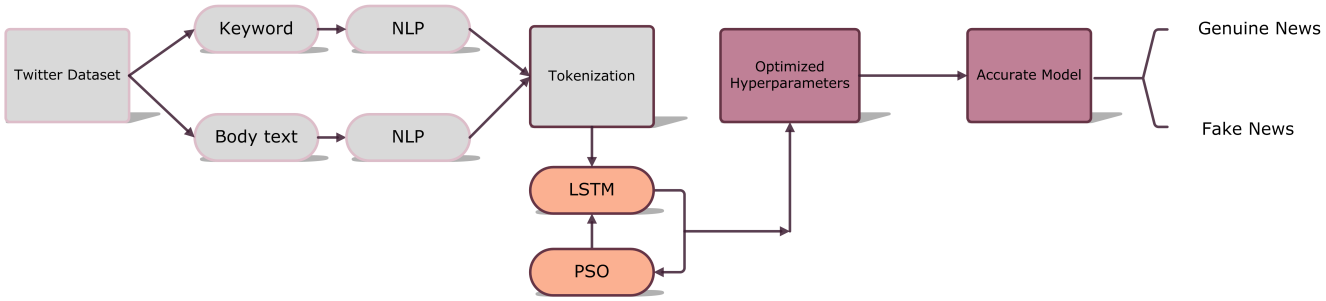


Fig. 2. LSTM-PSO Model for Hyperparameters Search and Fake News Detection

dimensional search space. Two parameters of LSTM are given as an input to PSO for the iterations and to come up with the most accurate position vectors that results in the most accurate fake news detection model and without the hassle of hyperparameters tuning since the task is performed by PSO algorithm. The pseudo code for the algorithm is given below [15].

1) *for each particle i in S do*
2) *for each dimension d in D do*
3) **initialize all particles position and velocity**
4) $x_{i,d} = Rnd(x_{min}, x_{max})$
5) $v_{i,d} = Rnd(-v_{max}/3, v_{max}/3)$
6) *end for*
7) **initialize particles best position**
8) $pb_i = x_i$
9) **update the global best position**
10) $if\, f(pb_i) < f(gb) then$
11) $gb = pb_i$
12) *end if*
13) *end for*

Line 1 of the pseudo-code initializes the swarm particles with control parameters in lines 2 to 5. Line 8 initializes the particle's best position that overwrites value based on the global best position vector declared in line 11. The search space converges to a position vector where the global best optimal solution arrives while iterating. The two parameters of LSTM given to PSO are the neurons and epochs.

IV. Experiments and Results

Particle swarm intelligence algorithm acts as a heuristic search to come up with a guess for the global maximum parameters. There are two main features $\mathcal{U}$ and $\mathcal{V}$ given to the LSTM model for training. $\mathcal{U}$ is a vector representation of categorized tweet hashtag while $\mathcal{V}$ is a vector representation of the body of tweet data. The evolutionary algorithm runs through a three-dimensional search space to optimize hyperparameters such as the number of neurons and number of epochs for which the model is most accurate. For the test and train of the model, disaster tweets dataset [3] has been used. To train the model, LSTM and PSO algorithms are used to adapt optimized parameters. Accuracy as a parameter is used to identify true or fake news. It is a ratio of correct predictions to the number of total samples. For the unique dataset containing over 11,000 tweets, to avoid over generalization and under performance, data is split in the ratio of 80:10:10 for test and train of the model. The global maximum result from the heuristic search of PSO with train, validation and test partition for LSTM is shown in Table III.

TABLE II
Data Attributes

Attribute	Type
Keyword	Text
Text	Body of text
Label	1 for True; 0 for Fake

TABLE III
Data Partition

Model	Train Accuracy	Validation	Model Accuracy
LSTM	0.9943	0.819	0.838

TABLE IV
Data split for Test and Train of the model

No. of Neurons	epoch	Trainable Parameters	Batch Size	Accuracy
60	6	141,241	40	83.80%

V. Conclusion

The training data contains 942 unique tokens and a total of 11,370 word vectors. Table IV shows the augmented sparsity of the Vanilla RNN, and Bidirectional LSTM split for test size, train size, and validation accuracy on disaster news dataset [3]. The text converted to vectors for the deep representation of machine learning models for sustainable reliability and accuracy as discussed in III-A. Hybridizing LSTM with the swarm intelligence algorithm improves the performance of the fake news detection model since it automates the hyperparameters tuning. Input parameters of LSTM are searched by the PSO algorithm to achieve global maximum values of epoch and LSTM neurons.

This work proposed NLP and artificial intelligence deep learning models to detect fake and real news. To generate the model, NLP feature engineering first cleans the data, word-level embedding applied to the keyword and body of the tweet, then the result is given as an input to the embedding layer of LSTM. This process augments the time and space search for hyperparameters using a random search of PSO. For future work, data size needs to increase, and a more robust search algorithm needs to develop for the hyperparameters.

REFERENCES

[1] D. Paskin, "Real or fake news: Who knows?" *Social media and society*, vol. 7, pp. 252–273, 2018.

[2] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, May 2017. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211

[3] V. Stepanenko, "Disaster tweets," kaggle, Nov. 2020, updated version 3. [Online]. Available: https://www.kaggle.com/vstepanenko/disaster-tweets/

[4] A. Yu. (2019, Feb.) Getting a machine to do my english homework for me. Website. Last accessed 20 January, 2021. [Online]. Available: https://towardsdatascience.com/getting-a-machine-to-do-my-english-homework-for-me-5d339470fe42

[5] A. Zheng, N. Shelby, and E. Volckhausen, "Evaluating machine learning models," in *Evaluating Machine Learning Models*, 2015.

[6] L. Borges, B. Martins, and P. Calado, "Combining similarity features and deep representation learning for stance detection in the context of checking fake news," *J. Data and Information Quality*, vol. 11, no. 3, Jun. 2019. [Online]. Available: https://doi.org/10.1145/3287763

[7] V. Veitch, D. Sridhar, and D. M. Blei, "Adapting text embeddings for causal inference," 2020.

[8] H. Karimi, P. Roy, S. Saba-Sadiya, and J. Tang, "Multi-source multi-class fake news detection," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1546–1557. [Online]. Available: https://www.aclweb.org/anthology/C18-1131

[9] P. Bahad, P. Saxena, and R. Kamal, "Fake news detection using bi-directional lstm-recurrent neural network," *Procedia Computer Science*, vol. 165, pp. 74 – 82, 2019, 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050920300806

[10] S. Gilda, "Evaluating machine learning algorithms for fake news detection," *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, pp. 110–115, 2017.

[11] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, p. 2267–2273.

[12] Y. Kim, "Convolutional neural networks for sentence classification."

[13] S. Alam and A. Ravshanbekov, "Sieving fake news from genuine: A synopsis," 2019.

[14] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to generate reviews and discovering sentiment," 2017.

[15] D. J. Toal, N. W. Bressloff, A. J. Keane, and C. M. Holden, "The development of a hybridized particle swarm for kriging hyperparameter tuning," *Engineering Optimization*, vol. 43, no. 6, pp. 675–699, 2011. [Online]. Available: https://doi.org/10.1080/0305215X.2010.508524