Assessing the Spreading Behavior of The Covid-19 Epidemic: A Case Study of Turkey

1st Erdem Demir
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
demir.erdem@std.izu.edu.tr

2nd Muhammed Nafiz Canıtez
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
canitez.muhammed@std.izu.edu.tr

3rd Mohamed Elazab
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
elazab.mohammed@std.izu.edu.tr

4th Alaa Ali Hameed
Department of Computer Engineering
Istinye University
Istanbul, Turkey
0000-0002-8514-9255

5th Akhtar Jamil
Department of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
0000-0002-2592-1039

6th Abdullah Ahmed Al-Dulaimi
Department of Electrical and Electronics
Karabuk University
Istanbul, Turkey
0000-0001-8741-9450

2022 2nd International Conference on Computing and Machine Intelligence (ICMI) | 978-1-6654-7483-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMI55296.2022.9873697

***Abstract*—Coronavirus (Covid-19) disease is a rapidly spreading type of virus that was discovered in Wuhan, China, and emerged towards the end of 2019. During this period, various studies were conducted, and intensive studies are continued in different fields regarding coronavirus, especially in the field of medicine. The virus continues to spread and is yet to be controlled fully. Machine learning is a well-explored field in the domain of computer science that can learn patterns based on existing data and make predictions on new data. This study focused on using various machine learning approaches for predicting the spreading behavior of the COVID-19 virus. The models that were considered include SARIMAX, Extreme Gradient Boosting (XGBoost), Linear Regression (LR), Decision Tree (DT), Gradient Boosting (GB), and Artificial Neural Network (ANN). The models were trained and then predictions were made by applying these models to the daily updated data provided by the Turkish Ministry of Health. Experiments on the test data showed that both XGBoost and Decision Tree models outperformed other models.**

***Index Terms*—COIVD-19 prediction, SARS-CoV2, machine learning, automatic prediction of COVID-19**

I. Introduction

Coronavirus is one of the main pathogens that target the human respiratory system and was first discovered in the 21st century [1]. Regarding history, severe acute respiratory syndrome-CoV, [SARS-CoV] was discovered in Guangdong, China in 2003 [2], and Middle East Respiratory Syndrome-CoV, [MERS-CoV] was discovered in the Arabian Peninsula in 2012 [3] are both types of coronavirus. After these two types, Covid-19 is considered the third biggest pandemic in the 21st century [4].

Centers for Disease Control and Prevention, [CDC] declared it a member of the CoV family and was given the name Covid-19. Later The World Health Organization (WHO) declared it a pandemic. Covid-19 is assumed that it was transmitted to a human being at the Huanan Seafood Market in Wuhan, China, and it is currently rapidly spreading. Covid-19 has symptoms on the human body such as coughing, fever, weakness, shortness of breath, and sore throat. The diagnosis of Covid-19 can be done by test kits developed specifically for this virus.

Precautions against Covid-19 can be taken by wearing masks, using disinfectants, and maintaining social distance. Simultaneously, a developed vaccine by an American company called Pfizer and a German company called BioNTech has a 95% effectiveness rate. Another vaccine developed by Russia, Sputnik V, has a 92% effectiveness rate. Also, scientists in China developed a vaccine called Sinovac (CoronaVac) that was based on Viral Vector and RNA techniques, it has a more than 50% effectiveness rate.

According to the current data provided by WHO, the number of confirmed cases worldwide on May 30, 2022, is 584 million, the number of deaths from the virus is 6.4 million, and the number of administered vaccines is around 4894 million [5]. As of May 30, 2022, the number of confirmed cases in Turkey is 16.2 million, the number of deaths from the virus is 99,678 thousand, and the total number of administered vaccines is around 150.2 million [6]. The number of cases and deaths worldwide continues to decrease, and the number of people vaccinated is increasing.

Artificial intelligence-based approaches can be used as a tool for the diagnosis and tracking of COVID-positive cases. Literature shows that a number of works have been dedicated to the analysis of COVID-related issues. In [7] authors proposed a machine learning-based framework for prediction of COVID positive cases from x-Ray images. In [8] a new deep learning based method called CoroDet is proposed for classification of COVID cases from chest X-Ray images. Similarly, [9] proposed deep learning models based on fuzzy color and stacking approaches for COVID-19 detection from X-ray images. Authors in [10] employed transfer learning for the detection of COVID cases using multimodal images. The three most commonly used images were used: X-Ray, Ultrasound, and

CT scan. A residual network pre-trained on ImageNet data set was used for the training and classification of COVID cases. Both supervised and unsupervised learning algorithms were used in [11] for COVID case identification. The results showed that supervised produced superior accuracy than unsupervised methods. In [12] five clinical symptoms of COVID positive cases are used as input for the machine learning model for its classification. The system can be used to help prioritize testing for COVID-19 with high chances of positive cases. For further details, refer to the paper [13] that provides a detailed account of recent works based on AI for COVID pandemic.

In this paper, six machine learning approaches are compared for COVID case prediction using real-time data. The performance of each model was then compared to find the best model for prediction. Based on the experimental results, we conclude that all models produced acceptable results but XGBoost and DT were more accurate for COVID prediction than the other four models.

The paper is organized as follows. Section II describes how the data was obtained and processed before being given to the model. Moreover, we provide a brief description of each model in this section. Section III provides a detailed account of the experiments and their results. Finally, the conclusion summarizes the work and provides a future perspective.

II. MATERIALS AND METHOD

This section provides description of the data set and the summary of the methods used in this study. The overall workflow of the proposed method is shown in Fig. 1.

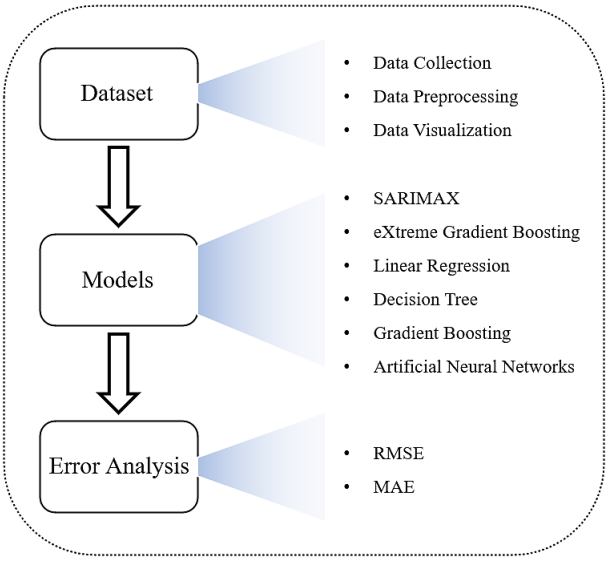

Fig. 1. Workflow of the proposed method.

A. Data Collection and Preprocessing

The data set of Covid-19 cases across Turkey was obtained from the Turkish Ministry of Health. This data is constantly updated daily and shared openly with the public [14]. This data set was integrated using Python's Selenium library. The daily trend of cases seems to be slowed down and the number of vaccinations is increasing. The data used starts from the first case of the data set on March 11, 2020, till the day we downloaded the data. Table I shows the details of the data set, i.e. variable names, their type, and the description of each feature.

The data we employ in this study is categorized as time series. Such data is stored by time in chronological order. Rows of data are ordered in a periodic (hour, day, month, year, etc.) loop. Time series can also be defined as simple data set in which events and operations that can be expressed numerically are associated with a timestamp [15].

B. SARIMAX

Machine Learning methods allow us to generate knowledge from data sets [16]. SARIMAX is a machine learning method that can be used to derive patterns from the input data. There are some specific models that can be used to deal with the time series data. One of the well-known models is called autoregressive integrated moving average (ARIMA). Researchers proposed a generalized model for the ARIMA known as seasonal-ARIMA termed SARIMA. It has the capability to deal with the seasonality within the data. It technically applies seasonality with autoregression (AR), a moving average (MA), and differentiation term.

SARIM, however, is not able to deal with the effect of exogenous variables [17]. A variation in SARIM was introduced to deal with the impact of exogenous variables and the model is called seasonal autoregressive integrated moving average with external or exogenous regressor (SARIMAX).

C. eXtreme Gradient Boosting

The XGBoosting algorithm has gained popularity for its simplicity and ability to produce high accuracy for various problems. It can be employed for different types of problems such as classification and regression tasks. Even it can also be used in clustering problems [18].

Boosted trees in the Extreme Gradient Boosting algorithm are split into regression and classification trees. The essence of this algorithm is based on optimizing the objective function value [19]. The XGBoosting algorithm can be very fast when implemented in parallel or distributed computing environments [20].

D. Gradient Boosting

Gradient boosting is a robust method in the domain of machine learning. It has been widely used for both classification and regression problems. Gradient boosting algorithm combines several weak learns to form a strong model. It employs optimization techniques that result in a minimum error. The weak learners are added iteratively that minimizes the loss.

Given a data set of N samples $D = \{x_i, y_i\}$, gradient boosting algorithm iteratively approximates a function $F(x)$ that maps

TABLE I
DATA SET DESCRIPTION USED IN THIS STUDY

Variable	Type	Description
Date	Float	The date the case was seen
Total Number of Tests	Float	Total number of validated tests
Total Number of Cases	Float	Total number of confirmed cases
Total Number of Deaths	Float	Total number of deaths due to the disease
Pneumonia Rate in Patients	Float	Pneumonia rate in daily cases
Number of Critically Ill Patients	Float	Number of cases in an intensive care unit (ICU)
Total Number of Recovered Patients	Float	Total number of recovered patients
Number of Confirmed Cases per Day	Float	Number of confirmed cases per day

input samples x to the output y by minimizing the loss function $L(x, F(x))$. Mathematically, we can express the equation as

$$F_m(x) = F_{i-1}(x) + w_i h_i(x) \tag{1}$$

where w_i is the weight associated with the function $h_i(x)$.

E. Linear Regression (LR)

LR models are simply used to find the relationship between variables [21]. The variable is called dependent and independent variables. Simple linear regression involves one dependent and one independent variable. Multivariable linear regression includes finding relationships between multiple variables. Simple linear regression between input data x and the output variable y can be expressed as:

$$y = mx + b \tag{2}$$

where b is the bias term. Similarly, a multivariable linear regression of three variables x, y and z can be represented as

$$f(x, y, z) = w_1 x + w_2 y + w_3 z + b \tag{3}$$

F. Decision Tree

The decision tree method tries to find the best order in predicting the target by performing many tests. Each test creates branches in the decision tree, and these branches cause other tests to occur. This continues until the test process terminates at a terminal node. The termination of the splitting and pruning operations, which are the two basic operations of the decision tree, is based on the stopping criterion applied. These operations and the stopping criterion can be briefly explained as follows.

• Splitting: This process is an iterative process that allows data to be broken down into smaller subsets. The model gradually divides the data in subset based on features. Each subset is further divided into small sets using further features until it reaches the unit level features. In each split, the variables are analyzed, and the best split is chosen.

• Pruning: After a tree is created, unwanted subtrees or nodes may be found. By removing them with pruning, the decision tree can be expressed in a more general form.

• Stopping criterion: Growing too deep trees may result in overfitting of data. Therefore, defining early stopping criteria may help overcome this issue and stop growing the trees further. The stopping criteria may be based on maximum depth, a number of elements in the split used error at each note, etc.

G. Artificial Neural Network (ANN)

ANN is a nonparametric machine learning approach. It has received a lot of attention due to its efficiency and high accuracy in understanding more complex patterns in the data. Various architectures have been proposed in the literature, but the most commonly used architecture is the multi-layer perceptron (MLP). The MLP architecture consists of a number of neurons arranged in layers. These layers are named input, hidden, and output layers. The input layer consists of the same number of neurons as the input features, the hidden layer consists of any number of neurons, while the output layers have the same number of neurons as the number of classes. The complexity of the network is proportional to the number of neurons at each layer, particularly at the hidden layer. Generally, the higher the number of neurons and layers, the network is able to learn more complex patterns [22]. However, for simple features, a relatively less number of neurons are selected to avoid overfitting.

The neurons in each layer are connected with neurons in the following layer via weights [23]. The working of the ANN is inspired by their biological counterparts. The brain has a massive number of parallel processing neurons that can process a large amount of information [24]. The neurons in the ANN receive input as a weighted sum of signals from other neurons at the previous layer and then apply an activation function. The main objective of the activation function is to introduce nonlinearity in the data. Neural networks are generally trained using the most popular gradient descent technique. The weights in the neural network are updated according to the following equation

$$w_i(n+1) = w_i(n) + \Delta w_i(n) \tag{4}$$

where $\Delta w_i(n)$ is weight correction term which can be obtained using

$$\Delta w_i(n) = \alpha x_i(n) e(n) \tag{5}$$

Where α is the learning rate and $e(n)$ is the epsilon. Learning rate controls the amount of error to be included in the weight updation. Higher value may help reach the optima faster while a small value can make the learning process slow. Great care

must be taken in selection of the learning rate because a large value can also introduce problems of skipping the optima.

H. Evaluation Metrics

The two most commonly used evaluation metrics were employed in this study for our proposed models. These two metrics include Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). RMSE and MAE, on the other hand, since they are error measures; low values mean high performance because they are inversely proportional to performance [25]. For example, if RMSE is equal to zero, the performance is very well. These metrics were calculated according to the following standard equations:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y_i} - y_i)^2} \tag{6}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\hat{y_i} - y_i| \tag{7}$$

where $\hat{y_i}$ and y_i represent model prediction and actual value, respectively.

III. Experimental Results

This section provides a detailed account of experiments performed on the data set using all models. As mentioned earlier, six conventional machine learning models were used in evaluation, which include SARIMAX, XGBoost, LR, DT, Gradient Boosting, and ANN.

For all models, the implementation was done using Python with Scikit Learn and Tensorflow environments. The models were run on a computer with 8GB RAM, Intel Core i7-7700Hz processor, and an NVIDIA ™ MX 450 Graphic Processor.

The data set was divided into two separate subsets: 80 % for training and 20 % for testing. Moreover, each model hyperparameters were obtained using grid search method. These optimal set of parameters were then used during training to find the optimal values. The trained models were then save and reused during testing for evaluation.

Evaluations were performed using RMSE and MAE. The predicted values of the model were then compared with the actual values. The model's performance was assessed using the RMSE and MAE error metrics according to equation (6) and (7). Table II summarizes the results obtained for each model and their corresponding error values in terms of RMSE and MAE. These results show that both XGBoosting and DT were more efficient for COVID-19 prediction than other models as they produced less RMSE and MAE. The SARIMAX model produced an RMSE value of 1945 and an MAE value of 1042. Similarly. for the XGBoosting model RMSE and MAE were 1500 and 845, respectively. The Linear Regression model produced an RMSE value of 4330 and an MAE value of 3183. The Decision Tree model produced an RMSE value of 1595 and an MAE value of 853. The Gradient Boosting model resulted in RMSE of 1720 and MAE of 946. Also for ANN RMSE value was 1912 and MAE as 1200. The Linear Regression model did not perform well for the purpose of this study.

Fig. 2-7 also shows the comparison of results for each model used. We can see that the results are confirming to the quantitative results shown in Table II. The visual results show verified number of real case vs the predicted cases by corresponding models. The real values are highlighted in green and predicted values are highlighted in red color.

It was observed that initially the number of cases were increasing exponentially then the cases started to decrease gradually. However, there was not the increase and decrease were not showing any uniform changes due to various reasons. In some cases, the cases were increasing sharping and sometimes they were decreasing drastically. These two situations were dependent on how the government was reactin to the situation. For instance, in hot spot areas, there were strict restrictions on movement for at least couple of weeks. Then drastic decrease was observed in such scenatios. The non regular changes also effected the performance of machine learning models. However, integrating some external factors into consideration, such as curfews and quarantine procedures, can have a significant impact on the prediction performance of models for Covid-19 cases. In light of these results, it's clear that good results can be obtained using real-time data. On the other hand, availability of a small size data set and the presence of many other factors that may affect the number of future cases, are making future predictions might be challenging.

TABLE II
COMPARISON OF MODELS IN TERMS OF RMSE AND MAE

Model	RMSE	MAE
Linear Regression	4330	3183
SARIMAX	1945	1042
Artificial Neural Network	1912	1200
Gradient Boosting	1720	946
Decision Tree	1595	853
eXtreme Gradient Boosting	1500	845

IV. Conclusion

The fast-spreading Covid-19 virus forced authorities to declare medical emergency in the whole world. Currently, the number of positive cases have decreased and the number of vaccinated people have increase. However, the danger is not over yet. Research regarding the virus is being conducted by different domains of science. This paper focused on tracking the spread behavior of the COVID-19 virus that can help authorities to make better decisions. Using Time Series, Machine Learning, and Artificial Neural Network models, research can be carried out in the field of computer science to handle the study of virus spread, proper prevention strategies, and virus treatments around the world. In this study, performance of six models were compared for the predictions of COVID-19 cases using the data provided by the Turkish Ministry of Health. These models include SARIMAX, Extreme Gradient Boosting (XGBoost), Linear Regression (LR), Decision Tree (DT), Gradient Boosting (GB), and Artificial Neural Network

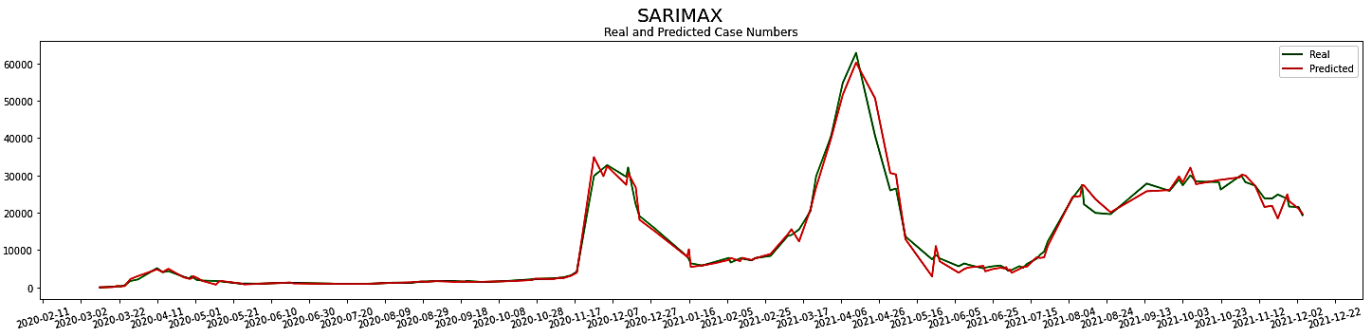

Fig. 2. Comparison of results SARIMAX model.

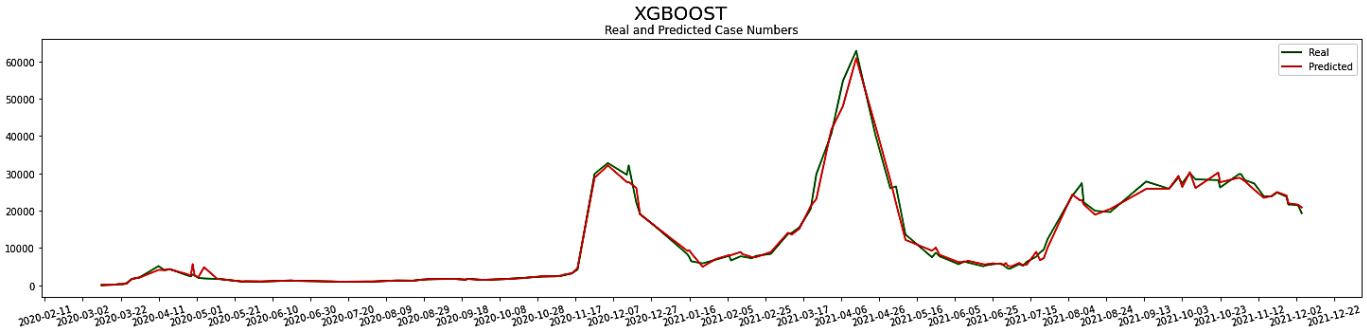

Fig. 3. Comparison of results XGBoost model.

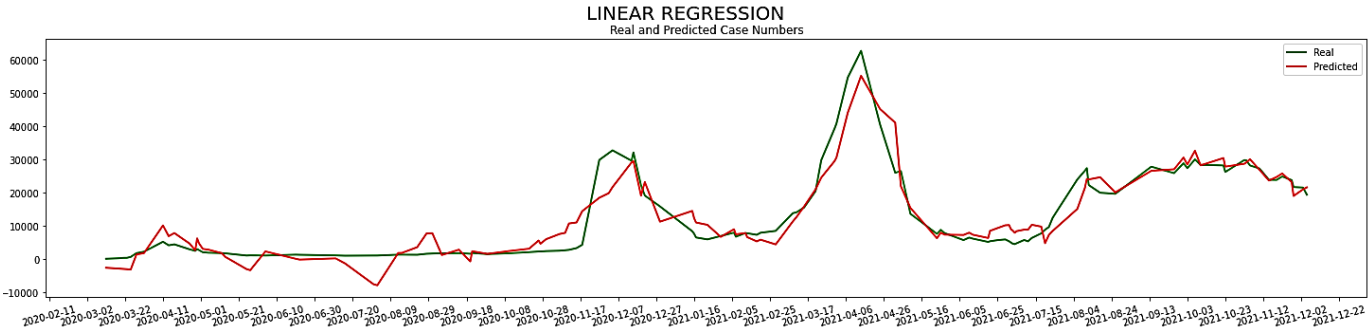

Fig. 4. Comparison of results Linear Regression model.

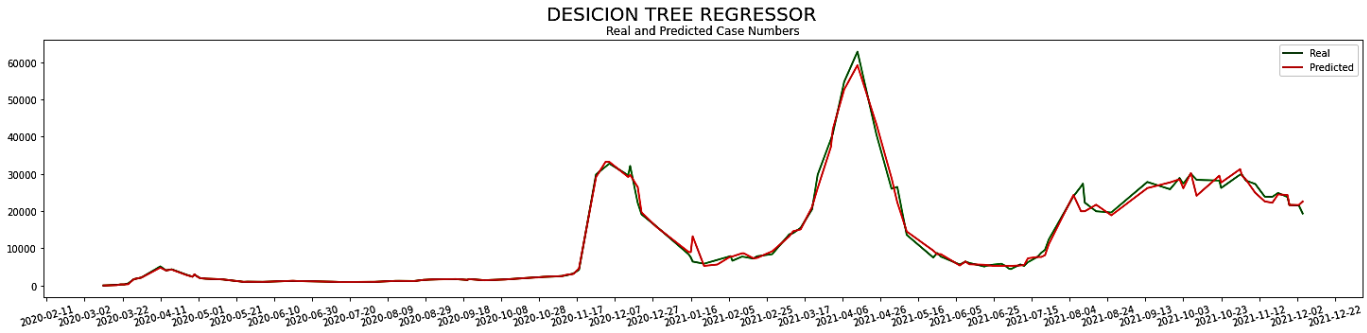

Fig. 5. Comparison of results Decision Tree model.

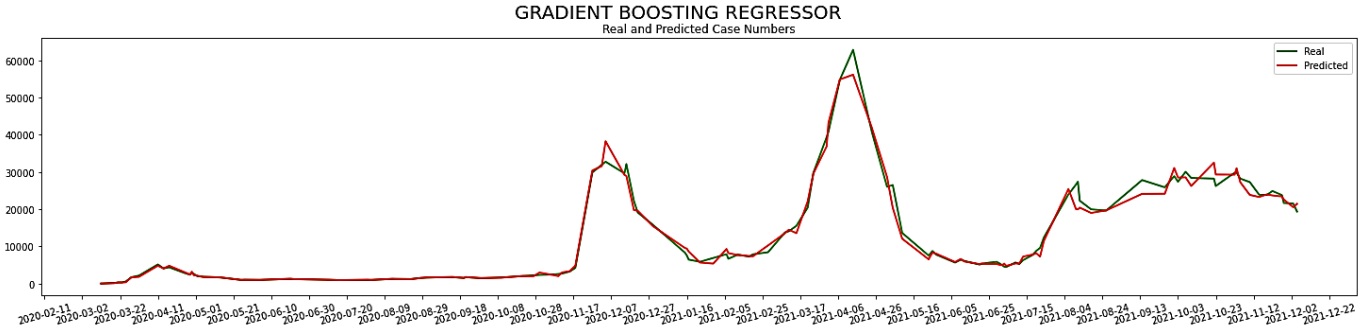

Fig. 6. Comparison of results Gradient Boost model.

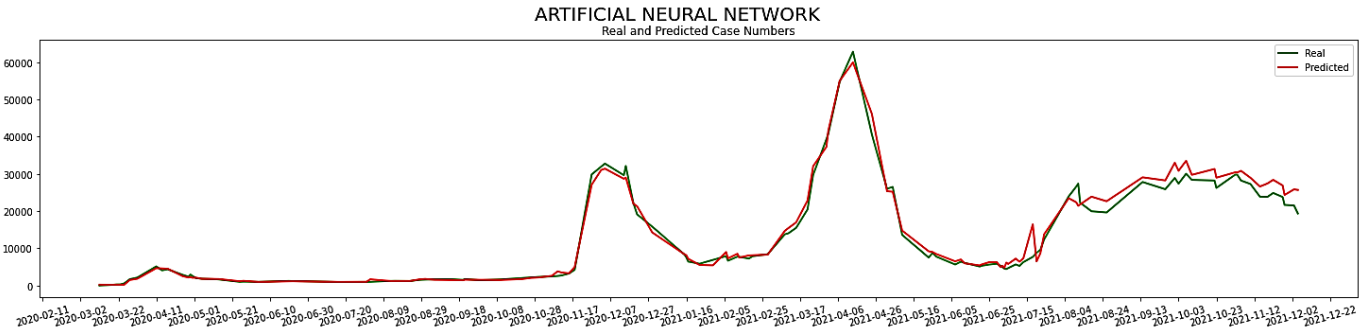

Fig. 7. Comparison of results ANN model.

(ANN). All models were efficient in tracking COVID-19 cases, however, XGBoost and DT showed promising performance. In future, we would like to include more data and apply deep learning based models for improving the overall accuracy of prediction. In addition, we plan to integrate IoT enabled devices to provide additional information such as body temperature that can help make better predictions.

REFERENCES

[1] K. Mcintosh, S. Perlman, and . Mand, *Coronavir uses, including severe acute respiratory syndrome (SARS) and Middle East respiratory syndrome (MERS)*. Elsevier Saunders, 2015.

[2] P. K. Chan and M. C. Chan, "Tracing the SARS coronavirus," *J. Thorac. Dis*, vol. 5, no. 2, 2013.

[3] R. J. De Groot, "Commentary: Middle east respiratory syndrome coronavirus (MERS CoV): announcement of the coronavirus study group," *J. Virol*, vol. 87, no. 14, pp. 7790–7792, 2013.

[4] E. R. Görkem and S. Ünal, "Koronavirüs salgını anlık durum ve ilk izlenimler," *FLORA*, vol. 25, no. 8, 2020.

[5] "WHO coronavirus (COVID-19) dashboard," https://covid19.who.int, accessed: 2022-8-13.

[6] "Covid19," https://covid19.saglik.gov.tr/, accessed: 2022-8-13.

[7] J. Rasheed, A. A. Hameed, C. Djeddi, A. Jamil, and F. Al-Turjman, "A machine learning-based framework for diagnosis of covid-19 from chest x-ray images," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 13, no. 1, pp. 103–117, 2021.

[8] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, and M. Z. Parvez, "Corodet: A deep learning based classification for covid-19 detection using chest x-ray images," *Chaos, Solitons & Fractals*, vol. 142, p. 110495, 2021.

[9] M. Toğaçar, B. Ergen, and Z. Cömert, "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*, vol. 121, p. 103805, 2020.

[10] M. J. Horry, S. Chakraborty, M. Paul, A. Ulhaq, B. Pradhan, M. Saha, and N. Shukla, "Covid-19 detection through transfer learning using multimodal imaging data," *Ieee Access*, vol. 8, pp. 149 808–149 824, 2020.

[11] A. S. Kwekha-Rashid, H. N. Abduljabbar, and B. Alhayani, "Coronavirus disease (covid-19) cases analysis using machine-learning applications," *Applied Nanoscience*, pp. 1–13, 2021.

[12] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of covid-19 diagnosis based on symptoms," *npj digital medicine*, vol. 4, no. 1, pp. 1–5, 2021.

[13] J. Rasheed, A. Jamil, A. A. Hameed, U. Aftab, J. Aftab, S. A. Shah, and D. Draheim, "A survey on artificial intelligence approaches in supporting frontline workers and decision makers for the covid-19 pandemic," *Chaos, Solitons & Fractals*, vol. 141, p. 110337, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077920307323

[14] "Genel koronavirüs tablosu," https://covid19.saglik.gov.tr/TR-66935/genel-koronavirus-tablosu.html, accessed: 2022-8-13.

[15] E. P. George, M. Box; Gwilym, C. Jenkins; Gregory, and M. Reinsel; Greta, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[16] T. M. Mitchell, *Machine Learning*, 1st ed. New York NY: McGraw-Hill, 1997.

[17] C. Chatfield, *Time-Series Forecasting*, 1st ed. New York NY: CRC, 2000.

[18] Mitchell, F. Rory, and Eibe, "Accelerating the xgboost algorithm using gpu computing," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 3, pp. 127–164, 2017.

[19] Z. Huiting, Y. Jiabin, and C. Long, "Short-Term load forecasting using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation," *Energies*, vol. 10, no. 8, pp. 1168–1188, 2017.

[20] T. . Chen and C. Guestrin, "XGBoost: A scalable tree boosting system"," in *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[21] A. Dey, "Machine learning algorithms: A review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[22] S. O. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson, Dec. 2010.
[23] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "{TensorFlow}: a system for {Large-Scale} machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.
[24] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002.
[25] W. Wang and Z. Xu, "A heuristic training for support vector regression," *Neurocomputing*, vol. 61, pp. 259–275, 2004, hybrid Neurocomputing: Selected Papers from the 2nd International Conference on Hybrid Intelligent Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231203005307