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.

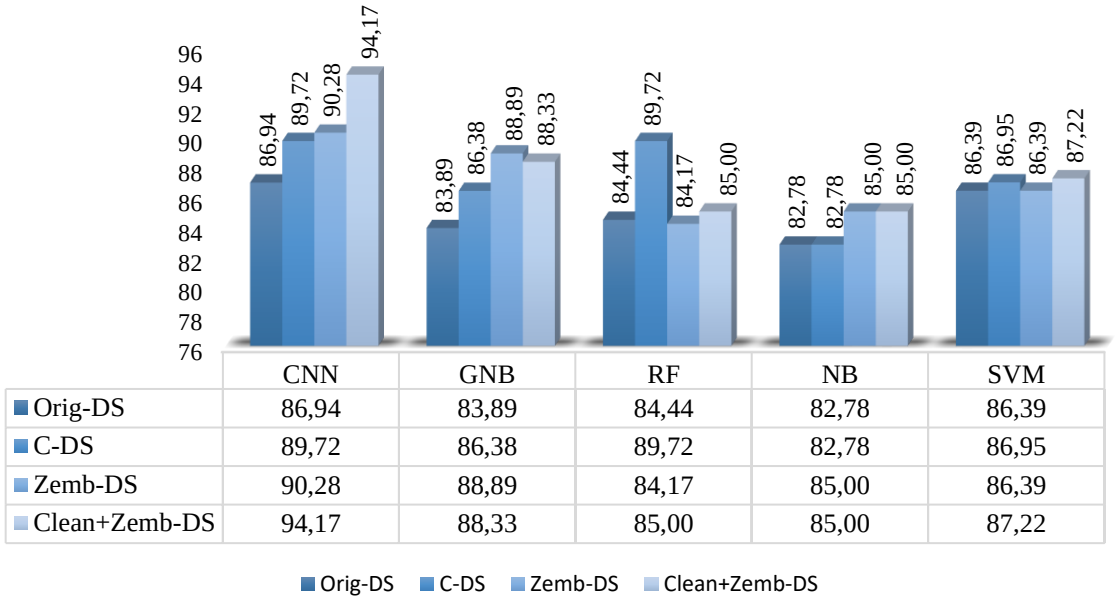

Fig. 1. Comparison of accuracy rates of CNN, GNB, RF, NB and SVM classification methods for each dataset.

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.

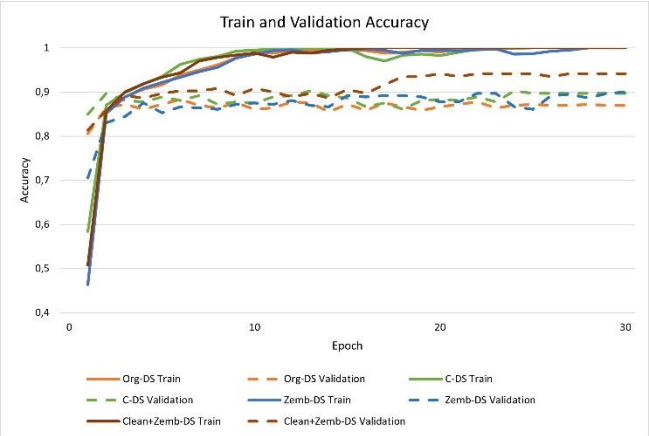

Fig. 2. CNN training and validation accuracy chart for each dataset.

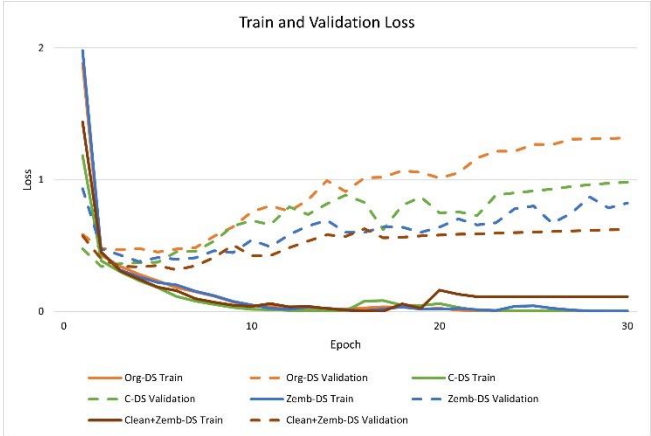

Fig. 3. CNN training and validation loss graph for each dataset.

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.