**JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES**
(ISSN : 1304-0448)
January 2022 Volume 15 Number 1
jast.hezarfen.msu.edu.tr

**Research Article**

# Multi-Class Document Classification Based on Deep Neural Network and Word2Vec

İlkay YELMEN[1*], Ali GÜNEŞ[2], Metin ZONTUL[3], Zafer ASLAN[4]

[1] *Istanbul Aydın University, Computer Engineering Department, 34295 Kucukcekmece, Istanbul, Turkey,* ilkayyelmen@stu.aydin.edu.tr, https://orcid.org/0000-0002-1684-9717
[2] *Istanbul Aydın University, Computer Engineering Department, 34295 Kucukcekmece, Istanbul, Turkey,* aligunes@aydin.edu.tr, https://orcid.org/0000-0001-6177-3136
[3] *Istanbul Ayvansaray University, Management Information Systems Department, 34020 Zeytinburnu, Istanbul, Turkey,* metinzontul@ayvansaray.edu.tr, https://orcid.org/0000-0002-7557-2981
[4] *Istanbul Aydın University, Computer Engineering Department, 34295 Kucukcekmece, Istanbul, Turkey,* zaferaslan@aydin.edu.tr, https://orcid.org/0000-0001-7707-7370
** Corresponding Author*

**Article Info**

***Received:*** *November 24, 2021*
***Accepted:*** *December 31, 2021*
***Online:*** *January 28, 2022*

**Keywords:** Document Classification, Multiclass Classification, Data Preprocessing, Word Embedding Methods, Machine Learning, Deep Learning

**Abstract**

With the increase in unstructured data, the importance of classification of text-based documents has increased. In particular, the classification of news texts and digital documentation provides easy access to the information sought. In this study, a large amount of news textual data was used. After the data set was preprocessed, Bag of Words (BoW), TF-IDF, Word2Vec and Doc2Vec word embedding methods were applied. In the classification phase, Random Forest (RF), Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Deep Neural Network (DNN) algorithms were applied. As a result of the experimental studies, using the Word2Vec method together with the DNN algorithm performed the best result.

**To Cite This Article:** İlkay YELMEN, Ali GÜNEŞ, Mehmet ZONTUL, Zafer ASLAN "Multi-Class Text Classification Based on Deep Neural Network and Word2Vec", Journal of Aeronautics and Space Technologies, Vol. 15, No. 1, pp. 59-65, Jan. 2022.

# Derin Sinir Ağı ve Word2Vec Tabanlı Çok Sınıflı Doküman Sınıflandırma

**Makale Bilgisi**

***Geliş:*** *24 Kasım 2021*
***Kabul:*** *31 Aralık 2021*
***Yayın:*** *28 Ocak 2022*

**Anahtar Kelimeler:** Doküman Sınıflandırma, Çok Sınıflı Sınıflandırma, Veri Ön İşleme, Kelime Temsil Yöntemleri, Makine Öğrenimi, Derin Öğrenme

**Öz**

Yapısal olmayan verilerin artmasıyla birlikte metin tabanlı belgelerin sınıflandırılmasının önemi artmıştır. Özellikle haber metinlerinin sınıflandırılması ve dijital dokümantasyon, aranan bilgilere kolay erişim sağlar. Bu çalışmada, büyük miktarda metinsel haber verisi kullanılmıştır. Veri seti ön işlemeye tabi tutulduktan sonra, Bag of Words (BoW), TF-IDF, Word2Vec ve Doc2Vec kelime temsil yöntemleri uygulanmıştır. Sınıflandırma aşamasında Random Forest (RF), Multilayer Perceptron (MLP), Support Vector Machine (SVM) ve Deep Neural Network (DNN) algoritmaları uygulanmıştır. Deneysel çalışmalar sonucunda DNN algoritması ile birlikte Word2Vec yönteminin kullanılması en iyi sonucu vermiştir.

## 1. INTRODUCTION

The Internet has been growing rapidly since 1981 [1]. In parallel with the increase in internet usage, a large amount of data is produced. According to Forbes data, text data produced in the near future will reach 40 zettabytes. In addition, while Google processes more than 40 thousand search results every second, this corresponds to approximately 3.5 billion searches per day [2].

A lot of textual data on the Internet is actually called unstructured data. As unstructured data increases, it becomes almost impossible to categorize this data manually. Research areas such as Natural Language Processing [3], Information Extraction [4] and Machine Learning [5] allow for automatic categorization of big data.

Text classification is a method of assigning predetermined one or more classes to a text data depending on its content. Text classification methods are used in many areas such as spam detection, e-mail filtering, news categorization, scientific articles indexing, etc. [6]. One of the good example of hybrid classifier systems usage can be found in [7].

Vectorization of words is as important as document classification algorithms. BoW is one of the most commonly used representation methods in document classification. Here, each text is expressed in N words. Each text has the term frequency (TF), inverse-term frequency (IDF), term-inverse-term frequency (TF-IDF) of the corresponding n-grams, etc. are represented using weighting methods [8].

Another representation method is the Word2Vec model, which is based on the principle of training words with an artificial neural network, which was developed in 2013 by Mikolov et al [9]. This model is related to the principle of predicting the target word based on the input words.

The rest of this article is structured as follows. In Section 2, literature review on text classification methods is discussed; then the details of dataset are presented in Section 3. The data preprocessing, word embeddings, classification and experiments and results are presented in Section 4, 5, 6 and 7 respectively. Finally, Section 9 concludes this paper.

## 2. LITERATURE REVIEW

Word embedding can be explained as the conversion of words from letters to numbers in vector form. Word embedding methods are preferred in this conversion process. When the articles in the literature are examined, it is accepted that word embedding is one of the most important points in text processing studies, especially with the latest developments in neural networks [8].

In another study [10], the authors combined semantic, syntactic, and lexical features to obtain better classification results. In addition to these, three different methods were used: Naïve Bayes, Nearest Neighbors (NN), SVM based on NGRAM bag and word bag models. In the study, it was also seen that combining semantic, syntactic and lexical features improved the classification results when using the SVM classifier.

Loni et al. [11], used a near Principal Component Analysis (PCA) size reduction method called Latest Semantic Analysis (LSA) to reduce the area of the feature size to a much smaller size. In addition, SVM and Back Propagation Neural Networks (BPNN) were used in their work. The results in their paper show that Backpropagation Neural Networks have better success rate than SVM.

Razzaghnooria et al. first used clustering algorithms to cluster words in their vocabulary and convert each question into a sparse vector. In the second method, they considered each feature vector as a linear combination of vectors of question words. The Word2vec method was used to extract the vector of the words and then the coefficients of the linear combination were determined using the tf-idf method. MLP and SVM were used for classification, and the highest accuracy was obtained from MLP with 72.46% [2].

Tilve and Jain [3] used VSM, Naive Bayes and Stanford Tagger for text classification algorithms on two different news datasets. When the 3 algorithms were compared, the Naive Bayes algorithm gave the best result. Gogoi and Sarma [4] emphasized the performance of using the Naive Bayes algorithm in document classification.

In a study on Turkish texts, the use of Word2Vec 60ord vectors was compared with the Bag of Words method. It has been stated that Word2Vec 60ord vectors give better results than the TF-IDF method [12].

Arshad has applied Logistic Regression (LR), Linear Support Vector Machine (SVC), Naive Bayes, BoW with Keras, LR with Doc2Vec and Word2Vec on Stack Overflow dataset which contain questions, answers, and tags. They have observed that these applied machine learning models give better result than human-generated baselines. The best result was obtained with 80% accuracy using Doc2Vec [13].

In another study on text analysis, the authors analyzed online reviews of users in the pharmaceutical field. In the research, LR, SVC, RF, Multinomial Naive Bayes, Extra Trees and Decision Trees algorithms were used in order to determine the medical status of users from their textual expressions. As a result, Linear SVC was found to be the most efficient model based on F1 Score, Recall and Precision metrics [14].

The motivation of this study is to classify texts with more than one class label by using different 60ite embedding methods together with deep learning and machine learning algorithms, apart from the methods that have been studied extensively in the literature. In the experiments conducted after the data collection and data preprocessing stages of this study, the best results were obtained when Word2Vec and DNN algorithms were used together.

## 3. DATASET

In this study, total 111.397 AG's news data were used [15]. The collected data is consisting of news title, description and class index (1-World, 2-Sports, 3-Business, 4-Sci/Tech).

## 4. DATA PREPROCESSING

In the data preprocessing phase, numeric characters, blockquotes, punctuation marks, links and stop words were removed in dataset. Then, all words were converted to lower case. After the normalization phase, word spelling correction was applied using the Python Symspell library. Finally, the words were taken to the roots to increase the success rate of the model. Since the rooting of the misspelled words would be wrong, after the spelling correction step, the rooting process was carried out using the snowball stem library. All data preprocessing steps are shown in Fig. 1.

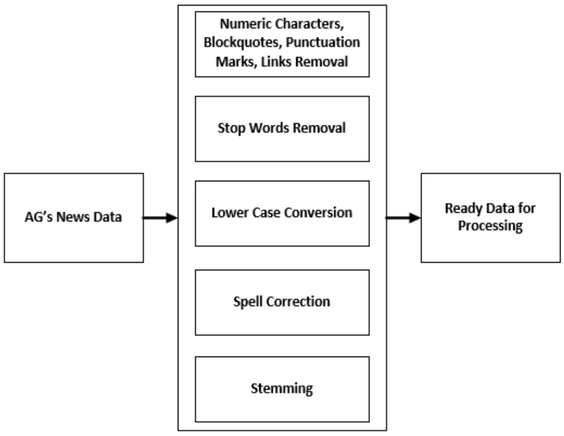


**Figure 1.** Preprocessing steps.

## 5. WORD EMBEDDINGS

### 5.1. Bag of Words (BoW)

The BoW method is a method that allows text to be converted to numbers [16]. In the BoW method, the entire text of the classification study is divided into terms. Each generated term is treated as an attribute. Then, the frequency of occurrence of each attribute, that is, the term, in the whole text is assigned as the value of the attribute. Thus, a categorical data is converted into numerical form.

### 5.2. Term Frequency Inverse Document Frequency (TF-IDF)

One of the numerical statistical method TF-IDF calculates the values of each word in a document as the inverse of the frequency of the word in a given document and the percentage of documents in which the word occurs. TF-IDF works by determining the relative frequency of words. In doing so, it works based on the inverse ratio of words in a particular document over the entire dataset. This calculation is done to determine how relevant a particular word is to a particular document. Words common to a single or small group of documents tend to have higher TF-IDF numbers than general words [17]. It is formulated in Eq. 1.

$$w_{t,d} = tf_{t,d} \times log \frac{N}{df_t} \tag{1}$$

where N is the total number of documents, $df_t$ is the document frequency of t and d denotes the each document.

### 5.3. Word2Vec

Word2Vec method was proposed by Mikolov et al [9]. This method establishes a close relationship between the word and its neighbors in a certain window size, and positions words that are close in meaning to each other in the vector space. It uses two different learning architectures to establish a meaning relationship. The first of these is the Continuous Bag of Words (CBoW) architecture. In this method, the word in the center of the window is tried to be guessed by looking at the neighbors of the word as close as the window size.

According to the Fig. 2, when any word is used as an input vector in a word dictionary consisting of 10,000 unique words, it is seen that the output of the network is a random vector containing the probability of each word in the dictionary and having a close meaning to the word.

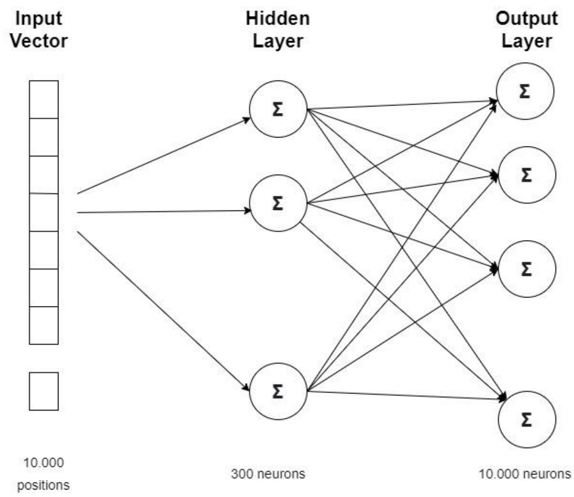


**Figure 2.** Word2Vec Model.

### 5.4. Doc2Vec

Doc2Vec has been developed to perform vector representation of sentences and documents in text pieces of variable size [18]. Doc2Vec also includes two different algorithms like Word2Vec. These algorithms are called Distributed Bag of Word (DBoW) and Distributed Memory (DM), respectively. The CBoW algorithm in Word2Vec has been updated to DBoW for Doc2Vec, and the SG algorithm to DM. The distributed memory model was used in the experimental studies, and 'Document ID' was added for Doc2Vec in addition to the Word2Vec algorithms as shown in Fig. 3.

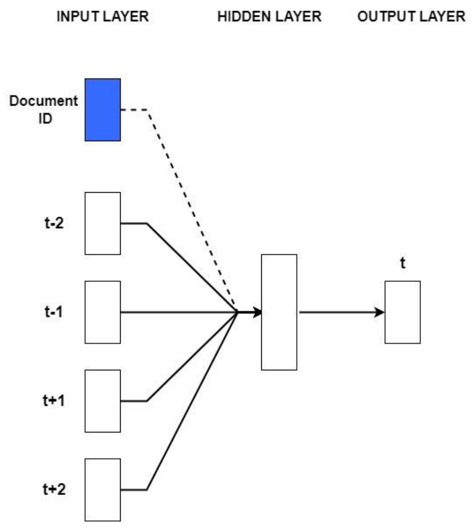


**Figure 3.** Doc2Vec Distributed Memory Model.

Many applications have been developed using Doc2Vec algorithms. Examples of these applications are author profile analysis [19], sentiment analysis and text classification [20].

# 6. CLASSIFICATION

## 6.1. Random Forest (RF)

Each tree in the RF algorithm was created from data that is selected randomly. For this reason, each tree has diversity within itself and decides within itself. After the RF algorithm creates a forest structure by learning from the training data, it tries the test data for each tree structure. Each tree makes its own evaluation for the test data and gives results. As a result, test data is classified according to majority voting. Decision trees have the feature of increasing model performance with their simple structure and flexibility [21].

## 6.2. Support Vector Machine (SVM)

SVM is a learning method developed in the field of statistical learning theory. SVM first transfers the data to a higher dimension where it can be separated linearly. It then provides the maximum boundary between them [21]. In general, SVM problems are divided into two as linear and non-linear. Many problems in daily life cannot be separated linearly. The purpose of using SVM in linear problems is to find a hyperplane passing through features. This hyperplane consists of two straight lines where the features of the classes are the farthest from each other.

## 6.3. Multilayer Perceptron (MLP)

MLPs are feedforward neural network models trained with a standard back propagation algorithm. They need the desired output values to be trained so they can learn how to transform the input data into desired outputs. With a variable number of hidden layers, they can predict almost any input-output map. MLP networks can model functions by determining the complexity of functions, with the number of hidden layers and the number of neurons in each layer. The number of hidden layers to use depends on the problem and the data type used for the models. The best starting point is to use the hidden layer, where the number of units for this layer should equal half the sum of the input and output units. [22].

## 6.4. Deep Neural Network (DNN)

Deep Neural Network is also called as DNN [23]. The neurons of the DNN do not contain convolutional units and they are fully connected. The depth of DNN specifically refers to the number of layers of the neural network. The original neural network has only an input layer, an output layer, and an implicit layer (perceptron) which could not perform complex operations. On the other hand, instead of sigmod using ReLU and other functions solves the of gradient disappearance problem that is the primitive form of the current DNN [24].

# 7. EXPERIMENTS AND RESULTS

Before performing the experiments, the raw data were refined from unnecessary expressions and stop words. Then, to obtain the high score in document classification phase, spell correction and stemming processes are applied in preprocessing phase.

In the experimental study, BoW, TF-IDF, Word2Vec and Doc2Vec word embedding methods and RF, SVM, MLP and DNN classification algorithms were used and the results were compared separately. Precision, Recall, F1-Score and Accuracy were used as evaluation metrics.

We have used max_features=500, min_df=5, max_df=0.7 as parametres in BoW ve TF-IDF methods. In Word2Vec sg=1, window=5, min_count=5, size=200, workers=100, iter=100 and in Doc2Vec dm=1, vector_size=200, epochs=25, window=8, workers=100 parameters were used.

In the classification phase, max_depth=2, random_state=0 were used in RF. In SVM max_iter=15000, kernel='linear', gamma='auto', MLP'de solver='lbfgs', activation='relu', max_iter=50, hidden_layer_sizes=(50,50,50), and in DNN, Dropout=0,5, node = 500, nLayers = 4 (hidden layer), epochs=50, batch_size=128 were used as parameter. It has been determined that these are the most suitable hyperparameters 62ort he best success rate.

The classification study using RF is shown in Table 1. In the BoW and TF-IDF methods, the metric values were far from each other and the result was the lowest. The best result in RF was obtained by using the Doc2Vec method with 89% accuracy.

**Table 1.** Classification with RF algorithm.

| Class No | Metrics | RF-BoW % | RF-TF-IDF % | RF-Word2Vec % | RF-Doc2Vec % |
|---|---|---|---|---|---|
| 1 | Precision | 0.89 | 0.90 | 0.81 | 0.91 |
| | Recall | 0.57 | 0.54 | 0.79 | 0.88 |
| | F1-Score | 0.70 | 0.68 | 0.80 | 0.89 |
| | Accuracy | 0.63 | 0.61 | 0.76 | 0.89 |
| 2 | Precision | 0.79 | 0.46 | 0.74 | 0.94 |
| | Recall | 0.68 | 0.96 | 0.95 | 0.96 |
| | F1-Score | 0.73 | 0.62 | 0.83 | 0.95 |
| | Accuracy | 0.63 | 0.61 | 0.76 | 0.89 |
| 3 | Precision | 0.73 | 0.72 | 0.77 | 0.84 |
| | Recall | 0.48 | 0.51 | 0.64 | 0.85 |
| | F1-Score | 0.58 | 0.60 | 0.70 | 0.85 |
| | Accuracy | 0.63 | 0.61 | 0.76 | 0.89 |
| 4 | Precision | 0.43 | 0.74 | 0.74 | 0.87 |
| | Recall | 0.79 | 0.43 | 0.67 | 0.87 |
| | F1-Score | 0.56 | 0.55 | 0.70 | 0.87 |
| | Accuracy | 0.63 | 0.61 | 0.76 | 0.89 |

The results of SVM are shown in Table 2, and accuracy values were found 74% in BoW, 82% in TF-IDF, and 88% in Word2Vec. The best result was obtained by using the Doc2Vec method with 91% accuracy.

**Table 2.** Classification with SVM algorithm.

| Class No | Metrics | SVM-BoW % | SVM-TF-IDF % | SVM-Word2Vec % | SVM-Doc2Vec % |
|---|---|---|---|---|---|
| 1 | Precision | 0.77 | 0.85 | 0.90 | 0.93 |
| | Recall | 0.80 | 0.82 | 0.87 | 0.89 |
| | F1-Score | 0.79 | 0.83 | 0.89 | 0.91 |
| | Accuracy | 0.74 | 0.82 | 0.88 | 0.91 |
| 2 | Precision | 0.79 | 0.83 | 0.94 | 0.95 |
| | Recall | 0.92 | 0.92 | 0.96 | 0.97 |
| | F1-Score | 0.85 | 0.87 | 0.95 | 0.96 |
| | Accuracy | 0.74 | 0.82 | 0.88 | 0.91 |
| 3 | Precision | 0.74 | 0.80 | 0.85 | 0.87 |
| | Recall | 0.51 | 0.77 | 0.84 | 0.87 |
| | F1-Score | 0.60 | 0.79 | 0.84 | 0.87 |
| | Accuracy | 0.74 | 0.82 | 0.88 | 0.91 |
| 4 | Precision | 0.65 | 0.81 | 0.85 | 0.87 |
| | Recall | 0.73 | 0.78 | 0.86 | 0.89 |
| | F1-Score | 0.69 | 0.79 | 0.85 | 0.88 |
| | Accuracy | 0.74 | 0.82 | 0.88 | 0.91 |

The results of MLP are shown in Table 3, and accuracy values were found 83% in BoW, 81% in TF-IDF, and 88% in Word2Vec. The best result was obtained by using the Doc2Vec method with 90% accuracy.

**Table 3.** Classification with MLP algorithm.

| Class No | Metrics | MLP-BoW % | MLP-TF-IDF % | MLP-Word2Vec % | MLP-Doc2Vec % |
|---|---|---|---|---|---|
| 1 | Precision | 0.86 | 0.80 | 0.91 | 0.93 |
| | Recall | 0.82 | 0.81 | 0.86 | 0.88 |
| | F1-Score | 0.84 | 0.81 | 0.88 | 0.91 |
| | Accuracy | 0.83 | 0.81 | 0.88 | 0.90 |
| 2 | Precision | 0.84 | 0.85 | 0.93 | 0.95 |
| | Recall | 0.92 | 0.87 | 0.96 | 0.98 |
| | F1-Score | 0.88 | 0.86 | 0.95 | 0.96 |
| | Accuracy | 0.83 | 0.81 | 0.88 | 0.90 |
| 3 | Precision | 0.81 | 0.78 | 0.84 | 0.87 |
| | Recall | 0.78 | 0.76 | 0.83 | 0.86 |
| | F1-Score | 0.79 | 0.77 | 0.84 | 0.87 |
| | Accuracy | 0.83 | 0.81 | 0.88 | 0.90 |
| 4 | Precision | 0.80 | 0.80 | 0.83 | 0.86 |
| | Recall | 0.80 | 0.78 | 0.86 | 0.89 |
| | F1-Score | 0.80 | 0.79 | 0.85 | 0.88 |
| | Accuracy | 0.83 | 0.81 | 0.88 | 0.90 |

The results of DNN are shown in Table 4, and accuracy values were found 88% in BoW, 88% in TF-IDF, and 91% in Doc2Vec. The best result was obtained by using the Word2Vec method with 92% accuracy. This is the best achievement in overall experiments. The best result was obtained with 50 epochs in DNN, and it was seen that the success was directly proportional with the increase in the number of epochs.

**Table 4.** Classification with DNN algorithm.

| Class No | Metrics | DNN-BoW % | DNN-TF-IDF % | DNN-Word2Vec % | DNN-Doc2Vec % |
|---|---|---|---|---|---|
| 1 | Precision | 0.90 | 0.90 | 0.95 | 0.92 |
| | Recall | 0.87 | 0.87 | 0.91 | 0.91 |
| | F1-Score | 0.89 | 0.89 | 0.93 | 0.92 |
| | Accuracy | 0.88 | 0.88 | **0.92** | 0.91 |
| 2 | Precision | 0.90 | 0.88 | 0.96 | 0.96 |
| | Recall | 0.92 | 0.94 | 0.99 | 0.97 |
| | F1-Score | 0.91 | 0.91 | 0.97 | 0.97 |
| | Accuracy | 0.88 | 0.88 | **0.92** | 0.91 |
| 3 | Precision | 0.85 | 0.86 | 0.89 | 0.88 |
| | Recall | 0.85 | 0.86 | 0.90 | 0.88 |
| | F1-Score | 0.85 | 0.86 | 0.90 | 0.88 |
| | Accuracy | 0.88 | 0.88 | **0.92** | 0.91 |
| 4 | Precision | 0.85 | 0.87 | 0.91 | 0.89 |
| | Recall | 0.87 | 0.84 | 0.90 | 0.89 |
| | F1-Score | 0.86 | 0.86 | 0.90 | 0.89 |
| | Accuracy | 0.88 | 0.88 | **0.92** | 0.91 |

## 8. ACKNOWLEDGMENTS

We wish to thank TUBITAK (2011-C Priority Areas Graduate Programme Scholarship) for financial support.

## 9. CONCLUSION

This article presents a study that aims to classify a multi-class textual document using word embeddings with machine learning and deep learning methods. In this study, 4 different experiments were performed and the results were compared. BoW, TF-IDF, Word2Vec and Doc2Vec word representation methods were used in all experiments. RF, SVM, MLP and DNN algorithms were used with these word representation methods, respectively, and the best results for 4 different class labels were obtained by using the DNN algorithm with Word2Vec as %92 accuracy score. According to the result, it has been seen that the deep learning classification algorithm is more successful with all word embedding methods than classical machine learning algorithms. We also found that if the number of epochs increases in the DNN, the accuracy also increases. It has been also observed that embedding vector representation methods (Word2Vec and Doc2Vec), which take into account the semantic relations between words, give better results than classical vector representation methods.

As future work, some feature extraction methods will be applied before using word embedding methods. In addition, by using an ontology dictionary and feature extraction techniques together, it is planned to narrow the word space and perform a more successful feature extraction process. In addition, experimental studies will be carried out using transformer models such as BERT and Transfer Learning.

## 10. REFERENCES

**[1]** "World Internet Statistics," [Online]. Available: https://www.internetworldstats.com/stats.htm, [Accessed: May 23, 2019].

**[2]** M. Kaytan, D. Hanbay, "Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines." *Anatolian Science-Journal of Computer Sciences*, vol. 2, no. 1, pp. 15-36, 2017.

**[3]** N. Indurkhya, F.J. Damerau, *Handbook of Natural Language Processing.* Chapman & Hall/CRC, 2010.

**[4]** C. D. Manning, P. Raghavan, H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.

**[5]** E. Alpaydin, *Machine learning : The New AI*. The MIT Press, 2016.

**[6]** J. F. De Paz, J. Bajo, V.F. López and J.M. Corchado, "Biomedic organizations: an intelligent dynamic architecture for KDD", *Information Sciences*, vol. 224, pp. 49–61, March 2013.

**[7]** V. Vapnik, *The Nature of Statistical Learning Theory*, 2000, pp. 1-15.

**[8]** G. Şahin, "Turkish document classification based on Word2Vec and SVM classifier," *25th Signal Processing and Communications Applications Conference (SIU), 2017*, pp. 1-4.

**[9]** T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space". Proceedings of Workshop at ICLR. Scottsdale, Arizona 2-4 May 2013.

**[10]** L. K.Şenel, V. Yücesoy, A. Koç, T. Çukur, "Interpretability Analysis for Turkish Word Embeddings," *26th Signal Processing and Communications Applications Conference (SIU), 2018* pp.1-4.

**[11]** B. K. Loni, H. Seyedeh and P. Wiggers, "Latent semantic analysis for question classification with neural networks," *2011 IEEE workshop on automatic speech recognition and understanding (ASRU), 2011*, pp. 437–442.

**[12]** M. Gogoi, S. K. Sharma "Document Classification of Assamese Text Using Naïve Bayes Approach," *International Journal of Emerging Trends & Technology in Computer Science*, vol. 30, no. 4, pp. 182-186, 2015.

**[13]** W. Arshad, M. Ali, M. Mumtaz Ali, A. Javed and S. Hussain, "Multi-Class Text Classification: Model Comparison and Selection," 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021, pp. 1-5.

**[14]** S. Joshi and E. Abdelfattah, "Multi-Class Text Classification Using Machine Learning Models for Online Drug Reviews," *2021 IEEE World AI IoT Congress (AIIoT)*, 2021, pp. 0262-0267.

**[15]** A. Anand "AG News Classification Dataset," *kaggle.com*, [Online]. Available: https://www.kaggle.com/amananandrai/ag-news-classification-dataset?select=train.csv [Accessed: Sept. 20, 2021].

**[16]** Z. S. Harris, "Distributional structure," Word, vol. 10, no. 2-3, pp. 146-162, 1954.

**[17]** J. Ramos, "Using tf-idf to determine word relevance in document queries," In Proceedings of the first instructional conference on machine learning 2003, vol. 242, pp. 133-142.

**[18]** Q. Le, T. Mikolov, "Distributed Representations of Sentences and Documents," *International Conference on Machine Learning 2014.*

**[19]** L. Chen, G. Feng, C. W. Leong, B. Lehman, M. M. Raugh, H. Kell, C. M. Lee and S. Y. Yoon, "Automated scoring of interview videos using Doc2Vec multimodal feature extraction paradigm," *2016 Proceedings of the 18th ACM International Conference on Multimodal Interaction, ACM.*

**[20]** D.W. Kim and M.W. Koo, "Categorization of Korean News Articles Based on Convolutional Neural Network Using Doc2Vec and Word2Vec," *Journal of KIISE,* vol. 44, no. 7, pp. 742-747, 2017.

**[21]** D. A. Demirci, *Vektör makineleri ile karakter tanıma. MSc* [Dissertation]. Istanbul: Yıldız Technical University, 2007.

**[22]** G. Panchal, A. Ganatra, Y. P. Kosta, and D. Panchal, "Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers," *International Journal of Computer Theory and Engineering,* vol. 3, no. 2, pp.332-337, 2011.

**[23]** G. E. Hinton, R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504-507, 2006.

**[24]** J. Huang, J. Zheng, S. Gao, W. Liu, and J. Lin, "Grid text classification method based on DNN neural network," *In MATEC Web of Conferences 2020,* vol. 309, p. 03016.

## CURRICULUM VITAE

**Ilkay YELMEN** received BS degree in Software Engineering from the Istanbul Aydın University in 2013 and the MS degree in Computer Engineering from the Istanbul Aydın University in 2016. He is currently pursuing the Ph.D. degree in Computer Engineering at the Istanbul Aydın University. His research interests are Machine Learning, Semantic Web and Natural Language Processing.

**Ali GUNES** received the BS degree from Middle East Technical University and Ph.D. degree from Anadolu University. He is currently working at Istanbul Aydın University as Professor. His research interests are Software Development, Biomedical Informatics and Database.

**Metin ZONTUL** received the BS degree in Computer Engineering from Middle East Technical University in 1993, the MS degree in Computer Aided Design, Production and Programming from Erciyes University in 1996 and Ph.D. degree in Numerical Methods from Cumhuriyet University in 2004. He is currently an Associate Professor at Istanbul Ayvansaray University. His research interests are Data Mining, Machine Learning, Information Retrieval and Natural Language Processing.

**Zafer ASLAN** received the BS, MS and Ph.D. degrees from Istanbul Technical University. She is currently working at Istanbul Aydın University as Professor. Her research interests are Atmospheric Sciences, Climatology, Wavelet, Applied Mathematics Renewable Energy.