VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

# Hate Speech Classification with Machine Learning and Ensemble Learning

Hüsnü BARAN[1], Muhammet Sinan BAŞARSLAN[1]*[1]

[1]Istanbul Medeniyet University, Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Istanbul, Türkiye

**Abstract**

It is becoming more and more apparent that social life has reached a breaking point with the unhealthy communication between people due to the technological developments of recent years. People are very tense and have unbearable emotions towards each other. The expression of these emotions has begun to be seen in social media applications. Factors such as pandemics and wars also contribute to the increase of this problem. In this study, after natural language processing techniques on Reddit, Twitter, and 4Chan data, texts were represented with text representations (TF-IDF, BoW, and Word2Vec CBoW and Skip-Gram). These representations were then classified as containing or not containing hate speech using machine learning (Decision Tree, K-Nearest Neighbor, Logistic Regression, Naive Bayes, and Support Vector Machine) and ensemble learning (AdaBoost, Hard Voting, Soft Voting, Stacking, and XGBooost) methods. The models were evaluated using Precision, Recall, F1 score, and Accuracy with 80%-20% training test separation. The best result was obtained with 97.20% Accuracy, 97.61% F1, 95.90% Recall, and 99.39% Precision with the model built using machine learning algorithms along with Stacking after Word2Vec CBoW. This study shows that the Word2Vec method, which is one of the prediction-based methods, gives good results even in unbalanced datasets.

**Makale Bilgisi**

*Başvuru: 22/12/2024*
*Kabul: 07/03/2025*

***Keywords:*** *Ensemble Learning, Hate speech, Machine Learning, Text Representation*

# Makine Öğrenmesi ve Topluluk Öğrenmesi ile Nefret Söylemi Sınıflandırması

**Özet**

Son yıllardaki teknolojik gelişmeler nedeniyle insanlar arasındaki sağlıksız iletişim, sosyal hayatın bir kırılma noktasına ulaştığını giderek daha belirgin hale getirmektedir. İnsanlar oldukça gergin ve birbirlerine karşı katlanılmaz duygular beslemektedir. Bu duyguların ifadesi, sosyal medya uygulamalarında görülmeye başlanmıştır. Pandemi ve savaşlar gibi faktörler de bu sorunun artışına katkıda bulunmaktadır. Bu çalışmada, Reddit, Twitter ve 4Chan verileri üzerinde doğal dil işleme teknikleri uygulandıktan sonra, metinler çeşitli metin temsil yöntemleriyle (TF-IDF, BoW, Word2Vec CBoW ve Skip-Gram) temsilleri çıkarılmıştır. Bu temsiller, nefret söylemi içerip içermediğine göre makine öğrenmesi (Karar Ağaçları, K-En Yakın Komşu, Lojistik Regresyon, Naive Bayes ve Destek Vektör Makineleri) ve topluluk öğrenme (AdaBoost, Hard Voting, Soft Voting, Stacking ve XGBoost) yöntemleri ile sınıflandırılmıştır. Modeller, %80-%20 eğitim-test ayrımıyla Doğruluk, hassasiyet, hatırlama ve F1 skoru kullanılarak değerlendirilmiştir. En iyi sonuç, Word2Vec CBoW temsili sonrası Stacking ile oluşturulan modelde %97.20 doğruluk, %97.61 F1, %95.90 hatırlama ve %99.39 hassasiyet ile elde edilmiştir. Bu çalışma, tahmin temelli yöntemlerden biri olan Word2Vec yönteminin, dengesiz veri setlerinde iyi sonuçlar verdiğini göstermektedir.

***Anahtar Kelimeler:*** *Topluluk Öğrenmesi, Nefret söylemi, Makine Öğrenmesi, Keilme Temsili*

*iletişim e-posta: muhammet.basarslan@medeniyet.edu.tr

## 1. Introduction

The concept of artificial intelligence, first expressed by John McCarthy and his colleagues at a conference at Dartmouth College in 1956, has played an increasingly important role in human life over time [1]. One of the areas touched by artificial intelligence is undoubtedly social media. AI algorithms are used to recommend user-based content by analyzing user behavior. It contributes to the social media experience by personalizing content to users' interests, preferences, and behaviours. This contribution can be determined by looking at the content a social media user has shared, their likes and comments, and their interests and preferences. Apart from the personal sharing of people, especially the false or misleading information/news given by the website, Twitter and other platforms where news is shared, which are responsible for public disclosure, cause major problems. Gathering people around false information or false news causes major problems. For this reason, detecting fake accounts that share this type of news, filtering unwanted content, and ensuring the safety of users also plays an important role in detecting hate speech and identifying situations that lead a certain segment of the public or a community to hatred.

Since the early 2000s, with the widespread use of the Internet, the impact of hate speech on social media platforms has become apparent. The only internationally recognized definition of hate speech is contained in the 1997 Recommendation of the Committee of Ministers of the Council of Europe on Hate Speech. Hate speech is defined as "all forms of expression which spread, incite, promote or justify racial hatred, xenophobia, anti-Semitism or other forms of hatred based on intolerance, including intolerance expressed by aggressive nationalism and ethnocentrism, discrimination, and hostility against minorities, migrants, and people of immigrant origin." This definition serves as a framework for identifying and understanding the scope of hate speech, emphasizing its detrimental impact on individuals and communities[1]. According to an international report, the number of people using social media worldwide has reached 4.76 billion [2]. Such a large population allows ideas to spread more quickly, so posts containing hate speech can reach a wider audience. Artificial intelligence technology is often used to detect such content and prevent it from reaching a wider audience. Among these technologies, text representation and natural language processing (NLP) methods such as Term Frequency-Inver Document Frequency (TF-IDF), Word2Vec play an important role in detecting and filtering content containing hate speech on platforms. For example, in 2020, Twitter introduced artificial intelligence tools that use methods such as TF-IDF and Word2Vec to detect hateful content. Thanks to these tools, the number of tweets containing hate speech dropped from more than 1 billion in Q1 2020 to 100 million in Q1 2023 [3]. These technologies can analyze large datasets to identify meaningful patterns and thus detect hateful content. The main contribution of this study lies in comparing the effectiveness of word-count-based text representation methods (TF-IDF, BoW) with prediction-based methods Continuous Bag of Words-(CBoW) and Skip-Gram for hate speech detection. By pairing these representations with single machine learning (ML) algorithms and advanced ensemble learning (EL) models, the research provides insights into achieving optimal performance, particularly when dealing with unbalanced datasets. This comparison addresses a critical gap in existing studies and underscores the potential of advanced text representations and learning techniques to improve detection Accuracy (Acc) across diverse social media platforms.

The second section of the study is a literature review. In the third section, the methodology, ML, EL, dataset, text representations, and experimental settings are mentioned. The experimental results are presented in the fourth section. In the last section, discussion and conclusion, a general evaluation is made and future goals are given.

## 2. Related works

Among ML methods such as Support Vector Machine (SVM), Decision Tree (DT) after Word2Vec and TF-IDF, they obtained 97.1% Acc results with DT after TF-IDF [4]. They obtained 77% Acc in the study of hate speech detection in Turkish tweets on the Istanbul Convention and 83% Acc in the joint dataset on the Istanbul Convention and migrants [5]. Jiang et al. achieved a test Acc of 0.818% in their classification study using Long Short-Term Memory

[1] Council of Europe, "Recommendation No. R (97) 20 of the Committee of Ministers to Member States on 'Hate Speech'," Committee of Ministers, Strasbourg, 1997. [Online]. Available: https://rm.coe.int/1680505d5b

(LSTM) after Bidirectional Encoder Representations from Transfers (BERT) [6]. They obtained 85.90% Acc in their hate speech study using LR and SVM on BERT [7]. They obtained 91% Acc in their study using n-gram text representation [8]. LR, Random Forest (RF), Naïve Bayes (NB), SVM, and Recurrent Neural Network obtained the highest overall Acc (87.78%) using BERT, while SVM obtained the best (84.66%) among traditional classifiers [9].

Ayo et al. presented a collection of hate speech benchmark datasets suitable for testing the efficiency of classification models. They also present the pros and cons of single and hybrid ML methods for hate speech classification. The paper also presents a generic metadata architecture for hate speech classification on Twitter to overcome the problems associated with Twitter data streams. It was observed that the developed generic metadata architecture outperformed all evaluation metrics for hate speech detection with 0.95, 0.93, 0.92, and 0.93 for Acc, precision, recall, and F1), respectively, compared to similar methods [10].

Abdurrahman et al. obtained 86.05% Acc in LightGBM with 70%-30% training test data to detect hate speech from Twitter [11]Putri et al. used ML algorithms such as NB, Multi-Layer Perceptron (MLP), AdaBoost, DT, and SVM on a dataset of 4,002 tweets related to politics, religion, ethnicity, and race in Indonesia. In their study, they show that the Multinomial Naïve algorithm produces the best model for the classification of hate speech with the highest recall value of 93.2%, which has an Acc value of 71.2% [12].

Pereira-Kohatsu et al. presented the HaterNet model, an intelligent system that detects and monitors the evolution of hate speech on Twitter, used by the National Office for Combating Hate Crimes of the Spanish Secretariat of State Security. Using social network analysis techniques, they created a new public dataset of 6000 expert-tagged tweets on hate speech in Spanish, the first intelligent system to monitor and visualise hate speech in social media. They compared several classification approaches based on different text representation strategies and text classification models. The best results were obtained by a hybrid of LSTM and MLP using TF-IDF-enriched embeddings of emoji and emoticon tokens as input, with an area under the curve (AUC) of 0.828, precision 0.784, recall 0.333, F1 0.467 [13].

MacAvaney et al. proposed a multi-view SVM approach that achieves near state-of-the-art performance while producing simpler and more easily interpretable decisions than neural methods. With this approach, they obtained 0.7469 Acc with SVM on Stormfront, 0.7190 Acc on HatEval, 0.5714 Acc on Trac, and, 0.8297 on the Hatebase Twitter dataset [14].

As seen in the literature, various ML and EL models have been created after TF-IDF, BoW, Word2Vec on hate speech related datasets. In this study, ML (NB, SVM, LR, K-Nearest Neighbor-KNN, DT) and EL (Extreme Gradient Boosting-XGB, Ada, Hard Voting, Soft Voting, Stacking) methods were used to create models with 80%-20% training and testing separation of the hate speech dataset.

## 3. Methodology

This section describes the text representation, dataset, ML and EL, experimental setup, and performance metrics.

### *3.1. Text representation*

Text representation is a topic that plays a critical role in the fields of NLP and natural language understanding, and is essential for achieving successful results in various applications. Representing text data in a meaningful way is crucial for modeling word-level meanings and relationships, for better understanding the content of a text, and for reflecting the rich structure of language in general [15]. Text representation is an important topic in NLP. Effective representation of text data is a basic requirement for many applications. In this study, two important methods for text representation, TF-IDF, BoW, and Word2Vec, are used.

#### *3.1.1 Word2Vec*

Word2Vec is one of the word embedding techniques and is used to learn word vectors. This method is based on the idea that words in a similar context will have similar vectors. The meaning of a word is represented in the vector space by associating it with the surrounding words. Using a learning-based approach, Word2Vec extracts the word embedding matrix from large text datasets and determines word similarities based on this matrix. Word2Vec has two methods, CBoW and Skip-Gram [16]. In this study, both are used with a vector size of 300.

### 3.1.2 TF-IDF

TF-IDF is a widely used method for determining the importance of words in a text. This method, which includes the terms TF and IDF, calculates the importance of a word by comparing its frequency in a document with its frequency in other documents.

In TF-IDF, the higher the frequency of a word in a document and the lower the frequency of that word in other documents, the higher the importance of that word is considered [17].

### 3.1.3 BoW

BoW is a text representation method widely used in text mining and NLP. BoW represents text as a vector by considering the frequency of words in a document. This representation method ignores word order and structure in a document and focuses only on understanding the number of occurrences of each word in the document. BoW has been used in many NLP applications [18].

## 3.2 Dataset

This dataset was open-sourced by Cooke [19] in 2022 and contains 3000 tagged comments and posts scraped from the social media sites Reddit, Twitter and 4Chan. In this dataset, 2400 comments are labelled as non-hateful, while 600 comments are labelled as hateful [19].

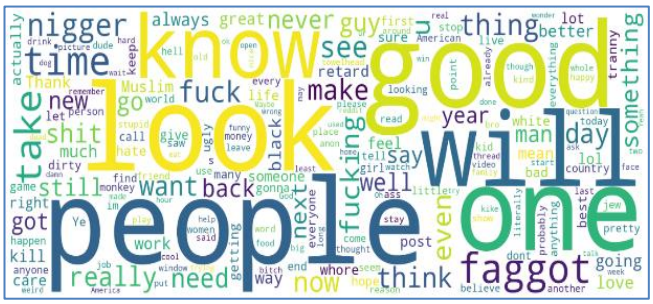


Figure 1. Word cloud of the dataset

Figure 1 presents a word cloud generated from the dataset, highlighting the most frequently occurring words. The visualization includes hateful words associated with offensive language, racism, and sexism (e.g., "n****r," "faggot," "monkey"), which represent the hateful content in the dataset. In contrast, neutral or positive words, such as "great" and "thank you," are also prominent, reflecting the diverse nature of social media language. This diversity poses a significant challenge for hate speech detection models, as they must distinguish between hateful and non-hateful contexts. The presence of overlapping vocabulary (e.g., "good," "people") in both hateful and neutral contexts underscores the importance of using effective text representation methods, such as Word2Vec, which can capture semantic nuances, to improve classification Acc.

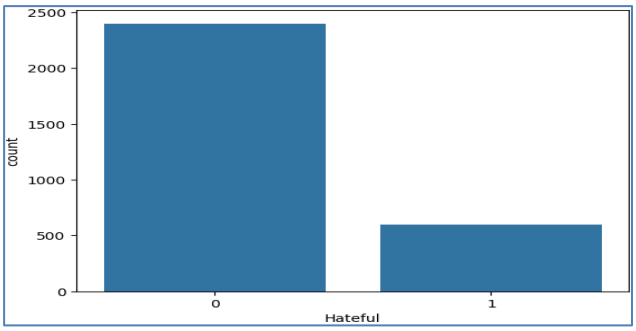


Figure 2. Dataset class distribution

Figure 2 illustrates the distribution of the two classes in the dataset: hateful (class 1) and non-hateful (class 0).

The dataset is highly imbalanced, with a significantly larger proportion of non-hateful examples. This imbalance reflects real-world conditions, where hate speech is relatively less frequent but disproportionately impactful. Such an imbalance presents two critical challenges for hate speech detection systems. First, ML models often exhibit a bias towards the majority class (non-hateful), as it dominates the dataset. This bias can result in suboptimal performance when detecting the minority class (hateful speech), which is the primary focus of hate speech detection efforts. Second, this imbalance highlights the importance of employing robust methods to mitigate bias and improve detection capabilities for the hateful class.

In this study, various text representation techniques, including TF-IDF and Word2Vec, are evaluated alongside advanced ensemble models like Stacking to address these challenges. By focusing on the impact of these methods in an imbalanced dataset, the study aims to identify effective strategies for accurately classifying hate speech without favoring the majority class.

## 3.3 Machine learning

ML is a discipline that gives computer systems the ability to learn from data. This approach allows computers to learn a specific task or problem without human intervention, using complex algorithms and statistical models. Essentially, ML focuses on gaining the ability to predict future events or perform a specific task by extracting patterns and relationships from large datasets.

Using various techniques such as classification, regression, and clustering, computers can perform complex learning processes and extract information from datasets. ML has achieved significant success in a wide range of applications, including medical diagnosis, financial forecasting, NLP, and gaming strategy. This discipline is finding an even wider range of applications with constantly improving algorithms and increasing computing power [20].

### *3.3.1 Naïve bayes*

NB is a probabilistic classification algorithm based on Bayes theorem and can be effectively used in classification tasks. It has four types: Multinomial, Bernoulli, Complementary, Categorical. The choice of these types can vary depending on the characteristics of the dataset and the requirements of the application. It is widely used for classification tasks due to its simplicity, efficiency, and ability to handle large datasets effectively[20].

### *3.3.2 Support vector machine*

SVM is a powerful supervised ML algorithm that can be used for both classification and regression tasks. It is particularly effective in high-dimensional spaces and is well-suited for handling both linear and nonlinear classification problems. The fundamental principle of SVM is to identify the optimal hyperplane that best separates the data points belonging to different classes in the input feature space.

The algorithm achieves this by maximizing the margin, which is the distance between the hyperplane and the closest data points from each class, known as support vectors. SVM is widely used in applications such as image classification, text categorization, and bioinformatics, making it a versatile and reliable algorithm for a variety of tasks [20].

### *3.3.3 K-nearest neighbor*

KNN algorithm is a simple and effective ML classification and regression algorithm. The basic idea is to use the influence of its k nearest neighbors to determine the class or value of a sample. KNN is a lazy learning algorithm that does not create models during the learning phase, but only when prediction is required.

KNN is considered a lazy learning algorithm, meaning that it does not build a model during the training phase. Instead, it memorizes the training data and makes predictions only when needed. This characteristic allows KNN to be straightforward and adaptable but can also make it computationally expensive for large datasets, as the prediction phase involves calculating the distances to all training samples. Despite its simplicity, KNN is highly effective for many practical applications, particularly when the dataset is well-structured and the number of features is not excessively large [16].

### *3.3.4 Logistic regression*

LR is a statistical model used for solving classification problems, despite its name suggesting it is a regression algorithm. It is widely applied, particularly in binary classification tasks, where the goal is to predict one of two possible outcomes. The algorithm works by modeling the relationship between the input features and the probability of a specific class using the logistic (sigmoid) function.

The logistic function maps predicted values to a range between 0 and 1, making it suitable for probabilistic interpretation [16].

### *3.3.5 Decision tree*

DT are supervised learning algorithms used for both classification and regression tasks. They are based on a tree-like structure, where decisions are made by applying a series of rules or constraints to the input data.

This hierarchical structure enables the algorithm to make predictions or classifications by progressively narrowing down the possible outcomes [21].

### *3.3.6 Random forest*

RF is a ML algorithm that combines the predictions of multiple DT to create a more robust, accurate, and generalizable model. Each DT in the RF is trained on a different subset of the dataset, which is typically generated using a technique called bootstrapping (random sampling with replacement).

The algorithm aggregates the predictions from all the DT through techniques like majority voting (for classification tasks) or averaging (for regression tasks), resulting in improved performance and reduced overfitting compared to DT. This ensemble approach makes RF highly effective for a wide range of ML problems. [21].

### *3.4 Ensemble learning*

EL refers to an approach in ML aimed at building a stronger and more generalizable model by combining a set of weak learners. This technique leverages methods such as voting, averaging, boosting, and bagging, which can be applied using

similar or different ML algorithms to improve overall performance. In this study, the EL methods employed include Voting, Stacking. These classifiers are described in this section [22].

### *3.4.1 Voting*

By combining the outputs of multiple learners, the voting method reduces the risk of overfitting to a single weak learner and enhances the overall robustness and Acc of the classification model. Voting is a key technique within the EL approach, widely used for classification problems.

In this method, predictions from multiple weak learners are aggregated to form a consensus, which is then used to make a final decision. This consensus can be achieved through [15]:

**Majority Voting (Hard Voting):** Each weak learner casts a "vote" for a specific class, and the class with the highest number of votes is selected as the final prediction. This approach relies on the collective agreement of the models [23].

**Probabilistic (Soft Voting):** Instead of relying solely on the majority, this method considers the confidence levels (or probabilities) of the predictions made by each weak learner. The final prediction is determined by averaging these probabilities or applying a weighted average [24].

By leveraging the diversity and complementary strengths of the single learners, the voting method ensures more reliable and accurate predictions.

### *3.4.2 Stacking*

Stacking is one of the EL methods. EL is an approach that aims to obtain a more robust and generalizable model by combining a number of different learning algorithms. These algorithms are usually different types of models or models trained with different hyperparameter settings [20].

Stacking takes the process of combining such different learning algorithms further by adding a second level model on top of the predictions of the first level models. The predictions of the first-level models are combined or weighted by the second-level model, thereby improving the overall performance [17], [25].

### *3.4.3 Boosting*

Boosting is an EL technique that enhances the performance of weak learners by combining them in a sequential manner to form a strong learner. Each weak learner is trained to address the mistakes of its predecessor by giving more focus to the instances that were previously misclassified. This is achieved by assigning higher weights to these difficult cases, ensuring that subsequent learners are better equipped to handle them.

The final model combines the predictions of all the weak learners, with more accurate learners contributing more to the overall prediction. Boosting is effective in reducing bias and variance, making it a versatile technique for complex datasets. Algorithms like AdaBoost, XGB, and Gradient Boosting are popular implementations of this approach [11]. XGB was used in this study.

**XGB** is a specialized and optimized version of the boosting framework. It builds on the principles of Gradient Boosting but incorporates several improvements to enhance computational efficiency and predictive Acc. XGB introduces regularization techniques like L1 (Lasso) and L2 (Ridge) penalties to prevent overfitting. It also includes features such as tree pruning using a depth-first approach, handling of missing data, and parallel processing, making it scalable and efficient for large datasets. XGB's ability to handle complex data and its speed have made it a favorite in ML competitions and practical applications, including fraud detection, recommendation systems, and financial modeling.

### *3.5 Experiment settings*

ML methods used in the study (DT, KNN, LR, RF, SVM, NB, XGB, Voting, and Stacking). The Acc, P, R, and, F1 metrics used to evaluate these models are described in this section.

### *3.5.1 Performance metrics*

Objectively evaluating the performance of classification models requires an understanding of key metrics such as Acc, precision (P), recall (R), and F1. Acc is the ratio of correctly classified instances to the total number of instances. Acc is given by equation (1) [26].

$$\text{Acc} = \frac{TN + TP}{TP + TN + FN + FP} \tag{1}$$

P measures the ratio of samples predicted to be positive by the model to true positives. P is given in equation (2) [26].

$$\text{P} = \frac{TN}{TP + FP} \tag{2}$$

R refers to the proportion of all true positive samples that are correctly predicted as positive. R is given by equation (3) [26].

$$R = \frac{TP}{TP + FN} \quad (3)$$

The F1 score represents the harmonic mean of P and R. These metrics are critical for identifying a model's strengths and weaknesses, and for understanding and improving its performance.

The mathematical formulations of each metric provide guidance in determining where the model is more successful or needs improvement. F1 is given in equation (4) [26].

$$F1 = 2 * \frac{P + R}{P * R} \quad (4)$$

## 3. Experimental results

The study was carried out on Google colab using Python language and libraries. As seen in the literature, ML (NB, SVM, LR, KNN, DT, KNN, DT) and EL (XGB, Ada, Hard Voting, Soft Voting, Stacking) methods were used to build models with 80%-20% training and testing separation of the hate speech dataset. The results obtained with ML and EL after BoW are shown in Table 1; ML and EL results obtained after TF-IDF are shown between Table 2 and Table 4.

Table 1. ML and EL performance results after BoW

| | Classifiers | Acc | F1 | P | R |
|---|---|---|---|---|---|
| ML | KNN | 85.00 | 91.60 | 94.60 | 84.50 |
| | SVM | **93.67** | 96.27 | **99.86** | 92.92 |
| | NB | 87.60 | **98.36** | 98.66 | 86.59 |
| | LR | 89.89 | 94.06 | 97.96 | 85.44 |
| | DT | 93.22 | 95.93 | 97.55 | 93.35 |
| EL | XGB | 92.22 | 95.42 | 99.05 | 92.05 |
| | Ada | 91.61 | 95.05 | **98.91** | 91.47 |
| | Hard Voting | 91.00 | 94.78 | 99.89 | 90.15 |
| | Soft Voting | 94.14 | 96.65 | 99.86 | 93.63 |
| | Stacking | **95.56** | **97.32** | 98.64 | **96.03** |

According to Table 1, EL is ahead in all metrics. However, SVM and DT are close to EL in some metrics. In detail about ML and EL;

- In the Acc metric, ML is ranked as SVM, DT, LR, NB, KNN. In the F1 metric, it is ranked as NB, SVM, DT, LR, KNN. In P, SVM ranks first, followed by SVM, NB, LR, DT and KNN. In R, the order is DT, SVM, NB, LR and KNN.
- Similarly, in Acc, F1, and R, EL is sorted as Stacking, Soft Voting, XGB, Ada, and Hard Voting. In P; Hard Voting, Soft Voting, Ada, XGB, and Stacking.

Table 2. ML and EL performance results after TF-IDF

| | Classifiers | Acc | F1 | P | R |
|---|---|---|---|---|---|
| ML | KNN | 85.67 | 91.99 | 96.9 | 85.17 |
| | SVM | **93.40** | **95.91** | **100** | **93.66** |
| | NB | 88.67 | 93.56 | 98.45 | 87.9 |
| | LR | 93.17 | 96.00 | 99.6 | 92.48 |
| | DT | 92.00 | 95.83 | 97.77 | 92.97 |
| EL | XGB | 93.36 | 96.08 | 97.03 | 92.93 |
| | Ada | 93.17 | 96.00 | 92.6 | 92.39 |
| | Hard Voting | 91.00 | 94.82 | 93.62 | 93.15 |
| | Soft Voting | 93.98 | 96.58 | **99.97** | **93.58** |
| | Stacking | **95.5** | **97.3** | 98.38 | 93.38 |

According to Table 2, EL is ahead in all metrics. However, SVM and LR are close to EL in some metrics. If we take a closer look at ML and EL;

- ML is sorted as SVM, LR, NB, DT, KNN in Acc, P metric. In F1 metric; SVM, LR, DT, NB, KNN. In R the order is SVM, DT, LR, NB, KNN.
- Similarly, in Acc and F1, EL is sorted as Stacking, Soft Voting, XGB, Ada, and Hard Voting. In P and R, the order is Soft Voting, Stacking, XGB, Hard Voting, Ada.

Table 3. ML and EL performance results after TF-IDF

| | Classifiers | Word2Vec Methods | Acc | F1 | P | R |
|---|---|---|---|---|---|---|
| ML | NB | Skip-Gram | 92.67 | 95.69 | 98.46 | 92.61 |
| | | CBoW | 92.89 | 95.81 | 99.60 | 92.42 |
| | SVM | Skip-Gram | 93.22 | 95.98 | 99.05 | 93.05 |
| | | CBoW | **94.83** | **96.93** | **99.79** | **94.10** |
| | LR | Skip-Gram | 92.56 | 95.64 | 99.73 | 91.86 |
| | | CBoW | 92.83 | 95.82 | 99.80 | **92.15** |
| | KNN | Skip-Gram | 84.67 | 91.59 | 99.19 | 85.07 |
| | | CBoW | 85.00 | 91.40 | 99.59 | 84.45 |
| | DT | Skip-Gram | 91.67 | 92.53 | 92.19 | 92.05 |
| | | CBoW | 93.19 | 92.56 | 98.05 | 92.18 |
| EL | XGB | Skip-Gram | 92.83 | 95.80 | 99.39 | 92.47 |
| | | CBoW | 93.33 | 95.86 | 99.46 | 92.52 |
| | Ada | Skip-Gram | 92.56 | 95.61 | 99.18 | 92.29 |
| | | CBoW | 92.97 | 96.84 | 99.59 | 94.24 |
| | Hard voting | Skip-Gram | 93.22 | 96.02 | 99.24 | 92.35 |
| | | CBoW | 94.00 | 96.10 | 99.80 | 92.67 |
| | Soft Voting | Skip-Gram | 93.33 | 96.08 | 99.19 | 92.46 |
| | | CBoW | 94.57 | 96.48 | 99.29 | 94.61 |
| | Stacking | Skip-Gram | 95.22 | 97.14 | 99.18 | 95.90 |
| | | CBoW | **97.20** | **97.61** | **99.89** | **99.18** |

According to Table 3, EL is ahead in all metrics. However, SVM and LR are close to EL in some metrics. CBoW is ahead of Skip-Gram in all metrics. If we take a closer look at ML and EL;

For ML;

- For Acc, the ranking after CBoW is SVM, DT, NB, LR, KNN, while for Skip-Gram it is SVM, NB, LR, DT, KNN.
- For F1, the order after CBoW is SVM, LR, NB, DT, KNN, while in Skip-Gram it is SVM, NB, LR, DT, KNN
- For P, the order after CBoW is SVM, LR, NB, KNN, DT while in Skip-Gram it is LR, KNN, SVM, NB, DT
- For R, the order after CBoW is SVM, DT, NB, LR, LR, KNN, while in Skip-Gram it is SVM, LR, NB, DT, KNN

In EL;

- For Acc; the order after CBoW is Stacking, Soft Voting, Hard Voting, XGB, Ada, whereas in Skip-Gram it is Sacking, Ada, Soft Voting, Hard Voting, XGB.
- For F1; the order after CBoW is Stacking, Hard Voting, Ada, Soft Voting, Soft Voting, XGB, whereas in Skip-Gram it is Sacking, Ada, Soft Voting, Hard Voting, XGB.
- In CBoW for P it is Stacking, Hard Voting, Ada, XGB, Soft Voting, Hard Voting, XGB, Ada while in Skip-Gram it is Stacking, Ada, Soft Voting, Hard Voting, XGB.
- In R, CBoW is Stacking, Soft Voting, Ada, Hard Voting, XGB, whereas in Skip-Gram it is Stacking, XGB, Soft Voting, Hard Voting, Ada.

## 4. Conclusion and discussion

According to the experimental results between Table 1 and Table 3, EL outperforms ML in all metrics and in all text representations (BoW, TF-IDF, and Word2Vec). In the case of Word2Vec, CBoW is ahead of Skip-Gram in all metrics. The reason why CBoW is ahead is due to the use of similar words in hate speech and the size of the dataset. This confirms the literature that CBoW performs well on small datasets and languages with frequent word usage.

In ML, SVM performed best on text representations (BoW, TF-IDF, and Word2Vec), while KNN performed worst. In EL, Stacking and Voting were slightly ahead of the others. Table 4 shows the comparison between the Acc results of similar studies in the literature and the best method in this study.

Table 4. Previous studies on dataset

| References | Model | Acc (%) |
|---|---|---|
| [4] | DT | 97.1 |
| [5] | BERT | 77.00 |
| [6] | LSTM | 81.8 |
| [7] | SVM | 85.90 |
| [8] | BERT | 91.00 |
| [9] | SVM | 84.66 |
| Presented Model | CBoW + Stacking | **97.20** |

This study demonstrates that utilizing EL methods in combination with state-of-the-art embedding techniques, such as Word2Vec (CBoW), leads to highly competitive results compared to previous research. As shown in Table 4, our stacking EL model with CBoW achieved 97.20% accuracy, outperforming most prior models. Traditional ML approaches, such as SVM, achieved 85.90% ([7]) and 84.66% ([9]), while deep learning models like LSTM reached 81.8% ([6]). Even BERT-based models, which are known for their strong contextual understanding, reported 77.00% ([5]) and 91.00% ([8]) accuracy, demonstrating that our approach provides a significant improvement. Additionally, while DT models achieved 97.1% accuracy ([4]), such models tend to suffer from overfitting and lack the generalizability of ensemble-based methods. The superior performance of our model can be attributed to the effective combination of advanced word representations (CBoW) and stacking ensemble learning, which enhances classification accuracy and robustness. These findings highlight that integrating embedding techniques with EL is a highly effective strategy, providing a strong alternative to both traditional ML and deep learning-based models.

The results highlight the potential of EL models to achieve superior performance by leveraging the strengths of multiple classifiers and high-quality embeddings. This approach sets the stage for future advancements, particularly through the exploration of transformer-based models such as BERT and its variants, which hold promise for further improving model accuracy and generalizability in complex tasks.

### Funding

The author (s) has no received any financial support for the research, authorship or publication of this study.

Authors' Contribution

The authors contributed equally to the study.

The Declaration of Conflict of Interest

No conflict of interest or common interest has been declared by the authors.

### References

[1] McCarthy J, Minsky ML, Rochester N, Shannon CE. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence". 2006.

[2] We are social. "Digital 2023, Global Overview Report". [Online]. Available: https://wearesocial.com/wp-content/uploads/2023/03/Digital-2023-Global-Overview-Report.pdf

[3] Vardal ZB. "Nefret Söylemi ve Yeni Medya". Maltepe Üniversitesi İletişim Fakültesi Dergisi, 2(1), 132–156, 2016.

[4] Saifullah S, Dreżewski R, Dwiyanto FA, Aribowo AS, Fauziah Y, Cahyana NH. "Automated Text Annotation Using a Semi-Supervised Approach with Meta Vectorizer and Machine Learning Algorithms for Hate Speech Detection". Applied Sciences, 14(3), 1078, Jan. 2024. doi: 10.3390/app14031078.

[5] Beyhan F, et al. “A Turkish Hate Speech Dataset and Detection System”. 2022. [Online]. Available: https://github.com/verimsu/

[6] Jiang Y, Dale R, Lu H. “Transformability, generalizability, but limited diffusibility: Comparing global vs. task-specific language representations in deep neural networks”. Cogn Syst Res, 83, 101184, Jan. 2024. doi: 10.1016/j.cogsys.2023.101184.

[7] Althobaiti MJ. “BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis”. International Journal of Advanced Computer Science and Applications, 13(5), 2022. doi: 10.14569/IJACSA.2022.01305109.

[8] Abdul Aziz NA, Maarof MA, Zainal A. “Hate Speech and Offensive Language Detection: A New Feature Set with Filter-Embedded Combining Feature Selection”. 3rd International Cyber Resilience Conference (CRC), IEEE, Jan. 2021, 1–6. doi: 10.1109/CRC50527.2021.9392486.

[9] Mercan V, Jamil A, Hameed AA, Magsi IA, Bazai S, Shah SA. “Hate Speech and Offensive Language Detection from Social Media”. International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), IEEE, Oct. 2021, 1–5. doi: 10.1109/ICECube53880.2021.9628255.

[10] Ayo FE, Folorunso O, Ibharalu FT, Osinuga IA. “Machine learning techniques for hate speech classification of Twitter data: State-of-the-art, future challenges and research directions”. Comput Sci Rev, 38, 100311, Nov. 2020. doi: 10.1016/j.cosrev.2020.100311.

[11] Abdurrahman MH, Irawan B, Setianingsih C. “A Review of Light Gradient Boosting Machine Method for Hate Speech Classification on Twitter”. 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), IEEE, Nov. 2020, 1–6. doi: 10.1109/ICECIE50279.2020.9309565.

[12] Putri TTA, Sriadhi S, Sari RD, Rahmadani R, Hutahaean HD. “A comparison of classification algorithms for hate speech detection”. IOP Conf Ser Mater Sci Eng, 830(3), 032006, Apr. 2020. doi: 10.1088/1757-899X/830/3/032006.

[13] Pereira-Kohatsu JC, Quijano-Sánchez L, Liberatore F, Camacho-Collados M. “Detecting and Monitoring Hate Speech in Twitter”. Sensors, 19(21), 4654, Oct. 2019. doi: 10.3390/s19214654.

[14] MacAvaney S, Yao H-R, Yang E, Russell K, Goharian N, Frieder O. “Hate speech detection: Challenges and solutions”. PLoS One, 14(8), e0221152, Aug. 2019. doi: 10.1371/journal.pone.0221152.

[15] Başarslan MS, Kayaalp F. “Sentiment analysis using a deep ensemble learning model”. Multimed Tools Appl, 83(14), 42207–42231, Oct. 2023. doi: 10.1007/s11042-023-17278-6.

[16] Mikolov J, Sutskever T, Chen K, Corrado GS, Dean. “Distributed representations of words and phrases and their compositionality”. Advances in Neural Information Processing Systems, 2013. [Online]. Available: https://proceedings.neurips.cc/paper

[17] Bafna P, Pramod D, Vaidya A. “Document clustering: TF-IDF approach”. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, 2016, 61–66. doi: 10.1109/ICEEOT.2016.7754750.

[18] Sreelakshmi K, Premjith B, Chakravarthi BR, Soman KP. “Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages Using Cost-Sensitive Learning Approach”. IEEE Access, 12, 20064–20090, 2024. doi: 10.1109/ACCESS.2024.3358811.

[19] Cooke S. “Labelled Hate Speech Detection Dataset”.

[20] Başa SN, Basarslan MS. “Sentiment Analysis Using Machine Learning Techniques on IMDB Dataset”. 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, Oct. 2023, 1–5. doi: 10.1109/ISMSIT58785.2023.10304923.

[21] Öztürk T, Turgut Z, Akgün G, Köse C. “Machine learning-based intrusion detection for SCADA systems in healthcare”. Network Modeling Analysis in Health Informatics and Bioinformatics, 11(1), 47, Dec. 2022. doi: 10.1007/s13721-022-00390-2.

[22] Polikar . “Ensemble based systems in decision making”. IEEE Circuits and Systems Magazine, 6(3), 21–45, 2006. doi: 10.1109/MCAS.2006.1688199.

[23] Ahmad I, Yousaf M, Yousaf S, Ahmad MO. “Fake News Detection Using Machine Learning Ensemble Methods”. Complexity, 2020, 1–11, Oct. 2020. doi: 10.1155/2020/8885861.

[24] Wang G, Sun J, Ma J, Xu K, Gu J. “Sentiment classification: The contribution of ensemble learning”. Decis Support Syst, 57, 77–93, Jan. 2014. doi: 10.1016/j.dss.2013.08.002.

[25] Mohammadifar A, Gholami H, Golzari S. “Stacking- and voting-based ensemble deep learning models (SEDL and VEDL) and active learning (AL) for mapping land subsidence”. Environmental Science and Pollution Research, 30(10), 26580–26595, Nov. 2022. doi: 10.1007/s11356-022-24065-7.

[26] Khaliki MZ, Başarslan MS. “Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN”. Sci Rep, 14(1), 2664, Feb. 2024. doi: 10.1038/s41598-024-52823-9.