IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 May 2024, accepted 5 June 2024, date of publication 17 June 2024, date of current version 26 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3415350

RESEARCH ARTICLE

So-haTRed: A Novel Hybrid System for Turkish Hate Speech Detection in Social Media With Ensemble Deep Learning Improved by BERT and Clustered-Graph Networks

AYSE BERNA ALTINEL[1], GOZDE KARATAS BAYDOGMUS[1], SEMA SAHIN[1], AND MUSTAFA ZAHID GURBUZ[2]

[1]Department of Computer Engineering, Faculty of Technology, Marmara University, Maltepe, 34854 İstanbul, Turkey
[2]Department of Computer Engineering, Doğuş University, 34775 İstanbul, Turkey

Corresponding author: Ayse Berna Altinel (berna.altinel@marmara.edu.tr)

This work was supported in part by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant 120E187, and in part by Marmara University [Bilimsel Araştırma Projeleri Koordinasyon Ofisi (BAPKO)] under Grant 10784.

ABSTRACT Hate speech on online platforms, characterized by discriminatory language targeting individuals or groups, poses significant harm and necessitates robust detection methods for digital safety. Recognizing the ease with which individuals can engage in such speech online, our study delved into detecting Turkish hate speech using deep learning algorithms and natural language processing techniques. We developed innovative methodologies, including a k-means+textGCN classifier with BERT, which marked the first such attempt in the literature, and explored multiple vector representation techniques such as Term Frequency, Word2Vec, Doc2Vec, and GloVe. Additionally, we investigated various learning algorithms and natural language processing techniques, conducting thorough evaluations on three distinct Turkish hate speech datasets. Notably, our newly presented algorithm exhibited superior performance, achieving an impressive F1-score of 87.81% on the 9K dataset, showcasing advancements in hate speech detection and contributing to a safer online environment.

INDEX TERMS Graph convolutional network, hate speech detection, machine learning, natural language processing, toxic speech, Turkish social media.

I. INTRODUCTION

It is obvious that communication methods are constantly changing from past to present. When we consider the concept of technology today, we can see that the concept of communication has merged with technology and shifted towards the social media side in parallel with advancements in technology. In recent years, there has been an exponential increase in the use of online platforms for communication, leading to the growth of social media networks [1]. There are both positive (constructive) comments and offensive (hate speech) comments made to people on social media. The most striking part of these comments is ''hate speech'', which will be explained in detail in the next section.

A. WHAT IS HATE SPEECH IN SOCIAL NETWORKS

Hate speech refers to any form of communication expressing hatred towards a specific group or individual based on attributes like race, religion, gender, or sexual orientation. Briefly, hate speech can be viewed as a form of digital bullying, facilitated by modern technologies [2].

It is a way of saying situations that people have difficulty in expressing in daily life or that they think they will get overreaction when they express it, by using a secret identity (nickname) when 'hate speech' is examined detailed. Because of such protective and facilitating factors, it is

The associate editor coordinating the review of this manuscript and approving it for publication was Omer Chughtai.

easier to make hate speech on online platforms [3]. The expressions that make up hate speech are to some extent personal and require a background study on the subject for their detection. Negative expressions containing hate speech in social media are constantly produced in different channels, and these contents are quickly distributed over the internet, again through social media, to provide access to a larger number of audiences. Such bullying is observed on social media messaging or gaming platforms, as well as on mobile phones. These types of bullying behaviors are aimed at intimidating, angering, frustrating, or embarrassing the targeted individuals, and they are reflected in a negative way towards the emotions of others [4]. The concept of hate speech, especially targeting children and young people, also affects other people who are exposed to it with different negative feelings. Therefore, daily interactions on social media, due to the high number of posts, are among the most primary factors for the increase in hate speech on social media.

The person who engages in hate speech often uses hate speech to more emphatically express their reaction to a situation or event. This behavior is supported by people who have the same thought but cannot express themselves in this way, which is the majority of the followers. In this case, it results in people who make hate speech gain more followers in a shorter time [3]. Figure 1 shows hate speech on social media and its effects on people. Hate speech on social media can turn into much bigger problems than unhappiness that will affect people's daily lives. According to Figure 1, emotional people are particularly affected by hate speech they see on social media. In addition, children or adolescents who have not completed their personal development at a sufficient level are influenced by the discourses on social media and shape their opinions without questioning what is right or wrong.

FIGURE 1. Illustration of hate speech on social media.

B. WHAT ARE THE CHALLENGES FOR HATE SPEECH DETECTION IN SOCIAL NETWORKS

Due to the vast volume of content generated on social media, manual detection of such materials is not practically feasible. Therefore, there is a need for systems that can automatically filter such content using artificial intelligence for effective identification [5]. This way, reliance on human effort is reduced in the detection of disturbing content.

Detecting hate speech in social networks poses a multi-faceted challenge owing to contextual ambiguity, the intricate nature of multimodal content, language evolution, and cultural diversity. The presence of sarcasm, irony, biases in training data, and the imperative to ensure user privacy during detection further complicates the task. Overcoming these challenges demands not just advanced machine learning (ML) techniques but also a commitment to ethical considerations and interdisciplinary collaboration [6], [7].

Despite the efforts of social media personnel to counter hate speech content through specific policies, the sheer volume of such content is making prevention increasingly challenging. Consequently, social media employees are compelled to formulate their own policies for the prevention of hate speech [8], [9].

We can address the issue of hate speech from two main perspectives: technical perspective and social perspective:

1) In technical perspective; one of the main challenges is accurately defining and identifying hate speech since between hate speech and free speech can be subjective and context-dependent, making it hard to create an objective set of criteria for recognition. Moreover, hate speech can take several forms, including subtle and coded language, making it challenging for automated systems to precisely distinguish and classify hate speech [10].
2) In a social perspective; one of the biggest challenges is addressing the diversity of languages, dialects and cultural nuances present. Different languages have different expressions and slang that can be used as hate speech. Another social challenge is the ever-evolving nature of hate speech. As social norms and language usage may change over time, hate speech evolves. There are also ethical concerns such as misrecognition of hate speech as well as suppression of freedom of expression. In short, hate speech recognition depends significantly on both the intention of the user and the context in which the speech is used [11].

Detecting hate speech on social media is a subject that has been studied by many researchers for numerous years. In this context, Figure 2 displays the distribution of SCI-E articles published in English on Web of Science (WoS) until January 25, 2024, over the past five years. To establish this distribution, a search was conducted on WoS using the keyword 'hate speech.' Since this study is not a systematic mapping study, a more detailed search was not conducted, but comprehensive analyses of this subject in the literature were also reviewed [12], [13], [14], [15].

It is known that the amount of hate speech on social media has increased, especially due to wars and immigrant movements in recent years. Since it is such a common and important problem, studies on English are increasing day by day. However, unfortunately this is not valid for Turkish.

There are only a handful of studies and shared data sets to detect the Turkish hate speech problem [8], [9].

By inspiring from this very important necessity, we aimed to develop a novel methodology based on recent technologies in order to detect hate speech in Turkish social media.

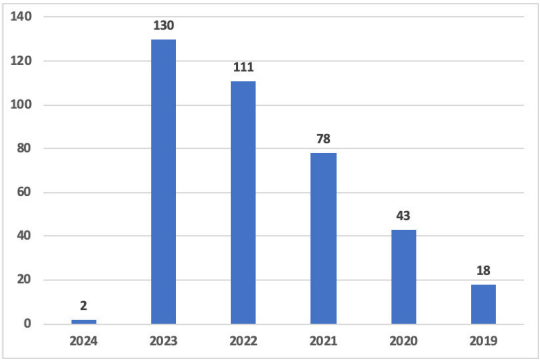

FIGURE 2. Number of papers on hate speech in WoS between 2019 and 2024.

C. WHAT ARE THE LIMITATIONS AND EXISTING GAPS IN TURKISH HATE SPEECH DETECTION IN SOCIAL NETWORKS

When examining recent research on hate speech detection in social media, it becomes apparent that studies in the English-Hindi languages [16], [17] dominate the literature. Upon closer inspection, it is evident that there are fewer studies conducted in languages other than English and Hindi. This can be attributed to the inadequacy of publicly available datasets in different languages and the structural complexities of various language constructs. Looking at studies on hate speech in the Turkish language [18], [19], it is apparent that the structural differences of the language and the limitations of available datasets necessitate a greater number of research endeavors. Structural differences of the Turkish, some words transferred from foreign languages to the Turkish language, special expressions used culturally, local words, differences such as accent, dialect, proverbs and idioms make it very tough to create a hate speech detection model and study in Turkish. It could be beneficial for the researchers summarize briefly the Structural differences between Turkish and other languages also in order to emphasize the contribution of this work:

1) There are differences in morphological analysis such as syntax, sentence structure etc. [20],
2) Turkish has a very complex grammar [20],
3) There are more punctuation marks which provide semantic differences among phrases [20],
4) Turkish has a more wealth of words and diversity in compare to other languages [20],
5) There are idioms and phrases specific to Turkish [21],
6) The suffixes and root structures of words are different from English [21].

The increased prevalence of online hate speech has led to significant concerns about the impact on public safety and social harmony. In some countries, as in Turkey, individuals or institutions exposed to hate speech are protected by law. The detection of hate speech in online platforms is, therefore, a critical task. To tackle this issue, researchers have developed various methods for automatically detecting hate speech in online content using ML. ML is a powerful tool that can be used to identify and classify hate speech accurately by analyzing large volumes of text data. Previous studies on hate speech detection using ML techniques have utilized different ML algorithms, including deep learning, unsupervised and supervised learning, and transfer learning to detect hate speech in online content [5], [6], [7], [8], [9].

The main objectives of this study:

- Focus on detecting hate comments in the Turkish language using three different datasets, with two datasets specifically prepared for this study.
- Conduct a comprehensive review of existing literature to examine various approaches to hate speech detection, including feature engineering techniques, machine learning (ML), and deep learning models.
- Employ several techniques for generating word embeddings, such as TF, Word2Vec (using CBOW and Skip-gram models), Doc2Vec (using PV-DM and PV-DBOW approaches), GloVe, and BERT.
- Utilize ML algorithms like Random Forest (RF), Naive Bayes (NB), Support Vector Machines, textGCN, and deep learning approaches like LSTM and BiLSTM, along with various feature engineering techniques like word and character n-grams and Count Vectorizer for the classification phase.

When the studies conducted in recent years on detecting hate speech on social media are scanned, the studies in English-Hindi languages [16], [17], [22], [23] stand out the most in the literature. There are also studies in different languages such as German and Italian [24], Bangladesh's Bangla [25], Arabic [26], [27], Kazakh [28], Romano-Urdu [29] and Indonesian [30]. This briefly shows us that there are fewer studies in other languages except English-Hindi. The reason for this is thought to be the inadequacy of publicly available data sets for different languages and structural difficulties in different language structures. For example, when we look at the studies on hate speech in Turkish [19], [31], [32], it is seen that there are relatively fewer studies compared to studies in other languages, due to the structural differences of the language and insufficient resources such as data sets, dictionaries, models. This situation is actually like a vicious circle; as the resources are less, fewer studies are being done and as a result, the number of resources does not increase significantly and ultimately researchers who want to work in this field unfortunately prefer to work on languages other than Turkish. Considering that Turkey ranks 7th in the world in terms of X platform usage,[1] this situation is not fair at all. Therefore, the main purpose of our study is to attempt to fill in the gaps, such as the lack of resources and models regarding

[1] https://twitter.com/DrDataStats

the Turkish hate speech problem, and to report the analysis results of our studies. To be more detailed, this study offers the following important contributions that are not available in the literature on the problem of Turkish hate speech:

- Presentation of two new Turkish hate speech detection datasets, which are valuable due to the scarcity of Turkish datasets on this topic. These datasets are shared as open source on Github for researchers in this field.
- Introduction of techniques such as BERT, GloVe, doc2vec, and textGCN for addressing Turkish hate speech detection, which have not been extensively explored in previous studies.
- Conducting analyses using a combination of ML and deep learning algorithms, as well as ensemble techniques, on Turkish hate speech datasets to explore and share the results obtained.
- Comparison of experimental results with those of an existing study, demonstrating the importance and effectiveness of the methodology used in this study for achieving superior performance in hate speech detection compared to previous research.

The rest of the article is organized as: Section II provides an overview of previous studies and available datasets on hate speech detection. Background technologies that are used in this study are presented in Section III. The methodology employed in this study is detailed in Section IV. Section V presents the experimental study and provides a detailed discussion of the results. Finally, conclusions of the research are presented in Section VI.

II. LITERATURE REVIEW

Almaliki et al. [33] identified hate speech on social media by introducing the Arabic BERT-Mini Model (ABMM). A total of 9352 Arabic tweets were gathered from X and classified into three distinct categories, namely normal, abusive, and containing hate speech. Word representations were created by using embedding techniques, and then BERT model is used to predict Arabic hate speech. They also used some ML models and deep learning models to compare their performances with the proposed ABMM model. Compared with these models, the proposed ABMM model performed better and achieved an accuracy rate of 98.60%.

Arshad et al. [34] proposed UHated which is a BERT-based methodology that utilizes transfer learning with RoBERTa to hate speech detection in Urdu language. They also create Urdu Hate Speech and Offensive Language Detection dataset which contains 7871 tweets. They used ML models with various feature engineering techniques like TF-IDF, n-grams, and Count Vectorizer. They used deep learning, traditional ML, and ensemble models then compared their performances with the proposed model. The results showed that RoBERTa outperforms other models and achieve a macro F1-score equal to 82.00%.

Aziz et al. [35] studied the automated identification of political hate speech in Roman Urdu. They prepared a newly created labeled dataset called RU-PHS which has 5002 instances and also included city-level information. Researchers developed three vectorization techniques, namely word2vec, TF-IDF, and fastText. The results demonstrate that a RF and the proposed feed-forward neural network using fastText word embeddings achieve an accuracy of 93.00% in distinguishing between politically offensive speech and neutral. They also used the spatial information in ArcMap to map the dataset. The statistical data analysis aided in identifying patterns and trends, and the cluster and hotspot analysis was useful in identifying the areas in Pakistan that are highly susceptible to hate speech. The results revealed that cities in Punjab were the most impacted and served as key locations for the generation of hate and sarcastic tweets.

Fernardo et al. [36] used ML and deep learning algorithms to detect hate speech in the Sinhala language. They used two different precollected datasets, one containing 1742 and the other 6345 observations. They used Bag of words, Word2Vec, TF-IDF, and FastText. Test dataset contains 400 observations that were labeled as hate and neutral. For the test set, FastText with Recurrent Neural Network (RNN) combination has the highest AUC ROC score of 0.71, accompanied by a 70.00% accuracy rate. They also conducted experiments for both word and character unigram, bigram, and trigram and stated that character n-gram features are effective in hate speech recognition. They also stated that to achieve optimal performance measurements, it is essential to fine-tune the hyperparameters in both ML and deep learning models.

Karayiğit et al. [37] used a pre-trained Multilingual Bidirectional Encoder Representations from Transformers (M-BERT) model to detect hate comments in the Turkish language, focusing on identifying comments related to homophobia as well as expressions involving sexism, defecation, and severe insults. They created Homophobic-Abusive Turkish Comments (HATC) dataset by gathering comments. To ensure the dataset's accuracy, they manually labeled the dateset at the sentence level, and then merged with their pre-developed Abusive Turkish Comments (ATC) dataset from their previous study. The HATC dataset contains a total of 19,827 neutral, 1,226 homophobic, and 10,237 hateful Instagram comments. They also used some ML models, deep learning models and ensemble classifiers to compare their performances with the proposed M-BERT model. As a result, M-BERT model outperformed the other models in classifying all categories.

Khanday et al. [38] aimed to leverage ML and ensemble learning (EL) techniques to detect hate speech during COVID-19. Using X's API and trending hashtags related to the pandemic, they extracted data and manually annotated tweets into two categories based on various factors. They extracted features by using TF-IDF, Tweet Length and Bag of Words. Results showed that the Decision Tree (DT) classifier was effective, with 97.00% recall, 98.00% precision, 97.00% accuracy, and 97.00% F1-Score. Stochastic Gradient Boosting classifier performed even better, outperforming all other ML classifiers with 97.00% recall, 99.00% precision, 98.04% accuracy, and 98.00% F1-Score.

Mayda et al. [31] conducted an automatic hate speech detection study on the Turkish language. Firstly, 1000 Turkish tweets were collected, and then they were labeled into three different classes (hate speech, aggressive expression, neither) using two evaluators. In cases where the annotators were indecisive, a third annotator's opinion was taken to produce a reliable result. After preprocessing the dataset, the final version was shared for use in other studies. Experiments were conducted using various ML algorithms such as NB, DT, Sequential Minimal Optimization (SMO), and RF, and the success rates of the results were compared according to the F-measure. Tests were performed using the WEKA tool. As a result, SMO algorithm provided the highest performance with a 79.90% F-measure value.

Baydogan and Alatas [39] aimed to detect hate speech content shared on online social networks quickly and effectively using artificial intelligence-based algorithms, including ML methods and artificial neural networks. A dataset consisting of 40,623 synthetic texts was classified into two labels, hate and nothate. Features representing the dataset were extracted using BoW, TF, and t-DM techniques. Experiments were carried out using ML algorithms such as Multinomial NB, Concept Adapting Very Fast DT (CVFDT), Lib-Support Vector Machine (Lib-SVM), DT-Part, and Multi-Layer Perceptron (MLP). CVFDT, DT-Part, and MLP algorithms had the highest accuracy rate of 80.00%, while MLP neural network had the highest F-measure value of 78.00%, CVFDT algorithm had the highest precision value of 81.00%, and CVFDT, DT-Part, and MLP algorithms had the highest recall value of 80.00%. The Lib-SVM algorithm had the lowest error rate.

Nergiz and Avaroglu [40] aimed to detect cyberbullying on social networks that contain Turkish comments using LSTM. The dataset was created by collecting 90,000 Turkish comments containing cyberbullying and 90,000 comments without cyberbullying from X, Instagram, and Youtube platforms. The Zemberek library was used for preprocessing the data. Three different models were created for training the LSTM using n-gram for Fasttext, Skip-Gram for Word2Vec, and PV-DBOW for Doc2Vec to convert words into numerical expressions. These models were named Model 1, Model 2, and Model 3 for Word2Vec, Fasttext, and Doc2Vec, respectively. In the LSTM classification model, the hidden layer employed the Sigmoid activation function, the dense layer employed the relu activation function, and the output layer employed the softmax activation function. In the experimental results, it was observed that Model 2 had the highest accuracy rate of 93.15%.

Samarasinghe et al. [41] proposed a solution for detecting hate speech in a given text corpus written in Sinhala Unicode using deep learning mechanisms. Their approach incorporated two convolutional neural networks (CNN) to classify a given text corpus as either containing hate speech or not. Furthermore, if the text corpus is identified as containing hate content, it will then be classified according to its level of severity, which can be used by authorities to make decisions. They used FastText word embedding. The results demonstrate an accuracy of 83.00% for classifying hate speech and 60.00% for classifying hate level.

Table 1 summarizes recent studies about hate speech on social media platforms. The first column in the Table 1 gives the reference number of reviewed articles.

III. EXISTING TECHNIQUES AND ALGORITHMS FOR HATE SPEECH DETECTION IN SOCIAL NETWORKS

In this section, information about the algorithms used in the study and the proposed method are given.

A. TEXT REPRESENTATION TECHNIQUES

In this section, information about text representation techniques are given. Table 2 contains a comparison of text representation techniques.

1) BIDIRECTIONAL LONG SHORT TERM MEMORY

Bidirectional Long Short Term Memory (BiLSTM) is a variant of the RNN architecture that leverages bidirectional processing using parallel LSTM networks to capture both past and future context in sequential data [42], [43]. By incorporating information from past and future context, BiLSTM enhances the model's understanding of temporal relationships, enabling it to make more accurate predictions or classifications. This unique bidirectional processing empowers BiLSTM to effectively model long-range dependencies and effectively handle sequential data across various domains, making it a powerful tool in tasks such as Natural Language Processing (NLP), speech recognition, and time series analysis. Figure 3 illustrates the archirecture of the BiLSTM.

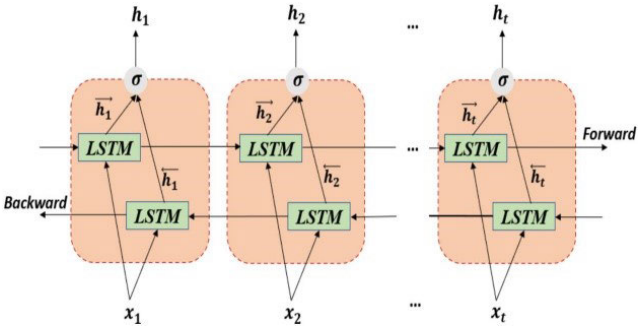

FIGURE 3. Architecture of the BiLSTM [44].

2) TERM FREQUENCY

Term Frequency (TF) is a quantitative metric used in NLP to measure the frequency of occurrence of a term within a document. It provides insights into the relative importance and significance of terms within the context of textual data.

TABLE 1. Previous studies on detecting hate speech.

Paper	Lang.	Year	Dataset	Category	Method	Perf.
[33]	Arabic	2023	X (9352 tweets)	Normal, abusive, hate speech	ABMM, some ML models and deep learning models	98.60% accuracy
[34]	Urdu	2023	X (7871 tweets)	Offensive and hate speech	UHated, some deep learning, traditional ML and ensemble models	82.00% F1-score
[35]	Roman Urdu	2023	X (5002 tweets)	Political hate speech	RF and the proposed feed-forward neural network using fastText word embeddings, some ML models and neural networks	93.00% accuracy
[36]	Sinhala	2022	Facebook, X, and Youtube	Hate speech, neutral speech	LR, SVM, Multinomial NB, RF, XGBoost, CNN, RNN, and LSTM	70.00% accuracy and 71.00% AUC ROC
[37]	Turkish	2022	Instagram	Homophobic, sexist, severe humiliation, and defecation expressions	M-BERT, some ML models, deep learning models and ensemble classifiers	homophobic F1-score: 82.64%, hateful F1-score: 91.75%, neutral F1-score: 96.08%, average F1-score: 90.15%
[38]	English	2022	X	Normal and hate speech	LR, SVM, Multinomial NB, DTs, Bagging, Adaboost, RF and Gradient Stochastic Boosting EL techniques	97.00% recall, 99.00% precision, 98.04% accuracy, and 98.00% F1-Score
[31]	Turkish	2021	X (1000 tweets)	Hate speech, aggressive expression, neither	ML algorithms such as NB, DT, SMO, and RF	79.90% F-measure value
[39]	English	2021	synthetic training dataset from kaggle	Hate and nothate	ML algorithms such as Multinomial NB, CVFDT, Lib-SVM, DT-Part, and MLP	78.00% F-measure value
[40]	Turkish	2021	X, Instagram, and Youtube	Cyberbullying and without cyberbullying	Three different models were created for training the LSTM using n-gram for Fasttext, Skip-Gram for Word2Vec, and PV-DBOW for Doc2Vec	93.15% accuracy
[41]	Sinhala	2020	Gossip sites	Hate and nothate	A deep learning mechanism that utilizes two CNNs	An accuracy of 83.00% for classifying hate speech and 60.00% for classifying hate level.

TABLE 2. Comparison of text representation techniques.

Technique	Description	Pros	Cons
BiLSTM	A type of RNN that processes sequences bidirectionally	Captures long-term dependencies, Effective for sequential data	Computationally expensive, Requires large amounts of data
Term Frequency	A simple method that measures the frequency of a term in a document or corpus.	Easy to implement, Provides basic word importance information	Ignores context, Does not consider word relationships
Word2Vec	A neural network-based technique that learns distributed representations of words based on their contexts in a corpus.	Captures semantic relationships between words, Space-efficient	Requires large corpus for training, Context window limitation
Doc2Vec	An extension of Word2Vec that learns fixed-length vector representations for entire documents.	Captures document semantics, Allows similarity comparisons	Training can be slow for large documents, Requires labeled data
GloVe	Global Vectors for Word Representation, a technique that learns word embeddings by factorizing word co-occurrence matrices.	Captures global word relationships, Space-efficient	Less effective for capturing word context compared to Word2Vec
BERT	It is an NLP model trained through language modeling and next sentence prediction tasks.	Pretraining, Open source implementation	Computational resources, Fine tunning overfitting Understanding limitations

3) WORD2VEC

Word2Vec is a neural network-based model in NLP that is used for generating distributed representations of words in a continuous vector space. It provides a powerful and effective way to capture semantic relationships and similarities between words based on their contextual usage within a large corpus of text.

There are two main architectures of Word2Vec: CBOW [46] and Skip-gram [47]. In CBOW, the model predicts the target word based on the surrounding context words. In contrast, Skip-gram predicts the context words given a target word. Figure 4 illustrates the two network archirecture of the Word2Vec model.

4) DOC2VEC

Doc2Vec is a neural network based model in NLP that enables the generation of dense vector representations for entire documents [48]. It extends the concept of word embeddings,

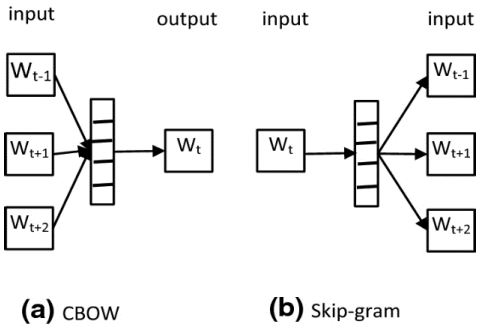

FIGURE 4. Network architecture of the Word2Vec Model: CBOW and Skip-gram [45].

as seen in models like Word2Vec, to capture the semantic meaning and contextual information of documents.

There are two primary architectures used in Doc2Vec: PV-DM and PV-DBOW. In the PV-DM architecture, the model predicts words within a document using both the surrounding words and the document label as context. In contrast, the PV-DBOW architecture predicts words based solely on the document label, disregarding the context of neighboring words. Figure 5 illustrates the two network archirecture of the Doc2Vec model.

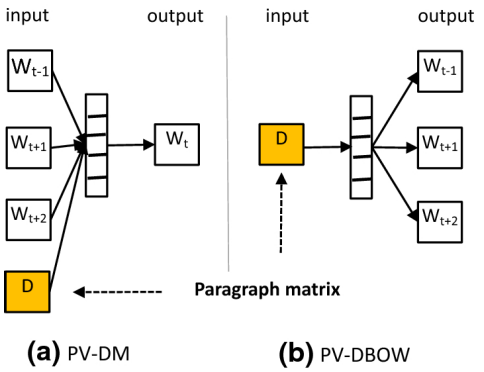

FIGURE 5. Network architecture of the Doc2Vec Model: PV-DM and PV-DBOW [45].

5) GLOVE

GloVe is a word embedding model in NLP that combines global word co-occurrence statistics with vector space representations [49]. By leveraging large-scale text corpora, GloVe generates dense vector representations for words that capture both semantic and syntactic information. GloVe stands out for its ability to produce high quality embeddings that encode both semantic and syntactic information, making it a powerful tool for various NLP tasks, such as word analogy, document classification, and sentiment analysis.

6) BERT

BERT is a neural language model that has transformed NLP [50], [51], [52]. By incorporating bidirectional context and the Transformer architecture, BERT generates contextually informed word representations. By training on vast amounts of unlabelled text, BERT learns to understand language semantics and has demonstrated remarkable performance across diverse NLP tasks.

B. CLASSIFICATION ALGORITHMS

In this section, information about classification algorithms are given.

1) DECISION TREE

Within its algorithm, a dataset is partitioned into subsets, which are further divided into smaller subsets through the application of specific decisions. Decision Trees (DT) are constructed using the "ID3" algorithm [53]. The ID3 algorithm employs the Entropy method in the creation of DTs. Entropy is utilized to construct frequency tables. As the training process commences, a reduction in entropy leads to the acquisition of information gain [53].

2) RANDOM FOREST

Random Forest (RF) is an ensemble learning technique widely employed in ML that combines the predictive capabilities of multiple DTs to enhance classification and regression tasks [54], [55]. Composed of an ensemble of DTs, RF operates by constructing multiple trees with random subsets of features and training data.

3) SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a powerful and widely utilized supervised learning algorithm in ML, designed for both classification and regression tasks [56], [57]. It aims to find an optimal hyperplane that maximally separates the instances of different classes while minimizing the classification error. The algorithm is implemented using a kernel.

4) NAIVE BAYES

Naive Bayes (NB) is a probabilistic ML algorithm that utilizes the principles of Bayesian statistics to perform classification tasks [58], [59]. Given a set of observed features, the algorithm aims to determine the most likely classes for a given instance by calculating the probabilities. These probabilities are computed by combining the prior probabilities of the classes with the likelihood probabilities of the features given each class.

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)} \tag{1}$$

where $P(C|X)$ represents the posterior probability of the class given the predictor, $P(X|C)$ represents the likelihood probability of the predictor given the class, $P(C)$ represents the prior probability of the class, and $P(X)$ represents the prior probability of the predictor (Equation 1).

5) LONG SHORT-TERM MEMORY

Long Short-Term Memory (LSTM) is a type of RNN architecture that addresses the limitations of traditional RNNs in capturing long-term dependencies and mitigating the disappearing gradient problem [2], [60]. It utilizes memory cells and gating mechanisms to selectively store, forget, and access information. The LSTM architecture is composed of essential components: the cell, forget gate, input gate, and output gate. The cell acts as the memory unit, while the input and output gates determine the input to and output from the cell. Figure 6 illustrates the archirecture of the LSTM.

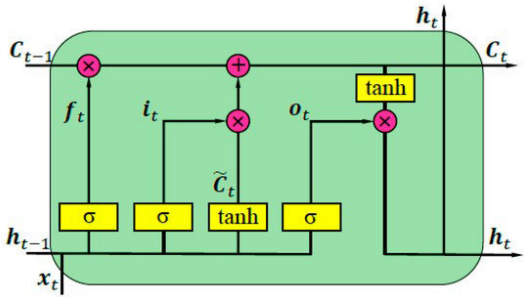

FIGURE 6. Architecture of the LSTM [61].

6) TEXTGCN

TextGCN is an advanced deep learning model designed for text data analysis [62]. It combines graph convolutional networks with NLP techniques to represent text as graphs, where nodes represent words or documents and edges capture semantic connections. By performing graph convolutions, TextGCN effectively integrates contextual information from neighboring nodes to generate enriched word or document embeddings.

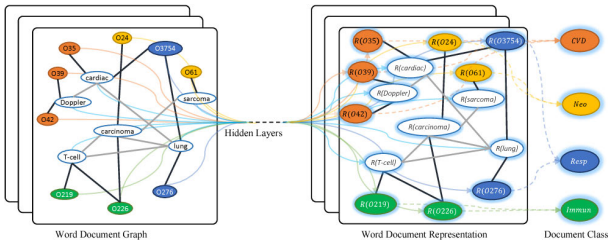

FIGURE 7. Schematic for text GCN [63].

Through the application of graph convolutional operations, TextGCN learns to aggregate and propagate information across the text graph, enabling it to capture intricate textual dependencies and improve the performance of various NLP tasks. Figure 7 illustrates the schematic representation of the Text GCN model.

$$Z = f(X, A) = p_2(A' p_1(A' X W_0) W_1) \tag{2}$$

where, $A' = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ denotes the normalized symmetric adjacency matrix W_i denotes represents weight matrices that are trainable through gradient descent, and p_i signifies the activation functions. Specifically, in the case of a two-layer TextGCN, p_1 is specified as the Rectified Linear Unit (ReLU) function, p_2 is designated as the Softmax function, and the chosen loss function is the cross-entropy error computed over all labeled instances [64].

7) K-NEAREST NEIGHBOR

K-Nearest Neighbor (KNN) algorithm is a ML technique employed in both classification and regression tasks. Its fundamental concept revolves around leveraging the nearest neighbors of a given instance to perform tasks of classification or value prediction [65]. The data is defined within the algorithm using the specified value of K, and it is delineated by considering its nearest neighbors. In the examined study, to avoid situations of parity, it is recommended that the value of K be an odd number.

8) LOGISTIC REGRESSION

When a logistic transformation is applied to the dependent variable, it can be categorized into two distinct types. Binary Logistic Regression (LR) is employed when dealing with scenarios where the dependent variable has two classes [66]. Another type, known as Multinomial LR, is utilized in cases where there are multiple categories within the label group [67]. Subsequently, our training set is educated based on the data points corresponding to these assigned values.

9) CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNN) consists of convolution, pooling and full link layers [68]. In the Convolution Layer, the attribute map of the class we want to train and about which we want to extract features is obtained. Then, the data with the attribute comes to the pooling layer. In the pooling layer, values are selected from the attribute layer according to which feature we want to train the data. It is determined which label the data coming to the full link layer will have. The pooling and convolution layer are used in feature extraction. Softmax and unit activation functions are used a lot in order to give better results in classification to full link layer.

C. ENSEMBLE LEARNING ALGORITHMS

Ensemble Learning (EL) involves combining multiple classification models to create a unified model. EL comprises two main models: Bagging and Boosting. In Bagging models, the dataset is randomly divided, creating multiple training subsets. These subsets are trained based on labels to produce various classification models [69], [70]. The resulting models are evaluated using a Voting Classifier. In Boosting models, unlike Bagging, the dataset is not divided, instead, it's iteratively trained using errors from previous models [69], [70]. Model evaluation is performed using the Voting Classifier, similar to Bagging.

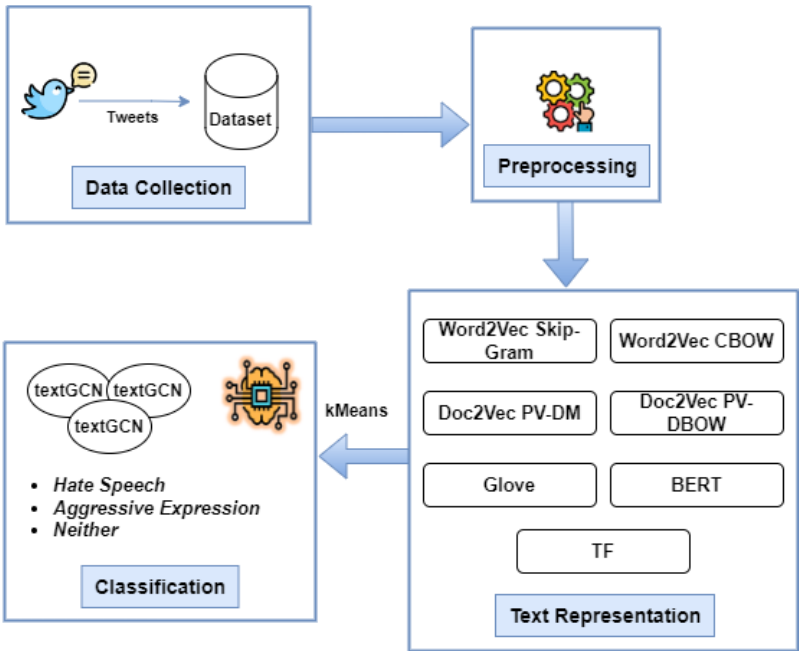

FIGURE 8. The general architecture of the proposed system.

1) EXTREME GRADIENT BOOSTING

Extreme Gradient Boosting (XGB), distinguished from other boosting algorithms, primarily differs in its training process involving the learning rate. This algorithm employs DT structures and initiates with an initial value prediction. Through the gradient boosting technique, it proceeds to the subsequent step, trying to predict with the knowledge it has gained [71]. Gains are calculated from the acquired trees, with the gain metric referred to as gamma. If the gamma value is 0, the gain is negative.

2) GRADIENT BOOSTING

Similar to XGBoost, Gradient Boosting (GB) executes the learning process using tree structures. The creation of trees persists until the predetermined count is reached. During the learning process, the learning rate from the preceding tree is taken into consideration [72].

3) ADAPTIVE BOOSTING

Adaptive Boosting (AB), often regarded as the pioneering boosting algorithm. AB employs tree structures to facilitate the learning process, and during the decision-making phase, it performs this task by computing the average of the weights generated within the constructed trees [73].

D. K-MEANS CLUSTERING METHOD

K-means Clustering Method (KCM), is a clustering algorithm that partitions a dataset into k distinct subgroups [74]. Each subgroup is grouped around a designated cluster center. Within the dataset, data centers are selected from predetermined random points. These points are associated with the cluster center that is closest to them. In this association process, cosine similarity computation is used for measurement [74]. By employing cosine similarity calculation and the designated random points as centers, data points are included in the clusters. Using this method, a dataset is clustered to find similarities.

IV. METHODOLOGY

There are four modules in our architecture.

1) Data collection module
2) Processing module
3) Text Representation module
4) Classification module

The general architecture of the proposed system is given in Figure 8. The details of these modules are given below.

A. DATA COLLECTION AND ANNOTATION MODULE

1) DATA COLLECTION

The first step in the process of creating a Turkish tweet dataset is to determine the keywords that will be used to collect tweets. For this purpose, the list of target groups in the ''Report on Hate Speech and Discriminatory Speech in the Media'' [75] published in 2019 within the scope of the ''Monitoring of Hate Speech in the Media Project'' carried out by the Hrant Dink Foundation was taken as basis. The project aims to scan the national and local print media in Turkey and identify texts containing discriminatory, marginalizing and targeting expressions. In line with these findings, media monitoring reports containing qualitative and quantitative analyzes for four-month periods are prepared.

Tweet before Preprocessing	Tweet before Preprocessing
in Turkish: "arslan bulut yazdı "300 metrekarelik konaklar yaptıralım suriyelilere..." akp iktidarı istanbul'un ve güneydoğu şehirlerinin demografik yapısını değiştirdi! Suriyeli çalıştıranlara teşvik bile veriyorlar! türk çalıştırırsan teşvik yok!" *(English translation: "arslan bulut wrote, "let's build 300 square meter mansions for syrians..." akp government changed the demographic structure of Istanbul and southeastern cities! they even give incentives to those who employ syrians! if you employ turks, there is no incentive!")*	in Turkish: "arslan bulut yaz metrekarelik konak yaptır suriyeli akp iktidar istanbul ve güneydoğu şehir demografik yapı değiştir suriyeli çalıştıran teşvik bile ver türk çalıştır teşvik yok." *(English translation: "arslan bulut write build a square meter mansion syrian akp government change the demographic structure of istanbul and the southeastern city give incentives to employ syrians even give incentives to employ turks there is no incentive")*
in Turkish: "kimin parası bu türk halkının parası peki bizim bu kadar suriyeli baktığımız yetmediği gibi bir de bahçeli ev mi yapacağız? türkiye tapusunu da verelim mi suriyelilere... hani şu yakaladığında arkadan vuran suriyelilere... besle kargayı oysun gözünü yapan suriyelilere..." *(English translation: "whose money is this, the money of the turkish people? well, as if taking care of so many syrians is not enough, are we going to build a house with a garden? should we also give the turkish title deed to the syrians... you know, those syrians who stab them from behind when they catch them... feed the crow to tear out your eyes types of syrians..."*	in Turkish: "kim para bu türk halk para peki biz bu kadar suriyeli baktık yetmedik gibi bir de bahçeli ev mi yap türkiye tapu da ver mi suriyeli hani şu yakaladık arka vuran suriyeli bes karga oy göz yapan suriyeli" *(English translation: "who is the money turkish people money then we looked at all these syrians it is not enough should you build a house with a garden will turkey give you the title deed you know this syrian, we caught the syrian who stabbed feed crows the syrian who made eyes.")*
in Turkish: "o zaman niye suriyeli çalıştırana teşvik ve mülteci entegrasyonu için okul yapıyorsunuz. gönderin gitsin tiksindik artık görmek istemiyoruz suriyeli mülteci" *(English translation: "then why are you building schools to encourage syrian employees and to integrate refugees? send them away, we are disgusted, we don't want to see syrian refugees anymore.")*	in Turkish: "o zaman niye suriyeli çalıştıran teşvik ve mülteci entegrasyon için okul yap gönder git tiksin artık görmek iste suriyeli mülteci" *(English translation: "then why do you encourage employing syrians and build schools for refugee integration and send them away let them go and be disgusted now want to see syrian refugees.")*

FIGURE 9. Data example before and after preprocessing from collected datasets.

During the tweet dataset construction phase, in addition to the names of the target groups that produced the most hate speech in national and local media sources specified in the relevant report, different keywords were also selected to capture such discourses more comprehensively. For instance, keywords such as ‘‘Müslüman (eng. transl.: *Muslim*)’’ and ‘‘ alevi (eng. transl.:*flame*’’ were used to monitor religious hate speech, and ‘‘kadın/kadınlar (eng. transl.: *woman/women*)’’ keywords were used to detect sexist hate speech. In addition, on the Within the scope of the study, the 25 keywords used to create the data set are as follows: suriyeli (eng. transl.: *syrian*), ermeni (eng. transl.: *armenian*), ingiliz (eng. transl.: *british*), kürt (eng. transl.: *kurdish*), yunan (eng. transl.: *greek*), yahudi (eng. transl.: *jewish*), rum (eng. transl.: *greek*), arap (eng. transl.: *arab*), alevi (eng. transl.: *flame*), mülteci (eng. transl.: *refugee*), kadın/kadnlar (eng. transl.: *woman/women*), ateist (eng. transl.: *atheist*), hıristiyan/hristiyan (eng. transl.: *christian/christian*), gavur (eng. transl.: *infidel*), göçmen (eng. transl.: *immigrant*), batılı (eng. transl.: *westerner*), fransız (eng. transl.: *french*), alman (eng. transl.: *german*), inli (eng. transl.: *chinese*), sırp (eng. transl.: *serbian*), müslüman (eng. transl.: *muslim*), rus (eng. transl.: *russian*), gay (eng. transl.: *gay*), eşcinsel (eng. transl.: *homosexual and bigot*) and yobaz (eng. transl.: *bigot*). KNIME tool was used for tweet collection. As a result of this study, 2K dataset and 9K datasets were created.

In this research study, we use three different datasets to conduct our experiments. These datasets are denoted as the 1K dataset, the 2K dataset, and the 9K dataset respectively. 2K dataset, 2049 tweets were collected between January 1, 2022, and February 27, 2022. For 9K dataset, 9818 tweets were collected between March 27, 2022, and April 3, 2022. 1K dataset was prepared Mayda et al. [31] and is publicly accessible. This dateset contains 1000 Turkish tweets which are labeled into three different classes, namely hate speech, aggressive expression, and neither, based on the assessments of two independent evaluators. In cases where the annotators were uncertain, a third independent annotator’s evaluation was taken to ensure a reliable outcome. Out of the 1000 tweets included in the dataset, 276 were classified as hate speech, 60 as aggressive expressions, and 664 as neither by the annotators. The details of 1K dataset are also shown in Figure 10.

2) DATA ANNOTATION

Within the scope of this study, a total of 3 annotators who are expert on this subject were involved in the data labeling process. First, all tweets were individually labelled by two independent annotators. After the independent labeling of the annotators was completed, these labels were compared by a third annotator and a consensus was tried to be reached by communicating and convincing each other on

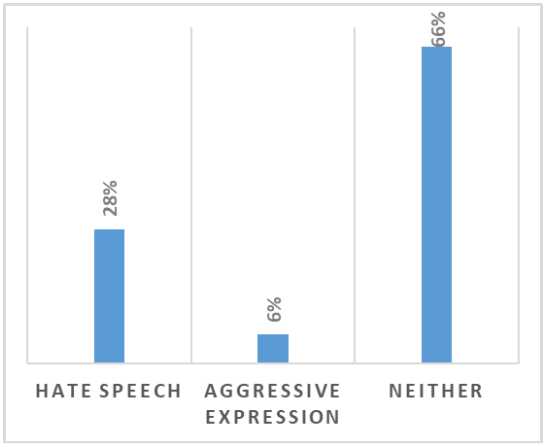

FIGURE 10. Class distribution of the 1K dataset.

the differently labeled data. Tweets for which there was no consensus on labeling were asked to a third annotator who do not see already given labels by previous annotators at all. Finally, after receiving the third annotator's labels, they were subjected to majority vote and a label was assigned to the relevant data.

In the labeling process, the following three class labels were used:

- "Hate": Tweets containing hate speech.
- "Offensive": Tweets that contain offensive expressions such as swearing and insults, but do not fall within the scope of hate speech.
- "None": Tweets that neither contain hate speech nor contain offensive language.

Additionally, the data labeled as hate speech was assigned a small subcategory label that expresses the category of hate speech, following the methodology used by Mayda et al. [8]. These subcategories include different areas such as ethnic, religious, gender or political/ideological. Subclass labels are as follows:

- "Ethnic": Tweets containing hate speech related to race or nationality.
- "Religious": Tweets containing hate speech related to belief, religion or sect.
- "Sexist": Tweets containing hate speech related to gender or sexual orientation.
- "Political": Tweets containing hate speech about political groups.

As in many parts of the world, Turkey has witnessed the historical journeys of different ethnic groups due to wars and migrations. For this reason, people from many different ethnic groups live in Turkey along with Turks. Some of these groups are as follows: Rom, Dom, Lom, Kurd, Armenian, Crimean Tatar, Uyghur, Georgian, Circassian etc. [31]. Many of these groups speak Turkish with some words and expressions specific to their culture. It is quite normal for these groups to have different behaviors, attitudes and sensitivities due to their cultural differences. Therefore, the definition of offensive speech or hate speech may vary depending on these groups. Consequently, for a more meticulous and realistic hate speech analysis, it may be necessary to make an analysis by taking into account the cultural values of these groups and include them under the subclass namely 'ethnic' in the hate speech model to be created.

As in the studies of Mayda et al. [8], no specific experiment was conducted on hate speech subclasses in this study. However, these subcategory labels can be used for more comprehensive analysis and review in future research.

Consequently by following above annotation process, we prepared the 2K and 9K dataset. 2K dateset contains 2049 Turkish tweets which are labeled into three different classes, namely hate speech, aggressive expression, and neither, based on the assessments of two independent evaluators. Out of the 2049 tweets included in the dataset, 376 were classified as hate speech, 385 as aggressive expressions, and 1288 as neither by the annotators. The details are also shown in Figure 11. 9K dateset contains 9818 Turkish tweets which are labeled into three different classes, namely hate speech, aggressive expression, and neither, based on the assessments of two independent evaluators. Out of the 9818 tweets included in the dataset, 1757 were classified as hate speech, 1614 as aggressive expressions, and 6447 as neither by the annotators.

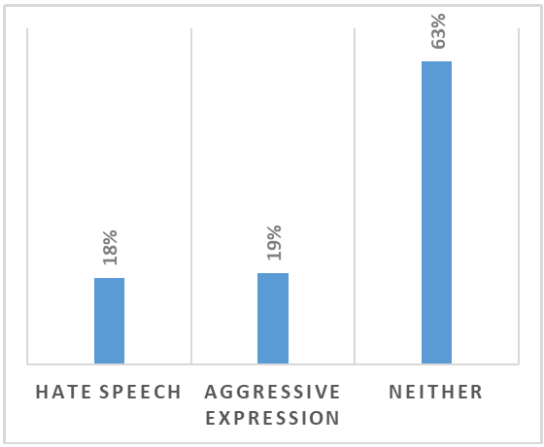

FIGURE 11. Class distribution of the 2K dataset.

The details are also shown in Figure 14. The data labeled as hate speech were also assigned at least one subcategory label, such as ethnic, religious, gender-based, or political/ideological, following the methodology employed by Mayda et al. [31]. However, this study also did not conduct experiments on hate speech subcategories. Instead, these subcategory labels could be utilized in future research for more comprehensive analyses and investigations.

There are some sample tweets which belong to different categories such as ''hate'', ''offensive'' and ''none'' from 9k dataset in Table 3.

B. PROCESSING MODULE

During the preprocessing phase, all texts in the dataset were initially converted to lowercase. Numbers, URLs, and usernames were removed. Feature sets included word unigrams and bigrams, character bigrams and trigrams, and tweet-specific features. The tweet-specific feature set comprised the number of likes, retweets, the follower count of the account posting the tweet, the total number of tweets shared by the account, the total number of likes, and the number of people the account is following [31].

To stem words and separate prefixes and suffixes, the open-source Turkish NLP library Zemberek [76] was utilized. When finding word n-grams, all punctuation marks were first removed. After words were stemmed and prefixes/suffixes were removed, the resulting stems, consisting of roots and derivational morphemes, were used as terms. Character n-grams were obtained using the data as is [31]. Figure 9 shows the tweet data in the dataset before and after data processing.

C. TEXT REPRESENTATION MODULE

This study conducts an in-depth exploration of hate speech detection methodologies by extensively reviewing existing literature. It covers diverse approaches, encompassing feature engineering techniques, ML models, and deep learning models. The examination includes various techniques for generating word embeddings such as TF, Word2Vec with CBOW and Skip-gram models, Doc2Vec utilizing PV-DM and PV-DBOW approaches, as well as GloVe and BERT.

D. CLASSIFICATION MODULE

During the classification phase, a range of ML algorithms, including RF, NB, SVM, textGCN as well as deep learning approaches such as LSTM and BiLSTM, are employed. Various feature engineering techniques, such as word and character n-grams, along with the utilization of Count Vectorizer, are applied in the analysis.

A hybrid framework integrating k-means and BERT has been devised, followed by the application of textGCN for classification purposes. In the hybrid structure, initially, the test set is partitioned into k subclusters using the k-means method. In the subsequent step, for each distinct subset characterized by specific features resulting from the division of the test set using K-means, the textGCN technique is employed. Finally, by summing up the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) derived from the subsets, a cumulative outcome is obtained for the entire test set.

In order to better understand the pseudo-code given in Figure 12, a flow chart is given in Figure 13. When Figure13 is examined, steps are seen more clearly. First,

Algorithm for So-haTRed:

Require: Collected data
Ensure: Build data matrix D
Clean and preprocess D

//Preprocessing and text representation phase

```
for i = 0 to N − 1 do
for j = 0 to N − 1 do
  if (D[i][j] > 0) then
    A:= BERT (D[i][j])
  else
    continue
  end if
end for
end for
return A
```

//clustering phase

```
for k cluster (C) in each c_j cluster do
    calculate similarity sim(x_i, center(c_j));
    if(sim>max_sim) then
        max_sim = sim;
        cluster(x_i) = j;
    end if
```

//classification phase

Ensure: Build Adjacency matrix G in Graph , Parameter Step for Graph G

```
    for i = 0 to N − 1 do
        for j = 0 to N − 1 do
            if (C[i][j] ≥ threshold and C[i][j] > 0) then
                continue
            else if (C[i][j] < V) then
                C[i][j] := 0
            end if
        end for
    end for
    P:=textGCN(C)

    return P

end for
```

FIGURE 12. Algorithm for So:haTRet.

dataset has been preprocessed. Then, text representations were performed. And then clustering was performed on text-transformed dataset. After clustering, classification is performed in each cluster, and in the final phase, classification results and performance metrics of each cluster are evaluated.

V. SYSTEM SPECS, EXPERIMENTAL RESULTS AND DISCUSSION

In this section, information about dataset, the experimental environment and the experimental results are given.

A. SYSTEM SPECS FOR EXPERIMENTAL ENVIRONMENT

All experiments detailed within this study were executed on a computer with Intel(R) CPU at 4.70 GHz with 64 GB of memory. After the feature sets were established, the classification stage was conducted utilizing both the WEKA Tool and Python. For morphological analysis of the words, Zemberek [76], an open-source Turkish NLP library, was

TABLE 3. Sample tweets in raw format from different categories in 9k dataset.

Category of Hate Speech	Tweet in Turkish	English Translation
Nefret (eng. transl.:Hate)	Ülkemde bu iğrenç mültecileri istemiyorum	I don't want these disgusting refugees in my country
Nefret (eng. transl.:Hate)	Bu adamla aynı ülkede doğmuş olmaktan nefes alıyor olmaktan utanıyorum Üzücü olan bi şekilde evlenip kendisi gibi insanlar büyütecek bu ve bunlar gibiler hiç bitmeyecek Bu zihniyet içler acısı	I am ashamed to be born in the same country with this man and to be alive. The sad thing is that he will marry and raise people like him. This and others like them will never end. This mentality is heartbreaking.
Nefret (eng. transl.:Hate)	Yobaz halk olduğu sürece nefes alamayacağız iğrenç yobazlar düşün yakamızdan	As long as there are bigoted people, we will not be able to breathe, disgusting bigots, get off our backs.
Saldırgan (eng. transl.:Offensive)	Kendi rahatını bozan bir gün gelir senin hayatını bozar suriyeli	The one who disturbs his own comfort will come one day and spoil your life, Syrian.
Saldırgan (eng. transl.:Offensive)	Nargilenizi de alın gidin İçiyormusunuz üstüne mi oturuyorsunuz ne yaparsanız yapın suriyeli	Take your hookah and go, are you smoking it, sitting on it, whatever you do, Syrian.
Saldırgan (eng. transl.:Offensive)	Bu bayan ne zamandan beri Suriyeli hayranı oldu. Bunun iki bayanın bi tuhaf şeysi gibi bir filminden anımsıyorum. Garibim parasız kaldı galiba...	Since when did this lady become a Syrian fan? I remember this from a movie like "Something Weird About Two Ladies". I guess my poor guy is broke...
Hiçbiri (eng. transl.:None)	Neçirvan Barzani Alman heyetle Şengal Anlaşmasını ele aldı	Nechirvan Barzani discussed the Shingal Agreement with the German delegation
Hiçbiri (eng. transl.:None)	Bu ümmetin başına Yavuz Sultan Selim gibi bir lider gerek	This ummah needs a leader like Yavuz Sultan Selim.
Hiçbiri (eng. transl.:None)	Haliçte haç çıkarma töreni düzenlendi	A cross-breeding ceremony was held at the Golden Horn

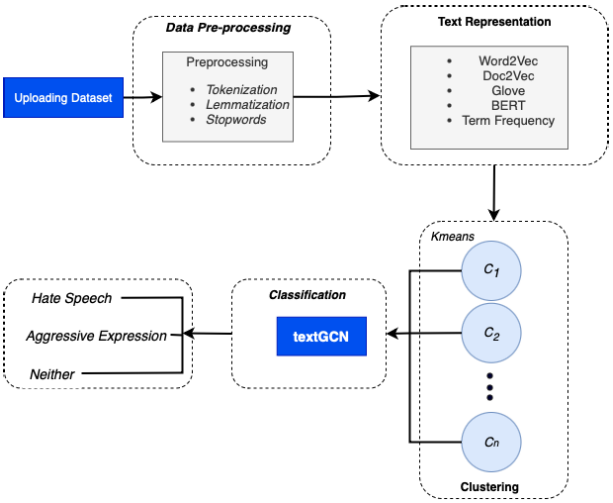

FIGURE 13. Flow chart of pseudo code.

used. Turkish hate speech datasets have been presented as open source on Github,[2] a web-based storage service.

Most of the matrix calculations were done by using Numpy.[3] For reproducibility, random seed values were used with convenient random seed functions that belong to Pytorch,[4] the standard random library of Python[5] and Pytorch Lightning[6] libraries we used especially for the preprocessing steps including splitting datasets and creating models. In experiments with the TextGCN algorithm, models were created using Pytorch, CUDA Module[7] of Pytorch was used for computations.

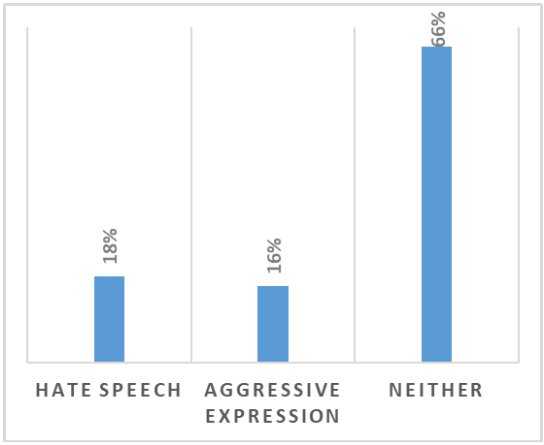

FIGURE 14. Class distribution of the 9K dataset.

B. EXPERIMENTAL RESULTS

The proposed classifiers were tested on the three Turkish hate speech detection datasets. The datasets utilized in the experiments exhibit class imbalance, and we present the

[2]https://github.com/mzahidgurbuz/Turkish-Hate-Speech-Detection.
[3]https://numpy.org/
[4]https://pytorch.org/
[5]https://www.python.org/
[6]https://www.pytorchlightning.ai/
[7]https://pytorch.org/docs/stable/cuda.html

experimental results using the F1 score which shown in Equation (3) [77]. Specifically, we assessed the F1 scores as percentages for classification algorithms utilizing various features on each dataset individually. Calculation of F1-score is important in classification problems because it combines precision and recall into a balanced metric using the harmonic mean, which is effective in evaluating models, especially in imbalanced datasets. It is preferred to measure the success of the model in almost all studies in the literature [78]. Also, word representations were generated using embedding techniques, and we employed various embedding techniques for the purpose of comparison.

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

Precision is the ratio of accurately classified tweets for a specific sentiment to the total number of tweets classified under that sentiment. Recall, on the other hand, is the ratio of accurately classified tweets for a particular sentiment to the total number of tweets that truly belong to that sentiment [77].

TABLE 4. Experimental results (F1 scores %) of the classification algorithms with *Word unigram* features on the 1K dataset.

Feature Set	SVM	NB	RF	LSTM	Bi-LSTM	Kmeans + textGCN
TF	72.39	62.91	56.69	66.63	66.72	**74.04**
Word2Vec CBOW	72.87	63.78	68.95	69.33	69.65	**73.80**
Word2Vec Skip-gram	75.59	66.79	72.04	70.22	71.07	**76.16**
Doc2Vec PV-DM	68.37	63.82	**70.35**	64.74	65.68	68.36
Doc2Vec PV-DBOW	75.36	65.66	69.50	65.68	66.55	**75.66**
GloVe	70.74	65.55	67.06	68.76	68.93	**71.22**
BERT	75.83	67.94	71.09	72.98	75.81	**77.43**

As indicated in Table 4, we assessed the F1 scores as percentages for classification algorithms that employed Word unigram features on the 1K dataset. In the experimental results, it is clearly evident that the k-means+textGCN classifier generally outperforms other ML methods. However, when using only the Doc2Vec with PV-DM for text representation method, RF yielded better results. In experiments conducted using various text representation methods, the optimal results, with an F-measure of 77.43%, were obtained in the experiment where the k-means+textGCN classifier was employed with BERT. This has a significant value since; k-means+text GCN classifier was employed with BERT has a performance gain on an existing study in the literature on Turkish hate speech detection [15]. The performance gain could be explained by the usage of more complex and advance algorithms which exposes semantic relationships with the words and sentences in the classification phase.berna

As indicated in Table 5, we assessed the F1 scores as percentages for classification algorithms that employed Word unigram features on the 2K dataset. In the experimental results, it is clearly evident that the k-means+textGCN classifier generally outperforms other ML methods. However,

TABLE 5. Experimental results (F1 scores %) of the classification algorithms with *Word unigram features* on the 2K dataset.

Feature Set	SVM	NB	RF	LSTM	Bi-LSTM	Kmeans + textGCN
TF	72.75	63.23	56.98	66.94	67.04	**74.38**
Word2Vec CBOW	73.23	64.10	69.29	69.67	69.99	**74.16**
Word2Vec Skip-gram	75.97	67.13	72.39	70.56	71.42	**76.53**
Doc2Vec PV-DM	68.71	64.14	**70.68**	65.06	66.00	68.70
Doc2Vec PV-DBOW	75.73	65.98	69.85	66.00	66.89	**76.04**
GloVe	71.07	65.88	67.38	69.11	69.26	**71.57**
BERT	76.20	68.26	71.44	73.34	76.18	**77.81**

when using only the Doc2Vec with PV-DM for text representation method, RF yielded better results. In experiments conducted using various text representation methods, the optimal results, with an F-measure of 77.81%, were obtained in the experiment where the k-means+textGCN classifier was employed with BERT.

TABLE 6. Experimental results (F1 scores %) of the classification algorithms with *Word unigram features* on the 9K dataset.

Feature Set	SVM	NB	RF	LSTM	Bi-LSTM	Kmeans + textGCN
TF	75.90	65.96	59.44	69.85	69.94	**77.60**
Word2Vec CBOW	76.39	66.86	72.28	72.69	73.02	**77.37**
Word2Vec Skip-gram	79.24	70.02	75.51	73.61	74.51	**79.84**
Doc2Vec PV-DM	71.68	66.90	**73.74**	67.87	68.86	71.67
Doc2Vec PV-DBOW	78.99	68.84	72.86	68.86	69.76	**79.32**
GloVe	74.15	68.72	70.30	72.08	72.27	**74.66**
BERT	79.50	71.23	74.53	76.50	79.48	**81.17**

As indicated in Table 6, we assessed the F1 scores as percentages for classification algorithms that employed Word unigram features on the 9K dataset. In the experimental results, it is clearly evident that the k-means+textGCN classifier generally outperforms other ML methods. However, when using only the Doc2Vec with PV-DM for text representation method, RF yielded better results. In experiments conducted using various text representation methods, the optimal results, with an F-measure of 81.17%, were obtained in the experiment where the k-means+textGCN classifier was employed with BERT.

In various experiments conducted with different datasets, it was observed that an increase in the volume of data consistently led to higher F1 scores across all experimental outcomes. Notably, the most elevated results were consistently obtained when experiments were conducted on the 9k dataset.

As indicated in Table 7, we assessed the F1 scores as percentages for classification algorithms that employed character trigrams + word unigram + word bigram + tweet features, features on the 1K dataset. In the experimental

TABLE 7. Experimental results (F1 scores %) of the classification algorithms with *Char. trigrams + word unigram + word bigram + tweet features* on the 1K dataset.

Feature Set	SVM	NB	RF	LSTM	Bi-LSTM	Kmeans + textGCN
TF	78.30	68.05	61.32	72.06	72.16	**80.07**
Word2Vec CBOW	78.81	68.98	74.57	75.00	75.33	**79.82**
Word2Vec Skip-gram	81.76	72.25	77.92	75.95	76.87	**82.37**
Doc2Vec PV-DM	73.96	69.03	**76.09**	70.02	71.04	73.95
Doc2Vec PV-DBOW	81.50	71.02	75.18	71.04	71.98	**81.84**
GloVe	76.51	70.90	72.54	74.38	74.56	**77.03**
BERT	82.02	73.48	76.89	78.94	82.00	**83.75**

results, it is clearly evident that the k-means+textGCN classifier generally outperforms other ML methods. However, when using only the Doc2Vec with PV-DM for text representation method, RF yielded better results. In experiments conducted using various text representation methods, the optimal results, with an F-measure of 83.75%, were obtained in the experiment where the k-means+textGCN classifier was employed with BERT.

TABLE 8. Experimental results (F1 scores %) of the classification algorithms with *Char. trigrams + word unigram + word bigram + tweet features* on the 2K dataset.

Feature Set	SVM	NB	RF	LSTM	Bi-LSTM	Kmeans + textGCN
TF	78.54	68.25	61.50	72.28	72.38	**80.32**
Word2Vec CBOW	79.05	69.20	74.80	75.22	75.56	**80.06**
Word2Vec Skip-gram	82.01	72.46	78.15	76.17	77.10	**82.62**
Doc2Vec PV-DM	74.18	69.23	**76.31**	70.23	71.25	74.17
Doc2Vec PV-DBOW	81.75	71.23	75.41	71.25	72.20	**82.09**
GloVe	76.73	71.12	72.75	74.60	74.78	**77.25**
BERT	82.27	73.70	77.12	79.17	82.25	**84.01**

As indicated in Table 8, we assessed the F1 scores as percentages for classification algorithms that employed character trigrams + word unigram + word bigram + tweet features, features on the 2K dataset. In the experimental results, it is clearly evident that the k-means+textGCN classifier generally outperforms other ML methods. However, when using only the Doc2Vec with PV-DM for text representation method, RF yielded better results. In experiments conducted using various text representation methods, the optimal results, with an F-measure of 84.01%, were obtained in the experiment where the k-means+textGCN classifier was employed with BERT.

As indicated in Table 9, we assessed the F1 scores as percentages for classification algorithms that employed character trigrams + word unigram + word bigram + tweet features, features on the 9K dataset. In the experimental results, it is clearly evident that the k-means+textGCN classifier generally outperforms other ML methods. However, when using only the Doc2Vec with PV-DM for text representation method, RF yielded better results. In experiments conducted using various text representation methods, the optimal results, with an F-measure of 87.81%, were obtained in the experiment where the k-means+textGCN classifier was employed with BERT.

TABLE 9. Experimental Results (F1 scores %) of the classification algorithms with *Char. trigrams + word unigram + word bigram + tweet features* on the 9K dataset.

Feature Set	SVM	NB	RF	LSTM	Bi-LSTM	Kmeans + textGCN
TF	82.10	71.33	64.28	75.54	75.65	**83.95**
Word2Vec CBOW	82.63	72.32	78.19	78.61	78.98	**83.68**
Word2Vec Skip-gram	85.72	75.74	81.68	79.61	80.59	**86.35**
Doc2Vec PV-DM	77.53	72.36	**79.76**	73.41	74.47	77.52
Doc2Vec PV-DBOW	85.45	74.45	78.81	74.47	75.46	**85.79**
GloVe	80.20	74.34	76.04	77.97	78.16	**80.77**
BERT	85.99	77.03	80.61	82.74	85.97	**87.81**

In various experiments conducted with different datasets, it was observed that an increase in the volume of data consistently led to higher F1 scores across all experimental outcomes. Notably, the most elevated results were consistently obtained when experiments were conducted on the 9k dataset. Upon thorough examination of feature sets applied to datasets, it is obvious that the combination of character trigrams, word unigram, word bigram, and tweet features yields superior results compared to the word unigram feature set.

TABLE 10. Experimental results (F1 scores %) of the ensemble classification algorithms with *Char. trigrams + word unigram + word bigram + tweet features* on the 9K dataset.

Feature Set %	Alg-1	Alg-2	Alg-3	Alg-4
Training: 30% Testing: 70%	**82.10**	71.33	64.28	75.54
Training: 45% Testing: 55%	**82.63**	72.32	78.19	78.61
Training: 60% Testing: 40%	**85.72**	75.74	81.68	79.61
Training: 75% Testing: 25%	77.53	72.36	**79.76**	73.41
Training: 30% Testing: 70%	**85.45**	74.45	78.81	74.47
Training: 45% Testing: 55%	**80.20**	74.34	76.04	77.97
Training: 60% Testing: 40%	**85.99**	77.03	80.61	82.74

As indicated in Table 10, we assessed the F1 scores as percentages for ensemble classification algorithms that employed character trigrams + word unigram + word bigram + tweet features, features on the 9K dataset. The ensemble classification methods employed are as follows:

- Alg-1: EL Algorithms RF, NB, SVM, kmeans+textGCN
- Alg-2: EL Algorithms SVM, bi-LSTM, LSTM, kmeans+textGCN

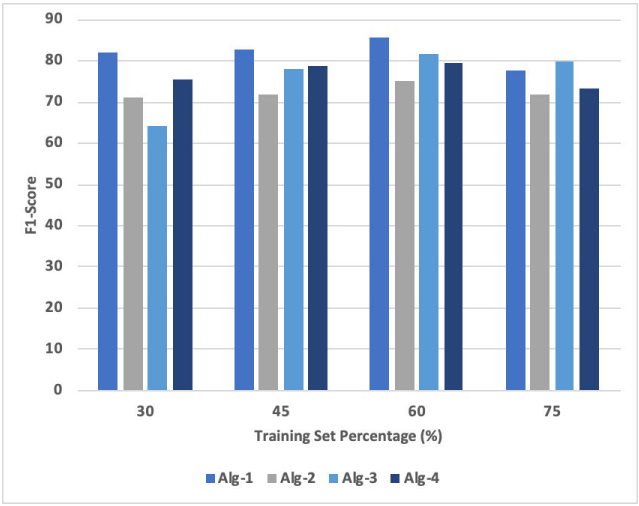

FIGURE 15. F1 scores of the ensemble classification algorithms with *Char. trigrams + word unigram + word bigram + tweet features* on the 9K dataset.

- Alg-3: EL Algorithms XGB, GB, AB
- Alg-4: EL Algorithms RF, DT, LR

In the experimental results, it is clearly evident that the Alg-1 classifier generally outperforms other ensemble classification methods. In the context of employing a training set comprising 75% of the data and a testing set comprising 25%, it was observed that Alg-3 exhibited better performance in comparison to alternative algorithms. In a series of experiments encompassing diverse training and test percentages, the Alg-1 classifier demonstrated optimal performance, an F-measure of 85.99% in the configuration employing a training percentage of 60% and a testing percentage of 40%. Figure 15 presents a graphical representation of the results detailed in Table 10.

It would also be reasonable to evaluate accuracy rates in Table 10 obtained as a result of the study from a statistical perspective. For this reason, ANOVA was preferred, which is method used to evaluate such studies [79], [80]. To perform an ANOVA on the accuracy rates of Table 10 algorithms, we'll follow these steps:

- H_0 which means accuracy rates of all algorithms are equal.
- H_1 which means accuracy rate is different from the others.

When the values in Table 10 are evaluated with ANOVA; it has been observed that these algorithms perform similarly at different training/testing separations.

C. DISCUSSION ABOUT EXPERIMENTAL RESULTS

This study represents a significant step forward in the challenging domain of hate speech detection in the Turkish language. By leveraging a combination of innovative techniques, diverse datasets, and sophisticated methodologies, we have not only contributed to the academic discourse but also provided practical insights and solutions for real-world challenges.

Our exhaustive literature review laid a solid foundation for exploring a wide range of hate speech detection methodologies, including traditional feature engineering, machine learning (ML), and deep learning models. The strategic selection and implementation of various word embedding techniques, such as TF, Word2Vec (CBOW and Skip-gram models), Doc2Vec (PV-DM and PV-DBOW approaches), GloVe, and BERT, enriched our understanding of text representation and its impact on hate speech detection accuracy.

The classification phase, characterized by the utilization of ML algorithms like Random Forest (RF), Naive Bayes (NB), Support Vector Machines (SVM), textGCN, LSTM, and BiLSTM, alongside advanced feature engineering techniques like word and character n-grams, and Count Vectorizer, showcased the versatility and effectiveness of our approach. Our rigorous evaluation using the F1 score metric provided nuanced insights into the performance nuances across different experimental setups and configurations.

Of particular note is the standout performance of the k-means+textGCN classifier with BERT, achieving an impressive F-measure of 87.81% on the 9k dataset. This finding underscores the efficacy of combining clustering techniques with advanced neural network architectures for hate speech detection, highlighting a promising avenue for future research and application.

Furthermore, our in-depth analysis of feature sets revealed the importance of holistic feature selection, with a combination of character trigrams, word unigrams, word bigrams, and tweet features consistently outperforming individual feature sets. This emphasizes the significance of context and multi-dimensional analysis in accurately identifying hate speech content.

Additionally, our comparative analysis demonstrated the superior and consistent performance of the Alg-1 classifier over alternative ensemble classification methods, showcasing its robustness and reliability across various experimental configurations. Notably, achieving an F-measure of 85.99% with a training percentage of 60% and a testing percentage of 40% on the 9k dataset further validates the effectiveness of our approach.

VI. CONCLUSION AND FUTURE DIRECTIONS

In conclusion, this study has delved into the critical task of identifying hate comments in the Turkish language, employing a multifaceted approach that integrates diverse datasets, advanced methodologies, and cutting-edge techniques. Our exploration covered a spectrum of methodologies, from traditional feature engineering to sophisticated machine learning (ML) and deep learning models. The presentation of experimental results using the F1 score metric showcases the effectiveness of our approach in hate speech detection.

Our experiments highlighted the potency of combining clustering techniques with advanced neural network

architectures, exemplified by the k-means+textGCN classifier with BERT achieving an exceptional F-measure of 87.81% on the 9k dataset. Furthermore, our analysis emphasized the significance of comprehensive feature selection in enhancing model performance compared to individual feature sets.

Looking forward, our future work will delve into several pivotal areas to propel the field of hate speech detection forward. These include exploring transfer learning techniques to improve model generalization, adopting inductive learning approaches to extract insights from unlabeled data, and leveraging transductive learning techniques for iterative model enhancement based on real-time data feedback. Additionally, we aim to explore the potential of large language models to conduct a more comprehensive examination of hate speech dynamics, while also addressing ethical considerations and biases to ensure fairness and accountability in our methodologies and implementations. Building a rule-based methodology for identification of hate speech in data collection and annotation module is another item in our future-work agenda.

By embarking on these avenues of future work, we aspire to contribute significantly to the ongoing endeavors aimed at combating hate speech online and fostering a more inclusive digital environment.

ACKNOWLEDGMENT

Points of view in this document are those of the authors and do not necessarily represent the official position or policies of TÜBİTAK.

REFERENCES

[1] Z. Zhang and L. Luo, "Hate speech detection: A solved problem? The challenging case of long tail on Twitter," *Semantic Web*, vol. 10, no. 5, pp. 925–945, Sep. 2019.

[2] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, 2017, vol. 11, no. 1, pp. 512–515.

[3] I. Gagliardone, D. Gal, T. Alves, and G. Martinez, *Countering Online Hate Speech* (Series on Internet Freedom). Paris, France: UNESCO, 2015, pp. 1–73.

[4] L. Silva, M. Mondal, D. Correa, F. Benevenuto, and I. Weber, "Analyzing the targets of hate in online social media," in *Proc. 10th Int. AAAI Conf. Web Social Media (ICWSM)*, 2016, pp. 687–690.

[5] T. Yıldız, S. Yıldırım, and B. Diri, "An integrated approach to automatic synonym detection in Turkish corpus," in *Proc. 9th Int. Conf. Adv. Natural Lang. Process. (NLP)*. Warsaw, Poland: Springer, Sep. 2014, pp. 116–127.

[6] Z. A. Güven, B. Diri, and T. Çakaloglu, "Comparison of topic modeling methods for type detection of Turkish news," in *Proc. 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2019, pp. 150–154.

[7] S. E. Seker and B. Diri, "Timeml and Turkish temporal logic," in *Proc. ICAI*, vol. 10, 2010, pp. 881–887.

[8] I. Mayda, Y. E. Demir, T. Dalyan, and B. Diri, "Hate speech dataset from Turkish tweets," in *Proc. Innov. Intell. Syst. Appl. Conf. (ASYU)*, Oct. 2021, pp. 1–6.

[9] M. F. Amasyalı and B. Diri, "Automatic Turkish text categorization in terms of author, genre and gender," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Berlin, Germany: Springer, 2006, pp. 221–226.

[10] L. B. Nielsen, "Subtle, pervasive, harmful: Racist and sexist remarks in public as hate speech," *J. Social Issues*, vol. 58, no. 2, pp. 265–280, Jan. 2002.

[11] A. B. Pawar, P. Gawali, M. Gite, M. A. Jawale, and P. William, "Challenges for hate speech recognition system: Approach based on solution," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Apr. 2022, pp. 699–704.

[12] Z. Mansur, N. Omar, and S. Tiun, "Twitter hate speech detection: A systematic review of methods, taxonomy analysis, challenges, and opportunities," *IEEE Access*, vol. 11, pp. 16226–16249, 2023.

[13] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, "Twitter and research: A systematic literature review through text mining," *IEEE Access*, vol. 8, pp. 67698–67717, 2020.

[14] M. Gaikwad, S. Ahirrao, S. Phansalkar, and K. Kotecha, "Online extremism detection: A systematic literature review with emphasis on datasets, classification techniques, validation methods, and tools," *IEEE Access*, vol. 9, pp. 48364–48404, 2021.

[15] M. A. Paz, J. Montero-Díaz, and A. Moreno-Delgado, "Hate speech: A systematized review," *SAGE Open*, vol. 10, no. 4, Oct. 2020, Art. no. 215824402097302.

[16] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, "A dataset of Hindi–English code-mixed social media text for hate speech detection," in *Proc. 2nd Workshop Comput. Model. People's Opinions, Personality, Emotions Social Media*, 2018, pp. 36–41.

[17] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, "Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113725.

[18] R. Nasiboglu and M. Gencer, "Adlandırılmış varlık tanıma modelleri ile türkçe sosyal medya metinlerinde küfürlü sözlerin sansürlenmesi," *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, vol. 23, no. 1, pp. 72–88, 2023.

[19] Ş. S. Yilmaz, O. İlyas, and H. Gökçen, "Twitter platformundan elde edilen türkçe saldırgan dil derlemi," *Mühendislik Bilimleri ve Araştırmaları Dergisi*, vol. 4, no. 2, pp. 304–316, 2022.

[20] K. Oflazer, "Turkish and its challenges for language processing," *Lang. Resour. Eval.*, vol. 48, no. 4, pp. 639–653, Dec. 2014.

[21] K. Tohma and Y. Kutlu, "Challenges encountered in Turkish natural language processing studies," *Natural Eng. Sci.*, vol. 5, no. 3, pp. 204–211, Nov. 2020.

[22] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schaefer, T. Ranasinghe, M. Zampieri, D. Nandini, and A. K. Jaiswal, "Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo–Aryan languages," 2021, *arXiv:2112.09301*.

[23] T. Y. S. S. Santosh and K. V. S. Aravind, "Hate speech detection in Hindi–English code-mixed social media text," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2019, pp. 310–313.

[24] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, "A multilingual evaluation for online hate speech detection," *ACM Trans. Internet Technol.*, vol. 20, no. 2, pp. 1–22, May 2020.

[25] T. Ghosh, A. A. K. Chowdhury, M. H. A. Banna, M. J. A. Nahian, M. S. Kaiser, and M. Mahmud, "A hybrid deep learning approach to detect Bangla social media hate speech," in *Proc. Int. Conf. Fourth Ind. Revolution Beyond*. Singapore: Springer, 2021, pp. 711–722.

[26] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in Arabic tweets using deep learning," *Multimedia Syst.*, vol. 28, no. 6, pp. 1963–1974, Dec. 2022.

[27] W. Aldjanabi, A. Dahou, M. A. A. Al-Qaness, M. A. Elaziz, A. M. Helmi, and R. Damaševič ius, "Arabic offensive and hate speech detection using a cross-corpora multi-task learning model," *Informatics*, vol. 8, no. 4, p. 69, Oct. 2021.

[28] D. Sultan, S. Mussiraliyeva, A. Toktarova, M. Nurtas, Z. Iztayev, L. Zhaidakbaeva, L. Shaimerdenova, O. Akhmetova, and B. Omarov, "Cyberbullying and hate speech detection on kazakh-language social networks," in *Proc. IEEE 7th IEEE Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2021, pp. 197–201.

[29] M. Bilal, A. Khan, S. Jan, and S. Musa, "Context-aware deep learning model for detection of Roman Urdu hate speech on social media platform," *IEEE Access*, vol. 10, pp. 121133–121151, 2022.

[30] T. Febriana and A. Budiarto, "Twitter dataset for hate speech and cyberbullying detection in Indonesian language," in *Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech)*, vol. 1, Aug. 2019, pp. 379–382.

[31] İ. Mayda, D. Banu, and T. Yildiz, "Türkçe tweetler üzerinde makine öğrenmesi ile nefret söylemi tespiti," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 24, no. 1, pp. 328–334, 2021.

[32] D.-S. Zois, A. Kapodistria, M. Yao, and C. Chelmis, "Optimal online cyberbullying detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2017–2021.

[33] M. Almaliki, A. M. Almars, I. Gad, and E.-S. Atlam, "ABMM: Arabic BERT-mini model for hate-speech detection on social media," *Electronics*, vol. 12, no. 4, p. 1048, Feb. 2023.

[34] M. U. Arshad, R. Ali, M. O. Beg, and W. Shahzad, "UHated: Hate speech detection in Urdu language using transfer learning," *Lang. Resour. Eval.*, vol. 57, no. 2, pp. 713–732, Jun. 2023.

[35] S. Aziz, M. S. Sarfraz, M. Usman, M. U. Aftab, and H. T. Rauf, "Geo-spatial mapping of hate speech prediction in Roman Urdu," *Mathematics*, vol. 11, no. 4, p. 969, Feb. 2023.

[36] W. S. S. Fernando, R. Weerasinghe, and E. R. A. D. Bandara, "Sinhala hate speech detection in social media using machine learning and deep learning," in *Proc. 22nd Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Nov. 2022, pp. 166–171.

[37] H. Karayiğit, A. Akdagli, and Ç. İ. Aci, "Homophobic and hate speech detection using multilingual-BERT model on Turkish social media," *Inf. Technol. Control*, vol. 51, no. 2, pp. 356–375, Jun. 2022.

[38] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, and S. H. Malik, "Detecting Twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 2, Nov. 2022, Art. no. 100120.

[39] V. C. Baydogan and B. Alatas, "Çevrimiçi sosyal ağlarda nefret söylemi tespiti için yapay zeka temelli algoritmaların performans değerlendirmesi," *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 33, no. 2, pp. 745–754, 2021.

[40] G. Nergiz and E. Avaroglu, "Türkçe sosyal medya yorumlarındaki siber zorbalığın derin öğrenme ile tespiti," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 31, no. 1, pp. 77–84, 2021.

[41] S. W. A. M. D. Samarasinghe, R. G. N. Meegama, and M. Punchimudiyanse, "Machine learning approach for the detection of hate speech in sinhala unicode text," in *Proc. 20th Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Nov. 2020, pp. 65–70.

[42] J. Deng, L. Cheng, and Z. Wang, "Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification," *Comput. Speech Lang.*, vol. 68, Jul. 2021, Art. no. 101182.

[43] S. Liu, K. Lee, and I. Lee, "Document-level multi-topic sentiment classification of email data with BiLSTM and data augmentation," *Knowl.-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105918.

[44] L. Shan, Y. Liu, M. Tang, M. Yang, and X. Bai, "CNN-BiLSTM hybrid neural networks with attention mechanism for well log prediction," *J. Petroleum Sci. Eng.*, vol. 205, Oct. 2021, Art. no. 108838.

[45] H. Xia, J. Weng, S. Boubaker, Z. Zhang, and S. M. Jasimuddin, "Cross-influence of information and risk effects on the IPO market: Exploring risk disclosure with a machine learning approach," *Ann. Oper. Res.*, vol. 334, nos. 1–3, pp. 761–797, Mar. 2024.

[46] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, "Bidirectional long short term memory method and Word2Vec extraction approach for hate speech detection," *IJCCS Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 2, p. 169, Apr. 2020.

[47] S. Al-Saqqa, A. Awajan, and B. Hammo, "Performance comparison of Word2Vec models for detecting Arabic hate speech on social networks," in *Proc. Int. Conf. Emerg. Trends Comput. Eng. Appl. (ETCEA)*, Nov. 2022, pp. 1–5.

[48] P. William, R. Gade, R. E. Chaudhari, A. B. Pawar, and M. A. Jawale, "Machine learning based automatic hate speech recognition system," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Apr. 2022, pp. 315–318.

[49] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and glove word embedding for offensive and hate speech text detection," *Proc. Comput. Sci.*, vol. 207, pp. 769–778, Jan. 2022.

[50] B. S. Sert, E. Elma, and A. B. Altinel, "Enhancing the performance of WSD task using regularized GNNs with semantic diffusion," *IEEE Access*, vol. 11, pp. 40565–40578, 2023.

[51] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Hate speech detection and racial bias mitigation in social media based on BERT model," *PLoS ONE*, vol. 15, no. 8, Aug. 2020, Art. no. e0237861.

[52] H. Saleh, A. Alhothali, and K. Moria, "Detection of hate speech using BERT and hate speech word embedding with deep model," *Appl. Artif. Intell.*, vol. 37, no. 1, Dec. 2023, Art. no. 2166719.

[53] M. Topaloğlu and G. Malkoç, "Decision tree application for renal calculi diagnosis," *Int. J. Appl. Math., Electron. Comput.*, vol. 4, no. 1, pp. 404–407, 2016.

[54] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016.

[55] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Syst. Appl.*, vol. 134, pp. 93–101, Nov. 2019.

[56] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6308–6325, 2020.

[57] D. C. Toledo-Pérez, J. Rodríguez-Reséndiz, R. A. Gómez-Loenzo, and J. C. Jauregui-Correa, "Support vector machine-based EMG signal classification techniques: A review," *Appl. Sci.*, vol. 9, no. 20, p. 4402, Oct. 2019.

[58] H. Gao, X. Zeng, and C. Yao, "Application of improved distributed Naive Bayesian algorithms in text classification," *J. Supercomput.*, vol. 75, no. 9, pp. 5831–5847, Sep. 2019.

[59] H. Parveen and S. Pandey, "Sentiment analysis on Twitter data-set using Naive Bayes algorithm," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 416–419.

[60] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and bi-LSTM," *Chaos, Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110212.

[61] Y. Xiao and Y. Yin, "Hybrid LSTM neural network for short-term traffic flow prediction," *Information*, vol. 10, no. 3, p. 105, Mar. 2019.

[62] Y. Yang, B. Wu, L. Li, and S. Wang, "A joint model for aspect-category sentiment analysis with TextGCN and bi-GRU," in *Proc. IEEE 5th Int. Conf. Data Sci. Cyberspace (DSC)*, Jul. 2020, pp. 156–163.

[63] H. Ren, W. Lu, Y. Xiao, X. Chang, X. Wang, Z. Dong, and D. Fang, "Graph convolutional networks in language and vision: A survey," *Knowl.-Based Syst.*, vol. 251, Sep. 2022, Art. no. 109250.

[64] Z. Wei, Z. Gui, M. Zhang, Z. Yang, Y. Mei, H. Wu, H. Liu, and J. Yu, "Text GCN-SW-KNN: A novel collaborative training multi-label classification method for WMS application themes by considering geographic semantics," *Big Earth Data*, vol. 5, no. 1, pp. 66–89, Jan. 2021.

[65] E. Taşcı and A. Onan, "K-en yakın komşu algoritması parametrelerinin sınıflandırma performansıüzerine etkisinin incelenmesi," *Akademik Bilişim*, vol. 1, no. 1, pp. 4–18, 2016.

[66] S. Şenel and B. Alatli, "Lojistik regresyon analizinin kullanıldığı makaleler üzerine bir inceleme," *J. Meas. Eval. Educ. Psychol.*, vol. 5, no. 1, pp. 35–52, 2014.

[67] H. Bircan, "Lojistik regresyon analizi: Tıp verileri üzerine bir uygulama," *Kocaeli Üniversitesi Sosyal Bilimler Dergisi*, vol. 8, no. 1, pp. 185–208, 2004.

[68] A. Dayan and A. Yilmaz, "Doğal dil işleme ve derin öğrenme algoritmaları ile makine dili modellemesi," *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, vol. 13, no. 3, pp. 467–475, 2022.

[69] V. Akram and P. Y. Taşer, "Telsiz duyarga ağlarda Byzantine saldırılarının topluluk öğrenme-tabanlı tespiti," *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, vol. 22, no. 66, pp. 905–918, 2020.

[70] A. B. A. Girgin and S. Sahin, "Improving the performance of sentiment analysis by ensemble hybrid learning algorithm with NLP and cascaded feature extraction," *Int. J. Adv. Eng. Pure Sci.*, vol. 35, no. 1, pp. 125–141, 2023.

[71] A. Hayri, "Xgboost ve mars yöntemleriyle altin fiyatlarinin kestirimi," *Ekev Akademi Dergisi*, vol. 83, no. 1, pp. 427–446, 2020.

[72] B. S. Sarikaya, "Aes algoritmasına yapılan zaman odaklı önbellek saldırılarının makine öğrenmesi ile tespiti," *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 13, no. 1, pp. 57–68, 2020.

[73] K. Seda and A. Sayar, "Yapay sinir ağları, destek vektör makineleri ve adaboost algoritması ile araç sınıflandırmasının değerlendirilmesi," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 29, no. 1, pp. 299–303, 2021.

[74] S. K. Çalışkan and İ. Soğukpınar, "Kxknn: K-means ve k en yakin komşu yöntemleri ile ağlarda nüfuz tespiti," in *Proc. EMO Yayınları*, 2008, pp. 24–120.

[75] H. D. Vakfı and İ. Engindeniz, "Medyada nefret söylemi ve ayrımcı söylem 2018 raporu," Hrant Dink Vakfı, HDV Yayınları, Istanbul, Türkiye, 2018.

[76] M. D. Akın and A. A. Akın, "Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi: Zemberek," *Elektrik Mühendisliği*, vol. 431, pp. 38–44, Jan. 2007.

[77] M. Bouazizi and T. Ohtsuki, "Multi-class sentiment analysis on Twitter: Classification performance and challenges," *Big Data Mining Analytics*, vol. 2, no. 3, pp. 181–194, Sep. 2019.

[78] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models," in *Proc. 1st Workshop Eval. Comparison NLP Syst.*, 2020, pp. 79–91.

[79] L. Stahle and S. Wold, "Analysis of variance (ANOVA)," *Chemometrics Intell. Lab. Syst.*, vol. 6, no. 4, pp. 259–272, Nov. 1989.

[80] R. G. Miller Jr., *Beyond ANOVA: Basics of Applied Statistics*. Boca Raton, FL, USA: CRC Press, 1997.

AYSE BERNA ALTINEL received the B.S. degree in computer engineering from Yeditepe University, İstanbul, Turkey, in 2004, the M.S. degree in computer engineering from Istanbul Technical University, İstanbul, in 2007, and the Ph.D. degree in computer engineering from Yıldız Technical University, İstanbul, in 2016. From 2006 to 2012, she worked as a Software and Test Engineer with Turkey İşbank and the Scientific and Technological Research Council of Turkey (TÜBİTAK). Currently, she is an Associate Professor with the Computer Engineering Department, Faculty of Technology, Marmara University. She is also the Founder and the Director of the Textual Data Analysis Research Laboratory (MetLab), Computer Engineering Department, Faculty of Technology, Marmara University. She is the author of more than 50 articles. Her research interests include textual data mining, natural language processing, social network analysis, machine learning, and bioinformatics.

GOZDE KARATAS BAYDOGMUS was born in İstanbul, Turkey, in 1991. She received the bachelor's degree from the Mathematics and Computer Science Department, Istanbul Kültür University, in 2009, the M.S. degree from the Computer Engineering Department, Istanbul Kültür University, in 2013, and the Ph.D. degree from the Computer Engineering Department, Marmara University. In 2015, she completed the master's thesis on NoSql Database Testing in Istanbul Kültür University and the Ph.D. thesis on Intrusion Detection Systems in Marmara University. She worked as a Research Assistant with the Department of Mathematics and Computer Science, Istanbul Kültür University. She is currently working as an Assistant Professor with the Computer Engineering Department, Marmara University. During the master's studies, she worked on distributed databases. She continues to work in the field of computer security. Her research interests include computer networks and security, machine learning, deep learning, cryptography, python programming, statistics, and graph theory.

SEMA SAHIN was born in Kadıköy, Turkey, in 1993. She received the bachelor's degree in computer engineering from Marmara University, Kadıköy, in 2016, where she is currently pursuing the master's degree in computer engineering. She is a Senior Java Backend Developer with over seven years of experience in the IT sector. In addition to her role as a Backend Developer, she has worked as a Full Stack Software Developer. Her research interests include artificial intelligence, machine learning, text mining, big data mining, social media mining, and sentiment analysis.

MUSTAFA ZAHID GURBUZ was born in Kahramanmaraş, Turkey, in 1982. He received the B.S., M.S., and Ph.D. degrees in mathematical engineering from Yıldız Technical University, İstanbul, Turkey, in 2015. From 2006 to 2014, he was a Research Assistant with the Computer Engineering Department, Doğuş University, İstanbul. From 2014 to 2016, he was a Lecturer. Since 2016, he has been an Assistant Professor with the Computer Engineering Department, Doğuş University. He has been the Head of the Artificial Intelligence Application and Research Center, Doğuş University, since 2024. His research interests include artificial intelligence, data science, optimization problems, and application development.