BİTLİS EREN ÜNİVERSİTESİ
FEN BİLİMLERİ DERGİSİ

ISSN: 2147-3129 / e-ISSN: 2147-3188

Article Type	: Research Article	**Year**	: 2025
Received	: February 11, 2025	**Volume**	: 14
Revised	: May 6, 2025	**Issue**	: 2
Accepted	: May 29, 2025	**Pages**	: 1077-1095
DOI	: 10.17798/bitlisfen.1637822		

HATE SPEECH DETECTION IN SOCIAL MEDIA WITH DEEP LEARNING AND LANGUAGE MODELS

Beste AKDİK [1], **Güncel SARIMAN** [2, *]

[1] *Muğla Sıtkı Koçman University, Computer Engineering Department, Muğla, Türkiye*

[2] *Muğla Sıtkı Koçman University, Information System Engineering Department, Muğla, Türkiye*

**** Corresponding Author:*** *guncelsariman@mu.edu.tr*

ABSTRACT

Nowadays, hate speech has started to spread rapidly with the increasing use of social media. Such abusive discourse can cause reputation damage and adversely affect psychological health. Large social media companies are trying to prevent this situation and increase their service quality with the increasing number of users every day. In this context, our study proposes a system that detects hate speech in texts and warns the user against hate speech. The project was implemented using machine learning, deep learning and language modeling techniques with a labeled hate speech dataset collected from various sources.

The results show that BERTweet and DistilBERT language models achieved 90% accuracy. On the other hand, although the success of the classical models was lower, they were more effective temporally.

Keywords: Natural language processing, Hate speech, Deep Learning, Language model.

1 INTRODUCTION

The development of technology has made the use of smartphones widespread and thus, social media can be accessed almost anytime and anywhere. For this reason, people spend more and more time on social media in order to interact and easily satisfy their desire to socialise. Through like and dislike buttons, private messages, comments, and conversations, users enable engagement with friends, family, and even complete strangers [1]. The increasing use of social media platforms and online tools creates fertile ground for the dissemination of negative

sentiments, and there is evidence that this can lead to serious threats and harm to social cohesion, individual welfare and freedom of expression. The fact that the user can hide his/her identity anonymously increases these discourses day by day and these discourses find a place in the literature as hate speech. Hate speech is defined as verbal or written communication that targets individuals on the basis of race, religion, ethnicity, gender or other characteristics, and has emerged as a major source of worries in the digital age [2].

Although large social media companies, whose number of users is increasing day by day, want to grow, they will want to prevent such discourses in order to provide better service and ensure user satisfaction. Research has shown that exposure to hate speech has real-world outcomes, such as increased anxiety, depression and fear among targeted groups. [3]. Additionally, the protection of the psychological health of the individual/user has a very important place in society. Various trainings and awareness-raising activities are being carried out to prevent hateful speech as it has become an increasingly common way of speaking in societies in recent years. However, since the education process is a long-term journey, it is important to produce software that can minimize the exposure and prevent these discourses, which we are mostly exposed to through media tools, from digital media. These software can detect sentences containing hate speech by analyzing the content of texts and speeches. Text-based analysis is performed for detection. In these studies where Natural Language Processing techniques are used, classical text analysis processes such as text cleaning and vectorization are run with various rule-based approaches, machine learning and deep learning algorithms, and hate speech is detected through classification. processes. [4,5]. For this reason, this study proposes a system that detects and categorizes hate speech in texts and warns the user against hate speech. It is aimed to apply the system to all kinds of social media platforms and to reduce hate speech for the benefit of society and individuals.

Natural language processing (NLP) refers to computer systems that analyze human language in many languages and try to understand or produce emotion. The input can be text, spoken language or keyboard input [6]. With NLP, it may be possible to translate from one language to another, to understand the content of the text. In addition, emotion analyses can be used to determine the weight of emotions in texts, speeches or to produce summaries of texts.

Hate speech detection also benefits from sentiment analysis studies. Sentiment analysis, a subfield of natural language processing (NLP), creates and uses models and methods to decide if the content of a text is subjective or objective, and if subjective, how this information is expressed. It examines the subjectivity of these texts as well as how strongly or weakly they are expressed. Subjective information in text analysis and NLP typically refers to natural language utterances that disclose thoughts and emotions on a specific issue. Sentiment analysis is the

process of automatically analyzing these expressions to determine the emotions they convey. Processing enormous quantities of data in this manner makes it feasible to track public opinion regarding public issues or the goods produced by different companies [7].

Detection of hate speech is made more complex by low-meaning word construction in sentences. In general, there are various methods available for hate speech classification, such as single and hybrid machine learning methods, but there can be some ambiguity when distinguishing between hateful, offensive and positive content. Along with the language model approach, the development of algorithms that look at semantic integrity has become more powerful with the transfer learning approach. The power of language models in hate speech detection is undeniable [8]. In this context, it is aimed to conduct a research comparing the success of classical machine learning methods and language models in terms of temporal and semantic accuracy in order to detect hate speech by training the dataset used in the study. Various natural language processing techniques such as machine learning, deep learning and language models are used in this study. Bidirectional Encoder Representations from Transformers (BERT), which are also frequently used in sentiment analysis, have been developed for Neural Machine Translation, but also for solving question answering, text classification, summarization and other problems [7]. BERT is a pre-trained language model with existing data. Therefore, it shows high performance in the field of natural language processing and has many variants. In this study, SVM (Support Vector Machine), RF (Random Forest) as a machine learning algorithm, LSTM (Long-Short Term Memory) as a deep learning algorithm, DistilBERT and BERTweet as a Language model are used. The dataset [9] used in this study was taken from a study in which sentences collected from various platforms were tagged and served. In our study 50.000 rows of data were used for train, test and validation.

The organization of the paper is as follows: Section 2 presents literature review, Section 3 presents the methodology, Section 4 presents the results and Section 5 presents the discussion. The last section includes a conclusion and suggestions for future development.

2 LITERATURE REVIEW

Hate speech detection has been the subject of a number of studies using automated methods. Machine learning, deep learning and language models are the traditional methods used to automatically detect hate speech. In this section, we review the literature on this topic and present the results obtained.

Irene et al. used supervised machine learning to learn a binary classifier for “racist” and “non-racist” hashtags by obtaining data from multiple Twitter accounts. The average accuracy

of the classifier on individual tweets was 76% [10]. S. Tulkens et al. present a dictionary-based approach for [11] racism detection. In addition to character n-grams, Zeerak et al. investigate the impact of non-linguistic features on the detection of hate speech. A glossary of the most descriptive terminology of the data is also provided [12].

Automatic expansion is also performed using the word2vec model. These dictionaries are trained and classified with Support Vector Machine. Davidson et al. made a significant contribution by introducing a dataset and several features specifically designed for hate speech detection [13]. Rios focuses on detecting and visualizing hate speech in social media tool that detects bad words. In the study, BERT, SVM, Convolutional Neural Networks (CNN) and Attention-Based Models were used. In the findings of the study, limitations such as the need for high accuracy rates in the classification process due to the diversity of hate speech expressions were identified [14].

Kar and Debbarma proposed the best feature extraction and hybrid diagonal gated recurrent neural network (FE-DGRNN) to detect whether common words used in different languages contain hate speech. The performance of the proposed technique is evaluated under 3 different tasks in English and 2 different tasks in German using the HASOC 2019 dataset. Looking at the results of the 3 tasks, F-measures of 94%, 95% and 91% were obtained for the English dataset, and F-measures of 90% and 91% were obtained for the German tasks [15]. Subramanian et al. use machine learning and deep learning models to present a thorough overview of current developments in sentiment analysis and hate speech identification in their survey study. They examined the many approaches and datasets employed in this area, as well as the challenges these models have when it comes to correctly detecting and categorizing hate speech and emotion in texts found online. It is urged to use the survey results to inform the creation of machine learning and deep learning solutions that are more successful in reducing hate speech and fostering an inclusive online community [16].

Chakravarthi et al. tried to identify abusive language in Tamil and code-mixed Tamil-English comments on YouTube. In the study, 4 datasets were used. Various machine learning classifiers and deep learning models were tested on these datasets. Classical word embedding models such as TF-IDF (Term Frequency — Inverse Document Frequency) and bag of words (BoW) were used in the study, along with deep learning models including multilingual transducers such as LSTM, CNN and MURIL, and TF-IDF and Word2Vec algorithms for feature extraction. The results obtained were validated using paired sample t-tests and SHAP values were used to explain the model predictions. The research addresses the important issue

of online harassment and aims to contribute to tools and methodologies for better moderation of abusive content in Tamil and similar low-resource languages [17].

The goal of Mousa et al.'s study is to identify several Arabic abusive language kinds on Twitter. Each tweet is categorized into one or more inappropriate language classes based on the words used, using a multi-parameter classification algorithm. Sentences classified as bullying, insulting, racist, vulgar, and non-offensive were separated out among the labels. The sentences were classified using the BERT family of language models (AraBERT, ArabicBERT, XLMRoBERTa, GigaBERT, MBERT, MBERT, and QARiB), CNN, and BiLSTM deep learning algorithms. With a 98.4% F1-score, the implementation of the cascade model—which was first introduced by ArabicBERT and subsequently by BiLSTM and RBF—produced the best result [18]. In their paper, Putra and Wang created a technique that uses CNN and LSTM to identify hate speech phrases in social media data. The model made advantage of the Davidson and TRAC-1 datasets to get beyond the drawbacks of conventional machine learning techniques. The Davidson dataset yielded a success rate of 73%, but the TRAC-1 dataset showed a success rate of 56% [3].

3 MATERIAL AND METHOD

This chapter provides an overview of the dataset, preprocessing processes and models. The flowchart of the proposed model and the parameters used are also shown at the end of the chapter.

3.1 Dataset

The dataset was taken from Devansh Mody and his team [9]. The dataset was created for hate speech detection from various sources such as Kaggle, GitHub and other websites and made available for projects to be developed in the field of NLP. The Content column in the dataset shows the sentences and the Label column shows whether the sentences contain hate speech or not in the form of 0-1. From this dataset 50.000 sentences and labels were extracted to be used in the study. The dataset is divided into 70% training (35.000), 20% testing (10.000) and 10% (5000) evaluation. Table 1 shows a sample image from the dataset.

This dataset has been used in various studies to evaluate the effectiveness of different machine learning and deep learning models for hate speech detection. In the study by Sharif et al. [19] a comprehensive dataset of 0.45 million comments from 18 different sources was used

and analyzed with CNN and BiLSTM based deep learning models. In the study, data diversity was managed by using the model fusion method and this approach provided higher generalization performance in hate speech detection compared to previous models. The results show that the proposed model achieves 89% accuracy rate and performs successfully in hate speech detection. Similarly, Riadi et al. [20] used a dataset of 5,000 tweets obtained from Indonesian Twitter. In the study, Support Vector Machines (SVM) algorithm was used to identify tweets containing hate speech and the model parameters were optimized with GridSearchCV method. According to the results, the RBF kernel-based SVM model was found to be an effective method for hate speech detection with 84% accuracy, 85% precision, 97% recall and 91% F1 score.

The dataset used in our study was evaluated to analyze the performance of various NLP models for hate speech detection, and it was observed that especially BERT-based models (BERTweet and DistilBERT) achieved higher accuracy rates compared to traditional machine learning methods. As in previous studies, the diversity and tag structure of our dataset makes it possible to test different modelling approaches and develop more successful methods for hate speech detection.

3.2 Pre-Processing Steps

In the preprocessing process, the words that consist of the dataset should be prepared for model training by cleaning, stemming, and splitting the sentences. For this reason, the following steps were performed in the following order and Table 1 shows the description and pseudo codes of these steps.

Table 1. Hate Speech Dataset.

Content	Label
::Please kill yourself.	1
-kelly this is yo momma speaking, please stop being a nerd.-	1
-Thank God, or we might still hear more of his crap.	1
, but be careful because I took a shit in it	1
:Sorry that I didn't answer earlier, as I am busy right now. But good work on the article scheme for the wikiproject	0
::Richard, we now do know officially that Harry Newcombe died last March 18th, Stephen Butcher died sometime la	0
:Right; well, as I was saying earlier I know the physical attacks of all the characters. Hobbits and Gimli push, Gand	0
::Ritz is German for crack.. that sounds right, if if you mean it as in the German word Ritz means crack in English. Y	0

3.2.1 Data Cleaning

The text cleaning process is performed in order to make the texts suitable for NLP models. In this study, firstly, it was checked whether the text inputs were in float type and converted to string format when necessary. Then, all texts were converted to lower case to ensure consistency. Unnecessary spaces in the texts were removed and HTML tags were completely removed and punctuation marks were removed. After removing all numbers and special characters from the text, reference numbers were also eliminated, leaving only words and whitespace.

3.2.2 Remove Stopwords

Words like "the", "are", "is", "and" and similar words have no importance in NLP. In text classification and sentiment analyses, these extra words are not given weight. Only the keywords that make up the topic should be found. Therefore, the more StopWords are removed, the better the results of the classification algorithms [21]. In this study, firstly, common stopwords in the English language are collected in a cluster. Then, the texts in the dataset were tokenized and divided into words. These tokenised words were filtered by comparing them with the set of meaningsless words and only meaningful words were left in the text.

3.2.3 Lemmatization

Lemmatization removes or modifies the suffix of the word to get down to the meaningful word base, called lemma [21]. In this study, similar to the stop words cleaning process, texts were tokenized and reduced to word level. For each word, lemmatization was applied using WordNet lemmatizer. The words were converted to their root forms and the meaning was preserved.

3.2.4 Expressive Lengthening Algorithm

In social media language, words are written exaggeratedly long to emphasize some words. In this paper, an algorithm is defined to normalize the exaggerated length of repeated characters in texts. The algorithm processes each word character by character and counts the repeated characters. When a certain threshold value (2 by default) is exceeded, the repeated characters are limited by this threshold value. In this way, repeated characters that are excessively long are shortened to a reasonable length.

Table 2. Pre-processing Steps for Hate Speech Detection in Social Media.

Pre-processing Step	Description	Pseudo Code
Clean Text	Prepares text for NLP models by removing HTML tags, punctuation marks, special characters, and numbers.	Procedure: clean_text(text) if text is float then convert to string convert to lowercase remove whitespace, HTML tags, punctuation, numbers, special characters replace multiple spaces with a single space return cleaned text
Remove Stopwords	Removes irrelevant words for classification (e.g., 'the', 'is'), leaving only meaningful keywords in the text.	Procedure: remove_stopwords(text) tokenize text into words for each word do if word not in stop_words then add to filtered_tokens return filtered tokens as a string
Lemmatization	Reduces words to their base or root form, preserving meaning.	**Procedure:** lemmatization(text) **tokenize** text into words **for** each word **do** **apply** lemmatizer to word **return** lemmatized tokens as a string

Expressive Lengthening Algorithm	Normalizes exaggerated character repetitions in social media text (e.g., 'soooo' becomes 'so').	Procedure: expressive_lengthening_algorithm (word, threshold=2)
		initialize shortened_word as an empty string, current_char as an empty string, char_count as 0
		for each char in word do
		if char == current_char then increment char_count
		else
		if char_count > threshold then append current_char * threshold
		else append current_char * char_count
		set current_char to char, reset char_count to 1
		return shortened_word

Apply All Pre-processing Steps	Sequentially applies each pre-processing step to prepare the text for analysis.	Procedure: preprocessing(text) set text = expressive_lengthening_algorithm(text) set text = clean_text(text) set text = remove_stopwords(text) set text = lemmatization(text) return text

The details of the steps followed in the pre-processing processes are given in psudo code. The preprocessing steps can be implemented more easily by using the information in Table-2.

3.3 Model Training

The preprocessed dataset is applied to model training by doing feature extraction. First, the model developed with traditional methods was run with deep learning and language model algorithms respectively. RF and SVM for the traditional model, LSTM for deep learning and BERTweet and DistilBERT for the language model were used. Figure-1 shows the model architecture with its details.

3.3.1 DistilBERT

The BERT-based distilBERT has 66 million parameters, whereas the BERT base has 110 million parameters. This makes DistilBERT 40% smaller and 60% faster than the BERT base [22]. Consequently, a faster and less costly model was developed while maintaining the information density of BERT. Diagram of BertBase and DistilBERT Model architecture is shown in Figure-1.

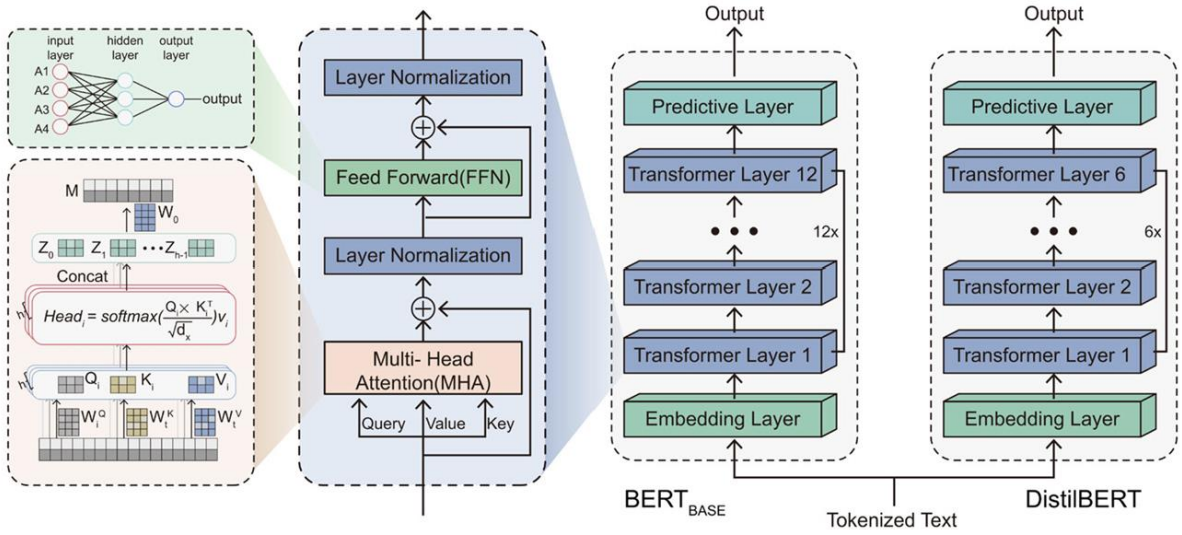

Figure 1. Diagram of BertBase and DistilBERT Model Architecture [23].

3.3.2 BERTweet

The BERT algorithm introduced by [24] was published in 2018 and has since deeply influenced the domain of Natural Language Processing. This neural network takes into account the bidirectional context in a large corpus to pre-train. Considering different objectives by masking words in a sentence or by predicting the next sentence in a text pair, the model can be fine-tuned for many other tasks. Twitter datasets collected have thus been pre-processed as required for the BERT training [25].

3.3.3 LSTM

LSTM is a recurrent neural network architecture that is designed to address the problem of learning long-term dependencies. LSTM was introduced by Hochreiter & Schmidhuber in 1997. Unlike RNNs, information is maintained in memory cells that control the flow of information. The architecture of an LSTM unit cell is built on four gating mechanisms that control the intuition, forgetting, and updating of the hidden state [26].

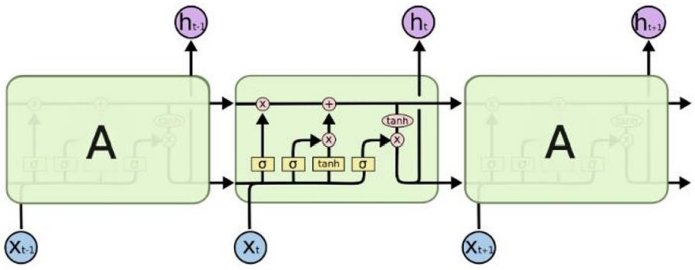

Figure 2. Diagram of BertBase and DistilBERT Model Architecture [27].

3.3.4 Word2Vec + Classifiers

This architecture breaks down dissimilar phrases into vectors of real numbers and is widely used in the sentiment analysis sector. A straightforward neural network with a single input, hidden layer, and output layer makes up the Word2Vec word embedding algorithm. Additionally, the Word2Vec embedding approach lessens the impact of popular words. Put another way, it concentrates on uncommon words while arbitrarily eliminating part of the text's repeated words. A meaningful and balanced vectorization of the words has been achieved [28]. In this study, Support Vector Machine and Random Forest classification algorithms were used together with this architecture.

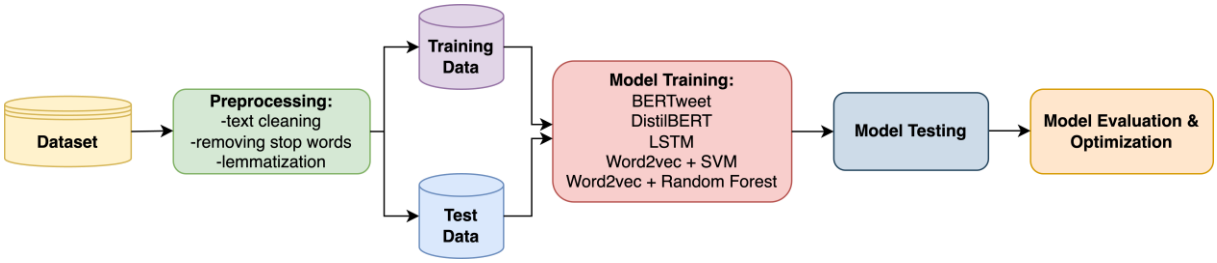

Figure 3. Proposed Model.

In our study, several machine learning and deep learning models were employed to detect hate speech on social media, including Support Vector Machine, Random Forest, Long Short-Term Memory, DistilBERT, and BERTweet and their performances were compared. As shown in Figure 3, firstly, preprocessing steps were performed on the dataset. After the preprocessing steps, the data was divided into three subsets as training, test and evaluation. BERTweet, DistilBERT, LSTM and Word2Vec-based SVM and Random Forest models were used in model training. Afterward, model evaluation & optimization and testing phases were carried out respectively. The outputs of the models were classified as 0: no hate speech, 1: hate speech. The parameters used in training the models and their values are given in Table 3. The results are presented in the Results section, comparing the accuracy and efficiency of each model in hate speech detection.

Table 3. Parameters of Training.

Architecture	Epoch	Labels	Batch Size	Window	Vector Size
BERTweet	4	2	16	-	-
DistilBERT	4	2	16	-	-
LSTM	4/16/32	-	16	-	-
Word2vec + SVM	-	-	-	5	250
Word2vec + RF	-	-	-	5	250

4 RESULTS

We report the experimental findings of our work on hate speech detection in social media comments in this section, according to the technique described in the previous part. Natural language processing techniques were employed to process and get a dataset from various platforms for the purpose of training and evaluating the models. The machine learning and deep learning models' performance metrics for this task are shown in Table 4.

Table 4. Experimental Results.

Architecture	Accuracy	Precision	Recall	F1-score	Runtime
BERTweet	0.90	0.90	0.90	0.90	38 min
DistilBERT	0.89	0.89	0.89	0.88	41 min
LSTM (4 Epoch)	0.82	0.85	0.77	0.81	2 min
LSTM (16 Epoch)	0.81	0.81	0.81	0.81	10 min
LSTM (32 Epoch)	0.81	0.81	0.79	0.80	18 min
Word2vec + SVM	0.77	0.79	0.77	0.77	10 sec
Word2vec + RF	0.79	0.79	0.79	0.79	10 sec

Table 4 summarizes the runtime required for training and finalization as well as performance metrics including Accuracy, Precision, Recall and F1-score. Among the evaluated models, BERTweet achieved the highest performance with Accuracy, Precision, Recall, and F1-score all at 0.90. Approximately 38 minutes were required for training and inference. DistilBERT followed close behind, achieving competitive results with an Accuracy of 0.89 and an F1-score of 0.88, processed in 41 minutes.

Comparatively, recurrent neural network models such as LSTM showed varying performance depending on the training epochs. The LSTM model trained over four epochs achieved 0.82 Accuracy and 0.81 F1-score, while the model trained over 16 epochs achieved 0.81 Accuracy and maintained 0.81 F1-score. The LSTM model trained over 32 epochs achieved 0.80 Accuracy and 0.80 F1-score. These models exhibited faster training times ranging from 2 to 18 minutes.

Furthermore, traditional machine learning approaches using Word2vec in combination with SVM and Random Forest classifiers achieved accuracies of 0.77 and 0.79, respectively, with fast inference times of around 10 seconds.

These results underline the effectiveness of transformer-based models such as BERTweet and DistilBERT in hate speech detection tasks and provide robust performance measures on multiple evaluation criteria.

Considering the model results, the BERTweet algorithm, which gives the most accurate and consistent results, was tested with 5 sentences randomly selected from the whole dataset to test the success of the model. In the study, 3 sentences were selected from the evaluation dataset. The last 2 sentences were randomly selected from the internet. Based on the accuracy of the results, it was not necessary to include more data in the paper. Model accuracy was obtained in the test sentences. Test, training and evaluation data are given in section 3.1. The results are given in Table-5. In the 5 sentences, our model predicted 4 sentences correctly, whereas only 1 sentence was incorrectly predicted.

Table 5. BERTweet Algorithm Test Results.

Text	Predicted Label	Actual Label	Status
Hi, please don't edit out my revised opinion of Whophd's comment on the chaser's war on everything talk. this time i make a valid point. the guy is a pompous fuckwit with a penis in each ear. Thanks.	1	1	True
The preceding unsigned comment was added by • 9 June 2006. : Unsigned, of course. Normally, I translate remarks made on my user talk page in foreign languages, but this ugly invective doesn't merit it. But I'll leave it here, because I think it says far more about the person who wrote it than it does about me.	1	0	False
20 April 2006 (UTC) Yes of course your right Aladdin, you have already refuted everything we have said and the only reason why we continue to edit is to disrupt.- | 09:07	0	0	True
And where the hell are you getting this? This is a completely ignorant point. I lived there, it definitely existed, fag.	1	1	True
== Tamar Braxton == U R retarded!!How can youtube not be a verifiable source. I put a clip of her hitting a D and E one octave above Soprano C(D7 and E7) and you say it's not a good source? What is then! I bet if I even got a video of her hitting whistle notes you'd say it wasn't verifiable. Of course it it. if you see or hear her hitting a whistle note, then she's a whistle registre singer. The same applies to all whsitle register singers!!!! U R just stupid!!!!! U and Mr. `I'll bring the food` who totally destroyed the Whsitle register singers category	1	1	True

The figures illustrate the performance evaluation of different models using multiple metrics. Figure 4 and Figure 5 provide both Precision-Recall and ROC curves for BERTweet and DistilBERT, highlighting their ability to balance precision and recall and showing their classification performance at varying thresholds.

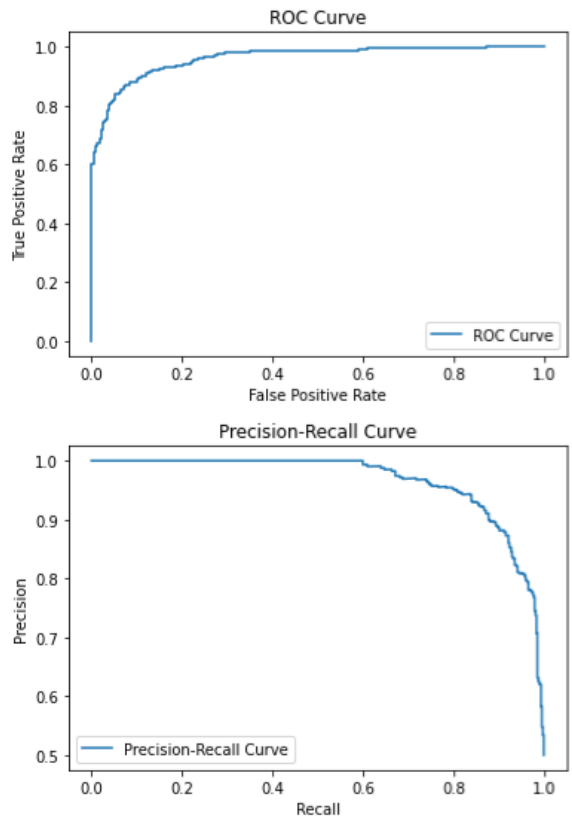

Figure 4. BERTweet Performance Metrics.

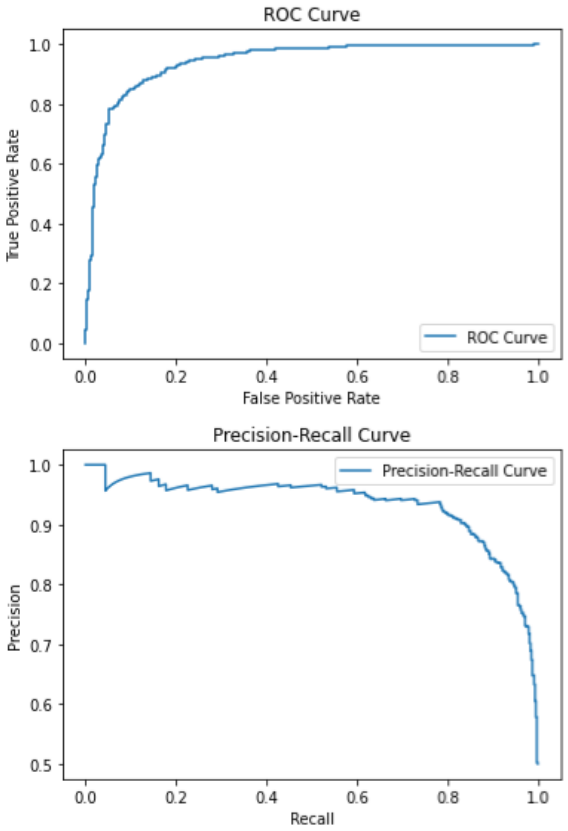

Figure 5. DistilBERT Performance Metrics.

The ROC curves for LSTM models trained with 4, 16, and 32 epochs (Figures 6-7-8) allow a comparative analysis of their performance over different training durations. Additionally, the training accuracy over epochs and the training-validation loss trends provide insights into the convergence behavior and potential overfitting of LSTM models.

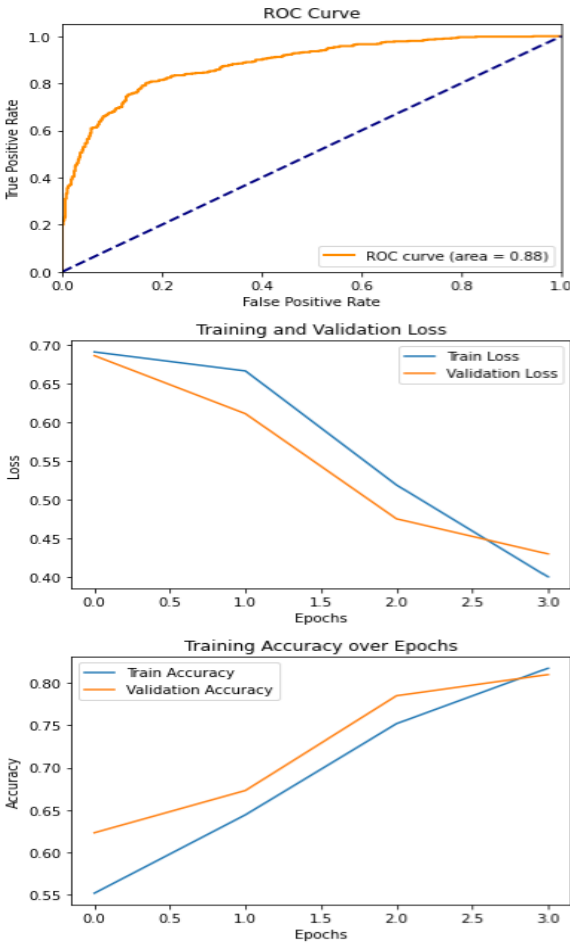

Figure 6. LSTM 4 Epoch Performance Metrics.

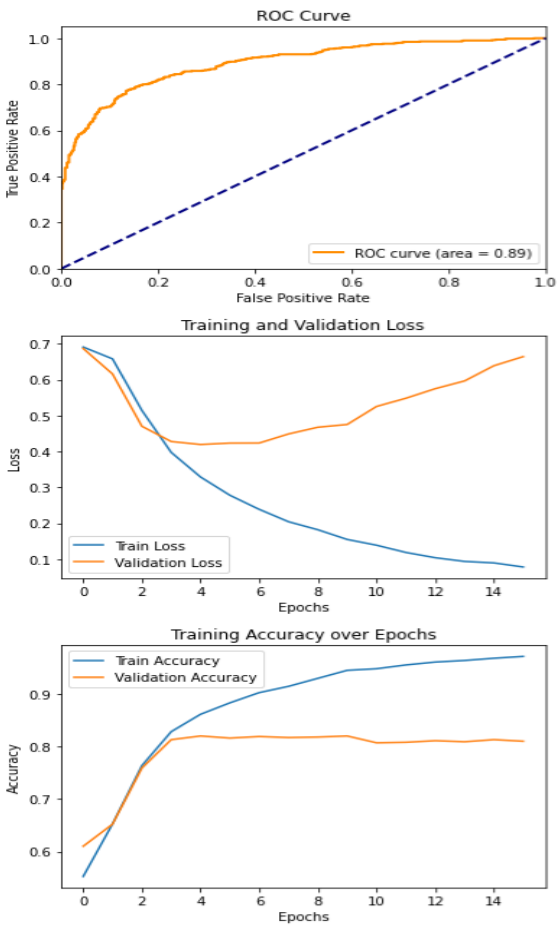

Figure 7. LSTM 16 Epoch Performance Metrics.

For traditional machine learning models, Random Forest and SVM, the ROC curves (Figure 9) are presented to assess their classification effectiveness.

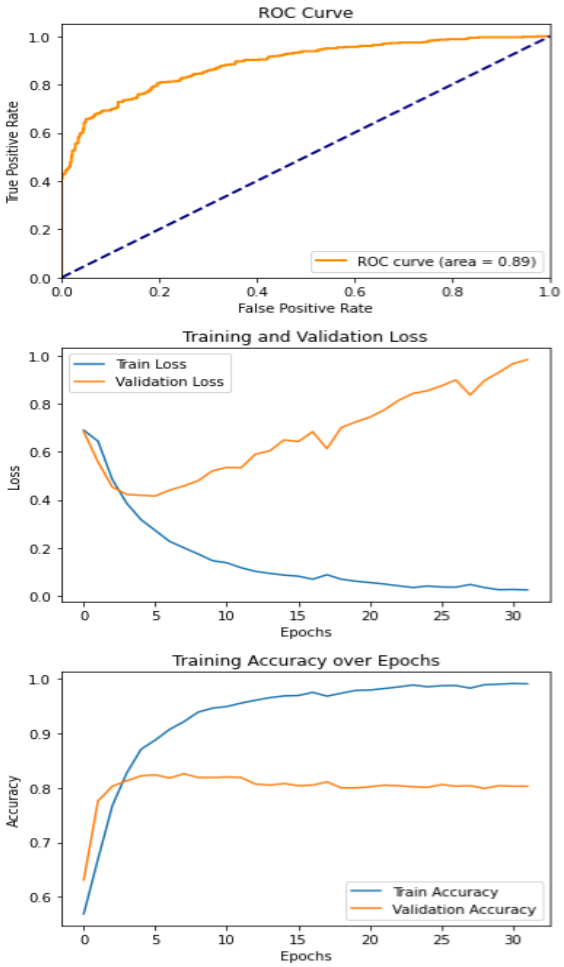

Figure 8. LSTM 32 Epoch Performance Metrics

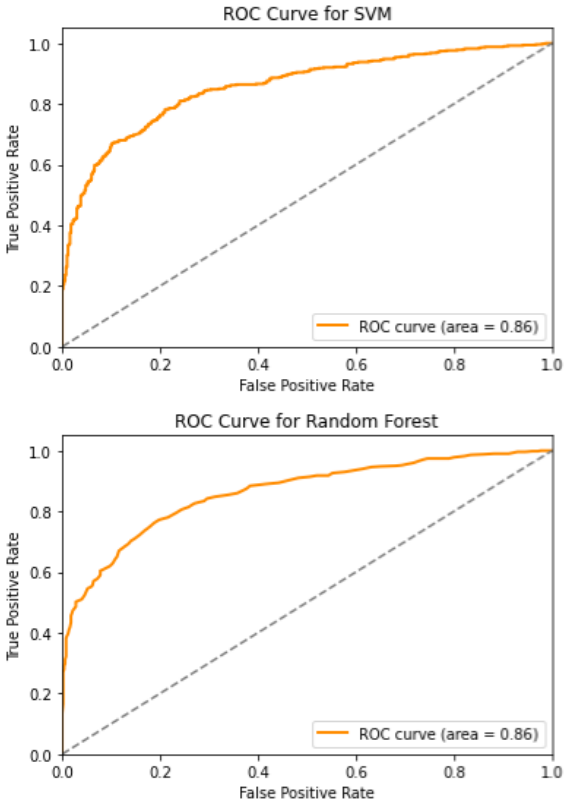

Figure 9. SVM and RF Performance Metrics

These visualizations collectively enable a comprehensive comparison of deep learning-based approaches against conventional machine learning models.

5 DISCUSSION

The results from our experiments reveal that BERT-based models (especially BERTweet and DistilBERT) outperform Word2Vec, SVM or Random Forest. In particular, BERTweet has shown to successfully handle the subtleties of social media texts with high precision, sensitivity and F1-score, achieving an accuracy of up to 90%. Although DistilBERT performed slightly lower worse than BERTweet, it showed competitive results and proved its effectiveness and efficiency in such models.

The LSTM model performed quite well with an accuracy of 85%, but showed slightly lower precision and sensitivity compared to the BERT models. This suggests that although

LSTM is able to capture sequential dependencies, it may not be as effective as BERT models, especially when using contextual embeddings in this task.

Compared to deep learning models, Word2Vec proved to perform better overall than conventional machine learning models like SVM or Random Forest. According to this, accurate hate speech detection in a variety of social media situations depends on the sophisticated contextual knowledge offered by pre-trained language models like BERT.

The incorrectly predicted the mispredicted examples during the classification process were examined and the sources of error were analysed. It was observed that the misclassified data were mostly ironic expressions or context-free hate speech. While the word has a negative meaning, sentences that may contain a sarcastic tone in context are included in the dataset. Such cases show that more context understanding should be developed during the training of the model.

In addition, when the model results are evaluated, it is seen that there is a success difference of approximately 10%, but there is a significant time difference. Although machine learning models are advantageous in terms of processing time, they lag behind language models in terms of accuracy. Since the primary goal of the study is to detect hate speech with as high accuracy as possible, BERT-based models with high accuracy despite the time difference are a more efficient choice.

In general, the BERTweet model gives the best results in terms of accuracy, while machine learning methods provide advantages in terms of processing time. However, when real-time analysis is required on dynamic and large-scale platforms such as social media, more lightweight models or optimisation techniques can be evaluated by considering the speed-accuracy trade-off.

6 CONCLUSION

Several machine learning and deep learning models have been proposed and evaluated to detect hate speech in social media. Our experiments show that BERT-based models, BERTweet and DistilBERT are highly effective in identifying hate speech with superior accuracy, precision, sensitivity and F1-scores compared to traditional methods. These findings emphasise the importance of advanced natural language processing techniques derived from pre-trained language models to address the challenge of hate speech detection on online platforms. Moreover, if such models are applied to social media platforms, instantaneous

detection and prevention of hate speech may be possible. This will contribute to a safer online experience for users.

This study makes an important contribution to the literature by comprehensively comparing both traditional machine learning methods and modern deep learning-based language models in the field of hate speech detection. In particular, the impact of language models optimized for the nature of social media data is evaluated, demonstrating their superior performance on social media content. Additionally, the computational costs and accuracy rates of transformer-based models have been analyzed alongside traditional methods to determine the most efficient approach in terms of speed and accuracy. The data preprocessing techniques used in the study were tailored to account for the unique structures of social media language, enhancing model performance. As a result, this research not only applies existing methodologies but also introduces an innovative approach to hate speech detection in social media, providing valuable insights for both academic and industrial applications.

In future studies, these models can be optimised and various approaches can be applied. In particular, further fine-tuning of BERT-based models and training them with domain-specific datasets can further improve the performance of these models. In addition, ensembles can be created by combining different language models to improve detection accuracy and efficiency.

The development of systems that analyse users' content in real time can enable the rapid detection and removal of hate speech content. The implementation of such systems can greatly reduce the spread of hate speech and harmful content in online communities. However, these technologies need to be implemented carefully, considering ethical and privacy concerns.

Pre-trained language models and powerful natural language processing techniques offer an effective method for detecting hate speech on social media platforms, marking a significant advancement in the creation of a respectful and safe online community.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

Artificial Intelligence (AI) Contribution Statement

This manuscript was entirely written, edited, analyzed, and prepared without the assistance of any artificial intelligence (AI) tools. All content, including text, data analysis, and figures, was solely generated by the authors.

Contributions of the Authors

B. AKDİK: Conceptualization, Literature Review, Model Implementation, and Experimental Design. Prepared the dataset and implemented the deep learning and language model pipelines for hate speech detection. Analyzed results and drafted the manuscript.

G. SARIMAN: Supervision, Methodology Design, Project Administration, and Validation. Critically reviewed and edited the manuscript. Contributed to experimental design, statistical analysis of results, and ensuring the scientific quality and integrity of the study.

REFERENCES

[1] S. V. Balshetwar and A. Rs, "Fake news detection in social media based on sentiment analysis using classifier techniques," Multimedia Tools and Applications, vol. 82, no. 23, pp. 35781-35811, 2023. https://doi.org/10.1007/s11042-023-14883-3

[2] N. Khanduja, N. Kumar, and A. Chauhan, "Telugu language hate speech detection using deep learning transformer models: Corpus generation and evaluation," Systems and Soft Computing, p. 200112, 2024. https://doi.org/10.1016/j.sasc.2024.200112.

[3] C. D. Putra and H.-C. Wang, "Advanced BERT-CNN for hate speech detection," Procedia Computer Science, vol. 234, pp. 239–246, 2024. https://doi.org/10.1016/j.procs.2024.02.170.

[4] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," PLOS ONE, vol. 14, no. 8, Article e0221152, 2019. https://doi.org/10.1371/journal.pone.0221152.

[5] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in Proceedings of the NAACL Student Research Workshop, Association for Computational Linguistics, 2016. https://doi.org/10.18653/v1/n16-2013.

[6] J. F. Allen, "Natural language processing," in Encyclopedia of Computer Science, 2003, pp. 1218-1222.

[7] M. A. Alonso, D. Vilares, C. Gómez-Rodríguez, and J. Vilares, "Sentiment analysis for fake news detection," Electronics, vol. 10, no. 11, p. 1348, 2021. https://doi.org/10.3390/electronics10111348

[8] Y. M. Ibrahim, R. Essameldin, and S. M. Darwish, "An adaptive hate speech detection approach using neutrosophic neural networks for social media forensics," Computers, Materials & Continua, pp. 1-10, 2024. https://doi.org/10.32604/cmc.2024.047840.

[9] D. Mody, Y. Huang, and T. E. A. de Oliveira, "A curated dataset for hate speech detection on social media text," Data in Brief, vol. 46, p. 108832, 2023.

[10] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 27, no. 1, pp. 1621-1622, 2013. https://doi.org/10.1609/aaai.v27i1.8539.

[11] S. Tulkens, L. Hilte, E. Lodewyckx, B. Verhoeven, and W. Daelemans, "A dictionary-based approach to racism detection in Dutch social media," arXiv preprint arXiv:1608.08738, 2016.

[12] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova, Author Profiling for Hate Speech Detection. [Online]. Available: http://arxiv.org/abs/1902.06734.

[13] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," arXiv preprint arXiv:1703.06707, 2017.

[14] A. Rios, "FuzzE: Fuzzy fairness evaluation of offensive language classifiers on African-American English," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, pp. 881-889, 2020. https://doi.org/10.1609/aaai.v34i01.5434.

[15] P. Kar and S. Debbarma, "Sentiment analysis & hate speech detection on English and German text collected from social media platforms using optimal feature extraction and hybrid diagonal gated recurrent neural network," Engineering Applications of Artificial Intelligence, vol. 126, p. 107143, 2023.

[16] M. Subramanian, V. E. Sathiskumar, G. Deepalakshmi, J. Cho, and G. Manikandan, "A survey on hate speech detection and sentiment analysis using machine learning and deep learning models," Alexandria Engineering Journal, vol. 80, pp. 110-121, 2023.

[17] B. R. Chakravarthi et al., "Detecting abusive comments at a fine-grained level in a low-resource language," Natural Language Processing Journal, vol. 3, p. 100006, 2023.

[18] A. Mousa, I. Shahin, A. B. Nassif, and A. Elnagar, "Detection of Arabic offensive language in social media using machine learning models," Intelligent Systems with Applications, vol. 22, p. 200376, 2024.

[19] W. Sharif, S. Abdullah, S. Iftikhar, D. Al-Madani, and S. Mumtaz, “Enhancing Hate Speech Detection in the Digital Age: A Novel Model Fusion Approach Leveraging a Comprehensive Dataset”, IEEE Access, vol. 12, pp. 27225–27236, 2024. Accessed: Mar. 30, 2025. [Online]. Available: https://doi.org/10.1109/access.2024.3367281

[20] I. Riadi, A. Fadlil, and M. Murni, "Identifying hate speech in tweets with sentiment analysis on Indonesian Twitter utilizing support vector machine algorithm," Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, vol. 9, no. 2, pp. 179–191, Oct. 2023. [Online]. Available: https://doi.org/10.23917/khif.v9i2.22470.

[21] A. Tabassum and R. R. Patil, "A survey on text pre-processing & feature extraction techniques in natural language processing," International Research Journal of Engineering and Technology (IRJET), vol. 7, no. 06, pp. 4864-4867, 2020.

[22] M. Hoekstra, Analyzing Personality Trait Intercorrelations: A Comparison between Model-Generated of Questionnaire-Derived Correlations (Master's thesis), 2023.

[23] J. Li, Y. Zhu, and K. Sun, “A novel iteration scheme with conjugate gradient for faster pruning on transformer models”, Complex & Intell. Syst., Aug. 2024. Accessed: Mar. 30, 2025. [Online]. Available: https://doi.org/10.1007/s40747-024-01595-w

[24] M. Md Abdul Qudar and V. Mago, "TweetBERT: A Pretrained Language Representation Model for Twitter Text Analysis," 2020. [PDF]

[25] M. V. Koroteev, "BERT: a review of applications in natural language processing and understanding," arXiv preprint arXiv:2103.11943, 2021. [PDF]

[26] K. Sheng Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," 2015. [PDF]

[27] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model”, Artif. Intell. Rev., vol. 53, no. 8, pp. 5929–5955, May 2020. Accessed: Mar. 30, 2025. [Online]. Available: https://doi.org/10.1007/s10462-020-09838-1

[28] P. Rakshit and A. Sarkar, "A supervised deep learning-based sentiment analysis by the implementation of Word2Vec and GloVe embedding techniques," Multimedia Tools and Applications, pp. 1-34, 2024.