International Journal of
Digital Communication and Analog Signals

ISSN: 2455-0329
Volume 11, Issue 2, 2025
July–December
DOI (Journal):10.37628/IJDCAS

https://journalspub.com/journal/IJDCAS/

Research IJDCAS

Use of Natural Language Processing for Hate Speech Detection on Social Media

Anshika Aneja[1], Apurva Garg[2], Aditya Nema[2], Ruchi Jain[1,*]

Abstract

The exponential growth of social media platforms has led to an increase in user-generated content, enhancing global connectivity but also facilitating the spread of harmful language, including hate speech. Addressing this issue requires robust, Naïve Bayes, Decision Tree, K-Nearest Neighbors, Linear Regression, and Random Forest – were tested, with Linear Regression achieving the highest accuracy automated systems for detecting and mitigating offensive content. This paper presents a comprehensive analysis of methodologies involving machine learning and natural language processing for hate speech detection, focusing on with a focus on Twitter as a data source because of its extremely dynamic and text-rich environment, we give a thorough review of approaches using machine learning and natural language processing for the detection of hate speech in this paper. Naïve Bayes, Decision Tree, K-Nearest Neighbors, Linear Regression, and Random Forest classifiers were among the algorithms that were analyzed and assessed. The ability of each of these methods to detect objectionable material in sizable datasets was evaluated. Outperforming all these models, Linear Regression outperformed the others with an accurate rating of 94%. The results of this study demonstrate how promising natural language processing is for improving the dependability of online content moderation systems. Automated systems can help create safer and more welcoming digital spaces by skillfully fusing statistical learning methods with linguistic analysis. However, there are still several restrictions, including how to deal with linguistic ambiguity, context-specific subtleties, and the ever-changing nature of online language. These difficulties highlight the need for more research that considers contextual embeddings, deep learning architecture, and hybrid strategies that combine supervised and unsupervised techniques.

Keywords: Social media platforms, hate speech, automated systems, machine learning (ML), natural language processing (NLP), Twitter data, linear regression, hate speech detection

***Author for Correspondence**
Ruchi Jain
E-mail: drruchij@lnct.ac.in

[1]Assistant Professor, Department of Artificial Intelligence and Machine Learning, Lakshmi Narain College of Technology and Science, Bhopal, Madhya Pradesh, India
[2]Student, Department of Artificial Intelligence and Machine Learning, Lakshmi Narain College of Technology and Science, Bhopal, Madhya Pradesh, India

Received Date: July 04, 2025
Accepted Date: September 13, 2025
Published Date: December 31, 2025

Citation: Anshika Aneja, Apurva Garg, Aditya Nema, Ruchi Jain. Use of Natural Language Processing for Hate Speech Detection on Social Media. International Journal of Digital Communication and Analog Signals. 2025; 11(2): 9–16p.

INTRODUCTION

The surge in social media usage over the past decade has made platforms, like Twitter, Facebook, and YouTube, vital for communication. These platforms facilitate global connections and the sharing of ideas but have also amplified the spread of harmful and abusive content. Hate speech – language that attacks or degrades individuals or groups based on attributes, such as gender, race, ethnicity, religion, or nationality – has become a pressing issue that can lead to psychological distress, social division, and violence [1].

The increasing reach and anonymity provided by social media have made hate speech more pervasive

and difficult to monitor using traditional moderation techniques. Manual moderation is labor-intensive, often inconsistent, and impractical for handling the vast amounts of data generated daily. This has prompted researchers and social media companies to explore automated solutions involving NLP and ML, which offer the potential for scalable, real-time hate speech detection.

Despite significant advancements, the task remains complex due to the nuanced nature of language and the varying forms hate speech can take. The challenge lies in differentiating between offensive but non-hateful language and outright hate speech, as well as adapting models to different cultural and linguistic contexts. This paper seeks to analyze and compare various approaches used in hate speech detection, examining their performance, limitations, and potential for real-world applications. By leveraging a combination of traditional ML algorithms and advanced deep learning models, we aim to outline the most effective strategies for building robust hate speech detection systems.

LITERATURE SURVEY

Extensive research has demonstrated the effectiveness of combining NLP and ML models in classifying hate speech. Early studies primarily focused on rule-based and basic ML approaches that relied on handcrafted features. These methods, while useful, often struggled with generalization and accuracy. As NLP evolved, so did the techniques used for hating speech detection. Researchers incorporated more sophisticated models capable of capturing deeper semantic and syntactic structures [2].

To improve the effectiveness of machine learning (ML) models for text classification and hate speech detection, pre-processing approaches are essential. By making sure that the textual input is cleaner, more standardized, and semantically relevant, processes, like lemmatization, stemming, tokenization, stop-word removal, and sentiment analysis, greatly increase the accuracy of the model [3].

In addition to lowering noise in the dataset, these pre-processing techniques assist the models in concentrating on the essential language elements required for successful classification. Pre-processing is an essential part of any natural language processing (NLP) pipeline since its quality directly affects how well later learning algorithms work [4].

Hybrid models emerged as a significant improvement over single-architecture model. By combining CNNs with LSTM networks, researchers were able to leverage the strengths of both: CNNs' ability to extract local features and LSTMs' capability to capture sequential dependencies. This approach has been particularly effective for languages with complex morphology and idiomatic expressions [5].

More recent advancements include the development of transformer-based models. BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al., brought a breakthrough by employing a bidirectional training approach that considers the context of words from both left and right. This model significantly improved the understanding of nuanced language, making it one of the most successful tools for text classification tasks [6]. Research has explored the use of character-level models, which enhance the granularity of word prediction and allow for a more detailed understanding of linguistic patterns. Unlike word-level models, character-level approaches can capture sub-word structures, prefixes, suffixes, and morphological variations, making them particularly effective for handling misspellings, slang, and creative language use that frequently appear in social media texts. This fine-grained representation strengthens the robustness of hate speech detection systems, especially when applied to noisy or unstructured datasets [7].

Ensemble learning and attention mechanisms have also gained traction as effective strategies for enhancing model performance. Combining multiple models allows researchers to harness their collective strengths, resulting in higher accuracy and more robust predictions. Attention mechanisms, particularly in models, like BiCHAT, have further improved the extraction of relevant information from text by emphasizing the most informative parts of the input sequence [8].

These developments underscore the importance of a multifaceted approach in designing hate speech detection systems. Future research continues to focus on refining these models to better handle diverse and large-scale data, incorporating cultural nuances, and reducing biases that may arise from training data.

METHODOLOGIES

- *Feature Engineering and Traditional ML Models:* Techniques, like word vectors (e.g., word2vec), n-grams, and distributional semantics, are used to enhance ML classifiers. For example, SVM classifiers achieved 79% accuracy on Twitter datasets using bigram features [9].
- *Deep Learning Models (CNN, RNN, BiRNN):* CNNs and RNNs are widely used in text classification, with CNN-LSTM hybrids proving effective even with parameter reductions. BiRNNs, shown in Figure 1 which capture both past and future context, improve contextual understanding, useful in abusive language detection [10].

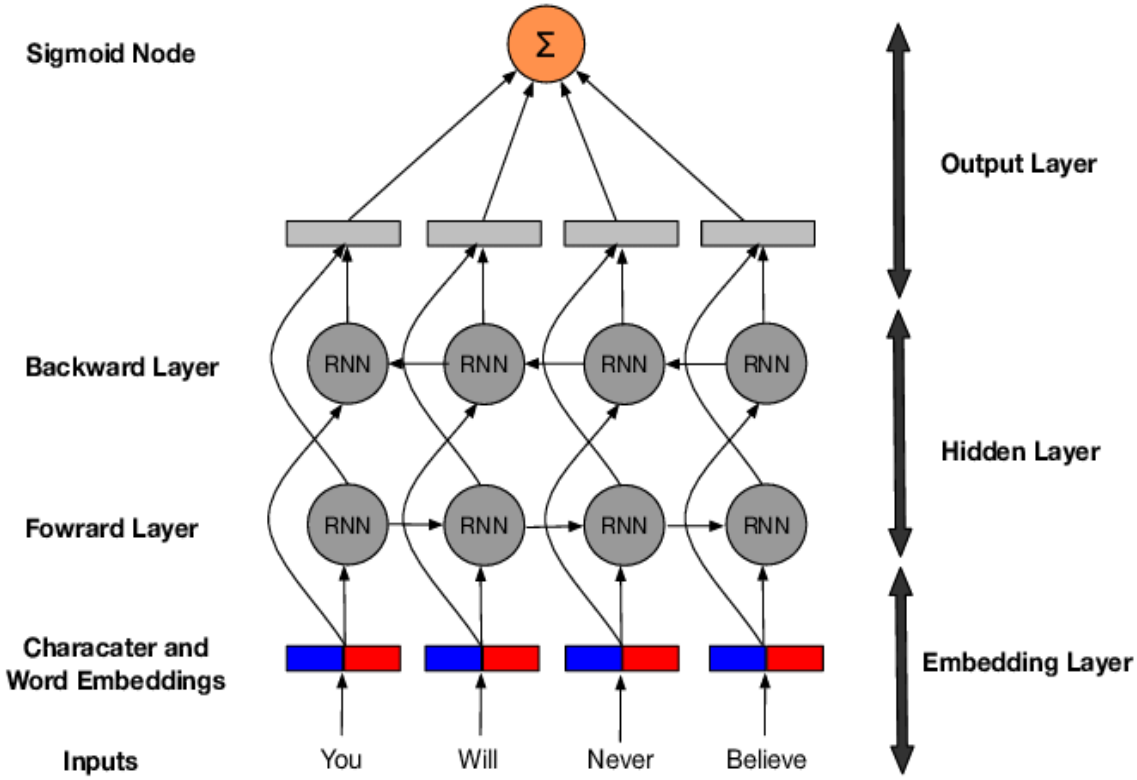

Figure 1. Architecture of a Bidirectional Recurrent Neural Network (BiRNN).

- *Embedding Models (Word2vec, AraVec):* Word embeddings capture semantic nuances crucial for abusive content detection. Models, like word2vec and domain-specific embeddings (e.g., AraVec for Arabic), enhance multilingual hate speech detection, yielding high F1-scores (e.g., 78.3%) shown in Figure 2.
- *Transformer Models (BERT, BiCHAT):* Transformer-based models, such as BERT and BiCHAT, outperform traditional ML models in precision for abusive language detection, particularly in multilingual contexts (e.g., 81.8% F1-score for hate speech).
- *Ensemble and Hybrid Models:* Hybrid models combining CNN and LSTM layers enhance contextual understanding by capturing both semantic and syntactic features, with BiLSTM-CNN achieving high accuracy in Twitter data classification.
- *Dataset-Specific and Multilingual Approaches:* Approaches tailored to specific languages or dialects (e.g., Canadian bilingual datasets) show high accuracy (e.g., 95.4% for offensive content) by leveraging localized embeddings (Table 1).

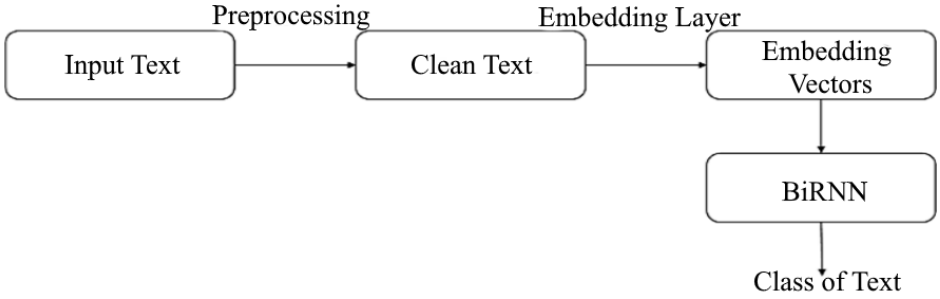

Figure 2. BiRNN-Based Text Classification Pipeline.

Example of Tweet from the Imported Dataset.

Table 1. Tweet from the imported dataset.

	Mean	Mean	Count	Max	75%	50%	25%	Min
Unidentified:0	12681.192027	12681.192027	24783.0	25296.0	18995.5	12703.0	6372.5	0.0
Record	3.243473	3.243473	24783.0	9.0	3.0	3.0	3.0	3.0
Hateful_speech	0.280515	0.280515	24783.0	7.0	0.0	0.0	0.0	0.0
Obscene_language	1.399459	1.399459	24783.0	9.0	3.0	2.0	0.0	0.0
None	1.113299	1.113299	24783.0	9.0	2.0	0.0	0.0	0.0
Group	0.462089	0.462089	24783.0	2.0	1.0	1.0	0.0	0.0

Table 2 shows significant information about many datasets, providing a quick summary of their properties and research importance.

Table 2. Description of the imported dataset.

ID	Column	Null/ non-null (0/1)	Count	Data type
0	Unidentified: 0	0	24,783	int64
1	record	1	24,783	object
2	hateful_speech	0	24,783	int64
3	obscene_language	0	24,783	int64
4	none	0	24,783	int64
5	group	0	24,783	int64
6	tweet	0	24,783	object

Table 3 shows few examples of dataset descriptions.

Table 3. A few examples of dataset characteristics.

ID	Record	Hateful speech	Obscene language	None
0	4	1	1	6
1	4	1	6	1
2	4	1	6	1
3	4	1	4	2
4	8	1	8	1

- *Challenges in Hate Speech Detection:* Variability in abusive language forms and platform-specific behaviors shown in Figure 3 pose challenges. Models need to generalize across contexts and platforms, requiring interdisciplinary solutions [1].
- *Performance Metrics:* Common metrices include accuracy, precision, recall, and F1-score, with CNN models achieving up to 90% F1-score, though simpler models, like Naïve Bayes, excel in specific contexts.
- *Data Collection and Preprocessing:* Large datasets (e.g., 24,783 Twitter instances) are gathered and split for training/testing, followed by cleaning processes to prepare for classification models.

- *Algorithms (Naïve Bayes, RF, DT, KNN, Linear Regression):* Models, like Naïve Bayes, Random Forest, Decision Tree, K-Nearest Neighbor (KNN), and Linear Regression, are evaluated for hate speech classification, each offering unique strengths across classification tasks.
- *Naïve Bayes:* Based on Bayes' theorem, this probabilistic classifier is simple yet effective, especially when feature independence is assumed.
- *Random Forest:* The mostly used ML technique known as random forest was developed by Leo Breiman and Adele Cutler, who combined the outcome of numerous decision trees to generate a single conclusion. Its popularity stems from its versatility and utility in resolving classification and regression problems [7].
- *Decision Tree:* A non-parametric method known for simplicity and ease of interpretation, but prone to overfitting. For classification and regression applications, non-parametric learning which is based on supervised learning is known as the decision tree. Its hierarchically arranged structure is made up of roots, branches, internal, and leaf nodes.
- *KNN:* It classifies data based on proximity to labeled points and is widely used for classification and regression, though it struggles with scalability for large datasets.
- *Linear Regression:* A case model with just one independent variable is simple linear regression. The variable's dependency is determined using basic linear regression.

  $y = \beta 0 + \beta 1x + \varepsilon,$

  where:
  y = The response variable (Dependent);
  x = The predictor variable (Independent) (1).

In simple regression, the impact of independent variables is differentiated from the interaction of dependent variables as shown in Figure 4.

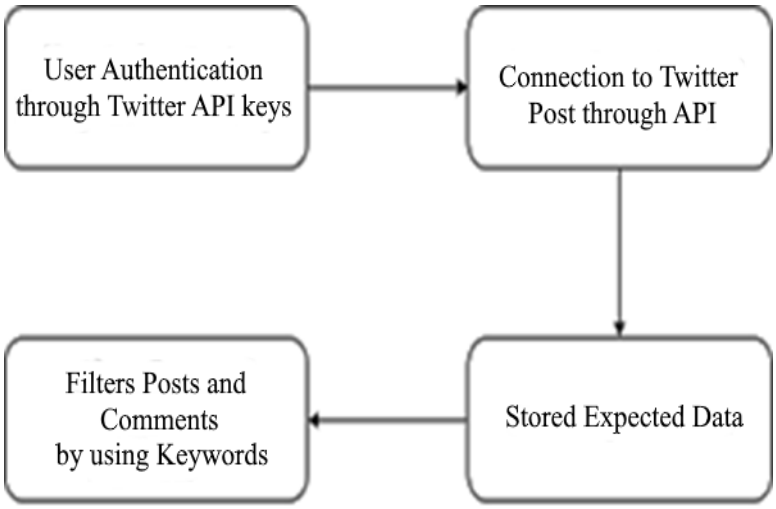

Figure 3. Data collection block diagram.

RESULTS

In the proposed method, the primary objective was to develop a system for classifying abusive and hateful language in Twitter data, employing NLP techniques with approaches such as a bag of words. The dataset initially used for this task was split into two different sets: (1) the training dataset and (2) the testing dataset, with an 80:20 allocation ratios. This division allowed for the evaluation and validation of the model's performance. To accomplish this, five different ML algorithms were employed, and their analysis is provided (Table 4).

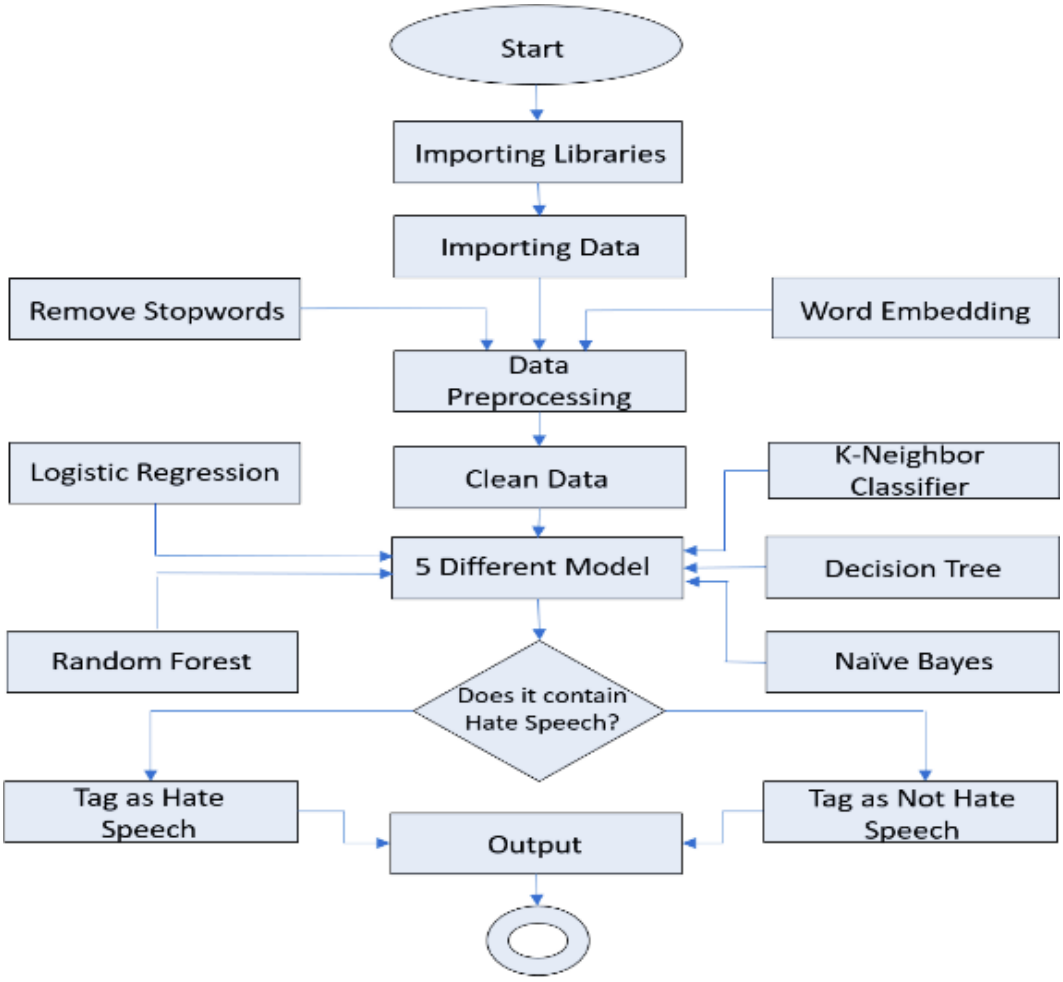

Figure 4. The proposed model's flowchart.

Table 4. Shows the Naïve Bayes model's accuracy.

	0.0	1.0	Accuracy	Weighted_avg	Macro_avg
Support	4678	279	4957	4957	4957
F1-Score	0.58	0.12	0.44	0.56	0.35
Precision	0.96	0.07		0.91	0.51
Recall	0.42	0.68		0.44	0.55

Table 5. Shows the Random Forest model.

	0.0	1.0	Accuracy	Weighted_avg	Macro_avg
Support	4678	279	4957	4957	4957
F1-Score	0.97	0.30	0.94	0.93	0.63
Precision	0.97	0.42		0.93	0.69
Recall	0.98	0.24		0.94	0.61

Table 6. Shows the Decision Tree model's accuracy.

	0.0	1.0	Accuracy	Weighted_avg	Macro_avg
Support	4678	279	4957	4957	4957
F1-Score	0.96	0.31	0.92	0.92	0.63
Precision	0.98	0.29		0.93	0.62
Recall	0.95	0.33		0.92	0.64

Table 7. Shows the K-Neighbor model's accuracy.

	0.0	1.0	Accuracy	Weighted_avg	Macro_avg
Support	4678	279	4957	4957	4957
F1-Score	0.58	0.12	0.44	0.56	0.35
Precision	0.42	0.68		0.44	0.55
Recall	0.98	0.07		0.91	0.51

Table 8. Shows the accuracy of the Linear.

	0.0	1.0	Accuracy	Weighted_avg	Macro_avg
Support	4678	279	4957	4957	4957
F1-Score	0.99	0.25	0.94	0.94	0.61
Precision	0.98	0.18		0.95	0.58
Recall	0.94	0.44		0.93	0.70

Classification report on all the different algorithms have been given above. Next, the process began by training each of these algorithms using the training datasets. Subsequently, these algorithms which have the highest accuracy during training were selected to further train and evaluate the test dataset on the model as shown in Tables 5-8. The results of these evaluations were then recorded. Table 9 shows the analysis of these five algorithms that were implemented for better accuracy.

Table 9. Final accuracy analysis results.

Model	Accuracy
The Linear Regression (LR)	94%
The K-Nearest Neighbor (KNN)	93%
The Random Forest (RF)	93%
The Naïve Bayes (NB)	45%
The Decision Tree (DT)	91%

Initially, the Naïve Bayes algorithm was applied to the test data, yielding an accuracy rate of 45%. However, this was surpassed by the Decision Tree algorithm, which accomplished a remarkable accuracy of 93%. Notably, both the Random Forest and KNN algorithms also exhibited the same accuracy rate of 93%. Of particular interest was the Linear Regression algorithm, which emerged as the most successful among all tested algorithms, achieving an accuracy rate of 94%. These findings are summarized, which provide a detailed and organized analysis of the performance of these five implemented techniques, highlighting their respective accuracy rates. In conclusion, this method demonstrated that the Linear Regression algorithm was the most effective in classifying abusive and hateful language in Twitter data, showcasing the potential of NLP techniques and ML in addressing online content moderation challenges.

CONCLUSIONS

This paper has illustrated the efficacy of NLP and ML techniques in detecting hate speech on social media platforms, with the Linear Regression model standing out as the top performer among traditional models. The hybrid CNN-LSTM models and transformer-based approaches also demonstrated strong potential, particularly for complex and multilingual data. Despite these successes, challenges remain in distinguishing between subtle and overt hate speech. Future work should aim to develop models that can better capture these nuances, explore larger and more diverse datasets, and implement real-time detection systems. Investigating more advanced transformer models, such as GPT and multilingual adaptations of BERT, will further contribute to improving the robustness of hate speech detection [6].

REFERENCES

1. Al-Makhadmeh Z, Tolba A. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing. 2020 Feb;102(2):501-22.

2. Kim Y, et al. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 1746–51.
3. Peters B, Martins AF. Beyond characters: Subword-level morpheme segmentation. InProceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology 2022 Jul (pp. 131-138).
4. Sanoussi MS, Xiaohua C, Agordzo GK, Guindo ML, Al Omari AM, Issa BM. Detection of hate speech texts using machine learning algorithm. In2022 IEEE 12th annual computing and communication workshop and conference (CCWC) 2022 Jan 26 (pp. 0266-0273). IEEE.
5. Rahman S, Jahan N, Sadia F, Mahmud I. Social crisis detection using Twitter based text mining-a machine learning approach. Bulletin of Electrical Engineering and Informatics. 2023 Apr 1;12(2):1069-77.
6. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. InProceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) 2019 Jun (pp. 4171-4186).
7. Breiman L, Cutler A. Random forests. Mach Learn. 2001;45(1):5–32.
8. Vapnik V. The nature of statistical learning theory. Springer science & business media; 2013 Jun 29.
9. Vapnik VN. The nature of statistical learning. 1998.
10. Mubeen M, Muskan A, Akram A, Rashid J, Alshalali TA, Sarwar N. Cyberbullying-related automated hate speech detection on social media platforms using stack ensemble classification method. Int J Comput Intell Syst. 2025;18(1):1–24.

