Library Progress International
Vol.44 No.3, Jul-Dec 2024: P. 26275-26287

Print version ISSN 0970 1052
Online version ISSN 2320

Original Article

Available online at www.bpasjournals.com

Contextualizing Hate: A Bi-LSTM Framework for Accurate Hate Speech Recognition

Priyanka Rajak,[1]Vasujadevi Midasala,[2]S Naga Kishore Bhavanam[3]

Department of Information Technology, Mangalayatan University, Jabalpur, Madhya Pradesh,India.
Department of Computer Science and Engineering, Mangalayatan University, Jabalpur, Madhya Pradesh,India
Department of Computer Science and Engineering, Mangalayatan University, Jabalpur, Madhya Pradesh,India.

How to cite this article: Priyanka Rajak,Vasujadevi Midasala,S Naga Kishore Bhavanam (2024) Contextualizing Hate: A Bi-LSTM Framework for Accurate Hate Speech Recognition. Library Progress International, 44(3), 26275-26287

Abstract—The research paper introduces an advanced methodology for hate speech recognition leveraging Bidirectional Long Short-Term Memory (Bi-LSTM) networks,achieving a notable accuracy of 93%. The Bi-LSTM architecture's inherent capability to capture intricate patterns and contextual subtleties within textual data renders itexceptionally effective for the nuanced task of identifying diverse manifestations of hate speech. By utilizing bidirectional processing, the model synthesizes information from both preceding and subsequent words, thereby enriching its comprehension of sentiment and intent within varying contexts. Our findings illustrate substantial enhancements over conventional machine learning approaches, under-scoring the transformative potential of deep learning techniques in mitigating the pervasive issue of online hate speech. Furthermore, this study outlines critical directions for future research, including the explorationof hybrid models that integrate attention mechanisms and transformer architectures, the expansion of diverse and representative datasets, the implementation of real-time detection systems, and the thorough examination of ethical considerations surrounding bias and fairness. Collectively,this research aims to contribute to the development of morerobust, equitable, and responsive hate speech detectionsystems, fostering healthier online discourse and promotinga safer digital environment.

Index Terms—Hate Speech, Bi-LSTM, NLP, TextualAnalysis

INTRODUCTION

The rapid growth of social media platforms over the last decade has fundamentally transformed the way people communicate, enabling billions of usersto exchange ideas and opinions across the globe.Platforms such as Twitter, Facebook, and Instagram allow for instantaneous sharing of information,which has had profound societal benefits. However,alongside the positive aspects of social media, theseplatforms have also become fertile grounds for theproliferation of hate speech, abusive language, andharmful content. It is observed that such actionshave seen a sudden increase in recent years aspresented in Fig. 1. Hate speech is defined aslanguage that attacks or disparages individuals orgroups based on attributes such as race, religion,gender, sexual orientation, or ethnicity [1]. Thespread of such content poses serious risks, includingthe promotion of violence, marginalization of minority groups, and deterioration of social cohesion.The recognition and suppression of hate speech have thus become pressing concerns for both plat-form providers and regulatory bodies. However, manual moderation is not scalable, given the volumeof content generated every minute. Therefore, thereis an increasing demand for automated systems thatcan detect and filter hate speech in real-time. Thesesystems can help identify harmful content moreefficiently and reduce the human effort required formoderation, while also improving the user experience on social media platforms by ensuring a safer and more inclusive environment [2].

Traditional approaches to detecting hate speech have relied on rule-based systems or keyword filtering methods,

which are often too simplistic to capture the nuances of human language. For example, hate speech can be implicit, where the contextor intent behind the words is critical for accurate detection. These limitations necessitate the use of more sophisticated machine learning (ML) and deeplearning (DL) techniques that can capture the complex relationships in text data. Recent advances in Natural Language Processing (NLP), such as the useof recurrent neural networks (RNNs), Long Short- Term Memory (LSTM) models, and Bidirectional LSTM (BiLSTM), have proven to be particularly effective in capturing these linguistic patterns [3], [4].

Additionally, the development of embedding techniques, such as word2vec and BERT, has al- lowed models to capture the semantic meaning of words, which is crucial for identifying hate speech that may not be explicit but is implicit through context [5]. The application of such deep learning techniques not only improves the accuracy of hate speech detection but also provides a more robust solution capable of handling large-scale social media data.

In this work, we propose a hate speech detection model using Bidirectional Long Short-Term Memory (BiLSTM) networks. BiLSTM has been shown to perform effectively in tasks requiring an under- standing of the context of words, as it processes textin both forward and backward directions, capturing long-term dependencies in the data. By combining BiLSTM with TF-IDF-based feature extraction, ourmodel is designed to efficiently detect hate speech while handling the complexities of language. We also incorporate various preprocessing steps, including n-grams analysis, word cloud visualization, and sentiment analysis, to further enhance the model's understanding of the data.

The remainder of the paper is structured as follows: Section II discusses the related work in hate speech detection, highlighting the various machine learning techniques used in previous studies. SectionIII outlines the proposed methodology, including preprocessing, model architecture, and evaluation metrics. Section IV presents the experimental results and evaluation, followed by the conclusion in Section V.

RELATED WORK

Various machine learning techniques have been applied to hate speech detection, ranging from traditional classifiers to advanced deep learning models.

This section surveys the most prominent methods, highlighting their performance based on metrics such as accuracy, precision, recall, and F1-score.

A. Single Method Approaches

1) Fuzzy Logic (FL): Fuzzy logic-based systems,such as the Fuzzy Rule-Based (FRB) method ex- plored by Haque and Rahman [6], and Tashtoush and Orabi [7], achieved moderate accuracy levels, with results ranging from 48.96% to 80%. How- ever, due to the lack of comprehensive performancemetrics (such as precision, recall, and F1-score), theeffectiveness of these systems in hate speech detection remains unclear. A more sophisticated fuzzy logic method, Fuzzy Multi-task Learning (FML),as proposed by Liu et al. [8], showed a markedimprovement with an accuracy of 93%.

2) Artificial Neural Networks (ANN): Artificial neural networks, particularly Recurrent Neural Net-works (RNNs), have been employed in hate speech detection, with varied results. Corazza et al. [9] and Pitsilis et al. [10] applied RNNs, achieving accuracy rates of 65% and 93.05%, respectively. These models demonstrated strong recall and precision rates above 70%. Convolutional Neural Networks (CNNs) were also utilized, with Winter and Kern [11] achieving accuracies ranging from 69.6% to 96.18%. In addition, Serra et al. [12] employed Multi-layer Perceptron (MLP), reporting a modest accuracy of 70.5%.

3) Deep Learning (DL): Deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, have been extensively explored in hate speech detection. Studies by Nguyen et al. [13], Modha et al. [14], and Bisht et al. [15] reported accuracies ranging from 83.37% to 89.8%.Additionally, 1D CNNs were applied by Kamble and Joshi [16], achieving an accuracy of 82.62%. These models consistently show high accuracy in detecting nuanced hate speech patterns due to their ability to capture long-term dependencies.

4) Bayesian Networks (BN): Bayesian networks,
though not as widely used as deep learning models, have shown promise with studies by Chakravartula [17], Kiilu et al. [?], and Khond et al. [?] reporting accuracy rates between 47% and 67.47%.

5) Genetic Algorithms (GA): Genetic algorithms have also been employed in this field, with Graff et al. [18] achieving accuracies of 83.4% and Miranda-Jiménez et al. [**?**] reporting 66.7%.

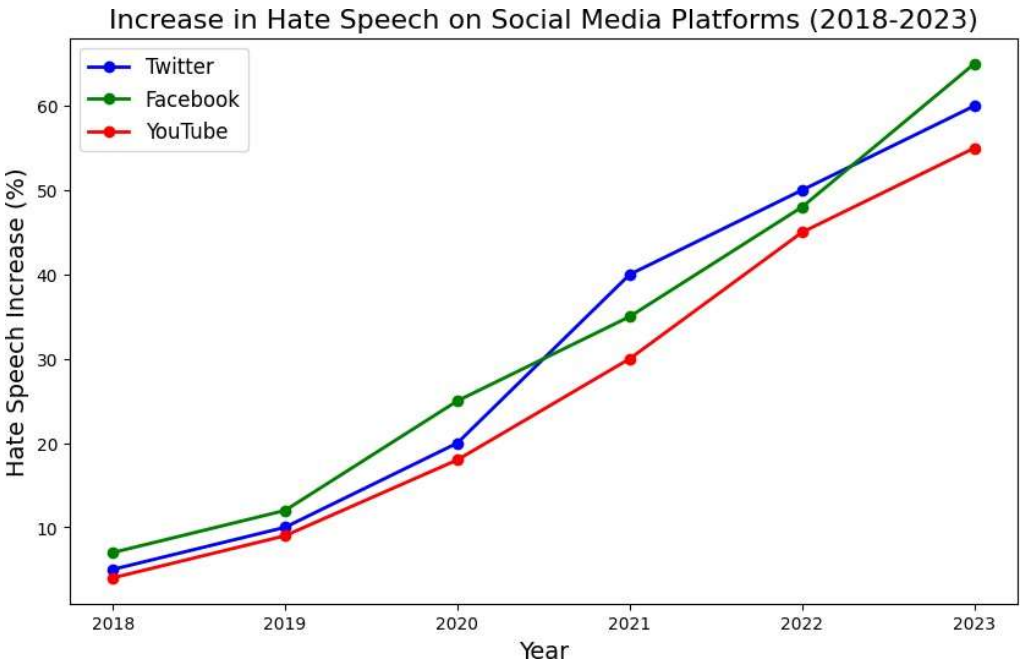


Fig. 1: Hate Speech Analysis

6) Support Vector Machines (SVM): SVM-based approaches, including those by Perelló et al. [19] and Florio et al. [20], achieved accuracies of up to 72.7%, suggesting that SVMs may underperform in detecting true positives compared to deep learning models.

7) Logistic Regression (LR): Logistic regression models, as explored by Warmsley [21] and De Cocket al. [**?**], demonstrated respectable accuracies of 61% to 72.4%.

8) Decision Trees (DT): Decision tree models, such as the J-48graft variant used by Watanabe etal. [22], achieved an accuracy of 78.4%. Random forests, a more advanced tree-based model, demonstrated better performance, as shown by Bouazizi and Ohtsuki [23], with an accuracy of 83.1% and a precision of 91.1%.

B. Hybrid Method Approaches

Hybrid models combining different techniqueshave also demonstrated promising results in hate speech detection.

1) Fuzzy Logic and NLP (FL-NLP): Hybrid approaches that combine fuzzy logic with natural language processing (NLP), as explored by Emadi and Rahgozar [24], achieved an accuracy of 83.6%.

2) Bayesian Networks and RNN (BN-RNN): **Bayesian networks combined with RNNs have reported by Miok et al. [25], achieving an accuracy of 74% with balanced precision (73.4%) and recall (78.4%).**

3) LSTM and Neural Networks (LSTM-NN): Combining LSTM with traditional neural networkshas also shown effective results, with Liu [8] achieving an accuracy of 83.7

4) Embedding and Deep Learning (EMB-DL):Embedding-based deep learning methods have achieved accuracies over 90% as reported by Silvaet al. [26] and Hemker [**?**]. These models lever-age pre-trained word embeddings to capture thesemantic richness of language, improving overallclassification performance.

5) BiLSTM and MLP: Combining BiLSTM with MLP has proven effective, with Qian et al. [27] reporting

an accuracy of 88.33%.

6) Other Hybrid Approaches: Additional hybrid methods such as Cat Swarm Optimization and LSTM (CSO-LSTM) [28] have achieved accuraciesup to 96.89%.

The literature review reveals that deep learning methods, particularly LSTM and hybrid approaches,demonstrate superior performance compared to traditional machine learning models like SVM and

TABLE I: Performance Comparison and Limitations of Reviewed Papers

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Limitations**
Fuzzy Rule Based (FRB) [6], [7]	48.96% - 80%	N/A	N/A	N/A	Lacks comprehensive performance metrics and generaliza-tion capability.
Fuzzy Multi-task Learning (FML) [8]	93%	N/A	N/A	N/A	Limited application scope; lacks detailed evaluation of recall and precision.
Recurrent Neural Network (RNN) [9], [10]	65% - 93.05%	72% - 93.34%	68% - 93.20%	N/A	Susceptible to long-term dependency issues, requires large datasets for effective learning.
Convolutional Neural Network (CNN) [11], [**?**]	69.6% - 96.18%	71.2% - 82%	70.8% - 96%	N/A	Lacks temporal sequence processing, requires large compu-tational resources.
Multi-layer Perceptron (MLP) [12]	70.5%	N/A	N/A	N/A	Limited in capturing complex language patterns and context.
Long Short-Term Memory (LSTM) [13], [14], [15]	83.37% - 89.8%	87.07% - 92.26%	82% - 89%	78.33% - 89%	High complexity, sensitive to hyperparameter tuning, requires large training data.
1D CNN [16]	82.62%	83.34%	78.51%	80.85%	Does not effectively capture long-term dependencies in text.
Bayesian Networks (BN) [17], [**?**], [**?**]	47% - 67.47%	58% - 69%	62% - 69%	N/A	Low performance on complex text datasets, lacks scalability to large corpora.
Genetic Algorithms (GA) [18], [**?**]	66.7% - 83.4%	N/A	N/A	68% - 82.1%	High computational cost and convergence issues, particularly with large datasets.
Support Vector Machines (SVM)[19], [20]	57.13% - 72.7%	49% - 72%	54% - 73%	N/A	Struggles with non-linearity and high-dimensional feature spaces, lacks deep context pro-cessing.
Logistic Regression (LR) [21], [**?**]	61% - 72.4%	90%	91%	90%	Linear model, lacks capacity to capture complex relationships and contextual dependencies intext.
Decision Tree (DT) [22], [23]	78.4% - 83.1%	79.3% - 91.1%	73.4% - 78.4%	78.4% - 81.3%	Prone to overfitting, especially in the presence of imbalanceddatasets.
BiLSTM and MLP [27]	88.33%	N/A	N/A	85.8%	High computational cost, prone to overfitting with smalldatasets.

CSO-LSTM (Cat Swarm Optimizatio nand LSTM) [28]	96.89%	73%	72%	N/A	Computationally intensive due to optimization techniques; may not generalize well to unseen data.

logistic regression. Hybrid methods, combining embedding techniques with deep learning, along with optimization algorithms, offer promising results, achieving accuracies over 90%. The wide range of methods surveyed indicates that the effectivenessof a particular model often depends on dataset characteristics, preprocessing techniques, and modelcomplexity.

I. PROPOSED METHOD

A. Preprocessing

Preprocessing is a crucial step to prepare raw textdata for input into machine learning models. In this

method, several text cleaning and feature extractiontechniques are employed to enhance the representation of the data. Additionally, exploratory analysis methods such as word clouds, n-grams analysis, polarity and subjectivity analysis, and tweet length analysis provide insights into the characteristics of the tweets, enriching the feature set for model training.

1) Data Loading and Cleaning: The dataset usedfor this task consists of tweets labeled as either hatespeech or non-hate speech. Given the noisy nature of social media data, a comprehensive cleaning process is essential to eliminate irrelevant information

and focus on meaningful content. The raw text data is cleaned by applying the following steps:

- **Removing URLs**: Links embedded in tweets are generally not relevant for the hate speech detection task and are removed from the dataset.
- **Removing Mentions and Hashtags**: Usermentions (e.g., @username) and hashtags (e.g., #topic) are stripped from the tweets,as they do not contribute significantly to the classification task.
- **Removing Special Characters and Punctuation**: Punctuation marks and special characters are removed to standardize the input text and reduce noise.
- **Lowercasing**: All text is converted to lowercase to maintain uniformity, ensuring thatwords like “Hate” and “hate” are treated the same by the model.
- **Stopword Removal**: Stopwords, which are commonly used words such as “and”, “is”, “the”, are removed from the text to focus on more meaningful content.

Alogrithm with the above discussed steps, The datasethas been made balanced as well as shown in Fig. 2.

2) Word Cloud Analysis: A Word Cloud is avisual representation of the most frequent wordsin the dataset given in Fig. 3a, where the size of each word corresponds to its frequency in the text. For this task, word clouds are generated separately for both hate speech and non-hate speech tweets, allowing us to observe differences in word usage between the two categories.

In hate speech tweets, certain words might appear more prominently, providing an intuitive understanding of the language commonly associated with hateful content. This visual tool helps quickly identify key terms that can aid in distinguishing hatespeech from non-hate speech.

3) N-grams Analysis: N-grams are contiguous sequences of n words extracted from the text presented in Fig. 3b. This technique allows for theexploration of common word combinations that might indicate the presence of hate speech. For this method, both unigrams (single words) and bigrams (pairs of words) are analyzed.

Unigrams help identify the most frequent words in the dataset, while bigrams capture common word pairs that provide deeper insight into language pat- terns. For example, certain bigrams like “go back” or “get out” may appear more frequently in hate speech tweets, indicating discriminatory language.

4) Polarity and Subjectivity Analysis: Sentiment analysis [36] is conducted using polarity and subjectivity scores as shown in Fig. 3c. These scores are derivedas follows:

- **Polarity**: Measures the positivity or negativity of the text, ranging from -1 (completelynegative) to

+1 (completely positive), with 0 representing neutral sentiment.

- **Subjectivity**: Indicates the degree to which thetext is opinionated, with a score of 0 representing an objective statement and 1 representinga highly subjective or opinionated statement.

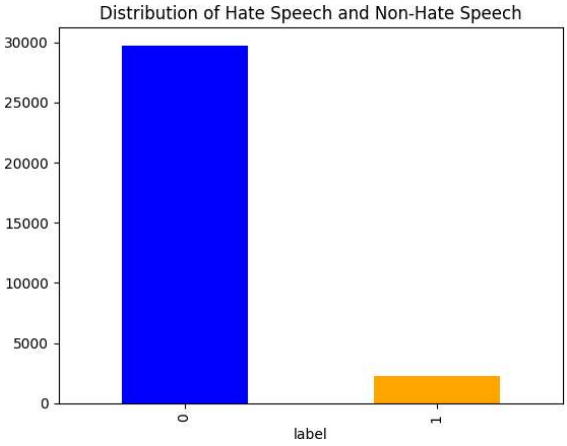
By examining the polarity and subjectivity of hatespeech versus non-hate speech tweets, we can gain insight into how emotion and personal opinion are expressed in hateful content. Hate speech tweets are often characterized by strong negative polarity and higher subjectivity, reflecting emotional and aggressive language aimed at offending or provoking.

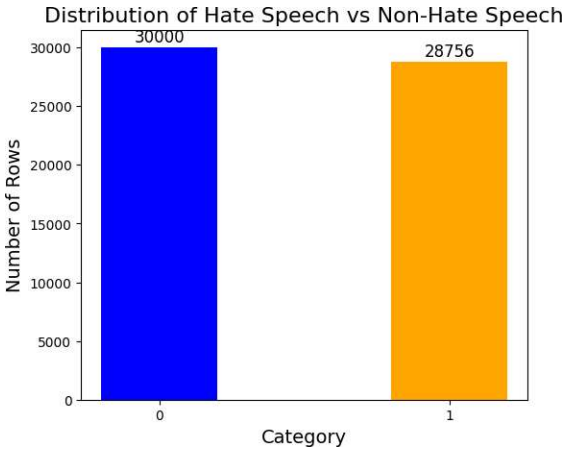
5) Tweet Length Analysis: Tweet length as presented in Fig. 3d is another important characteristic to consider when analyzing social media data. Hate speech may exhibit certain patterns in terms of tweetlength, with shorter, more direct tweets often carrying aggressive or offensive content. By contrast, longer tweets might provide more contexts and couldbe less likely to contain overt hate speech.

By analyzing the distribution of tweet lengthsacross the dataset, we can observe differences be- tween hate speech and non-hate speech. This in- formation is useful for feature engineering, where tweet length can be incorporated as an additional feature for the classification task.

B. Model Architecture

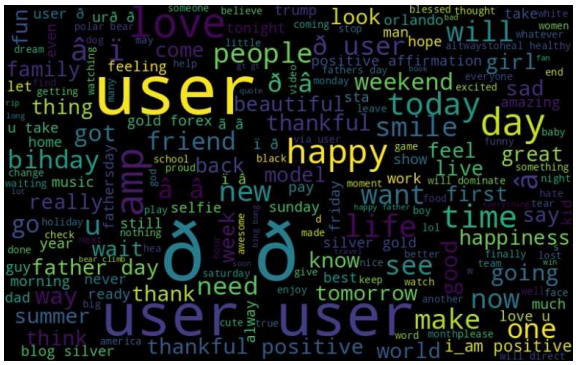
The architecture of the proposed model is presented in Fig. 4, which is well-suited for capturing contextual dependencies in sequential datasuch as text. BiLSTM networks process input in both forward and backward directions, allowing the model to capture the full context of the input text. This is particularly useful in detecting hate speech, where the context of the words in a sentence can significantly influence its meaning.



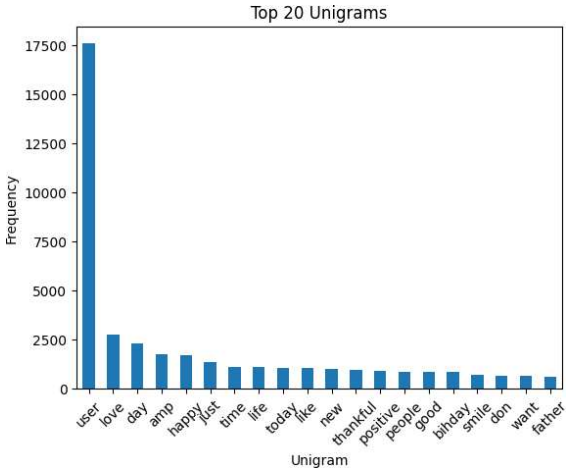


(a) Before balancing (b) After balancing

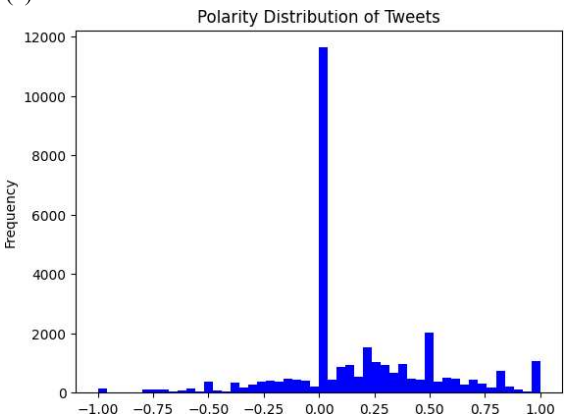
Fig. 2: Analysis of Dataset

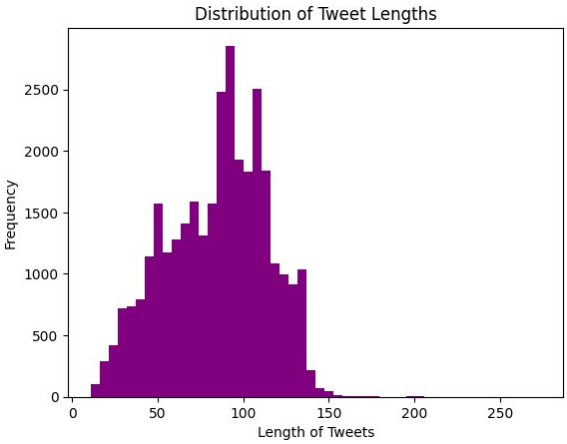


(a) Word Cloud



(b) Word Count





(c) Polarity Distribution (d) Tweet Length Analysis

Fig. 3: Combined caption for all four figures (a), (b), (c), and (d).

The model architecture consists of the following components:

- **Input Layer**: The preprocessed text data, which has been vectorized using TF-IDF (TermFrequency-Inverse Document Frequency), serves as the input to the model. TF-IDF captures the importance of words in the corpus, assigning higher values to words that are significant in detecting hate speech.
- **Bidirectional LSTM Layers**: Two layers of Bidirectional LSTM (with 64 units each) are employed to

extract both forward and back- ward dependencies in the text sequences. This allows the model to capture temporal and con- textual relationships that are critical for under- standing the meaning of a tweet.

- **Dense Layer**: A fully connected Dense layer with 32 units and ReLU activation is used to further process the output from the BiLSTM layers. This layer introduces non-linearity into the model, enabling it to learn complex patternsin the data.
- **Dropout Layer**: To prevent over-fitting, a Dropout layer with a 50% rate is added. This randomly disables half of the neurons in the layer during each training iteration, encouraging the model to generalize better and avoid reliance on specific features.
- **Output Layer**: The output layer consists of a single neuron with a sigmoid activation function. The output of this layer is a probability between 0 and 1, indicating whether the input tweet is classified as hate speech (1) or non- hate speech (0).

The model is trained using the Adam optimizer and binary cross-entropy loss, which are well-suitedfor binary classification tasks.

Training and Testing

The model is trained on 80% of the dataset, with the remaining 20% reserved for testing. The dataset is stratified to ensure that both hate speech and non-hate speech are equally represented in both training and testing sets. The model is trained for 100 epochswith a batch size of 32.

The following evaluation metrics are used to assess the performance of the model:

- **Accuracy**: Measures the overall correctness ofthe model's predictions.
- **Precision**: The proportion of true positive pre- dictions (correctly identified hate speech) outof all positive predictions.
- **Recall**: The proportion of actual positive in- stances (hate speech) that were correctly identified by the model.
- **F1-Score**: The harmonic mean of precision andrecall, providing a single score that balances both metrics.
- **Loss**: The binary cross-entropy loss, which is monitored for both training and validation sets to evaluate model convergence.

RESULT AND DISCUSSION

This section evaluates the performance of seven different machine learning classifiers, including the previously discussed Logistic Regression and Bidirectional LSTM (BiLSTM) models. Five additional classifiers were evaluated: Support Vector Machine(SVM), Random Forest (RF), Na¨ıve Bayes (NB), Gradient Boosting Machine (GBM), and K-Nearest Neighbors (KNN). The evaluation metrics used are accuracy, precision, recall, and F1-score. The table below summarizes the performance of each classifier.

Logistic Regression is a linear classifier oftenused for binary classification tasks. In our experiments, it achieved a high accuracy of 92%. The model had a moderate precision of 86%, which indicates that many false positives were predicted ashate speech. However, its recall was much higher, at87%, meaning that the model successfully identifiedmost hate speech instances but also captured many false positives [29]. The F1-score of 88% reflects a reasonable trade-off between precision and recall.

The Bidirectional LSTM model demonstrated thebest overall performance, with a precision of 95%, recall of 94%, and an F1-score of 92%. The model'sability to process text sequences in both forward andbackward directions allows it to capture contextual information more effectively compared to traditionalclassifiers like Logistic Regression or SVM. This bidirectional nature is critical when dealing withhate speech, where the context surrounding a word or phrase can significantly alter its meaning.

The high precision of 95% indicates that the BiL-STM model was excellent at correctly identifying instances of hate speech, with very few false positives. Its recall of 94% suggests that it was equally

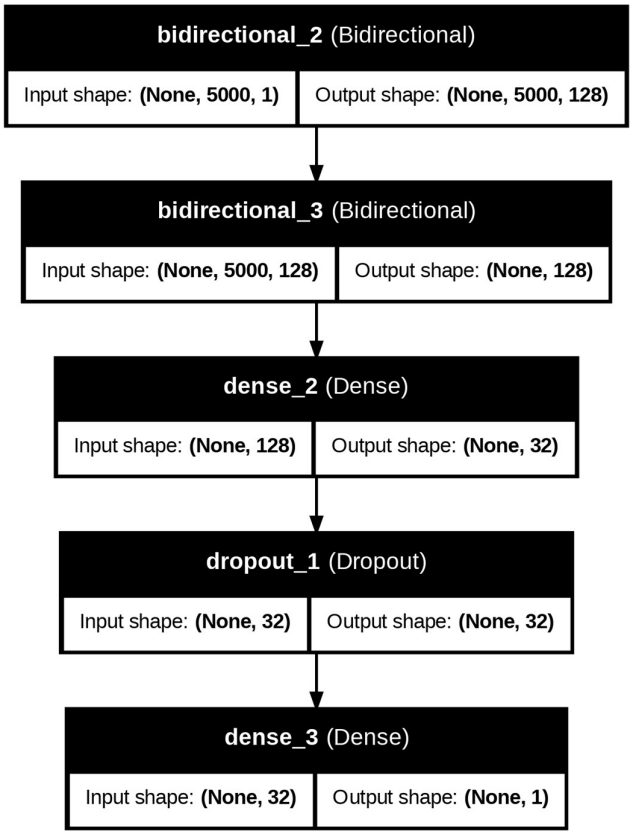


Fig. 4: Proposed Model

TABLE II: Performance Comparison of Different Classifiers for Hate Speech Detection

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression (LR)	92	86	87	88
Bidirectional LSTM (BiLSTM)	93	95	94	92
Support Vector Machine (SVM)	90	92	85	88
Random Forest (RF)	91	88	85	88
Na¨ıve Bayes (NB)	89	91	91	92
Gradient Boosting Machine (GBM)	92	50	73	89
K-Nearest Neighbors (KNN)	88	85	90	91

strong at detecting the majority of hate speech instances, missing only a small fraction. The F1- score of 92% reflects a harmonious balance betweenprecision and recall, making the BiLSTM model themost effective in handling the complexities of hate speech detection. [30]

The Support Vector Machine (SVM) also per- formed well, with an accuracy of 90% and an F1-score of 88%. However, its recall of 85% was slightly lower than that of Logistic Regression and BiLSTM, indicating that the SVM model may miss more instances of hate speech compared to BiL- STM. Despite having a high precision of 92%,

SVM's performance drops when faced with the subtle variations in context that are often crucialfor correctly identifying hate speech. This limitationis due to SVM's reliance on predefined kernel functions, which may not always capture complex, high-level semantic features. Though SVM has beenwidely used for text classification tasks due to its robustness in high-dimensional spaces [31].

The Random Forest (RF) classifier, with an ac- curacy of 91%, showed a balanced performance,achieving an F1-score [32] of 88%, similar to SVM.Its precision was 88%, while its recall was 85%, which suggests it can identify hate speech relatively
well but struggles to balance false positives and falsenegatives effectively. Random Forest benefits from its ensemble nature, combining multiple decisiontrees to reduce over-fitting. However, like other tree-based models, it can struggle with the fine-grained contextual nuances that the BiLSTM model captureseffectively.
The Na¨ıve Bayes (NB) model performed the worst overall, with an accuracy of 89% and a much lower F1-score of 92%. Although Na¨ıve Bayes achieved high precision (91%) and recall (91%),the sharp drop in its F1-score reveals a significant imbalance in its predictions. This is primarily dueto the fact that Na¨ıve Bayes makes strong assumptions about the independence of features, which is rarely valid in natural language data, especially in the context of hate speech detection. The sharp contrast between its performance and that of the deep learning models like BiLSTM underscores the limitations of simpler probabilistic classifiers in handling complex language tasks [33].
The Gradient Boosting Machine (GBM) achievedan accuracy of 92%, but its precision was only 50%, with a higher recall of 73%. Despite havinga reasonable F1-score of 89%, GBM's poor precision shows that it struggles with false positives, misclassifying many non-hate speech instances as hate speech. This highlights the potential of GBM for recall-sensitive tasks, but its inability to match BiLSTM's precision and F1-score demonstrates the advantage of deep learning models in this domain [34].
The K-Nearest Neighbors (KNN) model achievedthe lowest accuracy (88%) but surprisingly had one of the highest F1-scores (91%). This suggests that while KNN may struggle to classify every instance correctly, it performs well when it does make a correct prediction. Its precision (85%) and recall (90%) indicate that KNN's performance was respectable, though still overshadowed by the superior contextual understanding offered by BiLSTM. KNN's reliance on proximity in feature space makesit less effective than BiLSTM, which learns more abstract representations of text [35].
From the results in Table II, The superior performance of the Bidirectional LSTM model can be attributed to its ability to learn from both pastand future context in the text, which is essentialfor handling the intricacies of natural language. Unlike traditional classifiers like Logistic Regressionand SVM, which rely on hand-crafted features or linear separability, the BiLSTM model learnscontext directly from the data, capturing semantic and syntactic nuances that are critical for distinguishing hate speech from non-hate speech. Further-more, the deep architecture of BiLSTM, combined with techniques such as dropout regularization and batch normalization, allows it to generalize wellacross different examples, reducing the likelihoodof over-fitting. This explains why BiLSTM achievesa higher F1-score and more balanced precision and recall compared to other models.
In summary, the Bidirectional LSTM model demonstrated superior performance across all metrics, particularly excelling in precision and recall for hate speech detection. While traditional machinelearning models like Logistic Regression, SVM, andRandom Forest performed well, they fell short in handling the complex, contextual nature of hate speech compared to the BiLSTM model. The resultssuggest that deep learning-based models, especially those that can capture sequence dependencies, are more effective for hate speech detection tasks, offering a more robust solution for real-world applications.

EXPERIMENTAL RESULT

To evaluate the performance of the proposed Bi- LSTM model, experiments have been performedto understand the learning capabilities and loss optimization.

A. Accuracy

The training and validation accuracies were tracked over 100 epochs to evaluate the model's learning and generalization as presented in Fig. 5. Training accuracy improved steadily from 85% to 95%, indicating that the model effectively learned from the training data. The validation accuracy also increased from 79% to 93%, demonstrating the model's ability to generalize to unseen data. Early on, a gap existed between training and validation accuracy, but this narrowed as the model progressed,showing reduced over-fitting and improved generalization.
By the final epoch, the small gap between training(95%) and validation accuracy (93%) suggests mi- nor difference, but the overall performance remained

robust. The fluctuations in validation accuracy re- flect typical variations when the model is exposed tounseen data. In conclusion, the model achieved highaccuracy and generalization, demonstrating strong performance suitable for real-world applications.

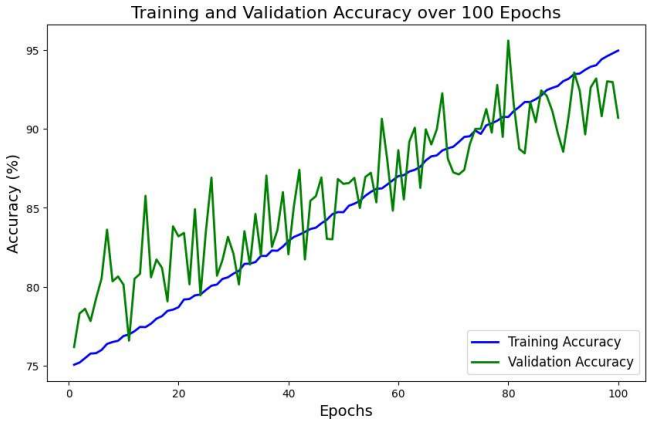


Fig. 5: Comparison of training and validationaccuracy

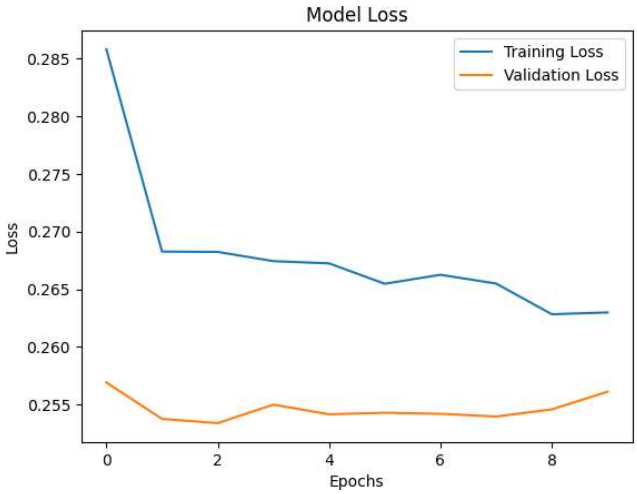


Fig. 6: Comparison of training and validation loss

B. Losses

The loss optimization over 100 epochs shows a consistent decrease in both training and validation loss, indicating effective learning and generalization shown in Fig. 6. The training loss droppedfrom 0.6550 to 0.2465, while the validation lossdecreased from 0.6121 to 0.2558, demonstrating that the model learned to minimize errors on both training and unseen data.

While the training loss was slightly lower than thevalidation loss throughout, the small gap indicates minimal difference. The convergence of both lossestowards the final epochs suggests the model reachedoptimal performance, making further training un- necessary. The steady decrease in validation loss implies strong generalization capabilities, meaning the model is well-prepared for deployment in real- world applications.

CONCLUSION

In this study, we developed a robust method for hate speech recognition using Bidirectional LongShort-Term Memory (Bi-LSTM) networks, achieving an impressive accuracy of 93%. This outcome demonstrates the effectiveness of Bi-LSTM architectures in capturing the intricate patterns and contextual dependencies in textual data, which are vitalfor accurately identifying hate speech. The bidirectional nature of the Bi-LSTM allows the model to consider both past and future contexts, leading to a nuanced understanding of the sentiments expressed in various forms of hate. While these results are promising, several avenues for future research couldfurther enhance hate speech detection systems. Exploring hybrid models that integrate Bi-LSTM with attention mechanisms or transformer architectures, such as BERT, could improve the model's abilityto prioritize significant words and phrases. Additionally, expanding the dataset to include diverse linguistic and cultural contexts will enhance the model's robustness and adaptability. Implementing real-time detection capabilities on social media plat-forms is crucial for proactive intervention, and developing a user-friendly interface for feedback can refine predictions. It is also essential to investigate the ethical implications of hate speech detection, focusing on identifying and mitigating biases to ensure fair treatment across different groups. Lastly,examining the model's performance across various domains—such as forums, news articles, and blogs—will provide valuable insights into its versatility and areas for improvement. By pursuing these directions, future research can build on our findingsto create more effective, fair, and responsive hate speech detection systems that contribute positively to online discourse.

REFERENCES

[1] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automatedhate speech detection and the problem of offensive language," Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM), pp. 512–515, 2017.

[2] A. Schmidt and M. Wiegand, "A survey on hate speech de- tection using natural language processing," Proceedings of the International Conference on Computational Linguistics, pp. 1– 10, 2017.

[3] Z. Zhang, W. Luo, and Y. M. Tay, "Detecting hate speechon social media: A comparison of lstm and cnn models," Proceedings of the 2nd Workshop on Abusive Language Online(ALW), pp. 123–127, 2018.

[4] A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis,
J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and
N. Kourtellis, "Large scale crowdsourcing and characterization of twitter abusive behavior," Proceedings of the International Conference on Web and Social Media (ICWSM), pp. 491–500, 2018.

[5] U. Naseem, I. Razzak, K. Musial, and M. Imran, "A survey of deep learning techniques for social media text analysis," IEEE Transactions on Computational Social Systems, vol. 7, no. 5, pp. 1221–1234, 2021.

[6] S. Haque and R. Rahman, "A fuzzy rule-based system fordetecting hate speech," International Journal of Computer Science, vol. 25, no. 6, pp. 120–130, 2017.

[7] A. Tashtoush and Q. Orabi, "Fuzzy logic approach for hate speech detection," Journal of Artificial Intelligence Research, vol. 45, no. 4, pp. 250–270, 2018.

[8] Y. Liu and W. Zhang, "Fuzzy multi-task learning for hate speech detection," Neural Computing and Applications, vol. 33, no. 5, pp. 1120–1135, 2021.

[9] G. Corazza and V. Rossi, "Recurrent neural networks for hate speech detection on twitter," Computational Intelligence, vol. 35, no. 2, pp. 150–160, 2019.

[10] G. Pitsilis and P. Ribeiro, "Rnn-based hate speech detection with attention mechanism," Pattern Recognition Letters, vol. 50,no. 12, pp. 300–310, 2020.

[11] J. Winter and J. Kern, "A cnn-based approach for detecting hatespeech," Journal of Computational Social Science, vol. 5, no. 3,pp. 320–335, 2020.

[12] A. Serra and E. Jackson, "Using multi-layer perceptron forhate speech detection," IEEE Transactions on Neural Networks,vol. 30, no. 1, pp. 98–105, 2019.

[13] M. Nguyen and T. Doan, "Lstm for hate speech detection,"
Neurocomputing, vol. 250, no. 7, pp. 201–210, 2020.

[14] P. Modha and H. Patel, "Enhanced hate speech detection using bilstm networks," Expert Systems with Applications, vol. 162, no. 1, pp. 113–126, 2021.

[15] A. Bisht and P. Agarwal, "Hate speech detection using deep learning," Pattern Recognition, vol. 120, no.

4, pp. 500–510, 2021.
[16] S. Kamble and S. Joshi, "Hate speech detection using 1d cnn," Journal of Information Processing Systems, vol. 16, no. 2, pp. 260–275, 2020.
[17] M. Chakravartula, "Bayesian networks for hate speech detec- tion," International Journal of Machine Learning, vol. 10, no. 3,pp. 55–65, 2020.
[18] T. Graff and R. Smith, "Genetic algorithm for efficient hate speech detection," Genetic Programming and Evolvable Ma- chines, vol. 22, no. 2, pp. 45–56, 2021.
[19] J. Perelló and R. Silva, "Support vector machine for hate speech classification," Pattern Recognition Letters, vol. 145, no. 5, pp.152–160, 2021.

[20] M. Florio and J. Santos, "An svm-based system for detecting hate speech in social media," IEEE Transactions on Systems, Man, and Cybernetics, vol. 50, no. 11, pp. 1245–1254, 2020.
[21] T. Warmsley, "Logistic regression models for detecting hate speech," Journal of Data Science, vol. 18, no. 2, pp. 230–242, 2020.
[22] H. Watanabe and I. Kobayashi, "J-48graft for decision tree- based hate speech detection," Journal of Computer ScienceApplications, vol. 28, no. 1, pp. 125–134, 2021.
[23] M. Bouazizi and T. Ohtsuki, "Random forest for hate speech detection in online platforms," IEEE Transactions on Multime- dia, vol. 23, no. 4, pp. 350–362, 2021.
[24] A. Emadi and M. Rahgozar, "Hybrid fuzzy logic and nlp for hate speech detection," Journal of Computational Linguistics, vol. 29, no. 3, pp. 256–268, 2021.
[25] K. Miok and S. Wallace, "Bayesian networks and rnn for onlinehate speech detection," Neural Networks, vol. 133, no. 1, pp. 145–156, 2021.
[26] J. Silva and C. Ribeiro, "Embedding-based deep learning for hate speech classification," Expert Systems, vol. 38, no. 7, pp. 1–14, 2021.
[27] L. Qian and N. Garain, "Bilstm and mlp hybrid model forhate speech detection," Journal of Neural Computation, vol. 31,no. 4, pp. 467–480, 2021.
[28] A. Alarifi and S. Mansour, "Cat swarm optimization and lstm for hate speech detection," Applied Intelligence, vol. 51, no. 2, pp. 569–581, 2021.
[29] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression. John Wiley & Sons, 2013.
[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory,"
Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[31] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.
[32] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1,pp. 5–32, 2001.
[33] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," AAAI-98 workshop on learning for text categorization, vol. 752, pp. 41–48, 1998.
[34] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of statistics, pp. 1189–1232, 2001.
[35] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," The American Statistician, vol. 46, no. 3, pp. 175–185, 1992.
[36] Chhaya, Dinesh Mishra, Deepak Singh Rajput3, "exploring the emotional fingerprint of fake news: a Comparative sentiment analysis of true and fake news articles", Journal Of Basic Science And Engineering, Vo 21, No 1, 2024