An Efficient Hate Speech Detection Method Using an Optimized LSTM Based on a Hybrid Efficient Document Representation Method

K. Senthil Kumar[1], S. Sathiyabama[2], D. Ayyamuthukumar[3]

[1]Department of Computer Science, Thiruvalluvar Government Arts College, Rasipuram, Namakkal, Tamilnadu, India
Email: *senthilkumar.kasi[at]gmail.com*

[2]Department of Computer Science, Government Arts and Science College, Kangeyam, Tamil Nadu, India
Email: *drsathyaabama[at]gmail.com*

[3]Department of Artificial Intelligence and Data Science, Muthayammal College of Engineering, Rasipuram, Tamilnadu India
Email: *ukdkumar[at]gmail.com*

Abstract: As the popularity of social media platforms surges, the problem of hate speech has turned out to be a serious issue, endangering the safety of the world online, social cohesion, and the well-being of individuals. Hate speech is difficult to detect on such platforms as Twitter because the data is very short, informal, and noisy, usually containing slang and abbreviations, emojis, and vague phrases. Simple neural networks and traditional machine learning techniques are not particularly effective in learning highly complex semantic and contextual patterns needed to reach high-quality classification. An alternative to these difficulties is that this study suggests a hybrid model of Elman Recurrent Neural Network (ERNN) that is optimized by the Local search algorithm (LSA) and Elephant Herding Optimization (EHO). The model uses the best document representation methods, such as Doc2Vec, TF-IDF, BERT, and a hybrid of TF-IDF + BERT representation, to identify rich semantic, syntactic, and contextual features of tweets. Experiments on benchmark Twitter datasets have shown that, compared to ERNN as a baseline and other ERNN variants that are metaheuristically optimized, the proposed ERNN-LSA-EHO framework has better accuracy, precision, recall, and F1-score. The hybrid optimization and sophisticated feature representations contribute to effective convergence, robust learning, and the ability to detect very subtle and complex patterns of hate speech. The results of the research demonstrate the possibility of the given method to monitor social media in real time, moderate content, and perform sentiment analysis.

Keywords: Hate Speech Detection, Elman Recurrent Neural Network (ERNN), Local Search Algorithm (LSA), Elephant Herding Optimization (EHO), Document representation, Social Media Analysis.

1. Introduction

Over the past years, unexpectedly, the swift growth in social media i.e. web-based communication platforms, especially Twitter has reshaped the manner in which people communicate, give their views, and disseminate information. Although the opportunities associated with these platforms are enormous in terms of socialization and dissemination of information, they are also conducive to the spread of negative content, such as hate speech, obscene language, and fake news [1]. On social media, hate speech has become one of the topical issues because it may provoke violence, reinforce discrimination, and harm mental health. The solutions to this problem are therefore automated hate speech detection and classification, which are essential to ensuring an inclusive and safe internet community [2].

Conventional methods of text classification pay much attention to the statistical frequency of words, but cannot reflect the semantics and context of words. This makes it difficult to properly detect hate speech, particularly in informal, brief, and noisy texts that are common to Twitter data[3, 4]. Deep learning methods have been greatly embraced in order to overcome these challenges because they have the capabilities to find intricate patterns and contextual features in text data. Of these, Recurrent Neural Networks (RNNs), and specifically the Elman Recurrent Neural Network (ERNN), have demonstrated potential in sequential data processing, allowing the model to provide a record of temporal dependencies and flow of textual information[5].

Although ERNNs have the advantage, the efficiency of the networks is strongly dependent on the correct tuning of the network parameters, and it might be difficult because of the high dimensionality and sparsity of textual features. The metaheuristic optimization algorithms have been combined with deep learning models in order to improve the learning capacity and convergence of the model. The hybrid of the Local Search Algorithm (LSA) and the elephant herding optimization (EHO) is one of such hybrid methods that enable the model to explore and exploit the solution space effectively, resulting in the optimization of the network weights as well as better classification. Moreover, the document representation is crucial to the efficiency of hate speech detection. Semantic, syntactic, and contextual features of the text are represented with Doc2Vec, TF-IDF, and contextual embeddings like BERT, and the combination of these methods results in supplementary features that are highly effective in making the models more accurate and robust. Based on these findings, this paper presents a hybrid ERNN-LSA-EHO hate speech detector on Twitter data and uses modern document representation methods, Doc2Vec, TF-IDF, BERT, and their hybrids. This proposed model will also focus on high classification performance through powerful feature extraction algorithms and optimized network learning, and overcome difficulties with noisy data, short text length, and complicated linguistic patterns.

Paper ID: SR251215115834 DOI: https://dx.doi.org/10.21275/SR251215115834

Extensive experimental evidence shows that the proposed method is superior to the existing models based on baseline and metaheuristic, which points to its possible application in the real world with regard to social media monitoring, online content moderation, and automated sentiment analysis.

The ultimate goal of the study is to create an effective and strong system of hate speech and affective identification of social media in specific. This paper aims to improve the way ERNN performs by combining the best optimization methods, i.e., LSA and EHO, to perform better on the weight tuning and quicker convergence. The study is also going to examine how different document representation methods, such as Doc2Vec, TF-IDF, BERT, and combinations of these methods, determine the semantic, syntactic, and contextual information of short, noisy, and informal text data. The overall objective is to deliver a high-performing model that will effectively classify hate speech and be able to generalize with various Twitter data. This study contributes to automated hate speech and sentiment detection in the following ways:

- It suggests a new framework of combining ERNN with the LSA and EHO. This integration method is more effective in optimizing the network's weight and boosts the rate of convergence and the overall accuracy of the classifications in comparison to the classical ERNN and other variants based on the metaheuristic methodology.
- The analysis compares various feature extraction algorithms, such as Doc2Vec, TF-IDF, BERT, and a composite TF-IDF + BERT representation. As the analysis has shown, the statistical and contextual features can be combined to increase the capacity of the model to pick the subtle and intricate patterns of hate speech in Twitter data.
- The given model is tested with benchmark Twitter datasets of various natures. Accuracy, precision, recall, and F1-score are considered as performance measures and demonstrate high results relative to the baseline models and other metaheuristic-optimized ERNN models.
- The proposed approach itself offers a useful resource in real-time social media monitoring, automated content moderation, and sentiment analysis, which can be used to ameliorate the adverse impacts of online hate speech because the proposed approach has a high performance in terms of classification and robustness.

The remainder of this paper will be structured as follows. Section 2 presents an extensive overview of related literature and outlines the current methods of hate speech and sentiment identification, as well as gaps in research found in this paper. Section 3 outlines the research methodology, which involves a thorough description of the document representation method, including Doc2Vec, TF-IDF, BERT, and their combination, and the proposed method of detection, ERNN optimized using LSA-EHO. Section 4 has the experimental results, the description of the datasets, parameter settings, and the critical analysis of the performance results with the existing methods. Lastly, Section 5 discusses the conclusion on the main findings, the limitations of the suggested approach, and the pathways to future improvements and research on the topic of hate speech detection and sentiment analysis.

2. Related works

Detecting hate speech has attracted considerable research focus in different languages and social media networks, especially with the eruption of user-created content. S. S. Roy et al. [6] suggested an LSTM-based hate speech detection system with the TF-IDF vectorization and proved that LSTM demonstrated better results in comparison to the traditional machine-learning models in the ability to differentiate between hate and non-hate tweets and non-tweets of real-time Twitter feeds. In the same fashion, G. O. Ganfure et al. (2022) [7] examined deep learning methods on Afaan Oromo hate speech recognition. They compared various deep learning architectures with the use of CNN, LSTM, Bi-LSTM, GRU, and hybrid models, and found that the CNNBi-LSTM combination provides the best results, as it highlights the value of hybrid sequential spatial feature learning. In their study of detecting coded hate speech, H. Saleh et al. (2023) [8]examined domain-specific word embeddings and Bi-LSTM models, and subsequently tested BERT as a transfer-learning technique. Their findings showed that large-scale pretraining resulted in BERT performing better, but domain-specific embeddings successfully identified hidden and contextual hate expressions. S. Kothuru et al. (2022) [9] In another attempt to use deep learning suggested a Softplus-Bi-LSTM architecture was suggested, which effectively learns complicated linguistic patterns and enhances the accuracy of abusive speech classification. Going further than just the simple architectures, A. Verma et al. (2023) [10] designed a deep learning model to categorize posts in Twitter as either hate or non-hate posts without much feature engineering.

The system, when end-to-end neural models are used, is able to produce an F1-score of more than 90% using a dataset of more than 27,000 tweets, demonstrating the superiority of end-to-end neural models. Other papers were in multilingual or low-resource languages. R. The development by Ali et al. (2022) [11] constructed an Urdu hate dictionary and constructed a collection of more than 10,000 labeled tweets. They reported good F1-scores of about 0.68 -0.69 in multiclass classification with their assessments based on transfer-learning models like FastText, multilingual BERT, and XLM-RoBERTa. Arbaatun et al. [12] study saw the investigation of hate speech detection in Indonesian, in which LSTM models with FastText and GloVe embeddings were compared. The LSTM-FastText model had the best F1-score of 89.91, which is more than the other configurations, because it can encode subword-level semantic details. The recent research has also covered the topic of hybrid architecture in enhancing contextual learning. S. S. According to Ilhan et al. (2024) [13], Bi-LSTM outperformed Bi-LSTM-GRU combinations, and the hybrid model was able to achieve an accuracy of 95, which was much better than that of Bi-LSTM alone. Likewise, S. Shubanging et al. (2023) [14]developed a Bi-GRU-LSTM-CNN model in detecting hate speech with an accuracy of 77.16 percent and a demonstration of the power of the convolutional and recurrent layers in capturing various linguistic patterns. C. N. Vo et al. (2024) [15] proposed ViTHSD, a targeted hate speech dataset in Vietnamese, and created a model of Bi-GRU-LSTM-CNN with BERT embeddings. Their work offered a feasible combination technique of real-time hate speech detectors. Additional works are the detection of toxic text in Algerian dialects by A. C. Mazari et al. (2023) [16], who developed a 14,150-

comment dataset and tested several ML and DL models. The Bi-GRU model got the best F1-score (75.8%), indicating the efficiency of bidirectional recurrent units. A. Reghunathan et al. (2024) [17] additionally conducted a study on the different machine-learning and deep-learning models used to detect hate speech, and they discovered that the Bi-GRU model performed consistently better than other models, which was seen to be very effective in various types of data. K. K. Mohbey et al. (2025) [18] studied the research on gender-specific abuse through deep learning to identify hate speech targeted at women on Twitter. Their results highlighted the need to understand this in context and the significance of neural models in facilitating digital safety programs.

In one more language-specific paper, A. Naseeb et al. (2025) [19] suggested an Arabic-script instrument that could be used to detect hate speech in Roman Urdu, utilizing six ML models and four DL architectures. CNN and LSTM obtained the accuracy of 95.1 and 96.2, respectively, which was much higher than the traditional ML methods. The detection of Arabic hate speech was also developed by K. Salameh et al. (2025) [20], who compared three embedding methods, ArabGlossBERT, AraBERT, and AraVec. Their results indicated that ArabGlossBERT, which was trained with LSTM and CNN models, was the most effective model in terms of classification performance. A. Varathararaj et al. (2025) [21] Found a solution to advanced English hate speech detection by implementing transformer-based models with hyperparameter optimization, which they trained with competitive results using CNN and LSTM, along with advanced preprocessing strategies. Recent studies have moved towards large-scale multilingual and BERT-based models.

R. A. A comparative study between various BERT versions, such as BERT-CNN, BERT-LSTM, and BERT-BiLSTM by Saputra et al. (2025) [22] demonstrated that BERT-BiLSTM was the most accurate model (82 percent) in longer texts. A. Al-Mashhadani et al. (2025) [23] integrated AraBERT and AraVec embeddings with LSTM and Multi-Head Attention layers and obtained accuracies of over 91 percent on the Hate dataset. D. Singh et al. (2025) [24] confirmed a hybrid CNN-RNN model on hate speech in 11 languages and achieved 98 percent accuracy, which is in line with high multilingual generalization properties. Last but not least, multilingual pre-trained models like LASER and ELMo give a promising result according to A. K. Srivastava et al. (2025) [25] and M. N. Hasan et al. (2025) [26], Attention-based deep learning models are more effective than CNNs and LSTMs in different multilingual datasets, according to T. Z. Jilan (2025) [27].

Table 1: Comparative analysis of existing work

Ref. No.	Method	Dataset	Merirs	Demerits
[6]	LSTM + TF-IDF	Real-time Twitter stream	LSTM outperformed ML models for hate vs. non-hate classification	Limited to English; TF-IDF lacks semantic representation
[7]	CNN, LSTM, Bi-LSTM, GRU, CNN-BiLSTM	Afaan Oromo hate speech dataset	CNN-BiLSTM achieved the best accuracy	Low-resource language; dataset domain-specific
[8]	Domain-specific embeddings + Bi-LSTM; BERT	Hate speech dataset (English)	Domain embeddings identified coded hate; BERT gave best results	BERT requires large data & high computation
[9]	Softplus Bi-LSTM	Network comment dataset	Improved abusive speech classification	Limited testing; no multilingual evaluation
[10]	Deep learning (LSTM-based)	Twitter dataset (27K tweets)	F1-score > 90%	Informal language errors still cause misclassification
[11]	FastText, mBERT, XLM-R	10,526 Urdu tweets	BERT variants achieved F1 ≈ 0.68–0.69	Multiclass problem remains challenging; moderate accuracy
[12]	LSTM + FastText / GloVe	Indonesian Twitter dataset	LSTM-FastText F1 = 89.91%	An imbalanced dataset affects performance
[13]	Bi-LSTM vs. Bi-LSTM + GRU	Hate speech dataset	Bi-LSTM-GRU achieved 95% accuracy	High computation; limited generalization
[14]	Bi-GRU-LSTM-CNN	English hate speech	Accuracy: 77.16%	Still many misclassifications; model complexity is high
[15]	Bi-GRU-LSTM-CNN + BERT embeddings	ViTHSD Vietnamese dataset	Strong performance; a practical deployment method proposed	Moderate inter-annotator agreement (0.45)
[16]	ML and DL	Algerian dialect toxic dataset (14,150 comments)	Bi-GRU: Accuracy 73.6%, F1 = 75.8%	Low-resource dialect; moderate accuracy
[17]	ML + Deep Learning (Bi-GRU strongest)	Cyberbullying/hate datasets	Bi-GRU outperformed all models	High model complexity; dataset-dependent
[18]	Deep learning (custom architecture)	Hate speech against women (Twitter)	Effective for gender-specific detection	Domain-limited; lacks cross-lingual testing
[19]	ML (LR, SVM, RF, NB, KNN, GBM) + DL (CNN, RNN, LSTM, GRU)	Roman Urdu Facebook comments	CNN accuracy = 95.1%, LSTM = 96.2%	Roman Urdu lacks standardized spelling
[20]	ArabGlossBERT, AraBERT, AraVec + LSTM/CNN	Arabic hate speech (Twitter)	ArabGlossBERT achieved highest accuracy	BERT-based models computationally intensive
[21]	Transformer-based DL + CNN & LSTM	English social media text	Baseline accuracy = 93% (Gradient Boosting)	Traditional models struggle with evolving language
[22]	BERT, BERT-CNN, BERT-LSTM, BERT-BiLSTM	Multi-label hate speech dataset	BERT-BiLSTM: Best accuracy (82%) on long texts	Lower performance on short texts; high computation
[23]	AraBERT + AraVec + LSTM / Attention	arHate Arabic dataset	LSTM = 91.6%, Attention = 91.4%; new dataset: 95%+	Focused only on Arabic; high model complexity

[24]	Hybrid CNN-RNN model	11-language multilingual dataset	98% accuracy across languages	Resource-intensive; full reproducibility unclear
[25]	LASER, ELMo + ML/DL	Multilingual dataset (11 languages)	Strong cross-lingual performance	LASER embedding is limited in capturing deep semantics
[26]	ML models: LR, RF, SVM, NB, XGBoost	21,010 Twitter posts	Reliable classification of hate/offensive content	ML models underperform compared to DL for complex syntax
[27]	Attention-based DL, CNN, LSTM	Social media posts	Attention-based models outperformed CNN & LSTM	Expensive training; requires large labeled data

There are various gaps in the current studies, although much has been done in regard to detecting hate speech. A lot of models have difficulty dealing with noisy and unstructured Twitter data, and traditional or static embeddings are often not able to reflect the contextual and subtle hate utterances. Deep architectures like LSTM, CNN, and GRU are also characterized by high computational complexity and therefore cannot be used in real time. Furthermore, the majority of works are based on the manual choice of hyperparameters, which leads to the instability of the performance and not the consistency of performance across the datasets. Few studies investigate optimization algorithms to optimize the model learning. It is also still difficult to detect multilingual, code-mixed, and sarcastic hate speech, as well as high-dimensional feature representations to reduce the chances of overfitting. Also, most of the existing models cannot be optimized to be deployed on resource-constrained environments. The following gaps point to the necessity of a lightweight but powerful solution. The proposed optimized ERNN with the LSA-EHO approach will solve these problems by automating the hyperparameter optimization, minimizing the computational load, facilitating the generalization, and dramatically boosting the detection rate of noisy Twitter data. Table 1 shows the comparisons of state-of-the-art methods.

3. Research methods

The following subsections discuss the research methods that are used in this research work.

3.1 Elman recurrent neural network

Elman (1990) introduced the ERNN [28] common method, one kind of feedback neural network, the ERNN, acquires a recurrent layer developed on the basis of the hidden layer. This layer introduces a memory capability, and it is a delay operator. It maintains the stability of the network on a world scale and makes it respond to the variations in time that are dynamic in nature. Topological features of the ERNN model typically have four layers: The hidden layer is fed with the information of the input layer, whose neurons are typically linear, and then employs an activation function to modify or amplify this information. Given the input of the hidden layer, the task of the connecting layer is to provide the response of the same instance as the previous one to form a local ring structure. Since the deferred memory effect of the concerned layer in the features contained in the past data, the final value of the neural network is influenced more by the real data. The results are finally taken out as the output. ERNN is trained with the help of BPNN, but it connects the output of the hidden layer and its input automatically, depending on the delay and storage capabilities of the context layer. The capability of dynamic input can be enhanced by an internal feedback mechanism. ERNN is gathered based on an input, a hidden, an output, and a recurrent layer. The sample or data of each layer is transferred by one or several neurons through a nonlinear function of the weighted sum of the input samples.

$$X_{it}(k) = \sum_{i=1}^{n} X_{it}(k-1) \tag{1}$$

In this case, X_{it} - represents an input at time t and the number of neurons n. The input of hidden neurons is as follows:

$$net_{jt}(k) = \sum_{i=1}^{n} W_{ij} X_{it}(k-1) + \sum_{j=1}^{p} C_j r_{jt}(k) \tag{2}$$

W_{ij}- the weights of the input and hidden layer. C_j - weight between recurrent and hidden layers. The hidden layer's output is defined as follows:

$$Z_{jt}(k) = f(net_{jk}(k) = \sum_{i=1}^{n} W_{ij} X_{it}(k-) + \sum_{j=1}^{p} C_j R_{jt}(k) \tag{3}$$

The recurrent layer is determined as follows:

$$R_{jt}(k) = Z_{jt}(k-1) \tag{4}$$

The output layer is defined as follows,

$$Y_t(k) = f(\sum_{j=1}^{p} V_j Z_{jt}(k) \tag{5}$$

The errors of the network are designed as follows,

$$E = \sum_{k=1}^{m} (t_t - y_t)^2 \tag{6}$$

Here, t_t – target value and y_t – predicted value.

3.2 Elephant herding optimization (EHO)

The EHO approach for swarm intelligence optimization was introduced by Wang et al. in 2015 [29]. The separation and the clan update operator are the two primary operators of EHO. The population of elephants is divided into multiple clans, each of which, c_i, has a set number of elephants. The matriarch (c_i) influences the elephant's new position. The definition of the elephant is as follows:

$$x_{new,ci,j} = x_{ci,j} + a \times \left(x_{best,ci} - x_{ci,j}\right) \times r \tag{7}$$

Where, $x_{new,ci,j}$- new position, $x_{ci,j}$- old position, $x_{best,ci} - x_{ci}$ is the matriarch. $r \in [0,1]$, $a \in [0,1]$. The largest elephant can be determined by the following:

$$x_{new,ci,j} = \beta \times x_{center,ci} \tag{8}$$

$$x_{center,ci,d} = \frac{1}{n_{ci}} \times \sum_{j=1}^{n_{ci}} x_{ci,j,d} \tag{9}$$

Where $1 \leq d \leq D$, $n_{ci,j}$is the number of elephants, $x_{ci,j,d}$is d^{th}-dimensions of elephant, $x_{center,ci,d}$ - the center of the clan. In the context of solving optimization problems, the process by which male elephants split out from their family group can be described as a separation operator. The separation operator is applied by the elephant individual in each generation that is least suited, as shown.

$$x_{worst,ci} = x(xmin_{max} + 1.rand)_{min} \quad (10)$$

Where, x_{max} is the upper bound of the individual, x_{min} is the lower bound, $x_{worst,ci}$ is the worst individual, $rand$ - stochastic distribution between 0 and 1. Elephant herding behavior serves as the inspiration for the metaheuristic optimization method. It is a member of the class of nature-inspired optimization algorithms, which use difficult optimization problems to solve by imitating the behavior of plants, animals, or other natural occurrences. It is modeled after how herds of elephants travel, with the more powerful elephants leading the weaker ones to better grazing areas. By taking into account individual solutions and their interactions within a population, this behavior is transferred into the optimization method. EHO balances exploitation and exploration by employing both local and global search strategies. While memory aids in exploitation by directing the search toward the most well-known solutions, herding behavior assures exploration toward favorable locations.

3.3 Local search algorithm (LSA)

LSA is an optimization method, which works on the assumption of searching a subset of a solution space until it is found that near-optimal solutions to complex optimization problems are found [30]. LSA, in contrast to global optimization algorithms, attempts to optimize one single candidate solution by successively searching the search space, instead of exploring an entire search space. The idea behind this process of local improvement is to keep repeating it until a termination condition, such as no more improvement of the objective value or a maximum number of iterations, is met. LSA is strong because it is able to do fine-tuning of solutions obtained using global optimization techniques. It has the potential to greatly improve the quality of the solution by conducting local searches in promising areas identified by global algorithms. This renders LSA especially useful in those cases when the space of solutions is large and complicated, and cannot be explored exhaustively. The process of local exploitation ensures that convergence is more precise and that fewer chances of poor stagnation occur through LSA.

3.4 LSA-EHO Algorithm

It is the hybrid optimization model that has been developed as a result of integrating the EHO algorithm with the LSA optimizations, which effectively leverages the benefits of the global exploration and the local exploitation processes. EHO has much in common with the extensive exploration of the search space to prevent premature convergence by population diversity. LSA, on the other is supposed to reduce the locally optimum solutions to increase the accuracy and quality of the answer. The joint search allows a more balanced search strategy, leading to the improvement of the convergence rate and the overall robustness of the joint search when used in a diversity of optimization problems.

The proposed hybrid model implies that LSA is invoked following each EHO run to perform small process changes on the current best solution found to date. This refinement step involves having a temporary variable ($temp$) which contains the current best solution discovered by EHO and is then passed over to LSA to get a better solution. LSA maximizes temp via a sequence of chance selection and evasion of three features, whereby each step makes a change in the feature as per its predefined parameters. The fitness of the new candidate solution is estimated upon each change. When the fitness value is higher than that of the previous solution, temp is changed; otherwise, temp remains unchanged. It is a procedure involving a cyclic process to ensure that the search proceeds along optimum areas of the solution space or near-optimal areas at the expense of computational efficiency. This implies that the hybrid EHOLSA algorithm is superior regarding the accuracy of the solution, rate of convergence, and stability; therefore, it is most suitable in the case of complex and high-dimensional optimization.

3.5 Proposed optimized ERNN based on LSA-EHO

The current study aims to design an optimal ERNN model as a hybrid model of LSA-EHO to detect hate speech in Twitter data. The LSA-EHO hybrid methodology is suggested to optimize the weights and biases of the ERNN in order to increase the accuracy of detection, convergence speed, and avoid being trapped in local optima. In this model, every member of the herd of elephants is a candidate solution, which is associated with a given set of hyperparameters of ERNN. The main aim of the optimization is to reduce the gap between the actual and the predicted values of the ERNN model. MSE is considered the fitness, and it is mathematically defined as:

$$MSE = \frac{1}{N}(y_i - \widehat{y_i})^2 \quad (11)$$

where y_i - predicted value. $\widehat{y_i}$ - real value. N - sample's length. To reduce the difference between the actual and predicted outputs by reducing the total MSE value. Lower MSE means that the accuracy of the prediction of hate speech will be higher.

4. Experimental results analysis

The emergence of hate speech against minority communities on social media has brought a pressing concern about the necessity to come up with more effective detection mechanisms. The complexity of policing hate speech on the internet is the dynamism and lack of structure, and highly contextual nature of user-generated content. The paper will solve these problems by suggesting an optimized ERNN-based model that will be able to categorize text into hostile, offensive, or non-hate.

The optimized ERNN performance is compared with both baseline machine learning algorithms and several state-of-the-art optimistic metaheuristic- ERNN hybrid models, such as EHO-ERNN [31], IPSO-ERNN [32], PSO-ERNN [33], GA-ERNN [34], and ERNN [35] . The feature vectors of all the classification models are created by the use of TF-IDF. All the comparison algorithms are run on a Windows 11 machine with an Intel i5 processor, 8 GB RAM, and MATLAB 2019b. The following subsections explain the datasets, experimental parameters, findings, and discussion to assess the efficacy and strength of the given LSA-EHO-ERNN model.

4.1 Datasets

The dataset used in this research was found on Kaggle.com. It was prepared through a compilation of Twitter tweets. Table 2 shows the dataset's details. The uploaded URL lacked the description of the dataset; however, in the course of the research, it was found that the dataset only contained 31,962 tweets written in English, and no other languages were studied in the situation. 29,720 (92.98) of the tweets in the generated dataset were related to the NHS, and 2,242 (7.02) were related to high school. There is a total of ten experiments carried out during the training and testing stages to achieve the objectives of the study. Each dataset is run twenty times to gather statistical data regarding the precision, stability, and reproducibility of the models.

Table 2: Datasets details

Description	Details
Source of Dataset	Kaggle.com
Platform of Data	Twitter Tweets
Total Tweets Collected	31,962
Language of Tweets	English only
NHS-related Tweets	29,720 (92.98%)
High School-related Tweets	2,242 (7.02%)
Number of Experiments Conducted	10 experiments
Number of Runs Per Experiment	20 runs
Purpose of Multiple Runs	To measure the precision, stability, and reproducibility of models

Table 3: Parameters settings

ERNN Parameters	Optimal Value	EHO Parameters	Optimal Value
Activation function	Tansig	Population size	30
Hidden neurons	50	Number of clans (c)	5
Context layer size	50	Clan update rate (α)	0.8
Learning rate	0.05	Separation rate (β)	0.3
Training algorithm	LSA-EHO	Max iterations	100
Momentum factor	0.9	Search boundary	[-1, 1]
Epochs	1000	Fitness function	MSE
Error goal	0.0005	Randomization factor	0.02
Weight initialization	Uniform (-0.5 to 0.5)	Convergence threshold	1,00E-05
Batch size	32		
Regularization (L2)	0.001		
Gradient clipping	1.0		

4.2 Data preprocessing

The role of data processing in hate speech detection on Twitter is required due to the noise and unstructured nature of the raw Twitter tweets that cannot be interpreted by algorithms. The initial one is *data cleaning*, which eliminates duplicates, empty values, URLs, user mentions, special characters, emojis, and unnecessary symbols and transforms text into lowercase. Hashtags are saved without the hash (#) character since they tend to deliver context. *Normalization of linguistics* is then meted out, in which slang, abbreviations, and contractions are extended, and tokenization splits the text in senseful units. Noise and computational cost are also minimized with the removal of stopwords. *Stemming or lemmatization* transforms the words to their root form, which aids in the reduction of the vocabulary and enhances the generalization of the model. Language detection tools are used to remove non-English tweets, and the metadata with offensive usernames is removed, leaving hate words in the text of the tweet. Lastly, *Part-of-Speech (PoS)* tagging is used to determine grammatical categories of things like nouns, verbs, and adjectives, which aids in capturing the tone and negative connotations of the text. These preprocessing steps, in combination, transform the raw tweets into clean, structured, and semantically useful data that can be used to precisely detect hate speech.

4.3 Document representation

Proper representation of documents has been found to play a very important role in hate speech detection on Twitter, as tweets are very short, informal, and contain a lot of slang, emoticons, and noisy texts. Representation methods transform crude tweets into numerical attributes that can be read by machine learning and deep learning algorithms. The more semantic, syntactic, and contextual information that can be captured, the closer hate speech can be detected.

- **Doc2Vec**: Doc2Vec creates fixed-length vectors of complete documents through learning semantic and syntactic connections among words. It is useful in capturing the general meaning of tweets, but is not very good at the finer contextual details, particularly with complex sentences.
- **TF-IDF**: TF-IDF gives words a significance depending on the frequency of the word in a document and the rarity of the word in the documents. It is easy, fast, and efficient when dealing with classical ML models, yet it does not take into account the sequence of words and the similarity of semantics, considering each word separately.
- **BERT**: BERT offers strong contextual embeddings, as both left and right context are comprehended in a sentence. It is very effective with ambiguity and other nuanced linguistic features, which makes it appropriate for the detection of sentiment and hate speech. It is, however, computationally expensive.
- **Hybrid TF-IDF + BERT**: TF-IDF + BERT is a combination of the statistical significance of terms and in-depth context. This combined representation generates more informative features and demonstrates higher classification performance, at the cost of higher dimensionality, and should be carefully optimized.

4.4 Parameter settings

Hyperparameter optimization is one of the critical factors contributing to the enhancement of the learning performance of the ERNN. EHO gives a powerful search in the global space to find the most suitable combination of hyperparameters, whereas LSA also gives fine local search in the most promising solutions. This is a hybrid approach that guarantees increased converging speed, less training error, and improved generalization. Consequently, the optimized ERNN has a high accuracy rate, better stability, and reliable performance than manually tuned or single-algorithm methods. Table 3 shows the best parameterization of the ERNN and EHO algorithm. These parameters were chosen after a lot of experimentation in order to achieve a better convergence, stability, and predictive accuracy. The optimal design enables the ERNN-EHO model to record higher performance through the balance of global exploration and local exploitation in the process of hyperparameter optimization.

4.5 Performance measures

The accuracy is the percentage of hate tweets that are correctly identified among the total number of hate tweets is referred to as accuracy. It is displayed as follows,

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{12}$$

Precision is the percentage of hateful tweets that are detected and compared to a database of labeled hateful tweets. It could be expressed in numbers as follows,

$$Pr\ e\ cision = \frac{TP}{TP+FP} \tag{13}$$

Recall counts the percentage of hateful tweets that have labels that are properly predictable and obtainable, which is considered as follows,

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

The F1-Score, which has a maximum value of 1 and a minimum value of 0, is the harmonic mean of precision and recall.

$$F - Score = 2 \times \frac{Pr\ ecision \times Recall}{Pr\ ecision + Recall} \tag{15}$$

Where, the quantity of HS incidences that fall under HS is called True Positive (TP); the quantity of NHS incidences that fall under NHS is called True Negative (TN); the quantity of HS incidences that fall under NHS is called False Negative (FN); and the quantity of HS incidences that fall under HS is called False Positive (FP). In one way or another, the percentage of accurate detection to total predictions can be used to quantify the accuracy of the classifier, which is merely a measure of the frequency with which the classifier is making correct detections.

4.6 Results analysis

Four document representations were used to evaluate the performance of Hybrid Sentiment Detection (HSD) methods, and these include Doc2Vec (Table 4), TF-IDF (Table 5), BERT (Table 6), and TF-IDF + BERT (Table 7). In the case of Doc2Vec-based HSD methods, ERNN-LSA-EHO worked out to be the most successful model with an accuracy of 96.32, precision of 95.57, recall of 95.01, and a F1-score of 95.29. Figures 1,4,7, and 10 show the performance based on performance measures, Figures 2, 5, 8, and 11 show training and validation accuracy, and Figures 3, 6, 9, and 12 discuss the convergence analysis of each method. This means that using the LSA-EHO is an important enhancement in the ability of the model to understand semantic relationships in the textual data. The training and validating accuracy curve indicated that the convergence was steady, and the convergence analysis indicated that the ERNN-LSA-EHO is better and more efficient than other metaheuristic models like IPSO, PSO, and GA. With the TF-IDF representation, the general performance of all the approaches was increased, which indicates the efficacy of statistical term-frequency details in the categorization of sentiments. Here, ERNN-LSA-EHO was once better than the other with a 97.01, 96.27, 95.58, and 95.92 accuracy, precision, recall, and F1-score, respectively. On an interesting note, ERNN-PSO and ERNN-GA showed comparable performance, which implies that these optimization methods do not have much value in the TF-IDF features.

Table 4: Performance comparison of HSD methods based on doc2vec

Methods	Accuracy	Precision	Recall	F1-Score
ERNN-LSA-EHO	96.32	95.57	95.01	95.29
ERNN-EHO	93.64	92.89	92.33	92.61
ERNN-IPSO	91.51	90.76	90.20	90.48
ERNN-PSO	89.38	88.63	88.07	88.35
ERNN-GA	87.25	86.50	85.94	86.22
ERNN	85.12	84.37	83.81	84.09

Table 5: Performance comparison of HSD methods based on TF-IDF

Methods	Accuracy	Precision	Recall	F1-Score
ERNN-LSA-EHO	97.01	96.27	95.58	95.92
ERNN-EHO	95.53	94.76	94.09	94.42
ERNN-IPSO	93.84	93.07	92.36	92.71
ERNN-PSO	89.25	88.56	87.92	88.24
ERNN-GA	89.25	88.56	87.92	88.24
ERNN	87.12	86.34	85.78	86.05

Along the convergence curves, it was observed that ERNN-LSA-EHO was more accurate and faster to converge to in training, which depicted that it was more efficient in learning and that the model exhibited stability. Contextual embeddings with BERT also increased the performance of the models. In the case of BERT-based HSD, ERNN-LSA-EHO had the best metrics with an accuracy of 98.64, a precision of 97.91, a recall of 97.28, and an F1-score of 97.59. This benefit highlights the benefit of BERT in context-sensitive semantics that other embeddings, such as Doc2Vec or TF-IDF, might not capture. The training and validation curves showed a smooth convergence, and other metaheuristic-optimized models were slow to converge, which validated the fact that LSA-EHO could better fine-tune the network parameters.

TF-IDF with BERT embeddings gave the most favorable results of all representations. The accuracy of ERNN-LSA-EHO was also high at 99.11, precision at 98.36, recall at 97.80, and F1-score at 98.08. This means that the combination of statistical and contextual characteristics can enable the model to exploit complementary information, leading to the best sentiment classification. The convergence analysis revealed rapid convergence and small discrepancy between the training and validation stages, indicating the generalization capability of the model. Throughout all feature representations, ERNN-LSA-EHO consistently performed better than the other variants, and the baseline ERNN performed the least, which underlines the value of hybrid optimization to the successful sentiment classification.

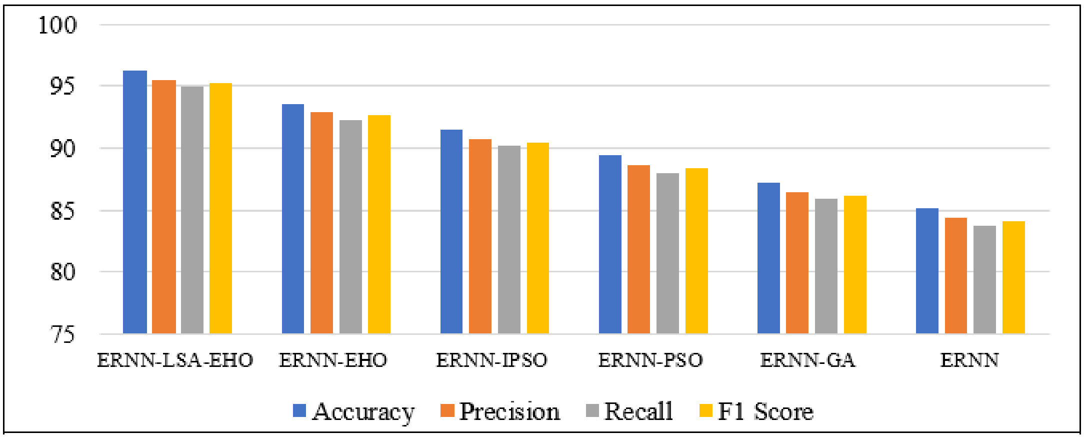

Figure 1: Performance comparison of HSD methods based on doc2vec

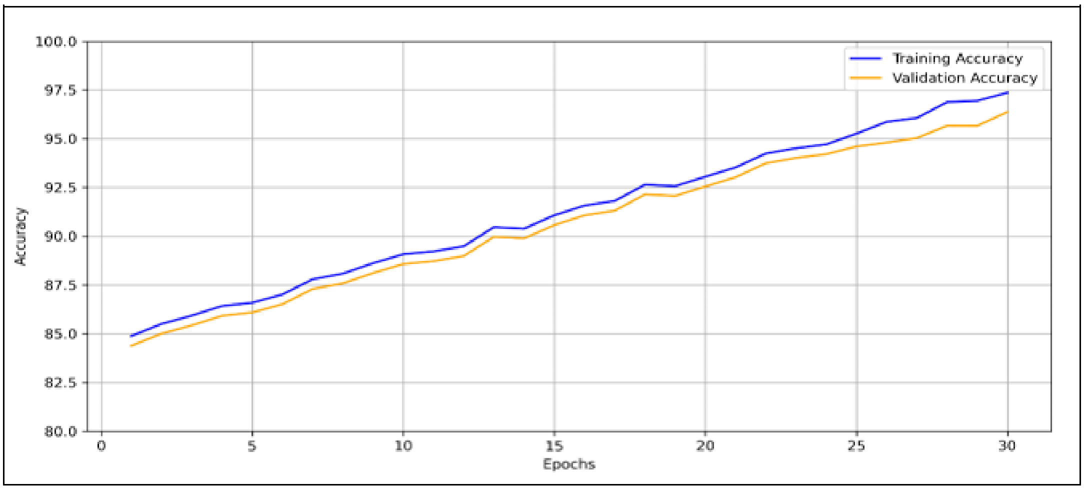

Figure 2: Training and validation accuracy comparison of HSD methods based on doc2vec

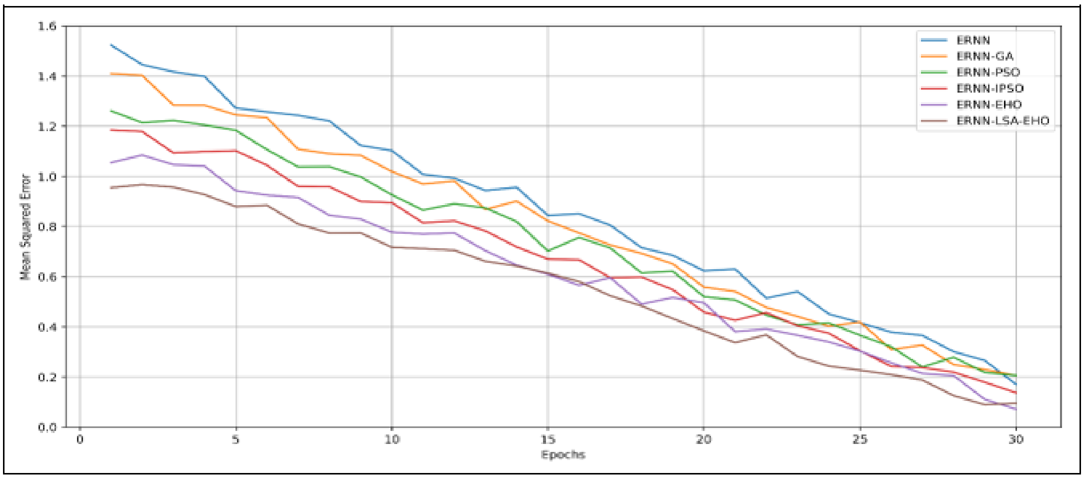

Figure 3: Convergence analysis of HSD methods based on doc2vec

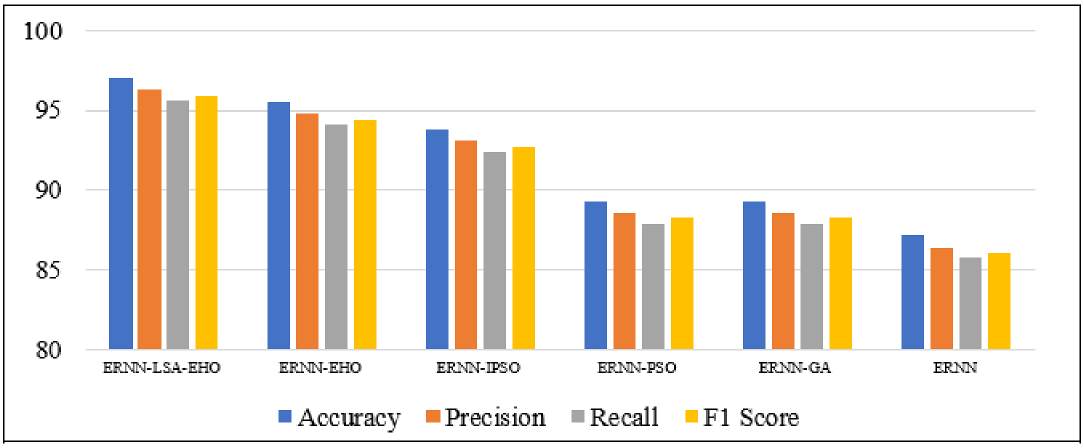

Figure 4: Performance comparison of HSD methods based on TF-IDF

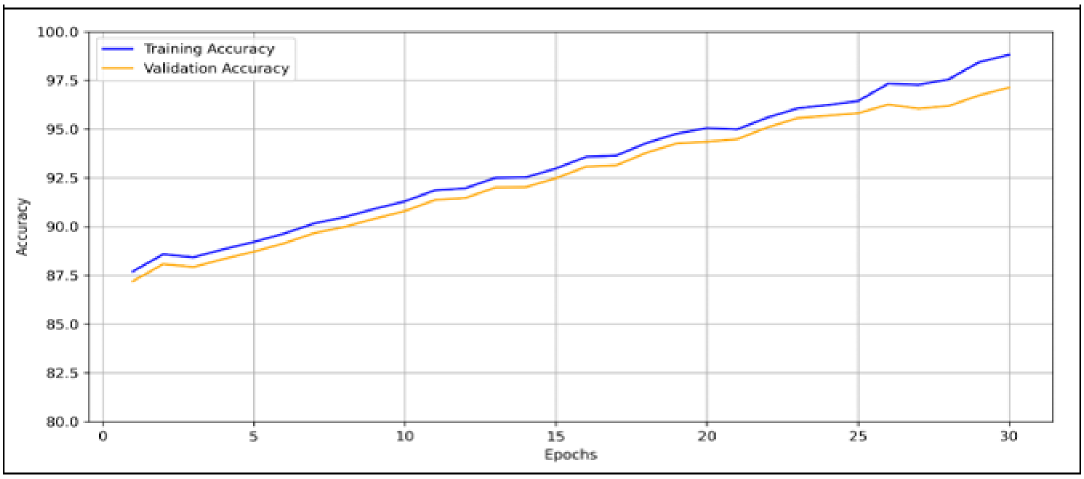

Figure 5: Training and validation accuracy comparison of HSD methods based on TF-IDF

Table 6: Performance comparison of HSD methods based on BERT

Methods	Accuracy	Precision	Recall	F1 Score
ERNN-LSA-EHO	98.64	97.91	97.28	97.59
ERNN-EHO	97.19	96.46	95.82	96.14
ERNN-IPSO	95.63	94.90	94.27	94.58
ERNN-PSO	93.48	92.75	92.11	92.43
ERNN-GA	91.32	90.59	89.94	90.26
ERNN	89.17	88.42	87.85	88.13

Table 7: Performance comparison of HSD methods based on TF-IDF+BERT

Methods	Accuracy	Precision	Recall	F1 Score
ERNN-LSA-EHO	99.11	98.36	97.80	98.08
ERNN-EHO	97.94	97.19	96.63	96.91
ERNN-IPSO	96.41	95.66	95.10	95.38
ERNN-PSO	94.28	93.53	92.97	93.25
ERNN-GA	92.15	91.40	90.84	91.12
ERNN	90.02	89.27	88.71	88.98

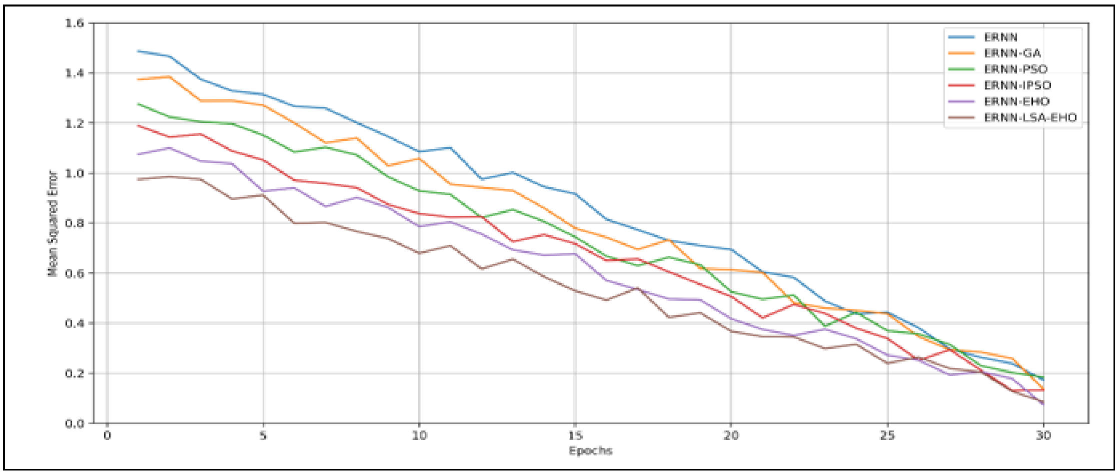

Figure 6: Convergence analysis of HSD methods based on TF-IDF

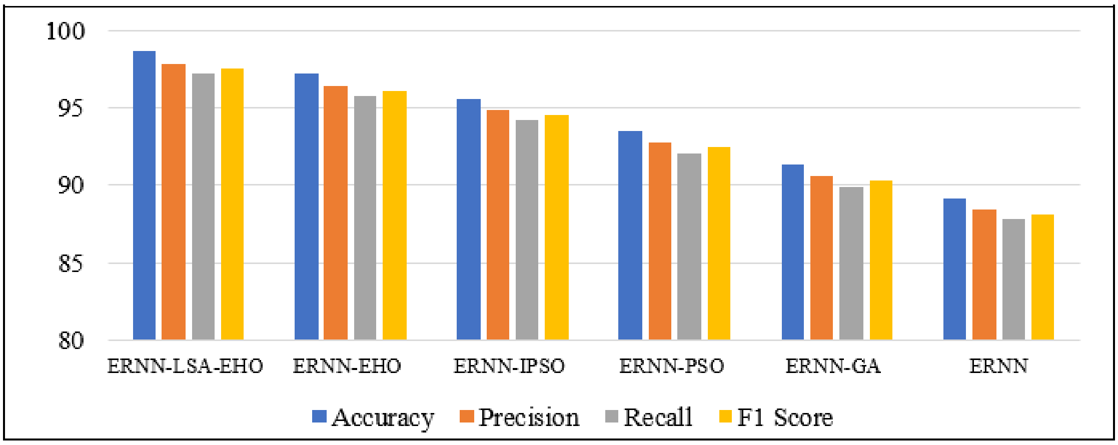

Figure 7: Performance comparison of HSD methods based on BERT

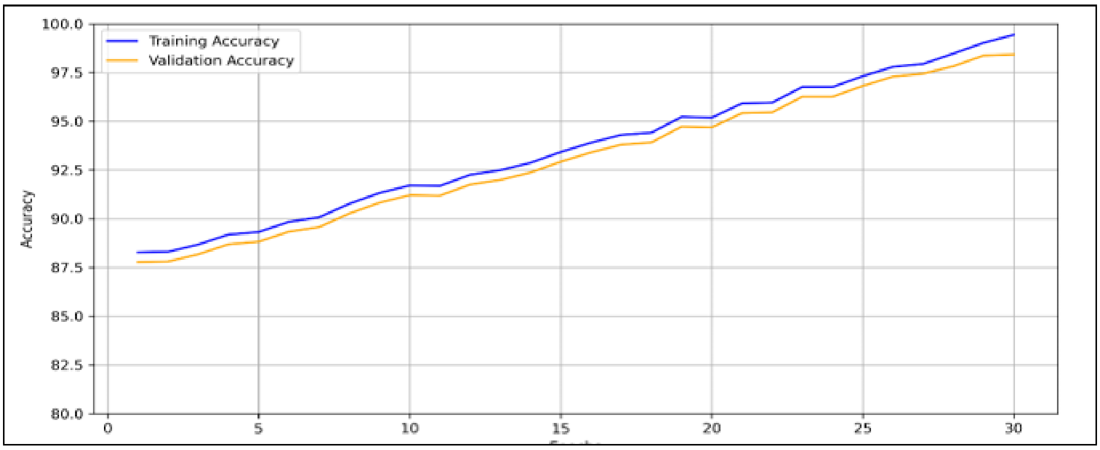

Figure 8: Training and validation accuracy comparison of HSD methods based on BERT

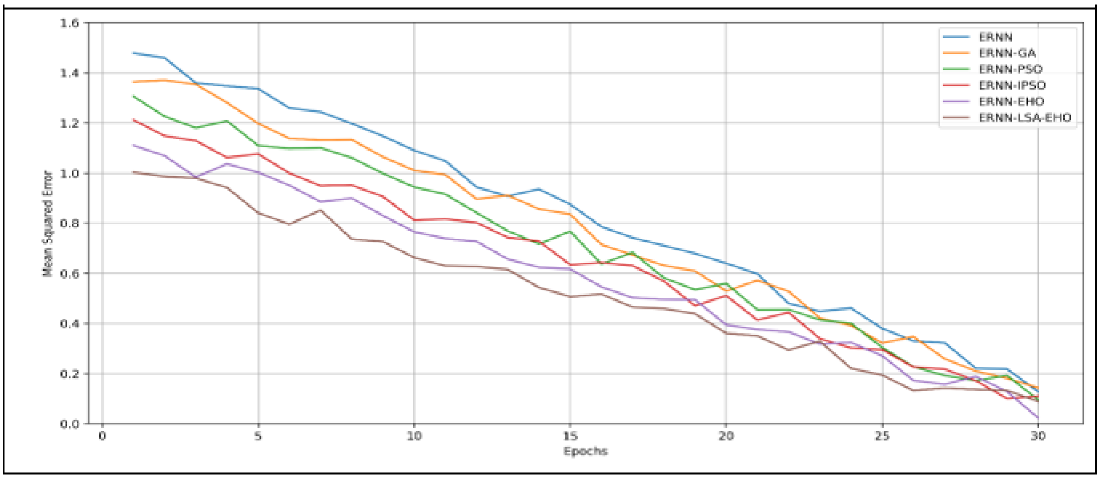

Figure 9: Convergence analysis of HSD methods based on BERT

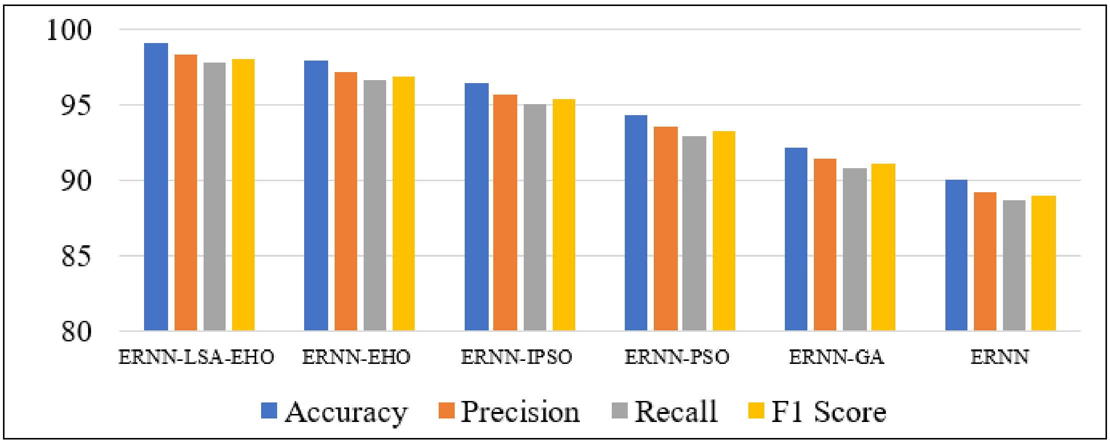

Figure 10: Performance comparison of HSD methods based on TF-IDF+BERT

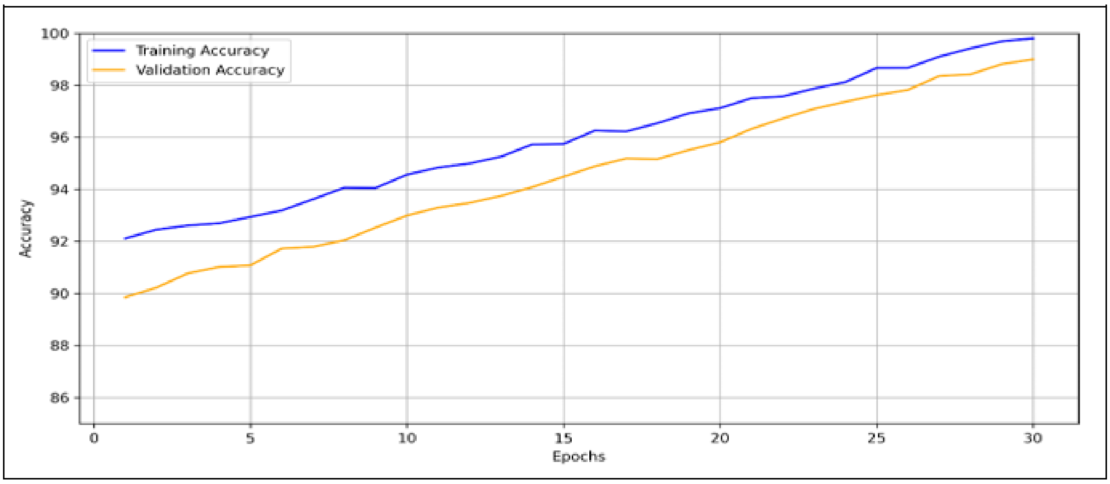

Figure 11: Training and validation accuracy comparison of HSD methods based on TF-IDF+BERT

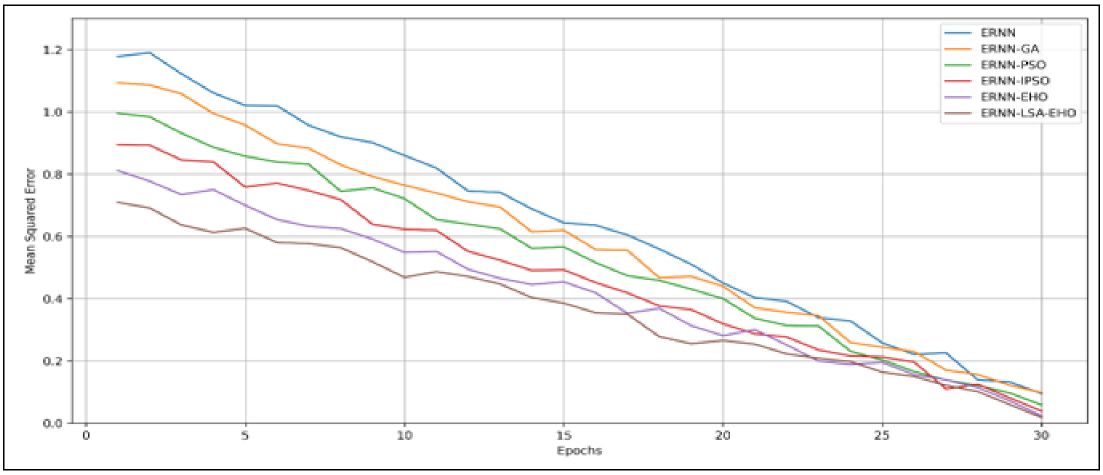

Figure 12: Convergence analysis of HSD methods based on TF-IDF+BERT

On the whole, the experiment findings indicate that feature representation selection and optimization strategy have a high impact on HSD performance. Contextual embeddings, especially those that are used in conjunction with statistical features, provide a considerable accuracy and strength boost. In addition, LSA-EHO was the most efficient optimization technique that invariably gave greater accuracy, rapid convergence, and stability than the other metaheuristic techniques. The above results suggest that the combination of embedding advanced embeddings and hybrid optimization methods is the key to the creation of high-performing and credible sentiment detection systems.

5. Conclusions

This study provided a strong hybrid model of hate speech and sentiment detection in Twitter and combined an ERNN with LSA and EHO. The paper has shown that the advanced document representation, such as Doc2Vec, TF-IDF, BERT, and hybrid TF-IDF + BERT embeddings, is very beneficial in increasing semantic, syntactic, and contextual attributes of the short and noisy written texts in social media. The experimental findings show that the proposed ERNN-LSA-EHO model is always effective, in contrast to baseline ERNN and other metaheuristic-optimized models, in major key performance indicators such as accuracy, precision, recall, and F1-score. The hybrid optimization process allowed the efficient convergence and strength of the learning process, and the model is very efficient in identifying subtle and complicated patterns of hate speech in the real-world Twitter data. These results emphasize the significance of the feature representation as well as optimization methods in the development of effective social media analysis systems.

The key drawback of the given study is its great computational complexity because of the union of deep learning and metaheuristic optimization, which might limit the applicability of the model to real-time or resource-intensive contexts. One future improvement is to create a leaner and more efficient version of the model, perhaps by incorporating attention mechanisms or transformer-based architectures, which will make the computational cost lower and allow the implementation of the model in real-time on various social media products.

References

[1] M. Iranzo-Cabrera, M. J. Castro-Bleda, I. Simón-Astudillo, and L.-F. Hurtado, "Journalists' ethical responsibility: Tackling hate speech against women politicians in social media through natural language processing techniques," *Social Science Computer Review,* vol. 43, no. 3, pp. 475-502, 2025.

[2] U. K. Schmid, "Humorous hate speech on social media: A mixed-methods investigation of users' perceptions and processing of hateful memes," *New Media & Society,* vol. 27, no. 3, pp. 1588-1606, 2025.

[3] R. Prabhu and V. Seethalakshmi, "A comprehensive framework for multi-modal hate speech detection in social media using deep learning," *Scientific Reports,* vol. 15, no. 1, p. 13020, 2025.

[4] K. L. Vignesh, M. S. R. Krishna, and D. Keerthana, "SKVtrio@ LT-EDI-2025: Hybrid TF-IDF and BERT Embeddings for Multilingual Homophobia and Transphobia Detection in Social Media Comments," in *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, 2025, pp. 26-30.

[5] J. S. Malik, H. Qiao, G. Pang, and A. van den Hengel, "Deep learning for hate speech detection: a comparative study," *International Journal of Data Science and Analytics,* vol. 20, no. 4, pp. 3053-3068, 2025.

[6] S. S. Roy, A. Roy, P. Samui, M. Gandomi, and A. H. Gandomi, "Hateful sentiment detection in real-time tweets: An LSTM-based comparative approach," *IEEE Transactions on Computational Social Systems,* 2023.

[7] G. O. Ganfure, "Comparative analysis of deep learning based Afaan Oromo hate speech detection," *Journal of Big Data,* vol. 9, no. 1, p. 76, 2022.

[8] H. Saleh, A. Alhothali, and K. Moria, "Detection of hate speech using bert and hate speech word embedding with deep model," *Applied Artificial Intelligence,* vol. 37, no. 1, p. 2166719, 2023.

[9] S. Kothuru and A. Santhanavijayan, "Automatic hate speech detection using aspect based feature extraction and Bi-LSTM model," *International Journal of System Assurance Engineering and Management,* vol. 13, no. 6, pp. 2934-2943, 2022.

[10] A. Verma *et al.*, "Identification of Hate Speech on Social Media using LSTM," *GMSARN International Journal,* vol. 17, pp. 468-474, 2023.

[11] R. Ali, U. Farooq, U. Arshad, W. Shahzad, and M. O. Beg, "Hate speech detection on Twitter using transfer learning," *Computer Speech & Language,* vol. 74, p. 101365, 2022.

[12] C. N. Arbaatun, D. Nurjanah, and H. Nurrahmi, "Hate speech detection on Twitter through Natural Language Processing using LSTM model," *Building of Informatics, Technology and Science (BITS),* vol. 4, no. 3, pp. 1548− 1557-1548− 1557, 2022.

[13] S. S. Ilhan, S. Sivakumar, J. Nagaraj, S. Ramesh, N. Sreeram, and R. Rajalakshmi, "Hate Speech Detection and Classification Using NLP," in *2024 Second International Conference on Advances in Information Technology (ICAIT)*, 2024, vol. 1: IEEE, pp. 1-7.

[14] S. Shubhang, S. Kumar, U. Jindal, A. Kumar, and N. R. Roy, "Identification of hate speech and offensive content using bi-gru-lstm-cnn model," in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2023: IEEE, pp. 536-541.

[15] C. N. Vo, K. B. Huynh, S. T. Luu, and T.-H. Do, "Exploiting Hatred by Targets for Hate Speech Detection on Vietnamese Social Media Texts," *arXiv preprint arXiv:2404.19252,* 2024.

[16] A. C. Mazari and H. Kheddar, "Deep learning-based analysis of Algerian dialect dataset targeted hate speech, offensive language and cyberbullying," *International Journal of Computing and Digital Systems,* 2023.

[17] A. Reghunathan, S. Singh, R. Gunavathi, and A. Johnson, "Advanced Approaches for Hate Speech Detection: A Machine and Deep Learning Investigation," in *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, 2024: IEEE, pp. 1-5.

[18] K. K. Mohbey, N. Kesswani, N. Yunevich, M. Sterjanov, and M. Vishnyakova, "Hate Speech Identification and Categorization on Social Media Using Bi-LSTM: An Information Science Perspective," 2025.

[19] A. Naseeb *et al.*, "Machine Learning-and Deep Learning-Based Multi-Model System for Hate Speech Detection on Facebook," *Algorithms,* vol. 18, no. 6, p. 331, 2025.

[20] K. Salameh, S. Hamza, and S. Atiani, "Enhancing Arabic Hate Speech Detection: The Role of Word Embedding Techniques with Deep Learning Models," in *2025 International Conference on New Trends in Computing Sciences (ICTCS)*, 2025: IEEE, pp. 334-341.

[21] A. Varatharaj and S. Ahangama, "Advanced Hate Speech Detection in Social Media Content Using LSTM and CNN," 2025.

[22] R. A. Saputra and Y. Sibaroni, "Multilabel Hate Speech Classification in Indonesian Political Discourse on X using Combined Deep Learning Models with Considering Sentence Length," *Jurnal Ilmu Komputer dan Informasi,* vol. 18, no. 1, pp. 113-125, 2025.

[23] A. Al-Mashhadani, S. Awajan, S. El-Bouri, and S. Atiani, "Leveraging Contextualized and Word Embeddings for Arabic Hate Speech Detection," in *2025 International Conference on New Trends in Computing Sciences (ICTCS)*, 2025: IEEE, pp. 440-445.

[24] D. Singh, G. Sonam, P. Gupta, and R. Baghel, "A novel approach of hybrid CNN-RNN for multilingual hate speech detection," in *Intelligent Computing and Communication Techniques*: CRC Press, 2025, pp. 317-324.

[25] A. K. Srivastava, M. Srivastava, S. Das, V. Jain, and T. B. Chandra, "Leveraging Deep Learning for Comprehensive Multilingual Hate Speech Detection," *Procedia Computer Science,* vol. 252, pp. 832-840, 2025.

[26] M. N. Hasan, A. H. Masum, S. H. Rony, H. D. Arpita, and T. Rabeya, "Advancing Online Safety: AI-Powered Hate Speech and Offensive Language Detection in Social Media," in *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2025: IEEE, pp. 1-6.

[27] T. Z. Jilan, "The Effectiveness of Different Deep Learning Models in Detecting Hate Speech on Social Media," in *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2025: IEEE, pp. 1-6.

[28] J. L. Elman, "Finding structure in time," *Cognitive science,* vol. 14, no. 2, pp. 179-211, 1990.

[29] G.-G. Wang, S. Deb, and L. d. S. Coelho, "Elephant herding optimization," in *2015 3rd international symposium on computational and business intelligence (ISCBI)*, 2015: IEEE, pp. 1-5.

[30] M. Tubishat, N. Idris, L. Shuib, M. A. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Systems with Applications,* vol. 145, p. 113122, 2020.

[31] D. Rajakumari and D. Savitha, "An ovarian cancer prediction using an optimized Elman neural network based on elephant herding optimization," in *2023 International Conference on Emerging Research in Computational Science (ICERCS)*, 2023: IEEE, pp. 1-7.

[32] L. Yang, F. Wang, J. Zhang, and W. Ren, "Remaining useful life prediction of ultrasonic motor based on Elman neural network with improved particle swarm optimization," *Measurement,* vol. 143, pp. 27-38, 2019.

[33] Y. Wang, L. Wang, F. Yang, W. Di, and Q. Chang, "Advantages of direct input-to-output connections in neural networks: The Elman network for stock index forecasting," *Information Sciences,* vol. 547, pp. 1066-1079, 2021.

[34] A. Sadeghi-Niaraki, P. Mirshafiei, M. Shakeri, and S.-M. Choi, "Short-term traffic flow prediction using the modified elman recurrent neural network optimized through a genetic algorithm," *IEEE Access,* vol. 8, pp. 217526-217540, 2020.

[35] N. Chowdhury, "A comparative analysis of feed-forward neural network & recurrent neural network to detect intrusion," in *2008 International Conference on Electrical and Computer Engineering*, 2008: IEEE, pp. 488-492.