International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I1P102

Volume 74 Issue 1, 25-38, January 2026


*Original Article*

# Nonlinear Seagull Optimized Bidirectional Recurrent Network for English Hate Speech Detection in Online Social Network

I. Imthiyas Banu[1], Velumani Thiyagarajan[2]

[1,2]*Department of Computer Science Rathinam College of Arts and Science, Coimbatore, Tamil nadu, India.*

[2]*Corresponding Author : Velumani46@gmail.com*

Received: 14 August 2025 Revised: 03 December 2025 Accepted: 25 December 2025 Published: 14 January 2026

***Abstract -*** *Online Social Network (OSN) provides services or sites to facilitate social interaction for identifying people's general attention, discussion, and exchanging information. The Proposed Nonlinear Evolutionary Seagull Optimized Bidirectional Gated Neural Network (NESO-BGNN) is introduced for hate speech detection in English in OSN. It comprises preprocessing, keyword extraction, and classification. First, Robust Scaling Normalization-based preprocessing is applied to handle outliers. Second, Nonlinear Evolutionary-based Seagull Optimization algorithm extracts optimal keywords for hate speech detection. Finally, the Bidirectional Gated Recurrent Neural Network (BGRNN) detects hate speech accurately with a lower misclassification rate. An experiment was carried out using the Hate Speech and Offensive Language Dataset with different factors.*

***Keywords -*** *Robust Scaling Normalization, Inter Quartile Range, Nonlinear Evolutionary, Seagull Optimization, Bidirectional Gated Recurrent Neural Network.*

## 1. Introduction

Social media computing provides efficient communication among various persons in an understandable way, specifically through social media platforms and chat forums. Enormous social websites and applications are designed to connect users and organizations for sharing information between them. In addition, families and friends are also making connections in the same area with the same interests.

Owing to this, social media has been regarded as one of the most distinguished benefactors to the freedom of speech postulates. There has been an outpouring in the employment of the Internet, together with social media platforms, that has resulted in an increase in online hate speech focused on individuals or groups. As a result, in recent years, hate speech has brought about one of the demanding issues that can spread at an accelerated speed on digital platforms, resulting in several issues like preconception, brutality, and even massacre.

With the growth of online social media platforms, hate speech detection has become a demanding issue for both individuals and society because of easy accessibility. Many individuals express their emotions, ideas, and feelings on social media sites like Reddit, Facebook, Instagram, YouTube, Twitter, etc. But people have exploited social media to convey hateful messages to certain groups to produce confusion. For the establishment of numerous authorities, manual identification of hate speech on several social media platforms is a difficult and heavy task to avoid confusion. However, several research efforts have been developed for detecting hate speech from online social networks.

The hate speech detection over the past few years has been reduced to a binary classification task; however, the misclassification rate and training time involved have been less concentrated. Hate speech detection has been reduced to a binary classification task; however, the Misclassification Rate (MR) and time were less concentrated. In [1], a Fine-grained cyberbullying Classification approach is described with Neutrosophic Logic within a Multi-Layer Perceptron (MLP) model. But, training time was not focused. Passion-Net was introduced [2] for the extraction of semantic and discriminative patterns.

However, MR was not concentrated. Survey on hate speech detection approach [3] presented. Fusion approach [4] and hate speech binary classification [5] were intended. Hate speech on Twitter was detected [6] by Machine Learning (ML) and Deep Learning (DL) techniques. NLP and DL were investigated in [7]. Binary classification method [9] was proposed for a social media plan. The transfer learning technique was applied [10] with less training time.

### *1.1. Contributions of the Work*

A novel contribution of NESO-BGNN is as follows:

- The Nonlinear Evolutionary Seagull Optimized Bidirectional Gated Neural Network (NESO-BGNN) method is developed to improve accuracy for English detection of hate speech.
- A novelty of Robust Scaling Normalization and Nonlinear Evolutionary-based Seagull Optimization algorithm to minimize MR and time.
- A novelty of the Bidirectional Gated Recurrent Neural Network is to obtain accurate precision and hate speech detection results based on the optimal keyword, improving the accuracy and reducing the time for the detection process.

The research gaps in online social networks comprise understanding the long-term effects of specific platforms and features, reconciling conflicting findings on mental health impacts, integrating technology behavior with experience perspectives, and exploring the nuanced influence of social media on academic performance and knowledge gaps.

## 2. Related Works

Malay hate speech detection using ML was proposed in [11]. The application of Bidirectional Encoder Representation [12] was employed to achieve a higher F1-score. Gated Recurrent Units (GRU) based Bidirectional Encoder Representations from Transformers (BERT) was proposed [13]. Transfer learning called t-HateNet was proposed in [10] for constructing a single representation of hate speech. Word2Vec was introduced in [14]. Deep learning methods were applied in [15]. A multilingual hate speech detection method employing fine-tuned transformers was proposed in [18]. An in-depth comparison of ML and DL for hate speech detection was investigated in [17, 18]. A review of hate speech employing optimization techniques, ML, and DL was presented in [8]. In [19], two transformer-based models were designed by extracting optimal keywords. Long short-term memory (LSTM) was presented in [20] to identify offensive or hate content in Bengali. A sequential model based on LSTM was designed in [21]. Time-consuming was investigated in [22]. Deep neural network-based multi-task learning was proposed in [23] for five classification tasks. However, multiple related classification result was not focused.

### *2.1. Problem Definition*

The Online Social Networks (OSNs) involve defining them as virtual communities for connection and information sharing, and a core problem is the management of associated risks and challenges, including user privacy, security threats like malware and identity theft, and the negative impacts of problematic use on mental health, such as anxiety, depression, and cyberbullying. To overcome these issues, the proposed NESO-BGNN is designed to achieve higher accuracy with minimum time for hate speech detection.

## 3. Proposal Methodology

Detection of social media comments in people is one of the most important tasks in social communication. With accurate hate speech detection, the proposed NESO-BGNN was introduced. The figure illustrates the overall architecture diagram of NESO-BGNN.

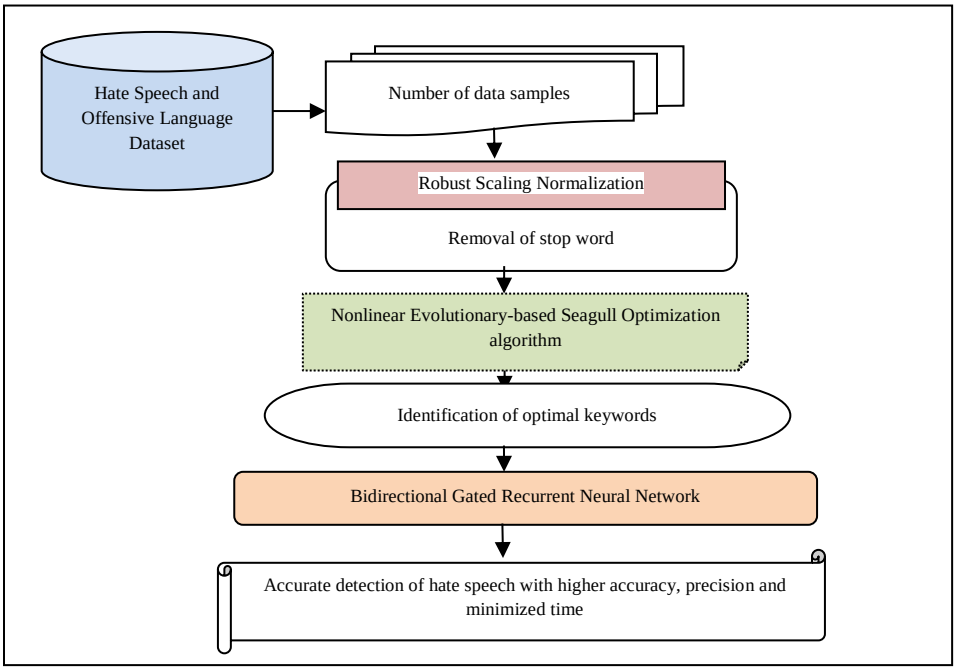


**Fig. 1 Process of NESO-BGNN on online social networks for hate speech detection**

From Figure 1, hate speech detection in English is effectively performed by NESO-BGNN. Initially, speech data samples are collected from the Dataset. Robust Scaling Normalization-based preprocessing eradicates repeated tweets. Nonlinear Evolutionary-based Seagull Optimization algorithm is applied to extract optimal keywords. A bidirectional Gated Recurrent Neural Network for accurate hate speech detection is performed with minimal training time.

### *3.1. Dataset Description*

NESO-BGNN implemented by Python- R Statistical Programming Tool. For performing hate speech detection, the Hate Speech and Offensive Language Dataset is considered. Here, a hate speech dataset is gathered from https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset. The dataset includes seven different attributes or features along with 25296 data samples. This dataset contains hate speech sentences in English and is confined to two classes, one representing hateful content. The total number of contractions considered in the dataset is 6403. The total number of bad words usually used in hateful content is 377. The text in each sentence of the final dataset, which is utilized for training and cross-validation, is limited to 180 words. The generated contractions dataset can be used for any projects in the area of NLP for data preprocessing. The augmented dataset can help reduce the number of out-of-vocabulary words, and the hate speech dataset can be used as a classifier to detect hate speech or non-hate content on social media platforms. The seven features are provided in a table.

**Table 1. Details of features or attributes**

| S. No | Features or Attributes | Description | S. No | Features or Attributes | Description |
|---|---|---|---|---|---|
| 1 | Index | Number of samples | 5 | Neither | number of CF users who judged the tweet to be neither offensive nor non-offensive |
| 2 | Count | Number of Crowd Flower users who coded each tweet (min is 3, sometimes more users coded a tweet when judgments were determined to be unreliable by CF) | 6 | Class | Class label for the majority of CF users. 0 - hate speech 1 - offensive language 2 - neither |
| 3 | Hate_speech | Number of CF users who judged the tweet to be hate speech | 7 | Tweet | text tweet |
| 4 | Offensive_language | Number of CF users who judged the tweet to be offensive | | | |

### *3.2. Robust Scaling Normalization-based Preprocessing Model*

A Robust Scaling Normalization process is employed to perform preprocessing. The data samples collected from the Hate Speech and Offensive Language Dataset comprise different forms of data features. Robust scaling is a method to regularize the range of features of data. During preprocessing, robust scaling is described as data normalization to identify missing data samples in OSN with estimation of the median and Inter Quartile Range (IQR) value. The processing of such samples introduces an error in the detection outcome of hate speech in English. Therefore, data preprocessing using the Robust Scaling Normalization method is required for obtaining better-quality samples for further use. This preprocessing is shown in Figure 2.

In Figure 2, preprocessing is used for processed data with stop data elimination. Assume hate detection dataset '$DS$' includes '$n$' number of data samples $s=\{s_1, s_2, ...,s_n\}$ and '$m$' number of features '$f=\{f_1, f_2, ...,f_m\}$'. Collected data samples and features arranged in matrix format.

$$A=\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1m} \\ s_{21} & s_{22} & \cdots & s_{2m} \\ s_{31} & s_{32} & \cdots & s_{3m} \\ s_{n1} & s_{n2} & \cdots & s_{nm} \end{bmatrix} \quad (1)$$

In (1), the sample data in OSN collected and arranged in matrix format is '$A$'. Samples are arranged in rows and columns where each row is samples '$n$', and each column is feature '$m$'.

After that, the mean value of the features' $\bar{f}$ in samples is estimated by using the following expression.

$$\bar{f}=\frac{f_1+f_2+f_3+...f_m}{m} \quad (2)$$

Where, '$f_1+f_2+f_3+...f_m$' is features and '$m$' is total features. Based on the mean value, the standard deviation feature '$SD$' is attained.

$$SD=\sqrt{\frac{\sum_{i=1}^{m}(f_i-\bar{f})^2}{m}} \quad (3)$$

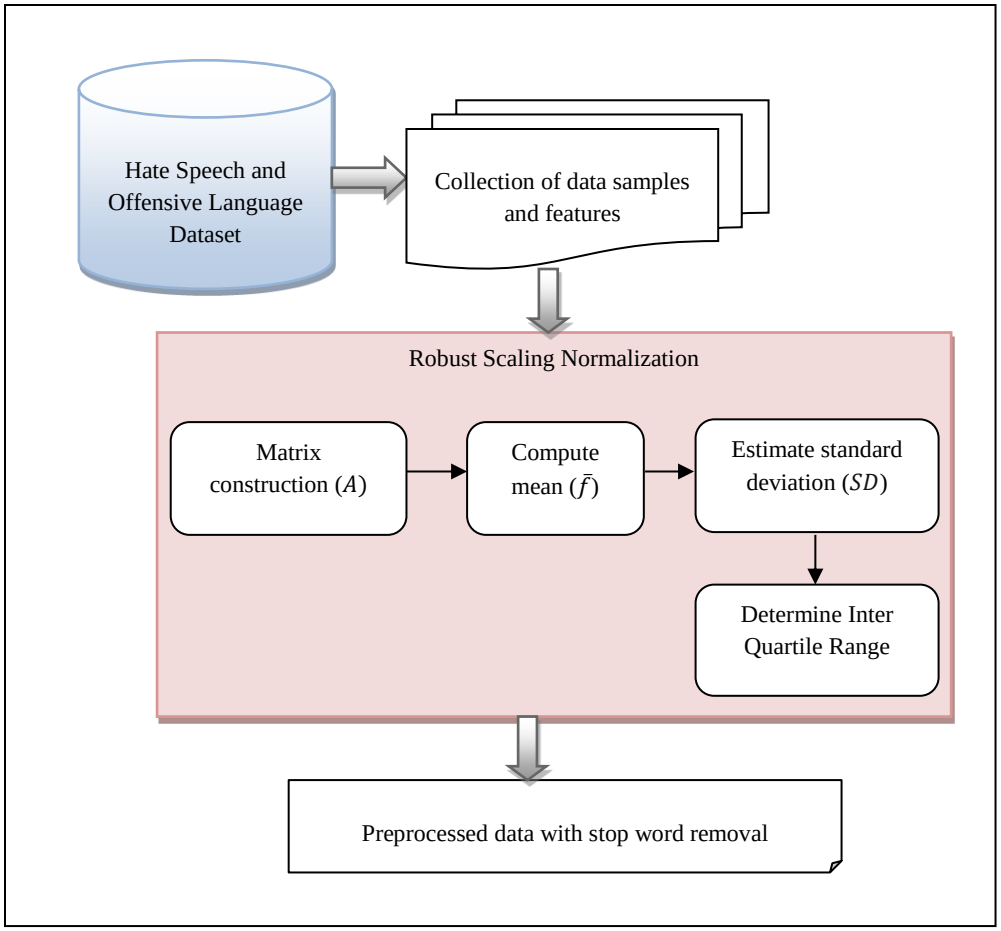


**Fig. 2 Block diagram of robust scaling normalization-based preprocessing**

Where, '$f_i$' is the number of input data features. The Inter Quartile Range (IQR) value for each feature is computed to determine stop words of English speech. Based on the calculated median and standard deviation value, IQR is measured with both the lower and upper halves of the data.

IQR is defined as the difference between the upper and lower quartiles. Input data sample features are arranged from ascending to descending order. Distributed data are segmented into four equal parts. The difference between the third quartile and first quartile is determined using the Interquartile range. A segmented set of data into quartiles is first quartiles, second quartiles, and third quartiles.

Quartiles are '$Q_1$', '$Q_2$' and '$Q_3$'. '$Q_1$' is the first half of rank-ordered data features, '$Q_2$' is the median value of data features and '$Q_3$' is the second half of rank-ordered data features. The interquartile range is illustrated in Figure 3.

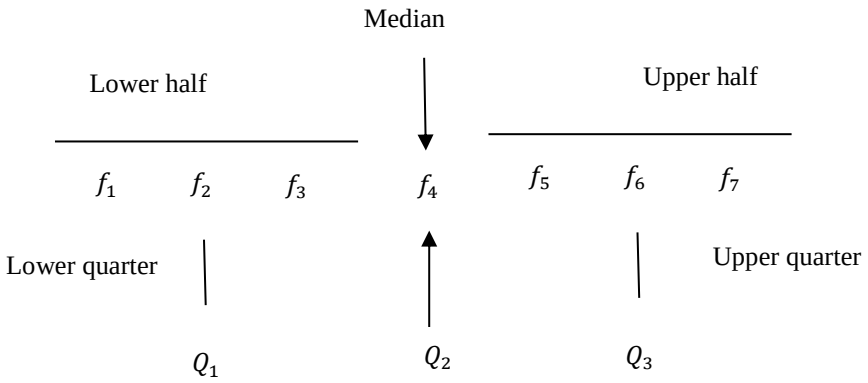


**Fig. 3 Interquartile range**

IQR is described in Figure 3 with a set of different data sample features.

$$IQR \simeq Q_2 = Q_3 - Q_1 \quad (4)$$

Where

$$Q_1 = \left(\frac{1}{4}\right)[(n+1)]^{th}\text{term},$$

$$Q_3 = \left(\frac{3}{4}\right)[(n+1)]^{th}\text{term}$$

Where (4), '*IQR*' is estimated based on '$n$' number of data sample features. This, in turn, stops the data from being eliminated and utilized to detect hate speech with minimized complexity. Therefore, all the data in the dataset is normalized for further processing. Pseudo-code for Robust Scaling Normalization-based preprocessing is described below.

**Algorithm 1: Process of Robust Scaling Normalization-based Preprocessing**

Input: Dataset '*DS*', Features '$f=\{f_1, f_2, ..., f_m\}$', Network Samples '$s=\{s_1, s_2, ..., s_n\}$'
Output: Efficient data preprocessing

Initialize '$m$' number of features and '$n$' number of data samples
Begin
1. For each data sample 's' with features '$f$'.
2. Construct the matrix in equation (1)
3. Find the mean of the features '$f$' given in equation (2)
4. Calculate the standard deviation given in equation (3)
5. Estimate the Interquartile range as in equation (4)
6. Identify stop words and eliminate them from the dataset
7. End for

End

### *3.3. Nonlinear Evolutionary-based Seagull Optimization Algorithm*

Keyword identification using the Seagull Optimization algorithm based on the obtained preprocessed data samples. An identified keyword helps to identify hate English speech among comments in OSN. The Seagull Optimization Algorithm (SOA) has been applied to online social networks primarily for tasks like hate speech detection and affect (emotion) classification. In these applications, the SOA is used as a powerful metaheuristic technique to optimize the performance and hyperparameters of machine learning and deep learning models.

The proposed Seagull Optimization algorithm is a meta-heuristic evolutionary optimization algorithm. Seagull reproduces natural behavior of seabirds that cover the globe, including skills of hunting, and they consume both fresh and salt water. Seagulls are omnivorous birds that feed on insects, fish, reptiles, amphibians, and earthworms. Considered seagulls are provided with different masses and lengths.

Enhanced Seagull Optimization is operated using evolutionary frontier curb processing and convergence speed. Estimation of seagull optimization provides migration and attacking behaviors of objects in the search space with enhanced results. According to the proposed optimization, seagull is linked to a number of data sample features in OSN. Consider the number of data sample features denoted as '$f=\{f_1, f_2, ..., f_m\}$'.Considered seagulls initialized with positions and velocities in the search space. After initialization, the position of the search agent '$P_{SA}$' is carried to avoid a crash between neighboring search agents.

$$P_{SA} = A * CP_{SA} \quad (5)$$

In (5), '$CP_{SA}$' is the current position of the search agent, and '$A$' is the movement behavior of the search agent in the given search space. It is estimated as follows.

$$A = f_n - \left(n * \left(\frac{f_n}{Max_{iteration}}\right)\right) \quad (6)$$

Where, '$f_n$' is features and '$n$' is data sample features. After the search space is updated to avoid collisions between neighbors.

For position updation, agents are forwarded towards the best neighbor agents (i.e., keywords). The movement of position '$M_s$' is expressed below.

$$M_s = b - (P_{BSA}(n) - P_{SA}(n)) \quad (7)$$

In (7), '$b$' is the behavior of the search agent, '$P_{BSA}(n)$' is the best search agent position. The behavioral position of the search space manages exploration and exploitation accurately. It supports identifying significant relevant keywords during the process of hate speech detection.

Optimal or near-optimal solutions of the search agent are identified through the evolutionary frontier curb processing model. The distance between search agents is estimated using convergence speed.

$$Dis = P_{SA} + M_s \quad (8)$$

Where '*Dis*' is the maximum function that helps to identify the optimal search agent (i.e., keywords). The result of distance values lies in the range from 0 to 1. If the distance is higher, then the feature is an optimal keyword for hate speech detection.

Otherwise, features are not significant keywords from the dataset to minimize the training time of hate speech detection. The Seagull Optimization Algorithm for optimal keyword extraction is described as follows.

**Algorithm 2: Process of Seagull Optimization algorithm**

Input: Dataset, preprocessed data sample features '$f=\{f_1, f_2, ..., f_m\}$'
Output: Extracted optimal keywords

Begin
1. For each feature in the dataset '$f_m$'
2. Measure the position of the search agent without collision, as in equation (5)
3. Measure the movement of the search agent as in equation (7)
4. Measure distance '*Dis*' using equation (8)
5. If (*Disishigher*) then
6. Features with optimal keywords are selected
7. Else
8. Keywords are not considered
9. End if
10. Select the optimal keywords and remove other features
11. Return ( optimal keywords )
12. End for

End

### *3.4. Bidirectional Gated Recurrent Neural Network*

Hate speech detection is carried out using B-GRNN. In B-GRNN, neurons like data sample keywords are interconnected with another layer to form a network.

If the link between two nodes is strong, B-GRNN provides an efficient result in hate speech detection. A bi-directional network is carried in two different data sequences, such as forward and backward series. In a forward series, data from the input layer is considered as input, and the hidden layer is processed to detect the final output. Similarly, the backward series is processed in the opposite direction by considering the current input sequence and the updated result of the hidden state output. Analyses of data features identified hate speech and were provided at the output layer. B-GRNN is shown in Figure 4.

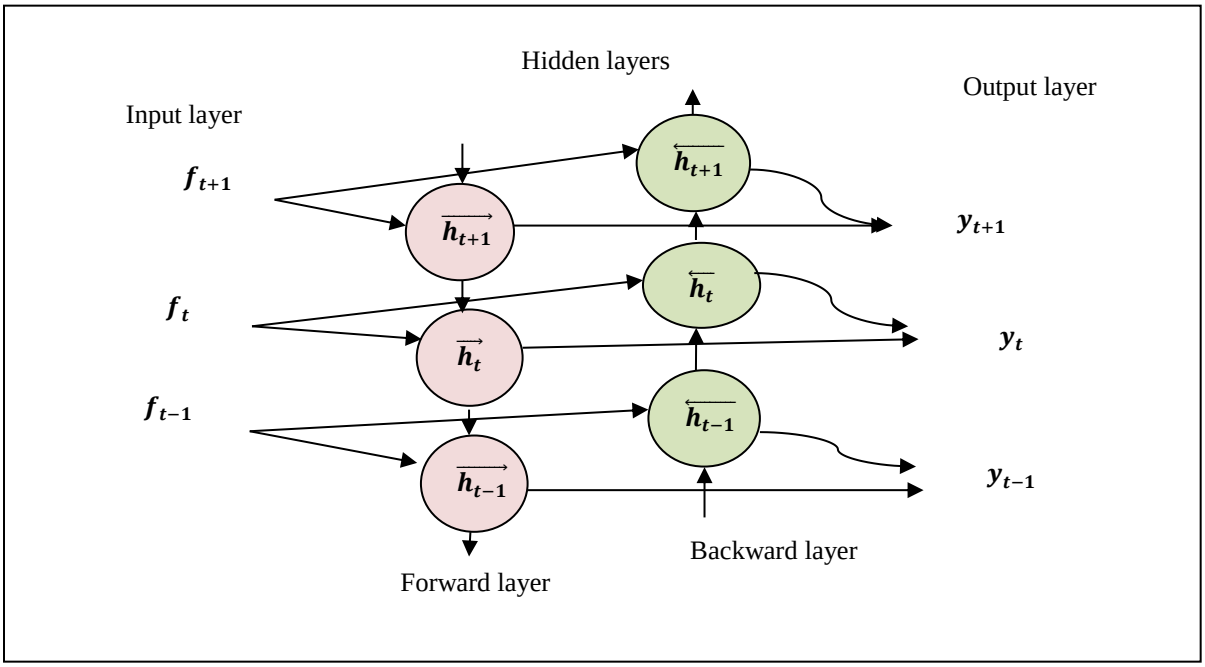


**Fig. 4 Process of B-GRNN**

Figure 4 shows the process of B-GRNN, which includes three different units. RNN includes a number of identified keywords of sample features in input units. In the hidden unit, input keywords are processed to classify hate and non-hate speech in English.

Extracted feature keywords '$f_1,f_2,f_3,...f_m$' are considered as input to the input layer. A neuron is considered an input that assigns the weight, and the bias returns the output. Therefore, the activity of a neuron is formulated,

$$y=[\sum_{m=1}^{n} f_m * W]+Bi \quad (9)$$

Where (9), the output of neuron '$y$' is obtained with feature keywords '$f_m$' multiplied with a set of weights '$W$' and bias value '$b$' that stored integer value '1'. Input keywords are transferred into the hidden layer for identifying hate and non-hate speech in social media. In the hidden layer, data is processed in both forward and backward directions. Hidden state data is presented at time instance '$t$'. At each time stamp, the recurrent process in the hidden unit is measured as follows,

$$\overrightarrow{h_t(forward)}=a(f_t*W_{fh}(forward)+h_{t-1}(forward)*W_{hh}(forward)+b_h(forward)) \quad (10)$$

$$\overrightarrow{h_t(backward)}=a(f_t*W_{fh}(backward)+h_{t+1}(backward)*W_{hh}(backward)+b_h(backward)) \quad (11)$$

From equations (10) and (11), an output of the hidden state at the time stamp '$t$' in forward '$\overrightarrow{h_t(forward)}$' and backward '$\overrightarrow{h_t(backward)}$' direction is determined. Here, '$a$' is the activation function, '$W_{fh}$' specifies weight among the input and hidden layer, '$W_{hh}$' denotes the weight of hidden layers, and '$b_h$' represents the bias value of the hidden layer. The final result of the output layer is determined by the softmax activation function. The result of the output state is formulated as.

$$y(t)=Softmax(h_t*W_{hh}+b_h) \quad (12)$$

Based on the activation result, the data keywords are classified as hate speech, offensive, or neither. This process is continually carried out to detect all hate and non-hate speech in online social networks. The detected speech results at the output layer are formulated as given below.

$$y(t)=\begin{cases}0 & \text{hatespeech}\\ 1 & \text{offenisve}\\ 2 & \text{neither}\end{cases} \quad (13)$$

From (13), '$y(t)$' denotes an output layer result based on the activation function. In this way, accurate classification of hate speech is done with higher accuracy.

The algorithmic process of a bi-directional gated recurrent neural network for hate speech detection is below,

**Algorithm 3: Bi-directional gated recurrent neural network**

Input: Identified feature keywords '$f_1,f_2,f_3,...f_m$'
Output: Detection of hate speech in English

Begin
1. Number of feature keywords'$f_1,f_2,f_3,...f_m$' taken at the input layer
2. For each feature keyword $f_m$
3. Formulate the activity of the neuron using (9)
4. End for
5. For each neuron activity –[hidden layer]
6. Analyses keywords in the forward direction using (10)
7. Analyses keywords in the backward direction using (11)
8. End for
9. Apply the softmax activation function using (12) at the output layer
10. If ($y(t)=0$ ) then
11. Keywords are detected as hate speech
12. End if
13. If ($y(t)=1$ ) then
14. Keywords are detected as offensive
15. End if
16. If ($y(t)=2$ ) then
17. Keywords are detected as neither
18. End if

End

## 4. Experimental Settings

Proposed NESO-BGNN, existing [1, 2] are implemented in Python with R Statistical Programming Tool using Hate Speech and Offensive Language Dataset.

## 5. Dataset Statistics

Common hate speech and offensive language datasets are often collected from social media platforms like Twitter, YouTube, and Facebook, and typically contain text data in English or other specific languages.

Size: Dataset sizes vary widely, from a few thousand samples to hundreds of thousands

Classes: The classification task is usually binary or multi-class.

Class Imbalance: A significant characteristic of these datasets is their inherent class imbalance, where hateful or offensive content is the minority class

Annotation: Datasets are typically human-annotated, often using multiple annotators and majority voting to determine the final label.

### *5.1. Preprocessing Steps*

Raw text data needs careful preprocessing to be usable by machine learning models. Key steps include:

Case Folding: Converting all text to lowercase to ensure uniformity.

Noise Removal: Removing irrelevant elements like URLs, user mentions (@usernames), HTML tags, and extra whitespace.

Punctuation and Special Character Handling: Removing or tokenizing special characters and punctuation marks, as they may not provide useful semantic value in standard models.

### *5.2. Experimental Design*

The experimental design focuses on ensuring robust and reliable model evaluation, especially given the class imbalance.

### *5.3. Data Splits*

Training, Validation, and Test Sets: The standard approach is to split the dataset into three subsets: a training set (e.g., 70-80% of data) for model training, a validation set (e.g., 10-15%) for hyperparameter tuning, and a held-out test set (e.g., 10-20%) for final, unbiased evaluation on unseen data.

Cross-Validation: To obtain more reliable results, k-fold cross-validation is often used, especially for smaller datasets.

Stratified Splitting: Crucially, data splitting (for both train/test splits and cross-validation) should be stratified to maintain the original proportion of each class in all subsets, which is vital for imbalanced data.

## 6. Performance Results and Discussion

NESO-BGNN and existing [1, 2] are discussed based on certain parameters.

Hate speech detection accuracy (HSDA): It is calculated as the ratio of data samples accurately detected as hate speech. The accuracy is formulated as given below.

$$\text{HSDA}=\sum_{i=1}^{n}\frac{S_{accdet}}{S_i}*100 \tag{14}$$

Where '$HSDA$' denotes a hate speech detection accuracy, $s_i$ denotes a data sample and '$S_{acc\ det}$' denotes a data sample accurately detected. It is measured in terms of percentage (%).

Precision: Precision is estimated as the number of correctly detected data samples against the total number of positive samples in the dataset. It is mathematically stated as given below.

$$\text{Precision}=\left(\frac{T_p}{T_p+F_p}\right)*100 \tag{15}$$

Where, '$T_p$' denotes a true positive (i.e., number of data samples that are correctly detected as hate speech), '$F_p$' denotes a false positive rate. It is measured in percentage (%).

Recall: It is measured as the ratio of the number of correctly detected data samples to the total number of negative data samples. It is mathematically expressed as given below,

$$\text{Recall}=\left(\frac{T_p}{T_p+F_n}\right)*100 \tag{16}$$

Where, '$T_p$' denotes a true positive (i.e., number of data samples that are correctly detected) and '$F_n$' denotes a false negative. It is measured in terms of percentage (%).

Detection Time (DT): It is defined as the amount of time consumed by the algorithm for detecting hate speech in English in an online social network.

It is measured in terms of milliseconds (ms). The formula for detection time is calculated as given below,

$$\text{DT}=\sum_{i=1}^{n}s_i*[\text{Time}(DS)] \tag{17}$$

Where, DT denotes a detection time, '$s_i$' denotes the number of data samples and Time (DS) denotes a time for detecting sample features.

Misclassification Rate (MR): It measures the ratio of the number of data samples inaccurately detected as hate speech to the total number of data samples. The misclassification rate is computed using the mathematical representation,

$$MR = \sum_{i=1}^{n} \frac{S_{inaccdet}}{S_i} * 100 \quad (18)$$

From (18), the misclassification rate '*MR*' is measured. Here, '$S_{inacc\ det}$' point out a number of data samples inaccurately detected and '$S_i$' indicates a total number of data samples. The misclassification rate is determined in terms of percentage (%).

**Table 2. Hate speech detection accuracy**

| Number of Data Samples | Hate speech detection accuracy (%) | | |
|---|---|---|---|
| | Fine-Grained Cyberbullying Classification | Passion-Net | NESO-BGNN Method |
| 2500 | 84 | 87 | 93 |
| 5000 | 85.3 | 87.65 | 93.7 |
| 7500 | 85.05 | 86 | 93.2 |
| 10000 | 84.63 | 86.2 | 93.06 |
| 12500 | 83.2 | 86.1 | 92.75 |
| 15000 | 82.6 | 85.6 | 92 |
| 17500 | 81 | 85.42 | 91.75 |
| 20000 | 80.6 | 85 | 91.6 |
| 22500 | 79.5 | 83.5 | 91 |
| 25000 | 81.6 | 84 | 92.3 |

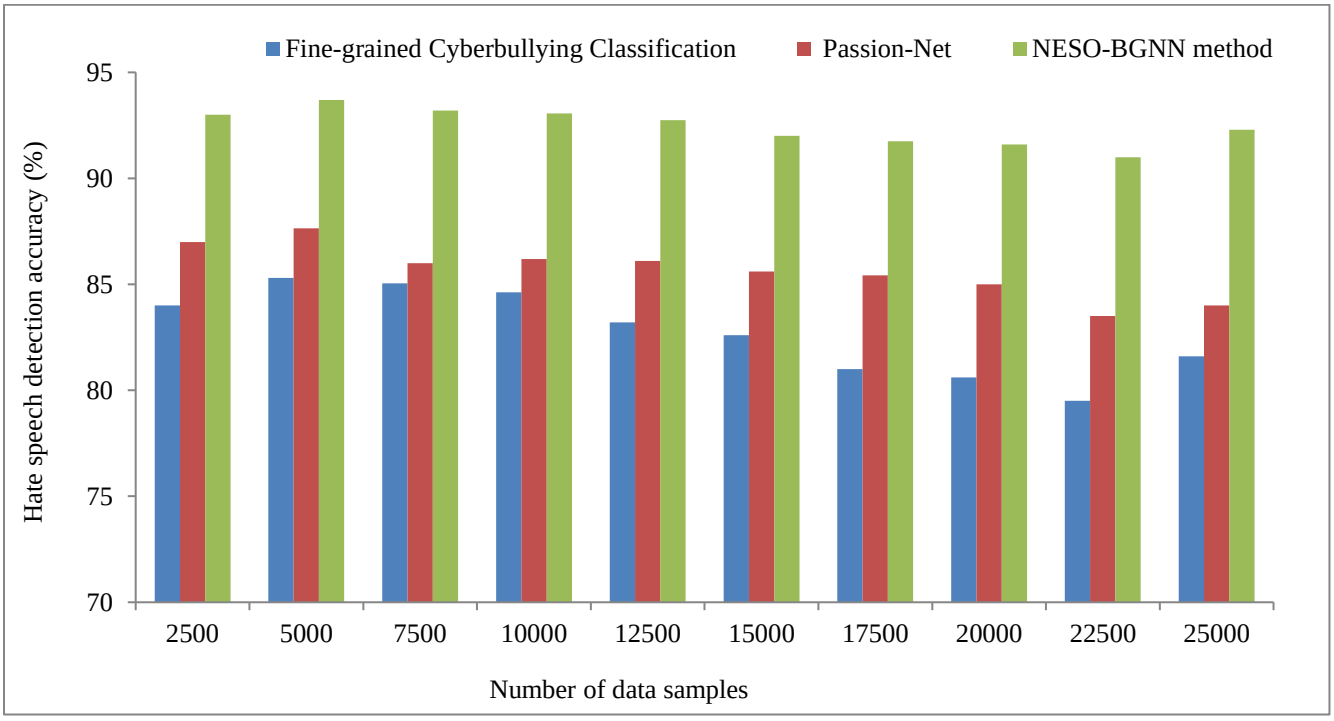


**Fig. 5 Graphical representation of Hate speech detection accuracy**

Figure 5 NESO-BGNN of accuracy improved by 12% and 8% over [1, 2].

**Table 3. Precision**

| Number of Data Samples | Precision (%) | | |
|---|---|---|---|
| | Fine-Grained Cyberbullying Classification | Passion-Net | NESO-BGNN Method |
| 2500 | 85.5 | 88.05 | 94.98 |
| 5000 | 85.96 | 88.36 | 95 |
| 7500 | 85.7 | 86.9 | 94.7 |
| 10000 | 85 | 87.54 | 94.36 |
| 12500 | 84.7 | 87 | 94.11 |
| 15000 | 83.14 | 86.9 | 93.8 |
| 17500 | 82 | 86 | 93 |
| 20000 | 81.9 | 86.1 | 92.5 |
| 22500 | 80.45 | 85.4 | 92.1 |
| 25000 | 82 | 84.6 | 93.44 |

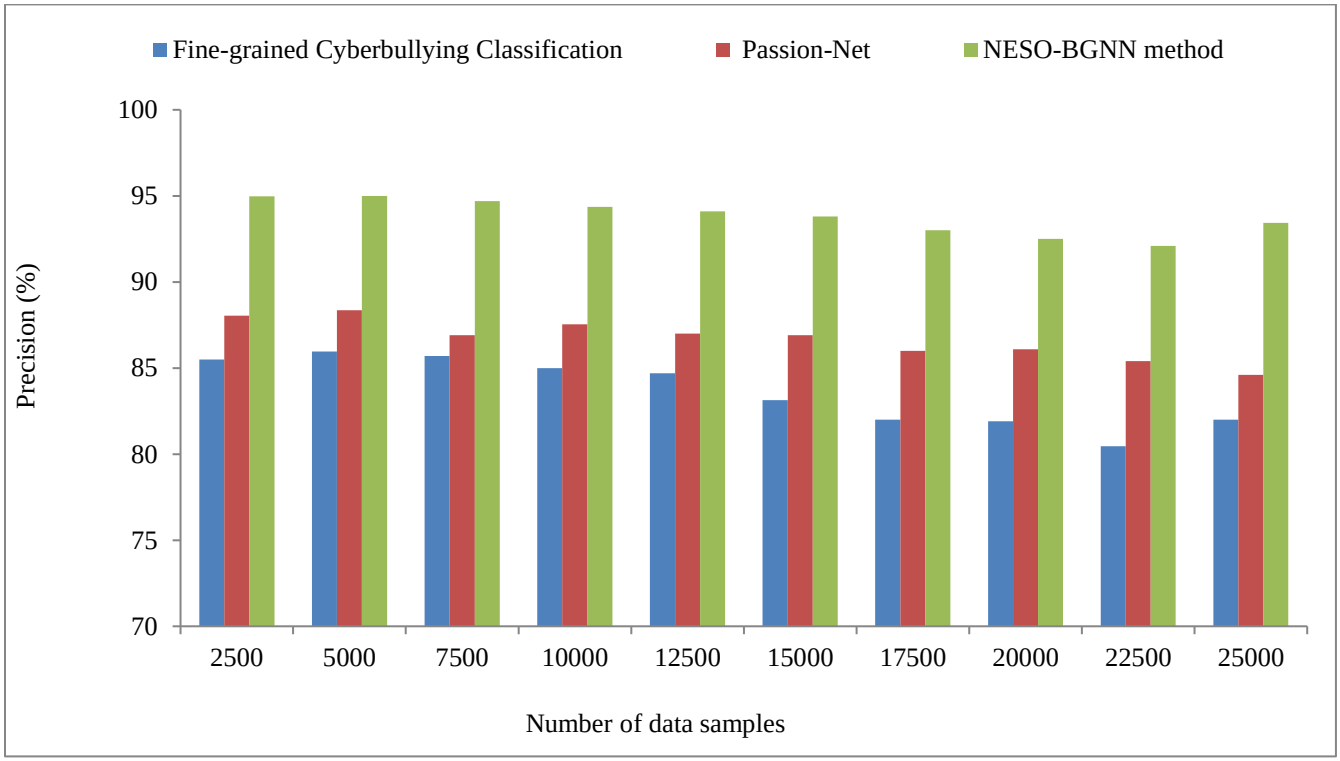


**Fig. 6 Graphical representation of Precision**

Figure 6 NESO-BGNN of precision enhanced by 12% and 8% over [1, 2].

**Table 4. Recall**

| Number of data samples | Recall (%) | | |
|---|---|---|---|
| | Fine-grained Cyberbullying Classification | Passion-Net | NESO-BGNN method |
| 2500 | 95.95 | 96.53 | 97.32 |
| 5000 | 95.12 | 96.24 | 97.21 |
| 7500 | 94.98 | 96 | 96.96 |
| 10000 | 94.75 | 95.89 | 96.24 |
| 12500 | 94.32 | 95.64 | 96.84 |

| | | | |
|---|---|---|---|
| 15000 | 93.68 | 95.5 | 97.27 |
| 17500 | 93.21 | 95.21 | 96.87 |
| 20000 | 92.8 | 95.44 | 96.57 |
| 22500 | 92.4 | 95.22 | 96.34 |
| 25000 | 92 | 95.24 | 96.15 |

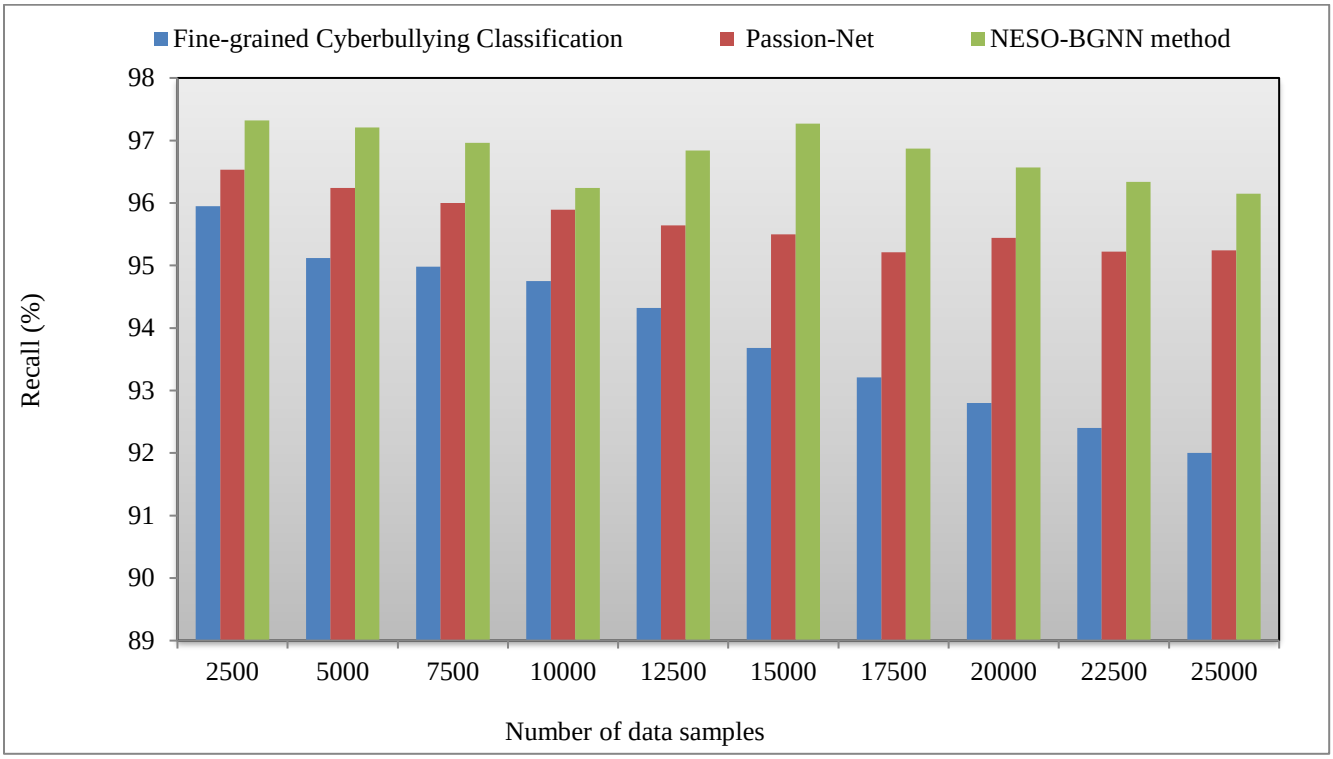


**Fig. 7 Graphical representation of recall**

Figure 7 NESO-BGNN of recall improved by 3% and 1.1% compared to [1, 2].

**Table 5. Hate speech detection time**

| Number of data samples | Detection time (ms) | | |
|---|---|---|---|
| | Fine-grained Cyberbullying Classification | Passion-Net | NESO-BGNN method |
| 2500 | 47.5 | 40 | 32.5 |
| 5000 | 51.2 | 43.2 | 35 |
| 7500 | 53.2 | 46.5 | 37.4 |
| 10000 | 55 | 48 | 39.2 |
| 12500 | 58.4 | 50.5 | 41.6 |
| 15000 | 61 | 52.8 | 43.5 |
| 17500 | 62.9 | 56 | 46 |
| 20000 | 64.5 | 59.7 | 48.2 |
| 22500 | 66.8 | 63.5 | 51.2 |
| 25000 | 68 | 65.3 | 53 |

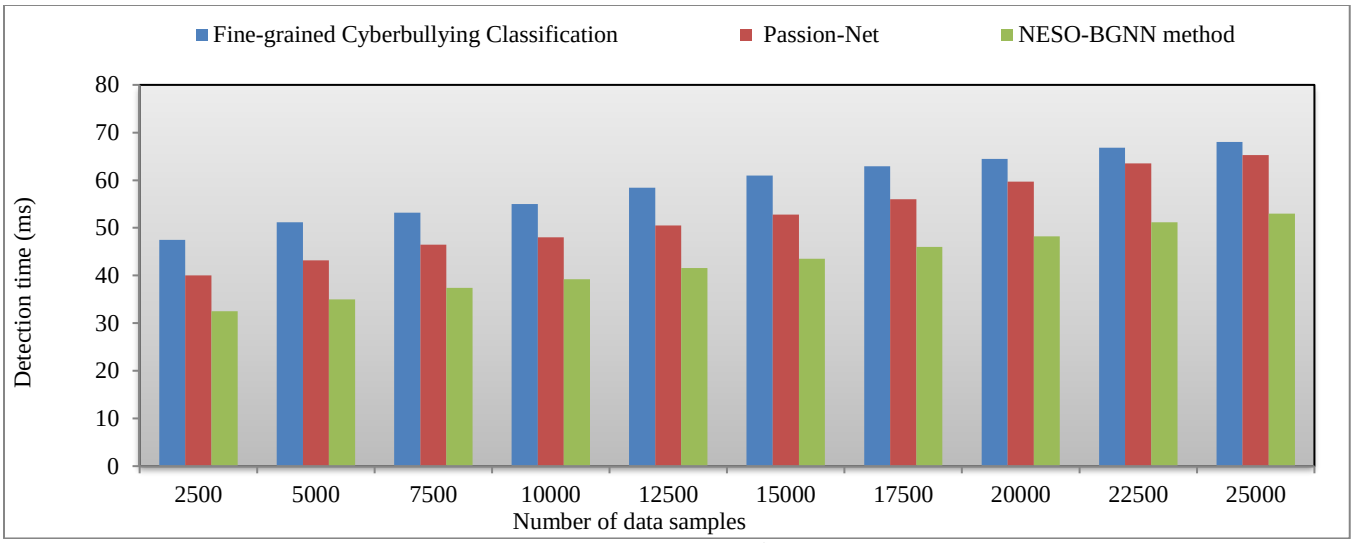


**Fig. 8 Graphical representation of detection time**

Time is shown in Figure 8. Detection time is minimized by 28% and 19% over [1, 2].

**Table 6. Misclassification rate**

| Number of data samples | Misclassification rate (%) | | |
|---|---|---|---|
| | Fine-grained Cyberbullying Classification | Passion-Net | NESO-BGNN method |
| 2500 | 16 | 13 | 7 |
| 5000 | 14.7 | 12.35 | 6.3 |
| 7500 | 14.95 | 14 | 6.8 |
| 10000 | 15.37 | 13.8 | 6.94 |
| 12500 | 16.8 | 13.9 | 7.25 |
| 15000 | 17.4 | 14.4 | 8 |
| 17500 | 19 | 14.58 | 8.25 |
| 20000 | 19.4 | 15 | 8.4 |
| 22500 | 20.5 | 16.5 | 9 |
| 25000 | 18.4 | 16 | 7.7 |

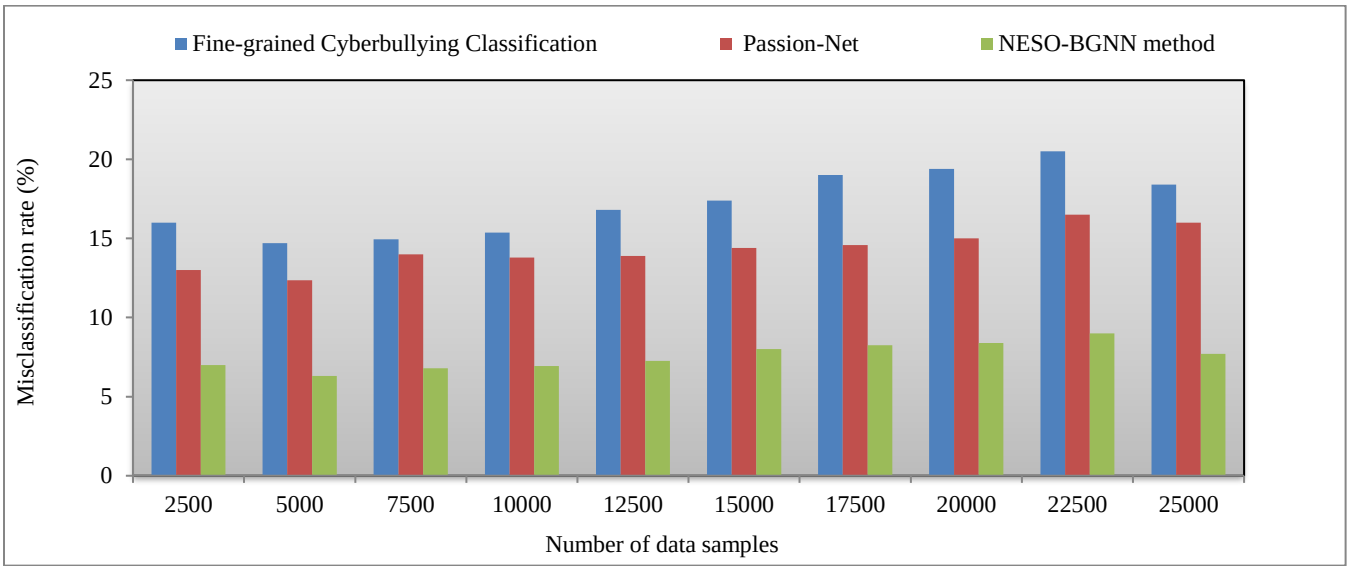


**Fig. 9 Graphical representation of misclassification rate**

Figure 9 depicts the misclassification rate. It was reduced by 56% and 47% over [1, 2].

### *6.1. Overall Comparison of the Proposed Model*

The overall comparison of the proposed method compared to existing [1, 2] was conducted on different parameters, namely, accuracy, precision, recall, time, and misclassification rate.

**Table 7. Overall comparison table**

| Parameters | Methods | | |
|---|---|---|---|
| | Fine-grained Cyberbullying Classification | Passion-Net | NESO-BGNN method |
| Accuracy | 82.74 | 77.04 | 92.49 |
| Precision | 75.42 | 86.68 | 93.69 |
| Recall | 93.62 | 95.69 | 96.77 |
| Time | 58.85 | 52.55 | 42.76 |
| Misclassification rate | 17.25 | 14.36 | 7.56 |

### *6.2. Ethical Considerations*

OSN includes privacy protection, preventing misinformation and cyberbullying, and maintaining accountability and transparency. Users must respect others' privacy, verify information before sharing, and be mindful of the impact of their posts, while platforms must address algorithmic bias and protect vulnerable users like minors. Companies also have ethical duties regarding data usage and authentic marketing. AI guidelines for online social networks are centered on ensuring transparency, accountability, fairness, privacy, and human oversight. Mitigation steps involve a combination of regulatory frameworks, technical controls (like data validation and stress testing), and human-in-the-loop processes to manage risks such as misinformation, bias, and harmful content. Mitigation involves policy, technology, and human processes. Steps to address risks include automated detection and human moderation for misinformation, regular audits for algorithmic bias, access controls for data privacy, detection for deepfakes, risk assessments for system failures, and clear policies for accountability.

## 7. Discussion

This study compares the proposed NESO-BGNN Technique with the existing Fine-grained Cyberbullying Classification [1] and Passion-Net [2] using Hate Speech and Offensive Language Dataset based on different parameters, namely, accuracy, precision, recall, training time, and misclassification rate with respect to data samples. Initially, preprocessing using First Robust Scaling Normalization was applied to the raw dataset employing median and IQR. In the Seagull Optimization algorithm, the Nonlinear Evolutionary algorithm was used to remove the keywords that are more suitable for hate speech detection. Finally, a BGRNN is designed to detect hate speech in a precise manner with a lower misclassification rate. The results confirm that the proposed NESO-BGNN improves the accuracy and precision by 10%, 2% of recall with 24% and 51% lesser time and misclassification rate compared to existing [1, 2] using the Hate Speech and Offensive Language Dataset.

## 8. Conclusion

The proposed NESO-BGNN method is introduced for detecting hate speech accurately with minimum time and MR. In NESO-BGNN, Robust Scaling Normalization, a nonlinear evolutionary-based seagull optimization algorithm, and BGRNN are employed for accurate detection of hate speech. Experiments are done with different metrics. The results of NESO-BGNN achieve higher accuracy with minimal time for hate speech detection compared to other methods. The limitations of online social networks have privacy and security risks like data breaches and fraud, negative mental health impacts such as increased anxiety, depression, and addiction, and social issues like the spread of misinformation, cyberbullying, and reduced face-to-face interaction. These platforms can also foster superficial connections and distract from real-world responsibilities, hurting productivity and overall well-being. In the future, online social networks are likely to feature an integration of AI, immersive experiences (VR/AR), and a shift toward niche communities and social commerce. AI will personalize content, automate tasks, and improve user safety, while AR and VR will create more three-dimensional and engaging virtual spaces. The landscape will also evolve towards smaller, more private groups and direct in-app purchasing, alongside a greater focus on user control over data and transparency.

## References

[1] Yasmine M. Ibrahim, Reem Essameldin, and Saad M. Saad, "Social Media Forensics: An Adaptive Cyberbullying-Related Hate Speech Detection Approach based on Neural Networks with Uncertainty," *IEEE Access*, vol. 12, pp. 59474-59484, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Faiza Mehmood et al., "Passion-Net: A Robust Precise and Explainable Predictor for Hate Speech Detection in Roman Urdu Text," *Neural Computing and Applications*, vol. 36, no. 6, pp. 3077-3100, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Francimaria R.S. Nascimento, George D.C. Cavalcanti, and Marjory Da Costa-Abreu, "Exploring Automatic Hate Speech Detection on Social Media: A Focus on Content-Based Analysis," *Sage Open*, vol. 13, no. 2, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Waqas Sharif et al., "Enhancing Hate Speech Detection in the Digital Age: A Novel Model Fusion Approach Leveraging a Comprehensive Dataset," *IEEE Access*, vol. 12, pp. 27225-27236, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Staša Vujičić Stanković, and Miljana Mladenović, "An Approach to Automatic Classification of Hate Speech in Sports Domain on Social Media," *Journal of Big Data*, vol. 10, no. 1, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Aigerim Toktarova et al., "Hate Speech Detection in Social Networks using Machine Learning and Deep Learning Methods," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, pp. 396-406, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Md Saroar Jahan, and Mourad Oussalah, "A Systematic Review of Hate Speech Automatic Detection using Natural Language Processing," *Neurocomputing*, vol. 546, pp. 1-30, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zainab Mansur, Nazlia Omar, and Sabrina Tiun, "Twitter Hate Speech Detection: A Systematic Review of Methods, Taxonomy Analysis, Challenges, and Opportunities," *IEEE Access*, vol. 11, pp. 16226-16249, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Olumide Ebenezer Ojo et al., "Automatic Hate Speech Detection using Deep Neural Networks and Word Embedding," *Computing and Systems*, vol. 26, no. 2, pp. 1007-1013, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Lanqin Yuan et al., "Transfer Learning for Hate Speech Detection in Social Media," *Journal of Computational Social Science*, vol. 6, no. 2, pp. 1081-1101, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Krishanu Maity et al., "A Deep Learning Framework for the Detection of Malay Hate Speech," *IEEE Access*, vol. 11, pp. 79542-79552, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ashwin Geet D'sa, Irina Illina, and Dominique Fohr, "Classification of Hate Speech Using Deep Neural Networks," *Data and Information Processing to Knowledge Organization: Architectures, Models and Systems*, vol. 25, no. 1, pp. 1-12, 2020. [Google Scholar] [Publisher Link]

[13] Ashfia Jannat Keya et al., "G-BERT: An Efficient Method for Identifying Hate Speech in Bengali Texts on Social Media," *IEEE Access*, vol. 11, pp. 79697-79709, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Nina Sevani et al., "Detection of Hate Speech by Employing Support Vector Machine with Word2Vec Model," *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Malang, Indonesia, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Kevin Usmayadhy Wijaya, and Erwin Budi Setiawan, "Hate Speech Detection using Convolutional Neural Network and Gated Recurrent Unit with FastText Feature Expansion on Twitter," *Scientific Journal of Electrical Engineering, Computer Science and Informatics*, vol. 9, no. 3, pp. 619-631, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Ehtesham Hashmi, and Sule Yildirim Yayilgan, "Multi-Class Hate Speech Detection in the Norwegian Language using FAST-RNN and Multilingual Fine-Tuned Transformers," *Complex & Intelligent Systems*, vol. 10, no. 3, pp. 4535-4556, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Alaa Marshan et al., "Comparing Machine Learning and Deep Learning Techniques for Text Analytics: Detecting the Severity of Hate Comments Online," *Information Systems Frontiers*, vol. 27, no. 2, pp. 487-505, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Malliga Subramanian et al., "A Survey on Hate Speech Detection and Sentiment Analysis using Machine Learning and Deep Learning Models," *Alexandria Engineering Journal*, vol. 80, pp. 110-121, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Gretel Liz De la Pena Sarracen, and Paolo Rosso, "Systematic Keyword and Bias Analyses in Hate Speech Detection," *Information Processing and Management*, vol. 60, no. 5, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Chandan Senapati, and Utpal Roy, "Bengali Hate Speech Detection using Deep Learning Technique," *CEUR Workshop Proceedings: Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation (FIRE-WN 2023)*, Goa, India, pp. 553-562, 2021. [Google Scholar] [Publisher Link]

[21] Ashraf Ullah et al., "Threatening Language Detection from Urdu Data with Deep Sequential Model," *Plos One*, vol. 19, no. 6, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] György Kovács, Pedro Alonso, and Rajkumar Saini, "Challenges of Hate Speech Detection in Social Media: Data Scarcity, and Leveraging External Resources," *SN Computer Science*, vol. 2, no. 2, pp. 4-15, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Shankar Biradar, Sunil Saumya, and Arun chauhan, "Fighting Hate Speech from Bilingual Hinglish Speaker's Perspective, a Transformer and Translation based Approach," *Social Network Analysis and Mining*, vol. 12, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]