Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.034798
Article

Improved Attentive Recurrent Network for Applied Linguistics-Based Offensive Speech Detection

Manar Ahmed Hamza[1,*], Hala J. Alshahrani[2], Khaled Tarmissi[3], Ayman Yafoz[4], Amira Sayed A. Aziz[5], Mohammad Mahzari[6], Abu Sarwar Zamani[1] and Ishfaq Yaseen[1]

[1]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[2]Department of Applied Linguistics, College of Languages, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Makkah, Saudi Arabia
[4]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[5]Department of Digital Media, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[6]Department of English, College of Science & Humanities, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa
Received: 27 July 2022; Accepted: 13 November 2022; Published: 28 July 2023

Abstract: Applied linguistics is one of the fields in the linguistics domain and deals with the practical applications of the language studies such as speech processing, language teaching, translation and speech therapy. The ever-growing Online Social Networks (OSNs) experience a vital issue to confront, i.e., hate speech. Amongst the OSN-oriented security problems, the usage of offensive language is the most important threat that is prevalently found across the Internet. Based on the group targeted, the offensive language varies in terms of adult content, hate speech, racism, cyberbullying, abuse, trolling and profanity. Amongst these, hate speech is the most intimidating form of using offensive language in which the targeted groups or individuals are intimidated with the intent of creating harm, social chaos or violence. Machine Learning (ML) techniques have recently been applied to recognize hate speech-related content. The current research article introduces a Grasshopper Optimization with an Attentive Recurrent Network for Offensive Speech Detection (GOARN-OSD) model for social media. The GOARN-OSD technique integrates the concepts of DL and metaheuristic algorithms for detecting hate speech. In the presented GOARN-OSD technique, the primary stage involves the data pre-processing and word embedding processes. Then, this study utilizes the Attentive Recurrent Network (ARN) model for hate speech recognition and classification. At last, the Grasshopper Optimization Algorithm (GOA) is exploited as a hyperparameter optimizer to boost the performance of the hate speech recognition process. To depict the promising performance of the proposed GOARN-OSD method, a widespread experimental analysis was conducted. The comparison study outcomes demonstrate

the superior performance of the proposed GOARN-OSD model over other state-of-the-art approaches.

Keywords: Applied linguistics; hate speech; offensive language; natural language processing; deep learning; grasshopper optimization algorithm

1 Introduction

Brand advertising and mass communication about the products and services of a company have been digitalized in this modern era. This phenomenon made several companies pay more attention to hate speech content than ever before [1]. Online hate speech content can be described as messages conveyed in discriminatory or pejorative language. Though the companies can control the content released on their social media channels and their website, it is impossible for them to completely control the online users' comments or their posts regarding their brand [2]. In simple terms, hate speech can be found in written communication, behaviour, or speech. It uses or attacks an individual or a group of people, or an organization using discriminatory or pejorative language about certain delicate data or protected features [3]. Such protected features include colour, religion, nationality, health status, ethnicity, disability, sexual orientation, marital status, descent, gender or race and other identity factors [4]. Hate speech is an illegal and dangerous act that should be discouraged at all levels. In addition to the content, both sounds and images are also utilized in the distribution of hate speech [5]. Hence, computer-based text classification is considered an optimal solution to overcome this issue.

There is no universal definition available for hate speech. Because precise and clear hate speech definition can shorten the annotator's effort, increasing the annotator's agreement rate [6]. However, it is hard to distinguish hate speech from normal speech. So, it is challenging to provide a universal and precise definition for hate speech [7]. Cyberbullying is a type of online-based harassment that involves repeated hostile behaviour towards a person or a group of individuals who are unable to defend themselves, mostly adolescents. This hostile behaviour is deliberately expressed upon the victims to hurt or threaten them [8]. Cyberbullying is also considered a type of hate speech if a victim's sensitive features are targeted during the assault. Hate speech can be differentiated from cyber-bullying in such a manner that hate speech affects an individual and has consequences for society or a whole group [9]. Hate speech is a complex and multi-faceted concept that is difficult to understand by computer systems and human beings. The prevailing literature contains numerous Deep Learning (DL) techniques for detecting hate speech [10].

In this background, the authors have offered various DL models with the help of Neural Network (NN) elements such as the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model. The existing DL methods generally utilize a single NN element and a set of certain user-defined features as additional features. In general, different DL elements are effective over different types of datasets. Automatic hate speech recognition becomes essential for avoiding spreading hate speech, mainly in social media. Several approaches have been presented to carry out the offensive speech detection process, comprising the latest DL-based models advancements. Various datasets have also been developed, exemplifying various manifestations of the hate-speech detection problem.

In the earlier study [11], the author used a multilingual and multi-task technique relevant to the newly-projected Transformer Neural Network to resolve three sub-tasks for hate speech recognition. These tasks are nothing but the tasks shared on Hate Speech and Offensive Content (HASOC)

detection in Indo-European languages during the year 2019. The author expanded the submissions to the competition using multitasking techniques. These techniques can be trained with the help of three techniques: multilingual training, multitask learning with distinct task heads, and the back-translation method. Khan et al. [12] introduced a hate detection technique for mass media with a multi-label complexity. For this purpose, the author devised a CNN-related service structure called 'HateClassify' to categorize the mass media content as hate speech, non-offensive and offensive.

In literature [13], the author presented a Transfer Learning (TL) method for hate speech recognition based on an existing pre-trained language method called BERT (Bidirectional Encoder Representations from Transformers) model. The study evaluated the presented method using two publicly-available Twitter datasets that had content with hate or offensive content, racism and sexism. Then, the author presented a bias alleviation system to mitigate the impact of the bias upon the trained set when fine-tuning the pre-trained BERT-related technique for hate speech identification. In the study conducted earlier [14], the researchers collected the hate speech content, i.e., English-Odia code-varied data, from a public page on Facebook. Then, the data were classified into three classes. The hate speech recognition models use a group of extracted features and Machine Learning (ML) method. A few approaches, such as Random Forest (RF), Support Vector Machine (SVM) and Naïve Bayes (NB), were trained well using the complete data with the extracted features relevant to word unigram, Term Frequency-Inverse Document Frequency (TF-IDF), bigram, combined n-grams, word2vec, combined n-grams weighted by Term Frequency Inverse Dense Frequency (TF-IDF) and trigram for datasets. Perifanos et al. [15] presented an innovative method for hate speech recognition by combining Natural Language Processing (NLP) and Computer Vision (CV) techniques to identify offensive content. This work focused on racist, hateful and xenophobic Twitter messages about Greek migrants and refugees. The author compiled the TL method and the finely-tuned Bidirectional Encoder Representations from Transformers (BERT) and Residual Neural Networks (Resnet) models in this method.

Das et al. [16] projected a common NLP tool, i.e., encoder–decoder-related ML approach, to classify Bengali users' comments on Facebook pages. In this study, the one-dimensional convolutional layer was employed to encode and extract the local features from the comments. At last, the attention systems such as the GRU-related decoders and the LSTM approach were employed to estimate the hate speech classes. Al-Makhadmeh et al. [17] presented a technique in which a hybrid of NLP and ML methods was used to determine the hate speech on mass media sites. In this study, the hate speech content was accumulated. The data went through different processes such as character removal, stemming, inflection elimination and token splitting, before executing the hate speech identification procedure. Then, the gathered data was scrutinized using an ensemble deep learning technique. The model identified the hate speech on Social Networking Sites (SNS) with the help of a proficient learning procedure that categorized the messages into hate language, neutral and violent.

Though several models are available in the literature for offensive speech detection, it is still needed to improve the detection performance. Since the trial-and-error hyperparameter tuning process is tedious, metaheuristic algorithms can be employed. Therefore, the current research article introduces a Grasshopper Optimization with an Attentive Recurrent Network for Offensive Speech Detection (GOARN-OSD) model for social media. The presented GOARN-OSD technique integrates the concepts of DL and metaheuristic algorithms for hate speech detection. In the proposed GOARN-OSD technique, data pre-processing and word embedding are carried out in the primary stage. The ARN model is utilized in this study for hate speech recognition and its classification. At last, the Grasshopper Optimization Algorithm (GOA) is exploited as a hyperparameter optimizer to boost the performance of hate speech recognition, showing the novelty of the work. A widespread experimental

analysis was conducted to establish the promising performance of the proposed GOARN-OSD technique.

2 Proposed Offensive Speech Detection Model

In this article, a new GOARN-OSD method has been developed to detect offensive speeches on social media. The presented GOARN-OSD technique integrates the concepts of DL and metaheuristic algorithms for hate speech detection. Fig. 1 illustrates the block diagram of the proposed GOARN-OSD approach.

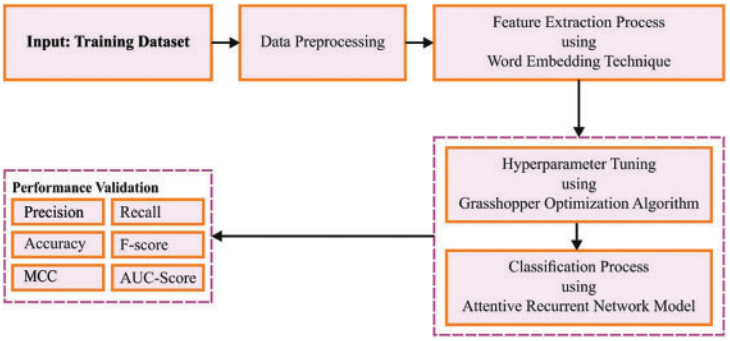

Figure 1: Block diagram of the GOARN-OSD approach

2.1 Data Pre-Processing

In the proposed GOARN-OSD technique, the data is pre-processed in the primary stage. The pre-processing step filters the irrelevant and noisy contents from the data. At first, all the duplicate tweets are filtered since it does not feed any dataset to the module. During the pre-processing stage, different Twitter-specific symbols and noises, namely, mention (@), retweets (RT), hashtags (#) and the URLs, are filtered. Furthermore, the alphanumeric symbols such as dots, ampersands, non-ASCII characters, stop-words and commas are filtered to prevent noisy content from being used. At last, the pre-processed tweets are transformed into lower case to avoid ambiguity.

2.2 Word Embeddings

Most DL approaches for text classification exploit the word embedding approach to extract effective and discriminative characteristics. It is a process of mapping the words into real, fixed and dimension vectors to capture the syntactic and semantic dataset for the words and initialize the weight of the first NNS layer. Its superiority considerably impacts the learner's performance [18]. The author-trained CNN classifier is used for contextualized word embedding (MBert and AraBert) and non-contextualized word embedding (FastText-SkipGram) processes.

FastText-SkipGram: It is trained using large Arabic corpora viz., a group of Wikipedia dump containing three million Arabic sentences, United Nations corpora of 6.5 million Arabic tweets, and other corpora of 9.9 million Arabic sentences. The embedded words contain word vectors with more than a million words under 300 dimensions.

Multilingual Bert: It is a contextualized word-embedding process that works on sub-word levels to generate the feature vectors for the words. Unlike the static non-contextualized word-embedding process, this Bert process captures both long- and short-span contextual dependencies in the input

text using a bidirectional self-attention transformer. In this work, the multilingual version of the Bert (MBert) is employed with a pre-trained monolingual Wikipedia corpus of 104 languages.

AraBert: This process employs a novel Arabic-contextualized word-embedding training process. AraBert is employed to accomplish a remarkable outcome in three Arabic NLP tasks and eight distinct data sets.

2.3 Hate Speech Classification Using ARN Model

The ARN model is utilized in this study to recognize and classify hate speech content. The Recurrent Neural Network (RNN) model efficiently mines datasets containing features. The hidden layer of the RNN model, with long-term sequence storage, has a loop that integrates the present moment's output with the following moment's input [19]. Consequently, the RNN model is considered fit to process the log datasets that differ with sedimentary faces in the in-depth direction. But, gradient explosion and disappearance problems tend to occur in real-time applications due to the fundamental structure of the RNN model. So, it has a memory function for short-term datasets. Regarding the issues mentioned earlier, the RNN variants of the Gated Recurrent Unit (GRU) and the LSTM technique are projected. The LSTM model has three gating units: the output, forget and input gates to update the input dataset and attain the capability of a long-term memory dataset. But the hidden unit of the LSTM model not only has various parameters and a complex structure and takes a long training time. In contrast to the LSTM system, the reset and update gates of the GRU model can shorten the training time, improve the network's generalization ability and decrease the number of network training parameters based on the assumption of guaranteeing the predictive performance.

The architecture of the GRU model integrates the output of the hidden state at -1 (h_{t-1}), the reset gate, the update gate (z_t), the input at t and the output of the hidden state at t as formulated below.

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right), \tag{1}$$

$$z_t = \sigma\left(W_z\left[h_{t-1},\ x_t\right] + b_z\right), \tag{2}$$

$$\widetilde{h}_t = \tanh\left(W_{\widetilde{h}}\left[h_{t-1} \circ r_t,\ x_t\right] + b_{\widetilde{h}}\right), \tag{3}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \widetilde{h}_t, \tag{4}$$

In these expressions, W_r, W_z, $W_{\widetilde{h}}$ and b_r, b_z, $b_{\widetilde{h}}$ denote the weights and the biases. Correspondingly, $\widetilde{h}_t$ indicates the novel hidden state at t, σ shows the logistic function sigmoid, and symbol $\circ$ characterizes the dot product. The reset gate control shows the amount of data preserved from the initial state. At the same time, the update gate conflicts with the reset gate function.

Even though the neural network efficiently handles the text classification tasks, a considerable shortcoming cannot be disregarded in the 'black box' methodology. In this study, a Bi-LSTM structure is described using an attention layer that permits the neork to consider words in a sentence based on their intrinsic importance. The attention LSTM mechanism can extract a portion of the subset of the presented input. At the same time, its importae can be understood on the phrase- or word-level significance of the specified queries. The intermediate outcomes of the NN output are exploited for an effectual feature selection (viz., attentiveord weight) to help the rule-based methodology construct an explainable and interpretable text classification solution.

Consider a sentence S is classified into t words, $= [I_1, \ldots, I_t]$ in which I_i signifies the $i{-}th.$ word. Further, $w_i \in R^d$ represents the vector depiction of the word I_i. In this study, the Bidirectional LSTM (Bi-LSTM) model is employed to summarize the annotation of the word embeddings from the

sentence. The Bi-LSTM model has a backward LSTM that can read from I_t to I_1 and a forward LSTM that can read from I_1 to I_t.

$$\begin{aligned}\vec{h}_i &= \overrightarrow{LSTM}\,(w_i)\,; i \in [1, t]\,,\\ \overleftarrow{h}_i &= LSTM\,(w_i)\,; i \in [t, 1]\,,\end{aligned} \tag{5}$$

Then, a word-level neural depiction is attained for the provided word I_i. by concatenating the backward hidden stage $\overleftarrow{h}_i$. and the hidden forward state $\vec{h}_i$ as follows.

$$h_i = \left[\vec{h}_i; \overleftarrow{h}_i\right], \tag{6}$$

The Bi-LSTM neural unit summarizes the data of the entire sentence, S. In the conventional LSTM method, the vectors $\vec{h}_t$ and $\overleftarrow{h}_1$ are generally concatenated as a text depiction that barely captures the data regarding the significance of every word in the entire sentence. Because the word that reflects the subject is mostly a keyword, while every word's signifance is distinct. As a result, an attention model is introduced in this study to tackle the importance of the 'hot' word to the meaning of the sentence. The subsequent formula is employed toerne t attention weight α_i between the sentence S and the $i-th$ word.

$$u_i = tanh\,(W_w h_i + b_w) \tag{7}$$

$$\alpha_i = softmax\,(u_i) = \frac{exp\,(u_i u_w)}{\sum_i exp\,(u_i u_w)} \tag{8}$$

$$h_a = \sum_{i=1}^{t} \alpha_i h_i \tag{9}$$

Now W_w and u_w denote the projection variables, b_w indicates the bias variable and h_a shows the resultant weight feature that reviews each word-level data. Next, h_a and LSTM forward and backward outcomes form the vector depiction of the text as given below.

$$s = \left[h_a; \vec{h}_t; \overleftarrow{h}_1\right], \tag{10}$$

In Eq. (10), s represents a high-level depiction of the sentence that is applied as a concluding feature for the prediction of a label y to the classification of a sentence using a *softmax* layer.

$$y = softmax\,(W_s s + b_s) \tag{11}$$

In Eq. (11), $\hat{y}$ represents the ground truth of a class label, whereas the training process aims to mitigate the cross-entropy error between $\hat{y}$ and the ground truth y for each trained dataset.

$$loss = -\sum_k \sum_j y^j log\hat{y}^j + \lambda \|\theta\|^2 \tag{12}$$

Eq. (12), k denotes the index of the sentence, j shows the index of the class, λ specifies the L_2-regularization term, and θ indicates the variable set.

2.4 Hyperparameter Optimization

At last, the GOA is exploited as a hyperparameter optimizer to boost hate speech recognition performance. In this work, the GOA method integrates the behaviour of Grasshoppers (GH) [20]. Grasshopper is a parasite that affects farming and agricultural practices. Its lifespan encompasses

three phases such as adulthood, egg and the nymph. In the first phase, the grasshoppers exhibit running and hopping behaviours in spinning barrels (at a slow motion with small increments) for ingratiation; they eat plants originating during their migration. The grasshoppers migrate to a long distance as colonies in their adult lifespan with long and unexpected movements. This phenomenon is shown in the following equation.

$$a_k = P_k + Q_k + R_k,\ k = 1, 2,\ \ldots, N \tag{13}$$

In Eq. (13), P_k represents the *k-th* GH social interaction.

$$P_k = \sum_{l=1, i\neq j}^{N} h\left(g_{kl}\right) \hat{g}_{kl},\ g_{kl} = |a_k - a_l| \tag{14}$$

In Eq. (14), g_{kl} represents the distance between *kth*, and the *lth* GHs, and h represent the social forces' function strength as follows:

$$h\left(b\right) = we^{\frac{-b}{m}} - e^{-b} \tag{15}$$

In Eq. (15), w and b show the attractive distance scale and attractiveness threshold. In Eq. (13), Q_k and R_k indicate the gravitational force and the wind direction of the *kth* GH.

$$Q_k = -q\hat{e}_q,\ R_k = y\hat{e}_s \tag{16}$$

In Eq. (16), q and y denote the constant gravitation and the drifting constant, while e_q and e_s symbolize a unit vector towards the earth's centre and the wind direction. In Eq. (13), the main optimization problem cannot be directly attributed to the solution; hence, it is rewritten as follows.

$$a_k = z\left(\sum_{l=1, k\neq l}^{N} z\frac{ub - lb}{2} h\left(|r_l - r_k|\right)\frac{r_l - r_k}{g_{kl}}\right) + \hat{Z}_g \tag{17}$$

In Eq. (17), *ub* and *lb* indicate the lower and the upper bounds of the search space, and Z_g signifies the optimum solution, h is shown in Eq. (15), and the gravity is considered. In contrast, the wind direction towards $\hat{Z}_g$ is frequently observed. Fig. 2 displays the flowchart of the GOA approach.

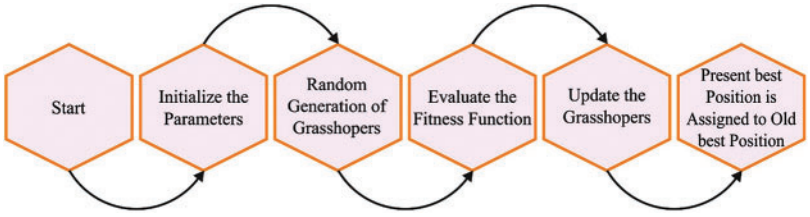

Figure 2: Flowchart of the GOA approach

Now, z refers to a critical factor that significantly decreases the zone of attraction, the comfort zone and the region of repellence, as given below.

$$z = z_{\max} - p\frac{z_{\max} - z_{\min}}{p_{\max}} \tag{18}$$

In Eq. (18), $z_{\max}$ and $z_{\min}$ denote the highest value (equivalent to 1) and least value (equivalent to 0.00001) of z correspondingly, p signifies the current iteration, and $p_{\max}$ symbolizes the highest number of iterations.

The pseudocode of the GOA approach involves Algorithm 1, whereas the GH begins with the implementation of an arbitrary population A-sized N. The subsequent phases characterize the fitness function calculation for each solution a_k, $k = 1, \ldots, N$. Then, the best solution $\hat{Z}_g$ is selected based on the best fitness function. The following two steps are expressed for each a_k:

1. Standardization of the distance for solution A within [1,4].
2. Update $a_k \in A$ using Eq. (17). The next step is to update the original iteration and repeat the previous steps till the ending condition is met.

Algorithm 1: Pseudocode of the GOA approach

1. Instantiation of the variable value of the size of population (N), $z_{\max}$, $z_{\min}$ and cumulative iteration number ($p_{\max}$)
2. Establish a population (A) randomly
3. Set the existing iteration $p = 1$
4. While $(p < p_{max\}})do$
5. Calculation of f fitness function
6. $\hat{Z}_g$ was exploited for the choice of the optimum solution
7. Updating of value Z based on Eq. (18)
8. *For* $k = 1 : Ndo$
9. Standardizes the distance amongst solutions in A within [1,4]
10. Updating of $a_k \in A$ based on Eq. (17)
11. End for
12. $p = p + 1$
13. End while
14. Return $\hat{Z}_g$.

3 Results and Discussion

This section examines the classification performance of the GOARN-OSD model on the Twitter dataset. The dataset has a total of 2,000 tweets under two class labels and the details about the dataset are depicted in Table 1.

Table 1: Dataset details

Class	No. of tweets
Hate	1000
Normal	1000
Total Number of Tweets	**2000**

In Fig. 3, the confusion matrices generated by the proposed GOARN-OSD model are depicted under distinct Training (TR) and Testing (TS) dataset values. On 80% of TR data, the GOARN-OSD model recognized 738 samples as hate class and 742 samples as normal class. Moreover, on 20% of TS data, the proposed GOARN-OSD approach categorized 205 samples under the hate class and 174 samples under the normal class. Along with that, on 70% of TR data, the GOARN-OSD approach classified 697 samples under hate class and 624 samples under normal class. Then, on 30% of TS data, the GOARN-OSD method recognized 293 samples as hate class and 270 samples as normal class.

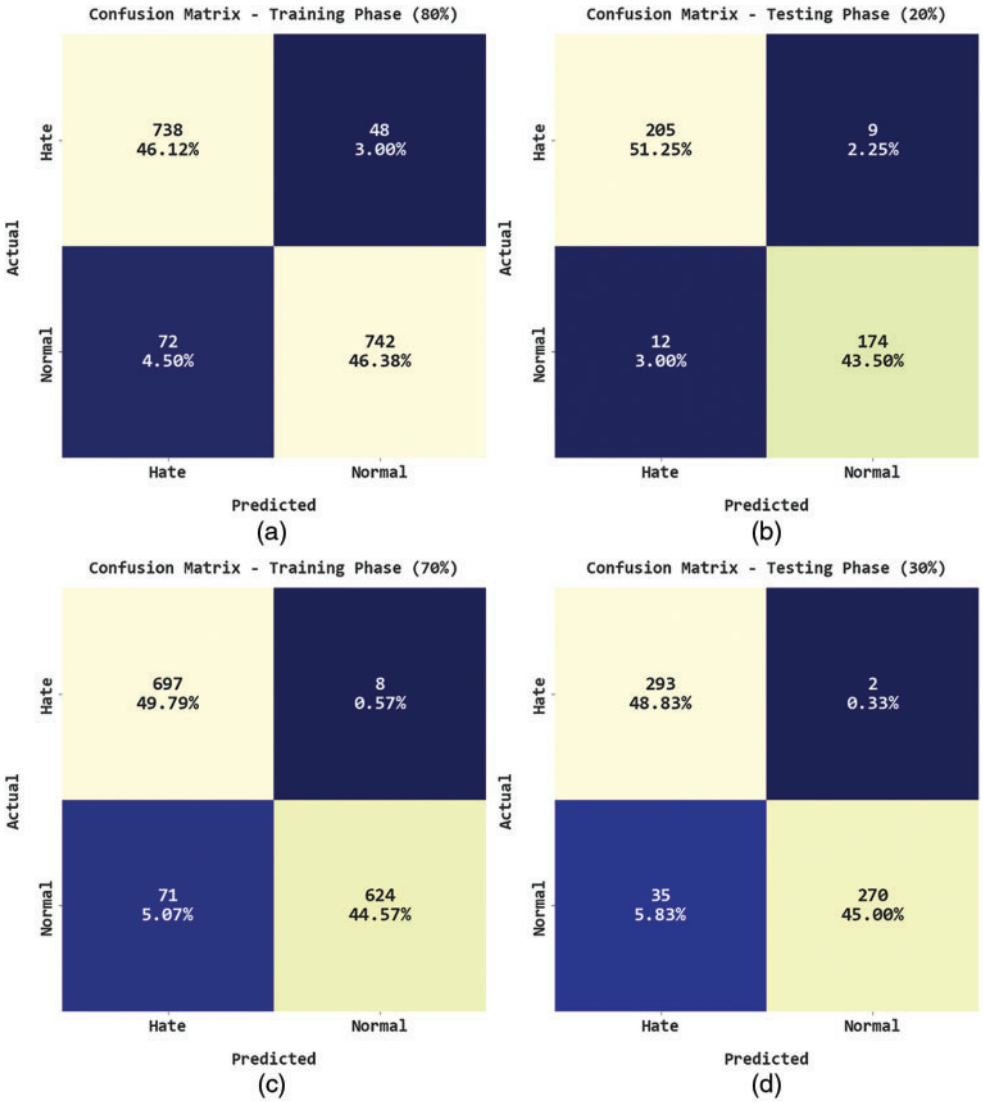

Figure 3: Confusion matrices of the GOARN-OSD approach (a) 80% of TR data, (b) 20% of TS data, (c) 70% of TR data, and (d) 30% of TS data

Table 2 presents the overall experimental results of the proposed GOARN-OSD method on 80% of TR data and 20% of TS datasets.

Table 2: Analytical results of the GOARN-OSD approach under 80:20 of TR/TS data

Labels	Accuracy	Precision	Recall	F-Score	AUC Score	MCC
Training Phase (80%)						
Hate	92.50	91.11	93.89	92.48	92.52	85.04

(Continued)

Table 2 (continued)

Labels	Accuracy	Precision	Recall	F-Score	AUC Score	MCC
Normal	92.50	93.92	91.15	92.52	92.52	85.04
Average	**92.50**	**92.52**	**92.52**	**92.50**	**92.52**	**85.04**
Testing Phase (20%)						
Hate	94.75	94.47	95.79	95.13	94.67	89.45
Normal	94.75	95.08	93.55	94.31	94.67	89.45
Average	**94.75**	**94.78**	**94.67**	**94.72**	**94.67**	**89.45**

Fig. 4 reveals the overall offensive speech classification performance of the GOARN-OSD model on 80% of TR data. The GOARN-OSD model recognized the samples into hate class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 92.50%, 91.11%, 93.89%, 92.48%, 92.52% and 85.04% respectively. Further, the proposed GOARN-OSD method categorized the samples as normal class with u_y $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 92.50%, 93.92%, 91.15%, 92.52%, 92.52% and 85.04% correspondingly. In addition, the proposed GOARN-OSD technique classified the samples into average class with *us*, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 92.50%, 92.52%, 92.52%, 92.50%, 92.52% and 85.04% correspondingly.

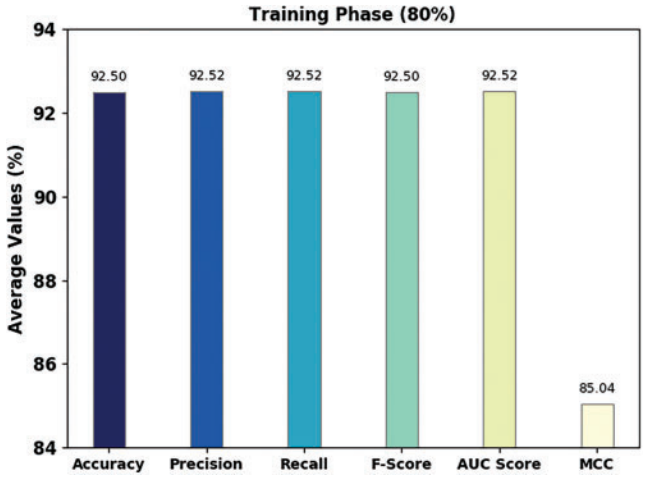

Figure 4: Average analysis results of the GOARN-OSD approach under 80% of TR data

Fig. 5 shows the complete offensive speech classification performance achieved by the proposed GOARN-OSD technique on 20% of TS data. The GOARN-OSD methodology recognized the samples into hate class with u_y $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 94.75%, 94.47%, 95.79%, 95.13%, 94.67% and 89.45% respectively. Further, the proposed GOARN-OSD technique recognized the samples into normal class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 94.75%, 95.08%, 93.55%, 94.31%, 94.67% and 89.45% correspondingly. Likewise, the GOARN-OSD approach classified the samples as average class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 94.75%, 94.78%, 94.67%, 94.72%, 94.67% and 89.45% correspondingly.

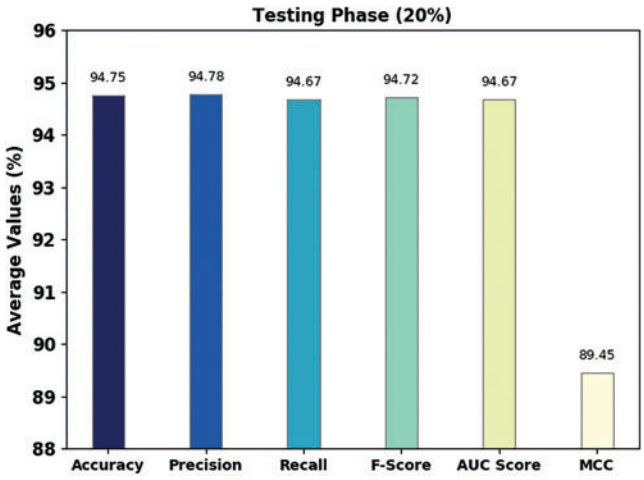

Figure 5: Average analysis results of the GOARN-OSD approach under 20% of TS data

Table 3 presents the detailed experimental results of the GOARN-OSD method on 80% of TR data and 20% of TS datasets. Fig. 6 displays the detailed offensive speech classification performance of the proposed GOARN-OSD technique on 70% of TR data. The GOARN-OSD approach recognized the samples under hate class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 94.36%, 90.76%, 98.87%, 94.64%, 94.32% and 89.07% correspondingly. Additionally, the proposed GOARN-OSD methodology recognized the samples as normal class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 94.36%, 98.73%, 89.78%, 94.05%, 94.32% and 89.07% correspondingly. Moreover, the GOARN-OSD algorithm recognized the samples as average class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC of 94.36%, 94.74%, 94.32%, 94.34%, 94.32% and 89.07% correspondingly.

Table 3: Analytical results of the GOARN-OSD approach under 70:30 of TR/TS data

Labels	Accuracy	Precision	Recall	F-Score	AUC Score	MCC
Training Phase (70%)						
Hate	94.36	90.76	98.87	94.64	94.32	89.07
Normal	94.36	98.73	89.78	94.05	94.32	89.07
Average	**94.36**	**94.74**	**94.32**	**94.34**	**94.32**	**89.07**
Testing Phase (30%)						
Hate	93.83	89.33	99.32	94.06	93.92	88.22
Normal	93.83	99.26	88.52	93.59	93.92	88.22
Average	**93.83**	**94.30**	**93.92**	**93.82**	**93.92**	**88.22**

Fig. 7 demonstrates the complete offensive speech classification performance achieved by the proposed GOARN-OSD approach on 30% of TS data. The GOARN-OSD methodology recognized the samples as hate class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such as 93.83%, 89.33%, 99.32%, 94.06%, 93.92% and 88.22% correspondingly. Also, the GOARN-OSD technique classified the samples under normal class with $accu_y$, $prec_n$, $reca_l$, F_{-score}, AUC_{score} and MCC values such

as 93.83%, 99.26%, 88.52%, 93.59%, 93.92% and 88.22% correspondingly. The proposed GOARN-OSD technique also recognized the samples as average class with $accu_y$, $prec_n$, $reca_l$, $F_{\_score}$, AUC_{score} and MCC values such as 93.83%, 94.30%, 93.92%, 93.82%, 93.92% and 88.22% correspondingly.

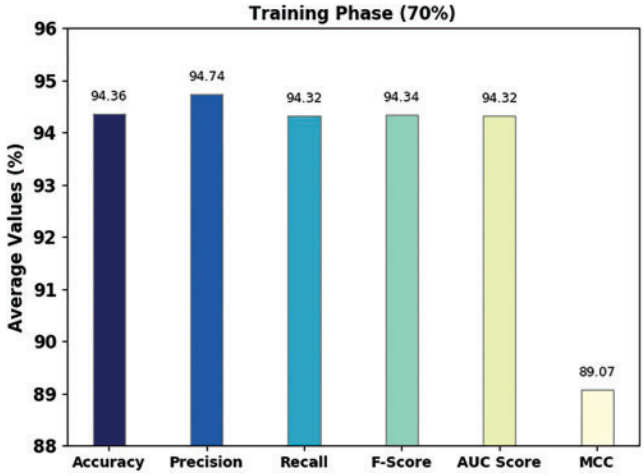

Figure 6: Average analysis results of the GOARN-OSD approach under 70% of TR data

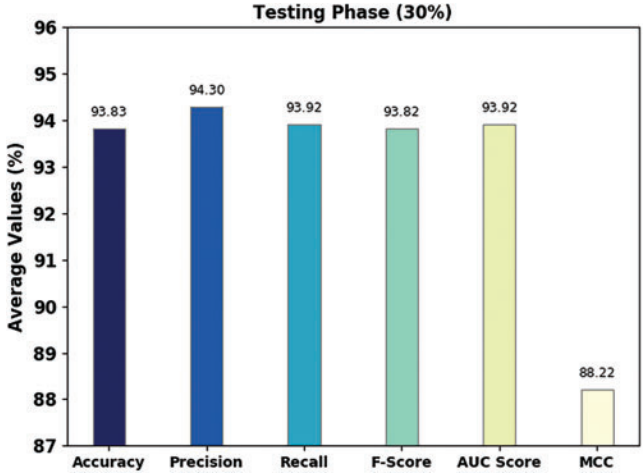

Figure 7: Average analysis results of the GOARN-OSD approach under 30% of TS data

Both Training Accuracy (TRA) and Validation Accuracy (VLA) values, gained by the proposed GOARN-OSD methodology on the test dataset, are shown in Fig. 8. The experimental results denote that the proposed GOARN-OSD approach attained the maximal TRA and VLA values whereas the VLA values were superior to TRA values.

Both Training Loss (TRL) and Validation Loss (VLL) values, obtained by the GOARN-OSD method on the test dataset, are displayed in Fig. 9. The experimental results represent that the GOARN-OSD technique exhibited the minimal TRL and VLL values while the VLL values were lesser than the TRL values.

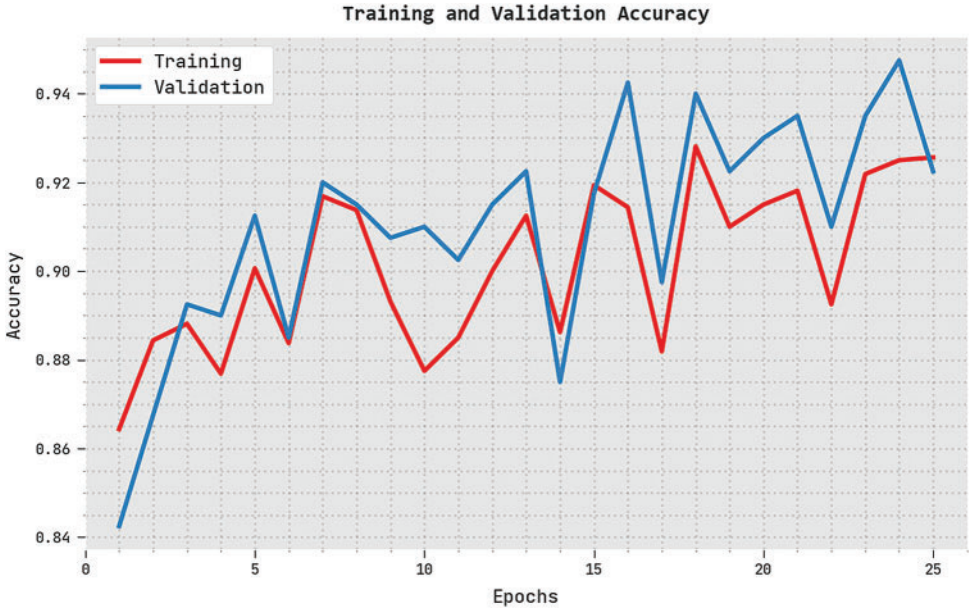

Figure 8: TRA and VLA analyses results of the GOARN-OSD approach

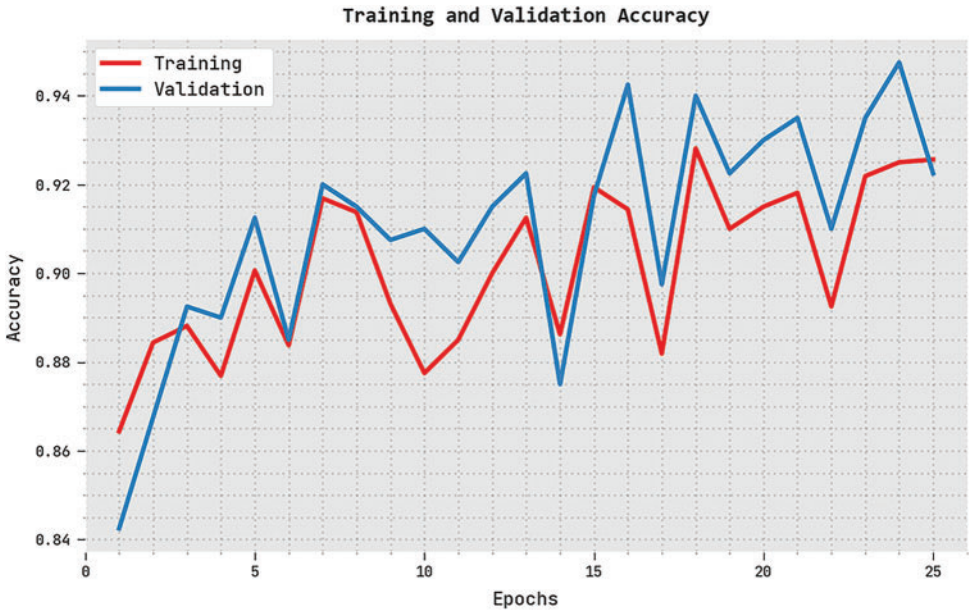

Figure 9: TRL and VLL analyses results of the GOARN-OSD approach

A clear precision-recall analysis was conducted upon the GOARN-OSD algorithm using the test dataset and the results are presented in Fig. 10. The figure denotes that the GOARN-OSD methodology produced enhanced precision-recall values under all the classes.

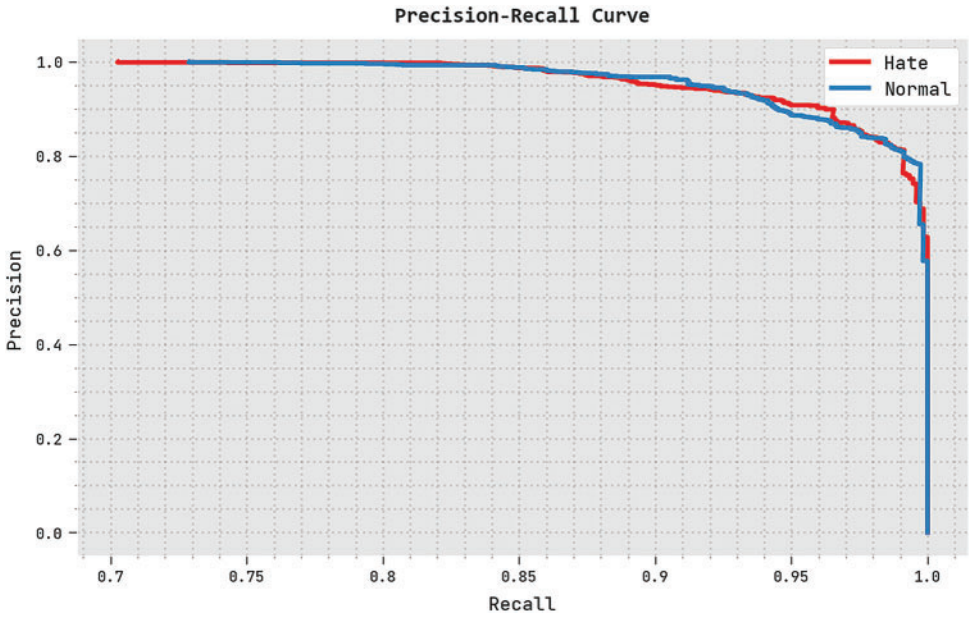

Figure 10: Precision-recall analysis results of the GOARN-OSD approach

A brief ROC study was conducted upon the GOARN-OSD technique using the test dataset and the results are portrayed in Fig. 11. The outcomes denote that the proposed GOARN-OSD method displayed its ability to categorize the test dataset under distinct classes.

The enhanced hate speech detection outcomes of the proposed GOARN-OSD model along with the results of other models were compared and the results are given in Table 4 and Fig. 12 [21]. These table values imply that the proposed GOARN-OSD model achieved a better performance over other models. For instance, in terms of $accu_y$, the GOARN-OSD model attained an improved $accu_y$ of 94.75% whereas the BiLSTM with deep CNN and hierarchical attention (BiCHAT), HCovBi-Caps, Deep Neural Network (DNN), BiLSTM and the GRU models produced the $accu_y$ values such as 89.77%, 79.97%, 65.24%, 69.36% and 63.46% respectively. In terms of $pren_c$, the proposed GOARN-OSD approach gained an improved $pren_c$ of 94.78% whereas the BiCHAT, HCovBi-Caps, DNN, BiLSTM and the GRU algorithms produced the $pren_c$ values such as 87.98%, 79.50%, 44.77%, 64.47% and 47.82% correspondingly. In terms of $reca_l$, the proposed GOARN-OSD approach reached an improved $reca_l$ of 94.67% whereas the BiCHAT, HCovBi-Caps, DNN, BiLSTM and the GRU approaches produced the $reca_l$ values such as 79.20%, 73.78%, 37.10%, 37.71% and 40.79% correspondingly.

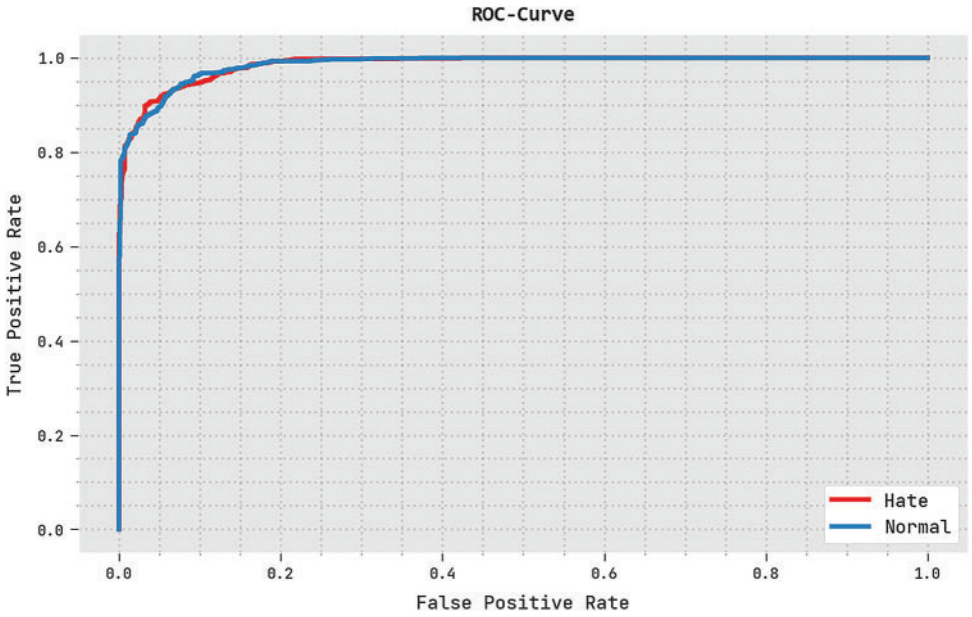

Figure 11: ROC analysis results of the GOARN-OSD approach

Table 4: Comparative analysis results of the GOARN-OSD approach and other recent algorithms

Methods	Accuracy	Precision	Recall	F-Score
GORAN-OSD	94.75	94.78	94.67	94.72
BiCHAT	89.77	87.98	79.20	83.31
HCovBi-Caps	79.97	79.50	73.78	76.17
DNN	65.24	44.77	37.10	39.54
BiLSTM	69.36	64.47	37.71	46.75
GRU	63.46	47.82	40.79	43.88

Finally, in terms of $F_{\_score}$, the GOARN-OSD technique attained an improved F_{score} of 94.72% whereas the BiCHAT, HCovBi-Caps, DNN, BiLSTM and the GRU algorithms achieved the $F_{\_score}$ values such as 83.31%, 76.17%, 39.54%, 46.75%, and 43.88%, respectively.

Therefore, it can be inferred that the proposed GOARN-OSD model produced enhanced offensive speech detection performance than the existing models.

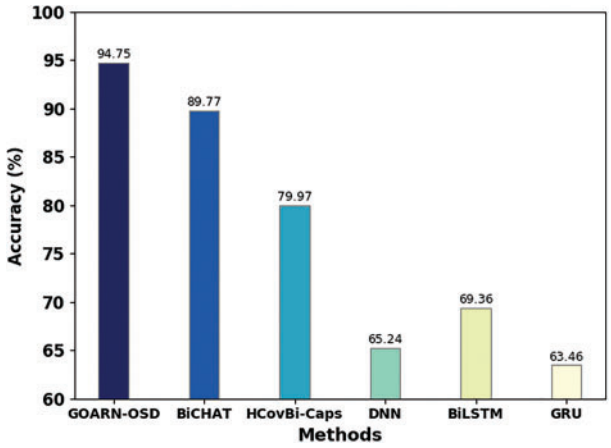

Figure 12: Comparative analysis results of the GOARN-OSD approach and other recent algorithms

4 Conclusion

In this article, a new GOARN-OSD technique has been developed for offensive speech detection on social media. The proposed GOARN-OSD technique integrates the concepts of DL technique and metaheuristic algorithm for the purpose of hate speech detection. In the proposed GOARN-OSD technique, the data is pre-processed and word embedding is carried out in the primary stage. For hate speech recognition and classification, the ARN model is utilized in this study. At last, the GOA approach is exploited as a hyperparameter optimizer to boost the hate speech recognition performance. To depict the promising performance of the proposed GOARN-OSD method, a widespread experimental analysis was executed. The comparative study outcomes established the superior performance of the proposed GOARN-OSD method compared to the existing approaches. In the future, the GOARN-OSD technique can be elaborated for sarcasm detection process too.

Acknowledgement: The authors thank to the support of Princess Nourah bint Abdulrahman University, Umm Al-Qura University, and Prince Sattam bin Abdulaziz University for their funding support to this work.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R281), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4331004DSR031). This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] G. Kovács, P. Alonso and R. Saini, "Challenges of hate speech detection in social media," *SN Computer Science*, vol. 2, no. 2, pp. 1–15, 2021.

[2] S. Modha, T. Mandl, P. Majumder and D. Patel, "Tracking hate in social media: Evaluation, challenges and approaches," *SN Computer Science*, vol. 1, no. 2, pp. 1–16, 2020.

[3] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal *et al.,* "Hatexplain: A benchmark dataset for explainable hate speech detection," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 35, no. 17, pp. 14867–14875, 2021.

[4] B. Vidgen and T. Yasseri, "Detecting weak and strong Islamophobic hate speech on social media," *Journal of Information Technology & Politics*, vol. 17, no. 1, pp. 66–78, 2020.

[5] N. Vashistha and A. Zubiaga, "Online multilingual hate speech detection: Experimenting with Hindi and English social media," *Information*, vol. 12, no. 1, pp. 5, 2020.

[6] M. Mondal, L. A. Silva, D. Correa and F. Benevenuto, "Characterizing usage of explicit hate expressions in social media," *New Review of Hypermedia and Multimedia*, vol. 24, no. 2, pp. 110–130, 2018.

[7] M. Corazza, S. Menini, E. Cabrio, S. Tonelli and S. Villata, "A multilingual evaluation for online hate speech detection," *ACM Transactions on Internet Technology*, vol. 20, no. 2, pp. 1–22, 2020.

[8] T. Davidson, D. Warmsley, M. Macy and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. of the Int. AAAI Conf. on Web and Social Media*, Canada, vol. 11, no. 1, pp. 512–515, 2017

[9] T. T. A. Putri, S. Sriadhi, R. D. Sari, R. Rahmadani and H. D. Hutahaean, "A comparison of classification algorithms for hate speech detection," in *Iop Conf. Series: Materials Science and Engineering*, India, vol. 830, no. 3, pp. 032006, 2020.

[10] G. K. Pitsilis, H. Ramampiaro and H. Langseth, "Effective hate-speech detection in Twitter data using recurrent neural networks," *Applied Intelligence*, vol. 48, no. 12, pp. 4730–4742, 2018.

[11] S. Mishra, S. Prasad and S. Mishra, "Exploring multi-task multilingual learning of transformer models for hate speech and offensive speech identification in social media," *SN Computer Science*, vol. 2, no. 2, pp. 1–19, 2021.

[12] M. U. Khan, A. Abbas, A. Rehman and R. Nawaz, "Hateclassify: A service framework for hate speech identification on social media," *IEEE Internet Computing*, vol. 25, no. 1, pp. 40–49, 2020.

[13] M. Mozafari, R. Farahbakhsh and N. Crespi, "Hate speech detection and racial bias mitigation in social media based on BERT model," *PLoS One*, vol. 15, no. 8, pp. e0237861, 2020.

[14] S. K. Mohapatra, S. Prasad, D. K. Bebarta, T. K. Das, K. Srinivasan *et al.,* "Automatic hate speech detection in English-odia code mixed social media data using machine learning techniques," *Applied Sciences*, vol. 11, no. 18, pp. 8575, 2021.

[15] K. Perifanos and D. Goutsos, "Multimodal hate speech detection in Greek social media," *Multimodal Technologies and Interaction*, vol. 5, no. 7, pp. 34, 2021.

[16] A. K. Das, A. Al Asif, A. Paul and M. N. Hossain, "Bangla hate speech detection on social media using attention-based recurrent neural network," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578–591, 2021.

[17] Z. Al-Makhadmeh and A. Tolba, "Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach," *Computing*, vol. 102, no. 2, pp. 501–522, 2020.

[18] S. Safari, S. Sadaoui and M. Mouhoub, "Deep learning ensembles for hate speech detection," in *32nd Int. Conf. on Tools with Artificial Intelligence ICTAI*, U.S., pp. 526–531, 2020.

[19] Z. He, C. Y. Chow and J. D. Zhang, "STANN: A spatiotemporal attentive neural network for traffic prediction," *IEEE Access*, vol. 7, pp. 4795–4806, 2018.

[20] B. S. Yildiz, N. Pholdee, S. Bureerat, A. R. Yildiz and S. M. Sait, "Robust design of a robot gripper mechanism using new hybrid grasshopper optimization algorithm," *Expert Systems*, vol. 38, no. 3, pp. e12666, 2021.

[21] S. Khan, M. Fazil, V. K. Sejwal, M. A. Alshara, R. M. Alotaibi *et al.,* "Bichat: BiLSTM with deep CNN and hierarchical attention for hate speech detection," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 4335–4344, 2022.