Denigration analysis of Twitter data using cyclic learning rate based long short-term memory

Suhas Bharadwaj Rajendra[1], Sampath Kuzhalvaimozhi[1], Vedavathi Nagendra Prasad[2]
[1]Department of Information Science and Engineering, The National Institute of Engineering, Mysuru, Affiliated to Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering, Vidyavardhaka College of Engineering, Mysuru, India

Article Info

Article history:

Received Feb 19, 2024
Revised Jul 16, 2024
Accepted Oct 1, 2024

Keywords:

Cyberbullying
Cyclic learning rate
Denigration
Long short-term memory
Twitter tweets

ABSTRACT

Technological innovation has given rise to a new form of bullying, often leading to significant harm to one's reputation within social circles. When a single person becomes target to animosity and harassment in a cyberbullying incident, it is termed as denigration. Many different cyberbullying detection techniques are carried out to counter this, concentrating on word-based data and user account features only. The main objective of this research is to enhance the learning rate of long short-term memory (LSTM) using cyclic learning rate (CLR). Therefore, in this research, cyberbullying in social media is detected by developing a framework based on LSTM-CLR which is more stable for enhancing classification accuracy without the need for multiple trials and modifications. The effectiveness of the suggested LSTM-CLR is assessed for identifying cyberbullying using Twitter data. The attained results show that the proposed LSTM-CLR obtains 82% accuracy, 80% precision, 83% recall and 81% F-measure in the classification of cyberbullying tweets, which is superior when compared with the existing multilayer perceptron (MLP) and bidirectional encoder representations from transformers (BERT) models.

Corresponding Author:

Suhas Bharadwaj Rajendra
Department of Information Science and Engineering, The National Institute of Engineering, Affiliated to Visvesvaraya Technological University
Belagavi, Karnataka
Email: suhasbr.research@gmail.com

1. INTRODUCTION

Denigration constitutes a form of cyberbullying where an individual purposefully undermines someone else's reputation or social connections by spreading unfavorable internet rumors or gossip. Character assassination of public figures such as politicians and celebrities are a common form of cyberbullying offense [1], [2]. Denigration is a prominent threat in contemporary society and has significant negative consequences for victims, being quite destructive because of the strong propagation and frequency [3]. Some recent efforts have addressed this issue with the use of several classical machine learning (ML) and deep learning (DL) methods to detect denigration. The power of social networking sites creates a fertile ground for spreading defamatory rumors, which is another form of denigration bullying. These targeted malicious remarks reflect amongst a huge number of recipients, and are a challenge to be rectified in best case scenario [4]. The models that are designed with the aim of unveiling and analyzing expressions of insults often found in such posts serve as instruments to help identify hate speech, insult and bullying [5]. This requires a mechanism of detection for mitigating of the underlying harmful effects [6]. An intrusion detection system (IDS) serves as a software application for monitoring data traffic flow across a network to identify potential instances of malicious behavior [7]. Recently, enhancements in cyberbullying classification have resulted from the automatic identification of denigration [8], [9]. Cyberbullying is linked to causing detrimental impacts on mental and physical health, academic performance, depression, and an increased risk of suicidal thoughts, as per various studies [10], [11]. As a result, the quick identification of denigration is critical to minimize its detrimental consequences on victims. Furthermore, the recurring nature of denigration makes it critical to notice and eliminate it at the earliest [12]. The goal of cyber aggression is to identify the aggressors, while also supporting the victims. The predominant approach in addressing cyberbullying has largely centered around scrutinizing and quantifying accuracy scores with the application of learning methodologies [13]. DL approaches such as convolutional neural network (CNN), long short-term memory (LSTM), and bidirectional long short-term memory (BiLSTM) have grown in favor of detecting cyberbullying [14]. Despite the numerous prevention and intervention approaches, cyberbullying action has not reduced in the last decade [15]. The current analyses have observed repeatedly detecting cyberbullying occurrences that are found to be effective in detecting cyberbullying. However, their accuracy is reduced once the data size is enlarged. Therefore, learning models may not be perfect in dealing with regular language uncertainties typical for cyberbullying [16]. To overcome the limitations of the existing methods, a DL model named LSTM which is integrated with cyclic learning rate (CLR) approach is proposed in this research for effective detection of denigration. Its effectiveness is measured by comparing it with the state-of-the-art methods namely, convolutional neural network (CNN), recurrent neural network (RNN), and gated recurrent unit (GRU).

Raj *et al.* [17] suggested a cyberbullying detection system using a deep learning framework by evaluating real-time tweets and posts on social media. Several neural networks were examined, and it was learned that the CNN-BiLSTM achieved preferable classification accuracy in detecting cyberbullying texts as it learnt the global features and long-term dependencies. However, this network required a large amount of data and time for training. Alduailaj and Belghith [18] developed a cyberbullying identification mechanism with a support vector machine using a real dataset from YouTube and Twitter. A natural language ToolKit was used for pre-processing the data, and the words were extracted based on different scenarios using term frequency-inverse document frequency (TF-IDF) and bag of words (BoW). The support vector machine (SVM) model was trained using real-time dataset and achieved better classification results. However, training the model with huge data accomplished better results. Murshed *et al.* [19] developed a hybrid DL model to identify cyberbullying using Twitter data. The presented approach was developed by integrating Elman-type RNN with an optimized Dolphin echolocation algorithm. This was done to fine-tune the Elman RNN's parameters and less training time. This approach resulted in the accurate detection of cyberbullying on social media. However, there was still an open research area for detecting cyberbullying from images, videos, and audio. Shelke and Attar [20] proposed a hybrid bidirectional LSTM with a multilayer perceptron (BiLSTM-MLP) model for cyberbullying detection using real-world and benchmark datasets on Twitter. BiLSTM was used for word embedding and was combined with multilayer perceptron (MLP) by using post-wise features, which improved the accuracy. This model mainly focused on text, content-based, and lexical category features for cyberbullying detection. However, the drawback was that it did not focus on multimedia-based features.

Raj *et al.* [21] proposed a mechanism to detect cyberbullying by using a hybrid bidirectional gated recurrent unit (Bi-GRU) and CNN-BiLSTM on real-world cyberbullying. Bi-GRU was used for text representation by using global vectors (GloVe), while CNN-BiLSTM was employed in the classification model. The model offered a robust mechanism with shallow neural networks to reduce the requirement of complex neural networks. But it also exhibited a limitation of being unable to achieve high accuracies when met with extensive datasets in supervised classification. Behl *et al.* [22] developed a MLP with an optimizer for the effective classification of Twitter tweets during the COVID-19 crisis. The three categories considered were resource needs, resource availability, and others. Better classification results were obtained by employing local interpretable model-agnostic explanations to examine the behavior of the proposed model. However, due to less training data, it displayed poor robustness with the training period of the classifier being higher. Deb and Chanda [23] compared the efficiency of the bidirectional encoder representations from transformers (BERT) embedding model to predict the disaster from Twitter data. The BERT embedding model was contrasted with the traditional contextual embedding models, where the findings showed that the BERT embedding model achieved superior results. However, transformer-based neural network models like BERT required a significant amount of memory storage for training, wherein the prediction accuracy diminished when the length of the tweets increased. Nasution and Setiawan [24] demonstrated a hybrid method called CNN and BiLSTM for enhancing cyberbullying detection on Indonesian Twitter. The major objective was to assess the presentation generated by FastText-enhanced feature expansion, and hybrid CNN and BiLSTM. Consequently, the outcomes confirmed the deletion of tweets comprising cyberbullying to be more precise and on target, developing a sense of security in consumers. Nevertheless, in some cases, there was a rise in the accuracy when other methods such as Allgram and Unigram+Bigram were deployed, as opposed to the proposed model.

From the overall analysis, it is clearly observed that the existing methods have major drawbacks such as the requirement of a large amount of data and time for training, lesser robustness of the system, high computational complexity, inconsideration of multimedia-based features, and low prediction accuracies. Also, the large memory storage demands of CNN or BiLSTM during the training stage broke the scalability and application of these models to large datasets. Furthermore, those techniques that only utilized text analysis were unlikely to detect those implicit clues that were part of multimedia content such as pictures and videos. As a result, the aforementioned methods were not very good at pinpointing cyberbullying offenders, thereby limiting the system's effectiveness. On the other hand, the number of inputs to data increased with large data sets which were relevant to some models, but were less effective. Although these approaches have made better results in detecting cyberbullying, an efficient and secure system is essential for formulating effective, scalable, and all-inclusive strategies that stop cyberbullying. To overcome the previous research's drawbacks, the proposed research is focused on developing a robust and efficient DL based LSTM-CLR denigration system to detecting cyberbullying. The major contributions of this research are specified as follows: i) Identifying cyberbullying on social media by developing a denigration detection system to address the pressing concerns on social media platform; ii) Reducing the risk of the proposed model being trapped in the local minima by combining CLR into LSTM training, which further facilitates the faster convergence by enhancing the optimization efficiency and cyberbullying detection capabilities; and iii) The learning rate of LSTM is increased with CLR, inturn increasing the model's capacity and providing a promising way to tackle online cyberbullying.

The rest of the paper is arranged as follows: the proposed methodology is explained in section 2. The process of LSTM with cyclic learning rate is explained in section 3, while the results and its comparisons are given in section 4. At last, the conclusion of this research paper is summarized in section 5.

2. PROPOSED METHOD

A DL-based LSTM classifier with CLR is proposed in this research to detect the denigration of persons through social media. The research proposes a deep learning-based LSTM classifier augmented with CLR to effectively detect instances of person denigration on social media platforms. By leveraging the LSTM's sequential modeling capabilities, the classifier analyzes textual data in context and captures nuances in language usage. Integrating CLR enhances the model's training dynamics, potentially improving its ability to generalize across varying levels of denigration expressions. The methodology involves collecting Twitter data, pre-processing the data, feature extraction, and classification using LSTM with CLR. The proposed framework's flow diagram is represented in Figure 1.

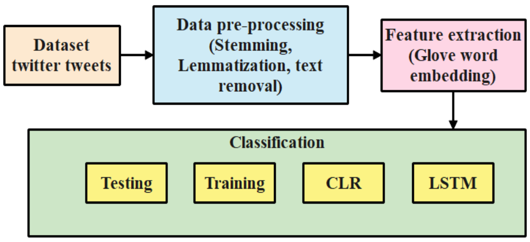

Figure 1. Flow diagram of the proposed denigration detection framework

2.1. Twitter data

The data is collected from the Twitter for detecting denigration and non-denigration events. The dataset consists of 2,000 comments, 1,000 comments with reputation rumors, and 1,000 comments with non-reputation rumors. The collected data is given as input to the pre-processing stage which is clearly explained in the following sub-sections.

2.2. Data pre-processing

It is an integral part of the natural language processing (NLP) to rebuild the original data into a meaningful format. Various methods such as stemming and lemmatization, and text removal are carried out as a part of pre-processing. Each of these techniques is discussed below.

2.2.1. Stemming and lemmatization

The tweets do not contain a standard format since a single word with the same meaning can be expressed differently. This problem is prevented by using lemmatization and stemming, where the process of lemmatization is to convert all words into their dictionary-based form known as a lemma. Stemming is the process of removing the word's last few letters to obtain a meaningful base. This contributes to enhancing the prediction accuracy of the system.

2.2.2. Stopword/text removal

Words such as "to," "me," "my," and "ours," and so on, are known as stop words which do not provide enough significance to the sentences and cause noise in the dataset. To remove these words, a Python package library called stopword is utilized [25]. The extraction of relevant features is performed after the removal of stopwords which is given as input to the extraction process where global vectors are applied.

2.3. Feature extraction

Once pre-processing is performed, a pre-trained word embedding model called Glove (2 billion tweets, 27 billion tokens, and 1.2 vocabularies) is deployed for generating a word vector matrix with 200 dimensions. Global vector (Glove) is a 2vec-based word representation that aids in the efficient learning of word embeddings from textual documents. The glove model is combined with TF-IDF to determine the proposed approach's effectiveness. TF-IDF presents a relative frequency of the word that is present in the textual document and the IDF is used in the process of scaling with the total count of documents. Each input word is represented as a token T and every individual word is transformed to a word vector of dimension d. So, the dimensionalities of each word vector are represented as R^d and the input text matrix created is denoted as $T = \{t_1, t_2, \dots, t_k\} \in R^{k\times d}$. Equation (1) represents the feature vector for document concatenation and word embeddings. The text representation is enhanced by combining the pre-trained Glove and TF-IDF weighing which is numerically represented in (2):

$$f_v = w_1 \oplus w_2 \oplus w_3 \dots \oplus w_{n-1} \oplus w_n \tag{1}$$

$$V_i = W_{d,t} \times f_v \tag{2}$$

where, the word vector matrix is stated as f_v, and the weighted value of the document and the term is stated as d and t, respectively. The suggested approach solves dimensionality issues related to high-dimensional matrix. Once the text representation from the word representation layer is received, the Gaussian noise and Gaussian dropout are generated. The Gaussian noise and Gaussian dropout processes are employed to regularize the model by rendering it less susceptible to overfitting. The LSTM classifier is trained with the CLR for effective classification, which is described in the following section [26].

3. CLASSIFICATION OF SENTIMENTS USING LONG SHORT-TERM MEMORY WITH CYCLIC LEARNING RATE

The extracted features are classified using LSTM network, where the input and output features are concatenated for the regularization of every layer. Unlike other classifiers, LSTM has the advantage of overcoming the overfitting problem, and the selected features are fed to the top layer to enhance the features. The LSTM has multiplicative cells formed by temporal and multiplicative units made up of various characters that handle data stream in the memory block. Figure 2 illustrates the architecture of LSTM model. Three gates namely, forget gate f_t, input gate i_t, and output gate o_t play a significant role in storing the memory components and regulates the information flow.

The output gate provides the final output, the forget gate selects which data to erase from the cell, and the input gate selects which data to add to the cell state. These gates help in transmitting the data and save the memory components. The processing of nodes in LSTM with three gates is given through (3) to (8):

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right) \tag{3}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

$$o_t = \sigma(W_0.[h_{t-1}, x_t] + b_0 \quad (7)$$

$$h_t = o_t * \tanh(C_t) \quad (8)$$

where, f_t denotes forget gate, σ denotes the sigmoid function, h_{t-1} denotes the hidden state of the prior layer, x_t is the input of the current layer, W states the weights, and b denotes the bias state. i_t is the input gate, C_t is the cell state in the next year, $\tilde{C}_t$ is the intermediate temporary state, C_{t-1} is cell state present in the preceding layer. o_t is the output layer and h_t is the hidden state of the next layer.

The purpose of the present method is to train LSTM using an LR that cycles through every batch. To calculate the LR of LSTM, the CLR is employed to vary the LR during training in a cyclic manner, typically by oscillating it between a lower and upper bound. This helps in achieving faster convergence and potentially finding better minima in the loss scenario. CLR allows the LR to vary during training, which helps the model converge faster. By using a higher LR during certain phases of training, the model makes larger updates to its parameters, potentially speeding up convergence. Higher LRs during cycles of training help explore the parameter space more broadly, while lower LRs allow for more fine-grained adjustments and exploitation of promising regions. The CLR consists of parameters such as batch size, step size, batch or iteration, cycle, and $base\_lr$ and max_lr. Considering k as epoch number from 1 to N epochs, each epoch's iteration is represented by the symbol t. The sequence $\{\theta_{k,t}\}_{k,1}$ generated as the output from the training is stated as given below:

$\{\theta_{1,1}, \theta_{1,2}, \ldots .\theta_{1,T}\}$ when $k = 1, 1^{st}$ epoch

$\{\theta_{2,1}, \theta_{2,2}, \ldots .\theta_{2,T}\}$ when $k = 2, 2^{nd}$ epoch

$\{\theta_{1,1}, \theta_{1,2}, \ldots .\theta_{1,T}\}$ when $k = l, l^{th}$ epoch

where, l denotes the most recent epoch N. The sequence that is formed after every epoch is $\theta_{k.t}$ wherein t has the value T. The current epoch's first iteration makes use of expressions given in (9) and (10).

$$\theta_{k,t} = \theta_{k-1,T} \quad (9)$$

$$\theta_{k,t+1} = \theta_{k,t} - \alpha_k \nabla L_{B_{k,t}}(\theta_{k,t}) \quad (10)$$

In (9), t denotes the next new epoch's first iteration with index k, and $k - 1$ states the prior epoch. Similarly, in (10), α_k and $\nabla L_{B_{k,t}}$ denote the gradients of the network layer with respect to epochs and iteration. $t + 1$ is an iteration of the next new epoch k with the LR. The LR for the subsequent iteration attained based on the prior iteration is represented in (11). The average stochastic gradient (sg) of an epoch is given by (12):

$$\alpha_{k+1} = \frac{\|s_k\|^2}{|s_k^T y_k|} \quad (11)$$

$$sg_{k,t+1} = (1-\gamma)sg_{k,t} + \gamma \nabla L_{B_t}(\theta_{k,t}) \quad (12)$$

where, $t = 1,2,3, \ldots, T$ and γ represent the pre-defined smoothing factors that regulate the degree of decay. The difference in the current gradients and prior epochs is represented in (13). To get a new LR for the current epoch, the average of all LRs from the prior epoch is used. The average finding formula is given in (14). The new LR for the iteration of the next epoch is given in (15).

$$y_k = sg_{k,T} - sg_{k-1,T} \quad (13)$$

$$\alpha_{avg} = \frac{1}{T}\sum_{i=1}^{T} \alpha_i \, \beta_i \quad (14)$$

$$\alpha_{new} = \begin{matrix} \alpha_{yk} \, S_k, & \alpha_{avg} > \alpha_{yk} \, S_k \\ \alpha_{avg}, & otherwise \end{matrix} \quad (15)$$

Furthermore, α_{avg} and $\alpha_{yk}\,S_k$ are compared to determine the α_{new} value for the following epoch in the LR scheduler. The same $\alpha_{yk}\,S_k$ is taken into account if α_{avg} is larger than $\alpha_{yk}\,S_k$, else α_{avg} is taken into account for the following epoch [27]. The values obtained account for anything minimal. As a result, training is performed more quickly with the following parameters.

- Batch size: Batch size determines the count of training samples for use in an iteration which is considered as 128 in this research.
- Step size: Step size determines the iteration count to complete half the cycle which is considered as 1 here.
- Batch or iteration: It determines the set of samples with a batch size of 128 with 100 iterations/batches to complete one epoch.
- $Base\_lr$: The base LR or the minimum LR considered in this research is 0.00001.
- Max_lr: The maximum LR considered is 0.05.

The performance of the proposed LSTM-CLR achieves better classification accuracy. The evaluation of the proposed model is performed on both training and testing data. The accuracy and loss of the proposed LSTM-CLR model on trained data gives rise to superior values, as observed in Figures 3 and 4 respectively. The pseudocode of proposed classifier is mentioned as Algorithm 1.

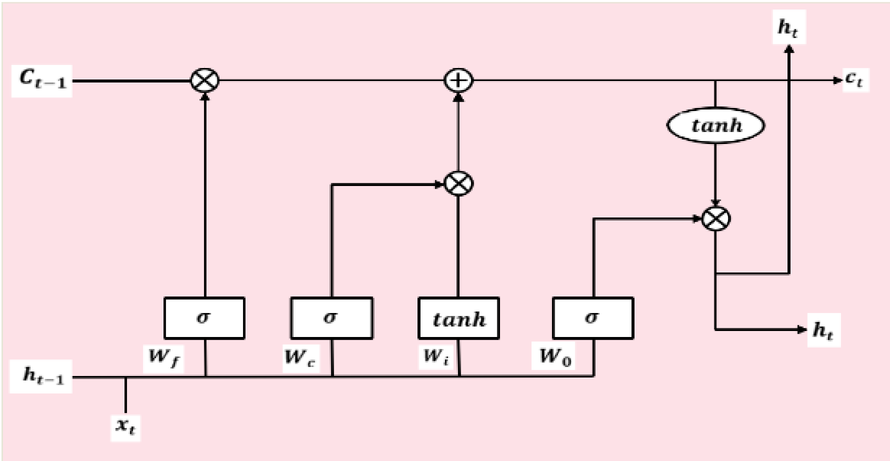

Figure 2. Architecture of LSTM

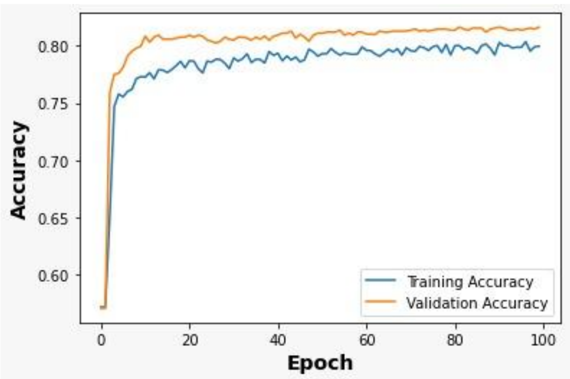

Figure 3. Accuracy evaluation for training and testing data

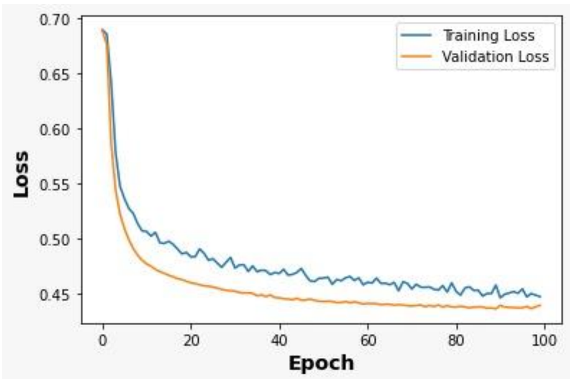

Figure 4. Loss evaluation of training and testing data

Algorithm 1. Pseudocode for proposed classifier

```
Input: Training dataset
Output
Normalize the dataset
Select the training feature size
for n epochs and batch size
```

```
Train the network (LSTM)
end for
Run Predictions using LSTM
Calculate the loss function
if cyclic_lr (epoch, current_lr)
base_lr=0.00001 # Initial LR
      max_lr=0.05 # Maximum LR
      step_size=1 # Number of epochs in half a cycle
cycle=floor (1+epoch/(2*step_size))
      x=abs (epoch/step_size-2*cycle+1)
     new_lr=base_lr+(max_lr-base_lr)*max(0, (1-x))
Return new_lr
End if
Train model with CLR
for epoch in range(epochs):
       current_lr=cyclic_lr (epoch, model.optimizer.lr. numpy ())
       model.optimizer.lr. assign(current_lr)
       fitting the model, the training and testing values
       Evaluating the parameter
End for
Return model
```

At first, the LSTM model's architecture is initialized by specifying the input shape, hidden layers, and output units. The CLR parameters are then specified as base LR ($base\_lr$), max LR (max_lr), and step size. The LR scheduler is then initialized with CLR: $clr = CyclicLR$ ($base\_lr, max\_lr, step\_size$). The LSTM is then compiled with the CLR scheduler. During each iteration, the CLR updates the LR, thereby helping the LSTM converge faster and explore a broader parameter space. After training is completed, the LSTM model is now fine-tuned using CLR. The model is then evaluated on the validation set for performance assessment. At last, the trained LSTM=CLR model is ready for making predictions on new sequences. CLR adjusts the LR during training in a cyclic pattern, enhancing the LSTM convergence. The base and max LRs are specified, allowing the model to explore a wider range of parameters. The LR scheduler is incorporated into the LSTM model's training process. Further, the CLR alternates the LR between the specified ranges during the training cycles. The oscillating LR helps escape local minima and accelerates convergence. The LSTM model is updated iteratively, adjusting weights based on the CLR schedule. The CLR-enhanced training process balances exploration and exploitation phases. After training, the LSTM-CLR model is expected to have enhanced generalization and improved performance. The evaluation on a separate validation set ensures the model's effectiveness. Finally, the trained LSTM-CLR model is ready for deployment and for making predictions on new sequential data.

4. RESULTS AND DISCUSSION

The proposed CLR-LSTM model is simulated on Anaconda Navigator 3.5.2.0 (64-bit), Python 3.7, OS: Windows 10 (64-bit), Processor: intel core i7, RAM: 16 GB. The performance is evaluated on the following metrics, as given in (16) to (19). The performance is measured in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) which are the positive and negative classes of true and false predictions.

$$Accuracy = \frac{(TP+TN)}{Total\ Instances} \tag{16}$$

$$Precision = \frac{TP}{(Predicted\ Instances=True)} \tag{17}$$

$$Recall = \frac{TP}{Actual\ number\ of\ Instances\ as\ True} \tag{18}$$

$$F - Measure = \frac{2*Precision*Recall}{Precision+Recall} \tag{19}$$

4.1. Quantitative analysis

The performance of LSTM with and without CLR is analyzed in this section respective to the traditional classifiers such as CNN, RNN, and GRU, as displayed in Tables 1 and 2. These tabular values are graphically represented in Figures 5 and 6. From Tables 1 and 2, it is clear that LSTM achieves better classification with CLR when compared to performing without integration of CLR. The LSTM model demonstrates the highest accuracy, precision, recall, and F-measure among the classifiers, showcasing its efficacy in handling sequential data with long-term dependencies. RNN and GRU perform preferably, but lag behind LSTM due to their limitations in capturing intricate temporal relationships. As seen in Table 2, the GRU classifier demonstrates better performance, while RNN and CNN fail in performance because of GRU and LSTM, indicating their comparative limitations in capturing patterns within the dataset. Overall, the table suggests that LSTM is effective for the denigration identification task and showcasing their robustness in handling sequential data with complex dependencies.

Table 1. Performance analysis of classifiers without CLR

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
CNN	71.35	72.73	75.65	74.55
RNN	73.57	74.23	77.48	75.65
GRU	76.23	76.34	78.44	77.67
LSTM	78.45	79.12	80.54	78.63

Table 2. Performance analysis of classifiers with CLR

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
CNN	74.27	73.27	77.44	76.29
RNN	77.73	75.27	79.42	77.94
GRU	79.56	77.41	80.56	79.85
LSTM	82.00	80.00	83.00	81.00

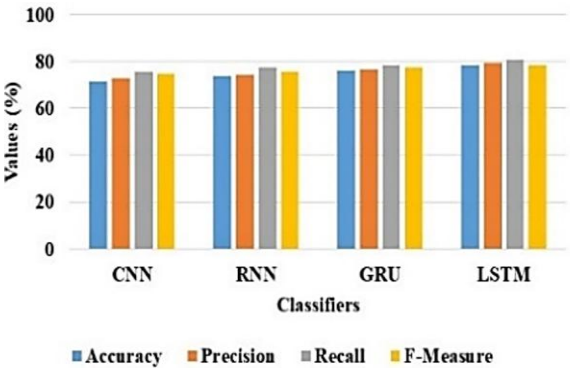

Figure 5. Performance analysis of classifiers without CLR

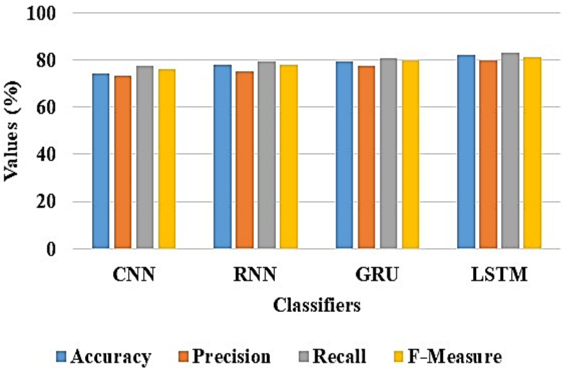

Figure 6. Performance analysis of classifiers with CLR

4.2. Comparative analysis

The LSTM-CLR performance is contrasted with the traditional methods such as MLP [22], BERT [23] and CNN-BiLSTM [24], as shown in Table 3. From Table 3, it is observed that the LSTM-CLR achieves better classification accuracy with 82% as the existing methods have limitations. MLP [22] has limitations of less training data, poor robustness of the system, and longer training period of the classifier. The BERT [23] has limitations of diminishing prediction accuracy when the length of the tweets is increased. These limitations are overcome in this research by introducing CLR to the LSTM for fast and accurate training.

Table 3. Comparative analysis of proposed LSTM-CLR with existing approaches

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
MLP (accuracy on mixed datasets) [22]	88	-	-	-
BERT [23]	79	-	-	74
CNN-BiLSTM [24]	80.55	-	-	-
LSTM-CLR	82	80	83	81

MLP's shortcomings include reliance on limited training data and extended training periods, with an emphasis on the need for more sophisticated models capable of handling the dynamic and diverse nature of social media content. While BERT initially keeps promise in understanding contextual terms in language, its performance fails with longer tweets, making it difficult for application in real-world scenarios where text lengths vary widely. The proposed method LSTM-CLR addresses these challenges by facilitating faster and more accurate training. CLR optimizes the learning rate during training, allowing the model to navigate complex data distributions more efficiently and adapt to varying tweet lengths. Consequently, LSTM-CLR not only overcomes the limitations of MLP and BERT, but also showcases the potential to enhance the efficacy of sentiment analysis and denigration detection in social media contexts.

4.3. Discussion

This research aims to address the issue of detecting person denigration on social media platforms by proposing LSTM-CLR to achieve better classification accuracy, as opposed to traditional methods such as MLP [23], BERT [23] and CNN-BiLSTM [25]. The main objective is to enhance the LSTM's learning rate (LR) with the help of CLR. Detecting and mitigating person denigration is crucial not only for protecting individuals' mental well-being, but also for fostering a safer and more inclusive online environment. Therefore, in this research, cyberbullying in social media is detected by developing a framework based on the LSTM. From the result analysis, the LSTM-CLR performance is contrasted with the traditional methods such as MLP [23], BERT [23] and CNN-BiLSTM [25]. From Table 3, it is observed that the LSTM-CLR attains commendable classification accuracy of 82%, precision as 80%, recall as 83%, and F-measure as 81%. While the existing methods MLP [23] attains 88% of accuracy; the existing BERT [23] attains 79% and 74% of accuracy and F-measure, respectively. On the other hand, the existing CNN-BiLSTM obtains an accuracy of 80.55%. These results prove importance of deploying adavanced technologies to tackle today's problems, alongside showing how deep-learning approaches make a significant contribution to the ongoing fight against cyberbullying detection. Still, a high LR sometimes causes varying loss function and problems with convergence as finding the global best is challenging. While a small LR slows down the network's learning speed and makes it hard to identify the global best. Therefore, the final convergence effect in the network model is significantly affected by LR, thereby making the setting of LR a major focus of the applied DL model. Also, the present study does not distinguish among the cyberbullying categories, consequently motivating to extend and study if the proposed LSTM-CLR can execute fine-grained cyberbullying classifications in the future.

5. CONCLUSION

The research findings offer an advancement in sentiment analysis, enhancing denigration detection on social media through LSTM-CLR. In the research field, this signifies a shift towards more sophisticated deep learning techniques and promises a safer online environment, fostering digital civility and countering cyberbullying with improved accuracy and efficiency. The further investigations are focused on the implementation of proposed method with multi-modal data. It is crucial to use the technology's potential for the good of the society as it develops further, therefore fostering a more secure and inclusive online community for all. Through the introduction of a novel framework that makes use of LSTM with CLR, this study adds to the ongoing efforts to tackle cyberbullying. The main goal is to increase LSTM's LR using CLR, which makes it easier to spot and deal with cases of cyberbullying on social media. Promising results are obtained when the proposed framework is evaluated against the existing approaches namely, MLP and BERT utilizing Twitter data as a benchmark. From the results, it is seen that the proposed LSTM-CLR accomplishes 82% accuracy, 80% of precision, 83% of recall and 81% of F-measure. This indicates how well the proposed framework works to identify tweets that involve cyberbullying. Moreover, exploring techniques to mitigate biases in training data and improve the model's resilience to adversarial attacks enhances its practical utility in real-world settings. Future research could also focus on developing ensemble methods or hybrid architectures that integrate multiple modalities for the practical implications of denigration detection.

REFERENCES

[1] A. P. Rodrigues *et al.*, "Real-time Twitter spam detection and sentiment analysis using machine learning and deep learning techniques," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–14, Apr. 2022, doi: 10.1155/2022/5211949.

[2] N. Pavitha *et al.*, "Movie recommendation and sentiment analysis using machine learning," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 279–284, Jun. 2022, doi: 10.1016/j.gltp.2022.03.012.

[3] P. Koukaras, C. Nousi, and C. Tjortjis, "Stock market prediction using microblogging sentiment analysis and machine learning," *Telecom*, vol. 3, no. 2, pp. 358–378, May 2022, doi: 10.3390/telecom3020019.

[4] Al-Khowarizmi, I. P. Sari, and H. Maulana, "Optimization of support vector machine with cubic kernel function to detect cyberbullying in social networks," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 22, no. 2, pp. 329–339, Apr. 2024, doi: 10.12928/TELKOMNIKA.v22i2.25437.

[5] R. H. Channegowda, P. Karthik, R. Srinivasaiah, and M. Shivaraj, "Customized mask region based convolutional neural networks for un-uniformed shape text detection and text recognition," *International Journal of Electrical and Computer Engineering*,

vol. 13, no. 1, pp. 413–424, Feb. 2023, doi: 10.11591/ijece.v13i1.pp413-424.

[6] N. Garg and K. Sharma, "Text pre-processing of multilingual for sentiment analysis based on social network data," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 776–784, Feb. 2022, doi: 10.11591/ijece.v12i1.pp776-784.

[7] N. Kewsuwun and S. Kajornkasirat, "A sentiment analysis model of agritech startup on Facebook comments using naive Bayes classifier," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 2829–2838, Jun. 2022, doi: 10.11591/ijece.v12i3.pp2829-2838.

[8] K. Naithani and Y. P. Raiwani, "Realization of natural language processing and machine learning approaches for text-based sentiment analysis," *Expert Systems*, vol. 40, no. 5, Aug. 2023, doi: 10.1111/exsy.13114.

[9] W. F. Satrya, R. Aprilliyani, and E. H. Yossy, "Sentiment analysis of Indonesian police chief using multi-level ensemble model," *Procedia Computer Science*, vol. 216, pp. 620–629, 2022, doi: 10.1016/j.procs.2022.12.177.

[10] I. Almalis, E. Kouloumpris, and I. Vlahavas, "Sector-level sentiment analysis with deep learning," *Knowledge-Based Systems*, vol. 258, Dec. 2022, doi: 10.1016/j.knosys.2022.109954.

[11] U. Naseem, I. Razzak, M. Khushi, P. W. Eklund, and J. Kim, "COVIDSenti: a large-scale benchmark Twitter data set for COVID-19 sentiment analysis," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 976–988, Aug. 2021, doi: 10.1109/TCSS.2021.3051189.

[12] N. Aslam, F. Rustam, E. Lee, P. B. Washington, and I. Ashraf, "Sentiment analysis and emotion detection on cryptocurrency-related tweets using ensemble LSTM-GRU model," *IEEE Access*, vol. 10, pp. 39313–39324, 2022, doi: 10.1109/ACCESS.2022.3165621.

[13] G. Chandrasekaran, N. Antoanela, G. Andrei, C. Monica, and J. Hemanth, "Visual sentiment analysis using deep learning models with social media data," *Applied Sciences*, vol. 12, no. 3, Jan. 2022, doi: 10.3390/app12031030.

[14] H. Saleh, S. Mostafa, A. Alharbi, S. El-Sappagh, and T. Alkhalifah, "Heterogeneous ensemble deep learning model for enhanced Arabic sentiment analysis," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103707.

[15] B. AlBadani, R. Shi, and J. Dong, "A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM," *Applied System Innovation*, vol. 5, no. 1, Jan. 2022, doi: 10.3390/asi5010013.

[16] H. Saleh, S. Mostafa, L. A. Gabralla, A. O. Aseeri, and S. El-Sappagh, "Enhanced Arabic sentiment analysis using a novel stacking ensemble of hybrid and deep learning models," *Applied Sciences*, vol. 12, no. 18, Sep. 2022, doi: 10.3390/app12188967.

[17] M. Raj, S. Singh, K. Solanki, and R. Selvanambi, "An application to detect cyberbullying using machine learning and deep learning techniques," *SN Computer Science*, vol. 3, no. 5, Jul. 2022, doi: 10.1007/s42979-022-01308-5.

[18] A. M. Alduailaj and A. Belghith, "Detecting Arabic cyberbullying tweets using machine learning," *Machine Learning and Knowledge Extraction*, vol. 5, no. 1, pp. 29–42, Jan. 2023, doi: 10.3390/make5010003.

[19] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, and H. D. E. Al-Ariki, "DEA-RNN: a hybrid deep learning approach for cyberbullying detection in Twitter social media platform," *IEEE Access*, vol. 10, pp. 25857–25871, 2022, doi: 10.1109/ACCESS.2022.3153675.

[20] S. Shelke and V. Attar, "Rumor detection in social network based on user, content and lexical features," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 17347–17368, Mar. 2022, doi: 10.1007/s11042-022-12761-y.

[21] C. Raj, A. Agarwal, G. Bharathy, B. Narayan, and M. Prasad, "Cyberbullying detection: hybrid models based on machine learning and natural language processing techniques," *Electronics*, vol. 10, no. 22, Nov. 2021, doi: 10.3390/electronics10222810.

[22] S. Behl, A. Rao, S. Aggarwal, S. Chadha, and H. S. Pannu, "Twitter for disaster relief through sentiment analysis for COVID-19 and natural hazard crises," *International Journal of Disaster Risk Reduction*, vol. 55, Mar. 2021, doi: 10.1016/j.ijdrr.2021.102101.

[23] S. Deb and A. K. Chanda, "Comparative analysis of contextual and context-free embeddings in disaster prediction from Twitter data," *Machine Learning with Applications*, vol. 7, Mar. 2022, doi: 10.1016/j.mlwa.2022.100253.

[24] M. A. S. Nasution and E. B. Setiawan, "Enhancing cyberbullying detection on Indonesian Twitter: leveraging fast text for feature expansion and hybrid approach applying CNN and BiLSTM," *Revue d'Intelligence Artificielle*, vol. 37, no. 4, pp. 929–936, 2023, doi: 10.18280/ria.370413.

[25] S. Bengesi, T. Oladunni, R. Olusegun, and H. Audu, "A machine learning-sentiment analysis on monkeypox outbreak: an extensive dataset to show the polarity of public opinion from Twitter tweets," *IEEE Access*, vol. 11, pp. 11811–11826, 2023, doi: 10.1109/ACCESS.2023.3242290.

[26] M. Ibrahim, S. Gauch, O. Salman, and M. Alqahtani, "An automated method to enrich consumer health vocabularies using GloVe word embeddings and an auxiliary lexical resource," *PeerJ Computer Science*, vol. 7, pp. 1–28, Aug. 2021, doi: 10.7717/peerj-cs.668.

[27] D. Vidyabharathi, V. Mohanraj, J. S. Kumar, and Y. Suresh, "Achieving generalization of deep learning models in a quick way by adapting T-HTR learning rate scheduler," *Personal and Ubiquitous Computing*, vol. 27, no. 3, pp. 1335–1353, Aug. 2023, doi: 10.1007/s00779-021-01587-4.

BIOGRAPHIES OF AUTHORS

Suhas Bharadwaj Rajendra is working as assistant professor in the Department of Information Science and Engineering at The National Institute of Engineering, Mysuru, since 2016. Obtained B.E. from Malnad College of Engineering Hassan in the year 2014 and M.Tech. from The National Institute of Engineering, Mysuru in the year 2016. Currently pursuing Ph.D. under Visvesaraya Technological University in the area of cyber security. His area of research interests are cyber security, machine learning, social networks, computer forensics. He can be contacted at email: suhasbr.research@gmail.com.

Sampath Kuzhalvaimozhi is working as a professor in Department of Information Science and Engineering. She obtained B.E. from Bharathidasan University and M.E. from PSG College of Technology and Ph.D. degree in field of cryptography from Visvesvaraya Technological University, Belagavi. Her area of research interests are cyber security, machine learning, social networks, and computer networks. She can be contacted at email: kuzhali_mozhi@nie.ac.in.

Vedavathi Nagendra Prasad is working as an associate professor in the Department of Computer Science and Engineering, Vidyavardhaka College of Engineering, Mysuru. She obtained Ph.D. in the year 2022 from Visvesvaraya Technological University, Belagavi and M.Tech. from Sri Jayachamarajendra College of Engineering, Mysuru, under Visvesvaraya Technological University, Belagavi. She has around 13.5 years of teaching experience. Her area of interest includes machine learning, artificial intelligence, web mining, semantic web, data mining, and social network analysis. She can be contacted at email: vedavathiresearch@gmail.com.