Received 9 August 2023, accepted 3 September 2023, date of publication 7 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3313113

RESEARCH ARTICLE

Cyberbullying Detection and Severity Determination Model

MOHAMMED HUSSEIN OBAID[1], SHAWKAT KAMAL GUIRGUIS[2], AND SALEH MESBAH ELKAFFAS[3]

[1]College of Science, Al-Nahrain University, Jadriya, Baghdad 64074, Iraq
[2]Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21544, Egypt
[3]College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 21532, Egypt

Corresponding author: Mohammed Hussein Obaid (M878224@gmail.com)

ABSTRACT Some teenagers actively participate in cyberbullying, which is a pattern of online harassment of others. Many teenagers are unaware of the risks posed by cyberbullying, which can include depression, self-harm, and suicide. Because of the serious harm it can cause to a person's mental health, cyberbullying is an important problem that needs to be addressed. This research aimed to develop a technique to identify the severity of bullying using a deep learning algorithm and fuzzy logic. In this task, Twitter data (47,733 comments) from Kaggle were processed and analyzed to flag cyberbullying comments. The comments embedded by Keras were fed into a long short-term memory network, composed of four layers, for classification. After that, fuzzy logic was applied to determine the severity of the comments. Experimental results suggest that the proposed framework provides a suitable solution to detect bulling with values of 93.67%, 93.64%, 93.62% achieved for the accuracy, F1-score, and recall, respectively.

INDEX TERMS Deep learning algorithm, severity of bullying, LSTM.

I. INTRODUCTION

With the prevalence of the Internet, social media has become a convenient and popular platform for people of all ages to communicate. However, social media has created several problems [1]. While these platforms enable people to communicate and interact in previously unthinkable ways, they have also led to malevolent activities such as cyberbullying. Cyberbullying is a type of psychological abuse with a significant impact on society [2]. It can be identified as a pattern of insulting messages that are posted repeatedly and that involve harsh or negative language [3].

Cyberbullying events have been increasing, primarily among young people, who often spend much of their time navigating between different social media platforms. Large social media networks such as Twitter are prone to cyberbullying because their widespread popularity can provide anonymity to abusers [2]. However, not all tweets using insulting words are abusive. There have been numerous studies on the automatic identification and prevention of cyberbullying, but there is still much work to be done to achieve a workable solution [4].

Cyberbullying detection is valuable because it assists in identifying and classifying cyberbullying activities, allows incidents to be dealt with after they have been identified, and helps internet users to take action to avoid becoming victims of cyberbullying [5]. The detection of cyberbullying occurring on social media platforms is difficult mainly because the interpretation of cyberbullying can vary from person to person, especially when classifying its severity: what might be a case of extreme severity for one person might not be for others.

To prevent cyberbullying incidents, a detection model should be able to take action immediately; however, this can be difficult to achieve in practice. Therefore, if a cyberbullying detection model can classify cyberbullying incidents among different severity levels, allowing incidents to be prioritized, the spread and influence of cyberbullying can be more effectively prevented. Cyberbullying detection tasks mainly focus on whether text contains cyberbullying content.

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

Determining the severity of cyberbullying content can assist in avoiding cyberbullying incidents and helping victims feel safe [1], [6].

The contribution of this study is the development of a classifier to detect cyberbullying and a comparison of its performance with other deep learning techniques, which will help to understand the limitations and advantages of the proposed method in text classification models. Additionally, to explore the patterns seen in cyberbullying victims, it may be useful to evaluate the degree of severity of a cyberbullying episode. Thus, creation of a technique to assess the seriousness of cyberbullying is important. The goal of this study is therefore to propose a model for detecting bullying by using a long short-term memory (LSTM) algorithm and classifying the level of severity as low, medium, or high using a fuzzy logic system.

II. RELATED WORKS

Many approaches have been proposed for the detection of cyberbullying. In [7], researchers reported using a Char-CNNS (character-level convolutional neural network with shortcuts) model to determine whether speech posted on social media sites contains cyberbullying. They used characters as the smallest learning unit, allowing the model to deal with intentional obfuscation and spelling errors in real-world datasets. In addition, a new Chinese Weibo comment dataset has been published specifically for cyberbullying detection, and experiments have been conducted using both the Chinese Weibo dataset and the English Tweet dataset. To solve the class imbalance problem, a focal loss function is used, and shortcuts are used to stitch different levels of features together to learn more granular bullying signals. The trial results show that their approach is competitive with cutting-edge approaches for detecting cyberbullying.

In [8], the authors investigated how well a method like fuzzy fingerprints works at spotting text-based cyberbullying in social media. When tested in a situation that is similar to real life, where cases of cyberbullying are less frequent than those without it, experiments reveal that the fuzzy fingerprints perform marginally better than baseline classifiers.

Yin et al. [9] used supervised learning to detect abuse across three social media networks – Kongregate (a chat-style community), Slashdot, and Myspace (both discussion-style communities) – as well as the content, sentiment, and contextual features of messages. As a classification tool, they used libSVM with a linear kernel. TFIDF weighting outperformed n-gram and profanity in the results.

Machine learning was applied to the detection of abusive Bangla text in Eshan and Hasan's project [10]. They compared the performance of various machine learning algorithms, namely multinomial naive Bayes (MNB), random forest (RF), and support vector machine (SVM). To create the dataset, they gathered data from the Facebook accounts of well-known Bangladeshi individuals. Special characters, such as "@" and "-", were eliminated in favour of Bengali Unicode characters. To validate their findings, they used the 10-fold cross-validation method. The machine detected 50% of the offensive terms. Additionally, three different types of string features – unigram, bigram, and trigram – were used in the trials. Using CountVectorizer and TfidfVectorizer, data were taken from each comment and vectorized. The results demonstrated that an SVM with a linear kernel consistently provided the highest accuracy level. Finally, they came to the conclusion that among all the techniques, trigram TF-IDF Vectorizer with an SVM linear kernel delivers the highest accuracy of 82%.

The authors of [11] developed a hybrid model using a Bi-GRU with self-attention followed by CapsNet for detecting cyberbullying in social media textual content. The proposed Bi-GAC model's performance was assessed using the metrics of F1-score and ROC-AUC curve. When compared to conventional models, the approach improved the F-scores for some MySpace and Formspring.me datasets by nearly 9% and 3%, respectively.

A benchmark corpus for cyberbullying detection in code-mixed language was created in [12], which investigated how code-mixed data can be handled effectively. They used the BERT language model, VecMap-based bilingual embedding, and a two-channel convolutional neural network (CNN) model. One channel receives the BERT language model, whereas the other receives the bilingual word embedding based on VecMap. Standard machine learning models, as well as deep neural network models such as CNN and LSTM, were used as baselines. Overall accuracy and F1-measure values of 81.12% and 81.03% were achieved, respectively.

Deep learning models like LSTM, bidirectional long short-term memory (BI-LSTM), recurrent neural network (RNN), bidirectional recurrent neural network (BI-RNN), gated recurrent unit (GRU), and bidirectional gated recurrent unit (BIGRU) models were used to detect cyberbullying in social media in [13]. These methods were applied to data on public Twitter comments, obtaining an accuracy of 90.4%, an improvement over state-of-the-art schemes.

The goal of [14] was to detect and prevent bullying on Twitter using two machine learning classifiers, SVM and naive Bayes, for training and testing. The results demonstrated that both naive Bayes and SVM were able to detect the true positives with 71.25% and 52.70% accuracy, respectively. However, SVM outperformed naive Bayes in similar work on the same dataset.

The authors of [15] used a CNN for classification. The CNN operated on many different layer types, each having different parameters to be set. Since manually adjusting the parameters would be difficult and slow, a metaheuristic optimization algorithm was incorporated to find the optimal or near-optimal values. OCDD advances the current state of cyberbullying detection by eliminating the difficult task of feature extraction/selection and replacing it with word vectors, which capture the semantic content of words, allowing the CNN to classify tweets in a more intelligent way than traditional classification algorithms. The CNN showed promising results when used for different text mining tasks;

however, it has not been implemented in the cyberbullying detection context.

An approach for the detection of cyberbullying was proposed in [16], in which a CNN was used and compared to traditional classification algorithms in the context of detecting cyberbullying in chats containing Hinglish (Hindi and English) code-mixed language.

Two important processes were studied in [26]: first, the process of forming a word representation, and second, the classification process for detecting bullying sentences. A separate pre-training process was performed to build a new representation of a term or word, using Word2vec. Two types of data were used in the pre-training process. The first type consisted of testing and training data, while the second type included the full dataset, totalling 26,800 unique Twitter sentences including the test and training data. The classification process uses three main algorithms that are popular for text classification: LSTM, bi-LSTM, and CNN. To create the dataset, 9,854 labelled sentences were extracted from 2,584 Twitter conversations. The dataset consists of 1,680 sentences labelled as bullying and 6,343 labelled as neutral. A total of 504 experiments were conducted in this research, using the preprocessing stage to determine the machine learning features, the dropout layer configuration, and the learning algorithm. An accuracy score of 90.57% was achieved, while the recall score for the bullying class reached 75.7%.

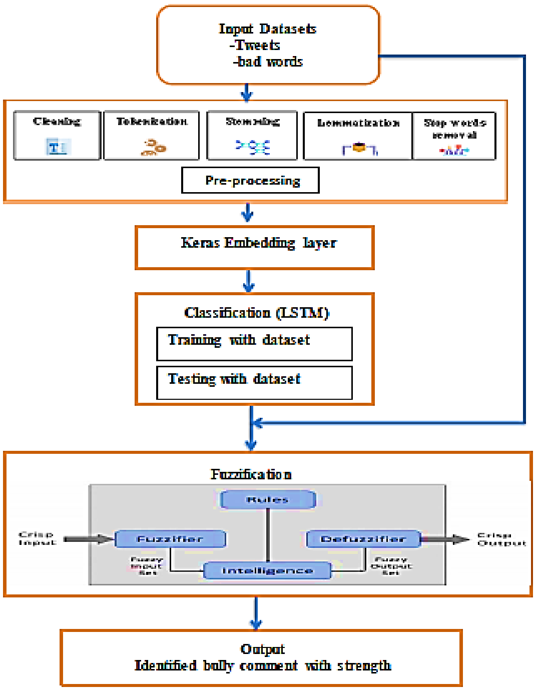


FIGURE 1. Architecture of the proposed model's.

III. METHODOLOGY

The architecture of the proposed model is described in this section. Figure 2 presents the four processes conducted in this study: (1) data collection, (2) data preprocessing, (3) LSTM model construction and cyberbullying severity classification, and (4) loss function and model optimization. The details of each process are described in the following subsections.

A. DATA COLLECTION

The proposed method was implemented in the Python programming language. Python packages like Numpy and Pandas, as well as the Tensorflow and OpenCV software, were used to preprocess the data [17]. Additionally, the Keras library was used to create the deep multichannel model [18]. The Kaggle dataset includes 47,733 tweets available at https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification, of which 39,748 were categorized as bullying and 7,985 as non-bullying. The class distribution of the dataset is displayed in Table 1. The dataset was divided into training and testing sets.

TABLE 1. Class distribution of the dataset.

Dataset	Label	Counts
(Tweet) 47.733	Bullying	39748
	Non bullying	7985

B. PREPROCESSING AND KERAS EMBEDDING LAYER

The preprocessing procedure included cleaning, stemming, and lemmatization. After this was completed, tokens could be extracted from the dataset. Tokenization is the process of extracting tokens, with sentences or paragraphs extracted from the data and output along with the entered text as separated words, characters, or sub words in the form of a list. These words then need to be converted to numerical vectors for the dataset to be represented as numerical data. Using Keras Layers, the features were vectorized, and each token's coefficient could be binary depending on the word count. The maximum length of sequences and the size of the vocabulary were both set at 4,500. Additionally, The Natural Language Toolkit (NLTK), one of the most popular and widely used NLP packages in the Python ecosystem, was used in this work. NLTK simplifies the removal of stop words, tokenization, tagging of parts of speech, and other tasks.

C. CYBERBULLYING DETECTION BY LONG SHORT-TERM MEMORY

Long Short –Term Memory (LSTM) is a temporal sequence simulation recurrent neural network (RNN) architecture. LSTM is more accurate than traditional RNNs due to its long-range dependencies. The problem of error backflow in the RNN architecture is caused by the use of backpropagation. Unlike in an RNN, LSTM's recurrent hidden layer is made up of distinct units referred to as memory blocks. Memory blocks are made up of memory cells with self-connections that store the network's temporal state, and special multiplicative units known as gates that control information flow. A forget gate scales the internal state of the cell before adding it as an input to the cell via the cell's self-recurrent connection, adaptively forgetting or resetting the cell's memory.

Each memory block's initial architecture included three gate types: an input gate controlled the flow of input activations into the memory cell, an output gate controlled the flow of memory cell activations into the rest of the network, and a forget gate controlled the flow of memory cell activations out of the network. As shown in Fig. 2, the cell acts as the memory, and the gates behave like neurons by computing an activation of a weighted sum and controlling the flow of values through the LSTM.

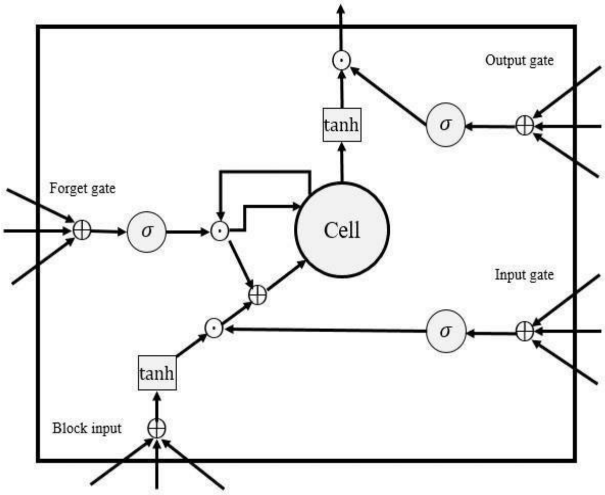


FIGURE 2. An LSTM block.

A basic LSTM network consists of an input (it), output (ot), and a forget gate (ft), The equations are shown as follows:

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

$$f_t = \sigma(W_i.\left[h_{t-1},x_t\right] + b_f) \tag{2}$$

$$o_t = \sigma(W_i.\left[h_{t-1},x_t\right] + b_o) \tag{3}$$

where h is used to characterize the state of the input, with h_{t-1} standing for the current state and h_{t-1} for the prior state, and x_t stands for an input text. The weights and biases for each gate are W and b, respectively. Here, s stands for the activation function, which in the case of the suggested model is the rectified linear unit (ReLU). ReLu Operation In contrast to Sigmoid and Tanh, which causes sparse neuron activation, which indicates that the neuron doesn't fire each time and that its value can be zero at some point. This property makes this activation function one of the most efficient ones for classification. The ReLU function is a different non-linear activation function that has gained popularity in the deep learning space. Because it does not fire every neuron at once, the ReLU function has an advantage over other activation mechanisms. This means that until the outcome of the linear transformation is less than 0, the neurons won't quit firing. The models are trained using a categorical cross-entropy loss function and backpropagation with the Adam optimizer. The Adam optimizer is more suitable for issues with a large amount of data or parameters because it is more computationally efficient, uses less memory, is invariant to gradient diagonal resizing, and is more memory-efficient. Long short-term memory (LSTM) models are designed to classify encoded documents as cyberbullying or not.

In this study, four hidden layers with 128 units, 64 units, 32 units, and 3 units were used to classify the comments. The models were trained using backpropagation with the Adam optimizer and a categorical cross-entropy loss function. The Adam optimizer is more computationally efficient, uses less memory, is unaffected by gradient diagonal resizing, and is well suited to problems with a large number of data or parameters.

D. PSEUDO-CODE

Pseudo-code summaries of the algorithms used in the experiments are presented below:

```
Input: List T=t1,t2,t3,...tn (n = number of comment)
Number f (f=number of filters)
Output: Number C=1 or 0 (0=no bullying,1=bullying)

Begin
Tokenize all text
Total vocab = length (Token)
Determine the maximum text length, len=max (length for t in T)
Split tweets for testing Train and Test
Create Embedding layer, Embedding(vocab, len)
lstm_model = Sequential()
lstm_model.add(LSTM(128))
lstm_model.add(Dense(128, activation='relu'))
lstm_model.add(Dense(64, activation='relu'))
lstm_model.add(Dense(32, activation='relu'))
lstm_model.add(Dense(2, activation='softmax'))
lstm_model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy'])
lstm_model.summary()
End
```

E. BULLY STRENGTH (SEVERITY) DETERMINATION

Fuzzy logic was used to determine the severity of the comments, with fuzzy rule sets eventually being implemented using the output of the LSTM prediction model. Fuzzy rules work by means of a succession of if-then clauses. The output of the LSTM was subjected to the fuzzy rule sets shown in Table 2. The input dataset was divided into bullying categories by this classifier, and fuzzy rule sets were only applied to comments that include bullying. By defining the severity of the bullying, it is possible to predict whether there is any potential risk of further bullying, as well as whether there is any risk in real life that could occur in the near future. The bullying severity was classified as high, medium, or low.

The fuzzy result "High" can be interpreted as the most severe form of bullying, where there is a chance that it will continue and a chance that it will eventually manifest in real life. The bullying is neither extreme nor moderate in the case of the fuzzy output "Medium". Though it can be replicated virtually, there is a small potential future risk. To reduce bullying and the stress on the targeted individuals, action must be taken. The fuzzy output "Low" denotes that the

TABLE 2. The three level membership function for number of bad word.

No	Rule
Rules 1	IF n >= 1 AND n < 3, then Bad Words Count = Low.
Rules 2	IF n >= 2 AND n < 5, then Bad Words Count = Medium.
Rules 3	IF n >= 4 AND n < 7, THEN Bad Words Count = High

bully is only marginally aggressive and can be stopped. The number of offensive terms found in the comment is used to create fuzzy rule sets (counted by matching with a bad word list for different bully types about 1618 bad word from Kaggle at https://www.kaggle.com/datasets/nicapotato/bad-bad-words). Each comment made by a bully is counted for the number of insulting words, The LSTM algorithm used on the supplied dataset produced these comments. To determine the number of offensive terms, the dataset is compared with the comments discovered for various types of bullies. The three-level membership function for the number of problematic words has partial levels of 1–3 as low, 2–4 as medium, and 4–6 as high; this function is shown in Table 2.

F. PERFORMANCE EVALUATION

This section describes the evaluation metrics used to validate and assess the performance of the LSTM for classifying social media post content as cyberbullying or non-cyberbullying. Specificity, recall, precision, F1-score, and accuracy are the assessment metrics. Accuracy can be used to determine how many texts were correctly predicted out of all the texts in the dataset. It is calculated using the following equation:

$$\text{Accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{\text{Total Instances}} \tag{4}$$

Out of all the texts that are predicted, either correctly or erroneously, the precision metric allows us to count how many texts in a certain category were accurately predicted. The precision (P) is the percentage of correctly predicted positive cases:

$$\text{Precision} = \frac{\text{True Positive}}{\text{Predicted Instances} = \text{True}} \tag{5}$$

The proportion of positive instances that were correctly detected is known as the recall, also known as the true positive (TP) rate, and can be expressed as follows:

$$\text{Recall} = \frac{\text{True Positive}}{\text{Actual number of instances as True}} \tag{6}$$

The F1-score is also known as the F-score or F-measure. It is used as a benchmark to calculate the weighted average of precision and recall. The F1-score ranges from 0 to 1, with 1 indicating that the model has few false positives and few false negatives and is considered ideal.

$$F1\text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

IV. RESULTS AND DISCUSSION

A Python program was used to implement and test the suggested model. Python functions are used to break a program down into modules. This makes scaling, managing, and debugging the code simpler. Additionally, the laptop (DELL 3000 series) with the following specs was used to implement the system: Processor: 2.20 GHz Intel (R), Core (TM) i5-5200u. RAM: 12.0 GB. The operating system of the 64-bit variety. An operating system is for Microsoft Windows 10 is only available in one language. The proposed cyberbullying detection that depends on a deep learning algorithm (LSTM) has been tested. Based on its accuracy, the performance of the LSTM model is evaluated.

However, because word representations are a key component of many NLP systems, it is common to represent words as vocabulary indexes. To represent each of the words, word-embedding vectors were used for each token of the sentence. The Keras embedding model was used in this study, in which the embedding input length was set to 1000 and 100 vocabularies items were taken into consideration. The structure of the model illustrated in table.3.

TABLE 3. LSTM model structure.

Setting	LSTM
Epoch	40
Embedding	Keras
Layers	4 (128 filters, 64 filters, 32 filters, 3 filters)
Dense layer	3 with RelU activation function
Optimizer	Adam
loss function	categorical cross-entropy

A series of experiments were performed to evaluate the model's performance. The first experiments investigated the effect of different divisions of the data into training and testing sets. The dataset was split according to the following ratios to assess the resulting performance of the model: 80% training and 20% testing, 70% training and 30% testing, 60% training and 40% testing, and 50% training and 50% testing. The highest accuracy of approximately 93.6% was obtained for the 80:20 ratio with an AUC value 0.964. It was found that the performance of the model declines as the number of training samples increases.

The performance of the proposed model was also evaluated using other metrics like the precision, recall, and F-score. These metrics were employed because they could assess the model's performance even when the distribution classes were not evenly distributed. fig 3 shows the results.

Studies on deep learning algorithms frequently report "loss" figures. Loss is essentially a penalty for making an incorrect prediction. To be more specific, if the model's forecast is correct, the loss value will be zero. Therefore, the objective is to obtain a set of weights and biases that minimize the loss value. The loss and the accuracy for the datasets when the dataset was split in an 80:20 ratio are shown in Fig. 4.

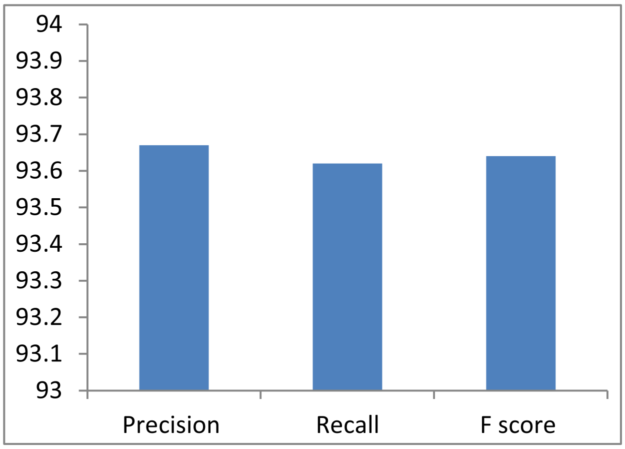


FIGURE 3. Performance evaluation of the proposed model on 80% training data.

TABLE 4. Performance of proposed model on different data set split ratio.

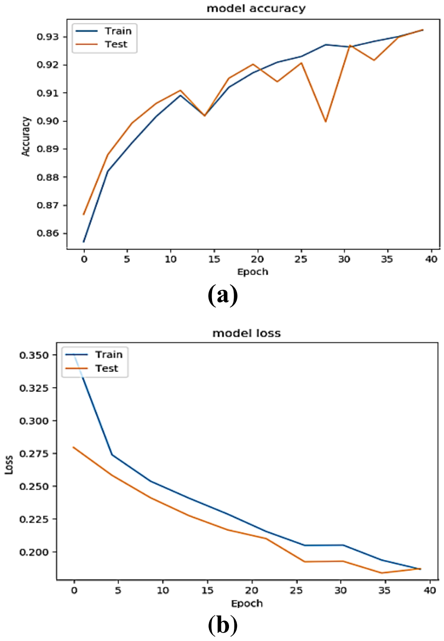
Data set	Accuracy			
	80% training 20% testing	70 %training 30% testing	60 %training 40% testing	50% training 50% testing
(Tweet)	93.67	90.4	88.75	85.34
AUC	0.9641			

Similar results were obtained in [24] by analyzing and uncovering cyberbullying textual patterns in Roman Urdu using RNN-LSTM, RNN-BiLSTM, and CNN models. It was stated that the best performance was achieved when the dataset was split into 80% training and 20% testing tests, with accuracies of 85.5% and 85%, respectively, and the corresponding F1-scores were 0.7 and 0.67 for the bullying class.

In [13], deep learning models such as LSTM, BI-LSTM, RNN, BI-RNN, GRU, and BI-GRU were applied to data on public comments on Twitter. The results indicated that the proposed mechanism was efficient for an 80:20 splitting ratio.

In [19], a hybrid deep learning model was created by combining a CNN for image-based prediction with an LSTM for text prediction. Compared to machine learning models, this model achieved more accurate predictions. The text was split into training and testing sets in an 80:20 ratio. After training a model with multiple text inputs, it reached 85° hardenability. It was concluded that development of an improved hybrid multi-input model predictive system can detect and prevent cyberbullying events more effectively on social media platforms.

The second experiment studied the influence of the epoch number on model performance. To achieve consistency in the evaluation parameters, the training process is repeated several times over a number of epochs. Epochs indicate how long the model is exposed to the training set for. The number of epochs is a hyper parameter that specifies how many times the learning calculation will loop through the entire training dataset. After one epoch, all examples in the training dataset have had a chance to update the internal model parameters. The obtained results illustrated in Table 4 show that the model's capacity for generalization grows as the number of epochs rises. However, a possible downside of having too many epochs is over-fitting. The model was run through various numbers of epochs, ranging from 5 to 40. Over 35 epochs, the model's performance stabilized, and after that point, any progress was almost nonexistent.



(a)

(b)

FIGURE 4. LSTM model accuracy (a) and loss (b) For the proposed model.

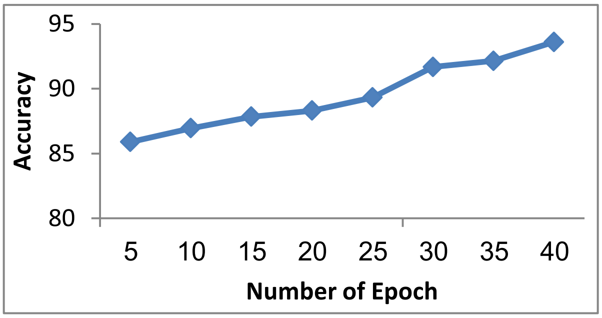


FIGURE 5. Model implementation over a number of epoch.

The model loss during training from 5 to 40 epochs is shown in Fig. 4. Indicating ideal model performance, the cross-entropy loss considered during setup converged satisfactorily over multiple epochs. The LSTM model exhibits a decreasing trend with respect to the precision value, so the more epochs, the more the accuracy tends to stabilize at a lower value with respect to its starting point. Similar results regarding epoch numbers were reported in [20].

For the models suggested in [21], the highest accuracy achieved was that of the LSTM model, which was approximately 93.84% with an F-score of 0.94 after 100 epochs, proving it to be an effective model to flag cyberbullying comments, with precision and recall values greater than the simple RNN and GRU network models.

By using an epoch number of 30, in [22], cyberbullying was detected in English-Hindi (En-Hi) code-switched text in an attempt to develop a new code-switched Twitter dataset by machine learning (SVM and logistic regression) and deep learning (multilayer perceptron, CNN, BiLSTM, BERT) algorithms. The deep ensemble model performed well on code-switched data, yielding a state-of-the-art F1-score of 0.93.

In the third experiment, the proposed system's performance was compared to that of other researchers, using some of their published papers' approaches [21], [23], [24], [25], [26], [28].

Despite the precision found in [23], an LSTM model with a relu activation and an Adam optimizer had an accuracy of 85% in detecting cyberbullying. Meanwhile, an RNN+LSTM model, with an accuracy of 91.82%, appeared to be the most effective choice in [21].

The results in [24], which addressed the issue of toxicity/cyberbullying detection in Roman Urdu text using deep learning techniques and advanced preprocessing methods, including the use of lexicons and resources typically developed for this work, showed that RNN-LSTM and RNN-BiLSTM performed best, with validation accuracies of 85.5% and 85% for the bullying class, respectively. It can be seen from Table 4 that the proposed method performs better than the previous methods in terms of accuracy.

Max-pooling was used in conjunction with a bidirectional LSTM network and attention layers in [25]. To determine the model's accuracy in identifying and classifying cyberbullying posts, it was tested on Wikipedia datasets. The approach outperformed competitors in terms of precision, recall, and F1-score, with values of 0.89, 0.86, and 0.88, respectively. The classification process in [26] is built around three popular text classification algorithms: LSTM, bi-LSTM, and CNN. The dataset contains 9,854 labeled sentences extracted from 2,584 Twitter conversations. The dataset includes 1,680 sentences labeled as bullying and 6,343 labeled as neutral. The steps included in the methodology involved 504 experiments that used the preprocessing stage to determine machine learning features, the dropout layer configuration, and the deep learning algorithm. The accuracy score obtained for the bullying class was 90.57%, while the recall score was 75.7%. In [28] an approach is proposed to cyberbullying detection in social media platforms by using the novel pre-trained BERT model with a single linear neural network layer on top as a classifier. The model is trained and evaluated on two social media datasets of which one dataset is small size and the second dataset is relatively larger size that may interpreted the reason of exceeded the accuracy obtained by our model.

A LSTM model is proposed for cyberbullying detection instead of a hybrid model such as that of the previous studies, and the present model differs from others in that it is a sequential type model with multiple (four) layers. On the other hand, in order to optimize the model even further, the particular embedding option (Keras) was used since it is task-specific; also, including ReLU as the activation layer for the hidden layer of the LSTM model to ensure that the activation function chosen has a significant impact on the performance will be done to enhance the model's.

TABLE 5. Comparison of accuracy results between proposed model and other studies.

Reference	Method	Accuracy
Raj *et al.*, 2022 [23]	LSTM	85%
Shylaja *et al.*, 2018 [21]	RNN+LSTM	91.82%
Dewani *et al.*, 2021 [24]	(RNN-LSTM), (RNN-BiLSTM)	85.5%,85%
Agarwal *et al.*, 2020 [25]	BiLSTM	89%
Anindyati *etal.*,2019 [26]	Deep learning	90.57%
Yadav *et al.*, 2020 [28]	pre-trained BERT model with a single linear neural network layer	98% (Formspring) and 96% (Wikipedia)
Proposed model	LSTM	**93.6%**

Because the Adam optimizer is more suitable for issues with a large amount of data or parameters and is more computationally efficient, uses less memory, and is invariant to gradient diagonal resizing, it was included in the study. We carry out extensive experiments with numerous time steps using a real-world dataset and following a time-aware evaluation that proves the performance improvement over baselines.

The final series of trials was performed to assess the system's complexity. A component of computational complexity theory called time complexity analysis is used to describe how computer resources are consumed by a program. In our model, the execution time for one epoch is given in Table 5, and the overall execution time was 926 s.

The obtained results were close to those of other studies such as [2]. In comparison to previous deep learning Bi-LSTM and RNN baseline models, it was noticed that the proposed DEA-RNN model requires a shorter training time. DEA-RNN took 248.52 s to train, compared to 349.1 s and 274.31 s for the baseline models based on Bi-LSTM and RNN, respectively.

An additional study [27] used an RNN-biLSTM model. They produced a sequential model with a maximum of 2,000 features in the embedding layer. The sigmoid activation

TABLE 6. Comparison of execution time between the proposed model and other studies.

Reference	Method	Execution time
[2]	DEA-RNN	248.52 s
	Bi-LSTM,	349.1s
	RNN	274.31s
[27]	RNN-biLSTM	13-15 min
[21]	RNN-LSTM	78 min
Proposed model	LSTM	926s

TABLE 7. Output of the fuzzy rule set.

Sentence	Out put	Strength of bully
Love that the best response to the hotcakes they managed to film was a non-committal "meh" from some adolescent.	**Non bullying**	**Neither**
When you accidently get high and call a girl gorgeous on bumble and then realize that sheâ€™s actually this horrible person who bullied me in high school help me	**Bullying**	**High bullying**
I want to slap that smirk off Kat's face. I know I know. Stand in line....	**Bullying**	**Low bullying**
@The_Loki_Jotunn @melisssugh Awesome. Will go into my "daily use" folder.	**Non bullying**	**Neither**
rape, self harm, suicide, and body shaming jokes aren't funny. Using gay as an insult isn't funny. – amina	**Bullying**	**High bullying**
RT @Ezmac_Thedream: Im not sexist, but women just can't drive	**Bullying**	**Low bullying**

function was used to create the hidden layer (H1). Adam optimization and a binary cross-entropy loss function were also used. Their model was run through various numbers of epochs, with an average execution time of 13 to 15 ms for each epoch. In [21], an RNN+LSTM model was used for cyberbullying detection, and it was found that the execution time for the model was 78 min.

A. SEVERITY DETERMINATION BY FUZZY LOGIC

In this study, a systemic framework was provided for identifying cyberbullying severity in dataset. To achieve this, first a deep learning classifier was built for classifying a comment as cyberbullying or not, then, based on fuzzy criteria, the severity was divided into multiple categories (low, medium, severe, or none). The primary goal and contribution of the current study was to offer a multi-class classification-based, systematic method for applying the severity of cyberbullying behavioral text.

Bullying comments are recognized and categorized in the LSTM model's output. The fuzzy system uses the comments associated with various bullies to determine the bullying's severity. The output for the fuzzy rule sets for the input test data is shown in Table 7. The evaluation of the fuzzy rule was done by human being, in which a randomly selected posts were chosen from dataset, it already known their classification then tested by the model and compared the obtained results with that in dataset.

V. CONCLUSION AND FUTURE WORK

A model structure is proposed for cyberbullying detection with multiple layers, which has a significant effect on its performance. Fuzzy rule sets are designed to specify the strength of different types of bullying. The limitations of the proposed model include it is not considered image and video cyberbullying detection, which means a post having only image or video are not a part of this research, combining the image with text has been found in cyberbullying posts. However, this study is limited to text oriented cyberbullying detection. Hence, the future scope of this research is always open to discussion as it has varied sub problems. The accuracy achieved by the proposed system was 93%, which can be improved by other combinations of the models can opt, and an ensemble system will form to achieve better prediction accuracy. This research achieved on tweet dataset so its recommended to be include other plate form like Instagram, Facebook and other so in future, the other components of social media posts, such as the user's information, network information, and any audio and video content of the post could also be explored for improving cyberbullying detection. Also including data from different social media platform to show how does the model work.

VI. CONFLICT OF INTEREST

Declarations of interest: none.

REFERENCES

[1] J. L. Wu and C. Y. Tang, "Classifying the severity of cyberbullying incidents by using a hierarchical squashing-attention network," *Appl. Sci.*, vol. 12, no. 7, p. 3502, 2022, doi: 10.3390/app12073502.

[2] 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.

[3] N. Haydar and B. N. Dhannoon, "A comparative study of cyberbullying detection in social media for the last five years," in *Al-Nahrain J. Sci.*, vol. 26, no. 2, pp. 47–55, 2023, doi: 10.22401/ANJS.26.2.08.

[4] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," in *Proc. 17th Int. Conf. Distrib. Comput. Netw.*, Jan. 2016, pp. 1–6, doi: 10.1145/2833312.2849567.

[5] N. Ayofe AZEEZ, S. Misra, O. Ifeoluwa LAWAL, and J. Oluranti, "Identification and detection of cyberbullying on Facebook using machine learning algorithms," *J. Cases Inf. Technol.*, vol. 23, no. 4, pp. 1–21, Jan. 2022, doi: 10.4018/JCIT.296254.

[6] A. Aggarwal, K. Maurya, and A. Chaudhary, "Comparative study for predicting the severity of cyberbullying across multiple social media platforms," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2020, pp. 871–877, doi: 10.1109/ICICCS48265.2020.9121046.

[7] N. Lu, G. Wu, Z. Zhang, Y. Zheng, Y. Ren, and K. R. Choo, "Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 23, p. e5627, Dec. 2020, doi: 10.1002/cpe.5627.

[8] H. Rosa, J. P. Carvalho, P. Calado, B. Martins, R. Ribeiro, and L. Coheur, "Using fuzzy fingerprints for cyberbullying detection in social networks," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2018, pp. 1–7, doi: 10.1109/FUZZ-IEEE.2018.8491557.

[9] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, "Detection of harassment on web 2.0," in *Proc. Content Anal. WEB*, 2009, vol. 2, pp. 1–7.

[10] S. C. Eshan and M. S. Hasan, "An application of machine learning to detect abusive bengali text," in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*. IEEE, Dec. 2017, pp. 1–6, doi: 10.1109/ICCITECHN.2017.8281787.

[11] A. Kumar and N. Sachdeva, "A bi-GRU with attention and CapsNet hybrid model for cyberbullying detection on social media," *World Wide Web*, vol. 25, no. 4, pp. 1537–1550, Jul. 2022, doi: 10.1007/s11280-021-00920-4.

[12] K. Maity, S. Saha, and P. Bhattacharyya, "Cyberbullying detection in code-mixed languages: Dataset and techniques," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 1692–1698, doi: 10.1109/ICPR56361.2022.9956390.

[13] S. Balakrishna, Y. Gopi, and V. K. Solanki, "Comparative analysis on deep neural network models for detection of cyberbullying on social media," *Ingeniería Solidaria*, vol. 18, no. 1, pp. 1–33, Jan. 2022, doi: 10.16925/2357-6014.2022.01.05.

[14] R. R. Dalvi, S. Baliram Chavan, and A. Halbe, "Detecting a Twitter cyberbullying using machine learning," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2020, pp. 297–301, doi: 10.1109/ICICCS48265.2020.9120893.

[15] M. A. Al-Ajlan and M. Ykhlef, "Optimized Twitter cyberbullying detection based on deep learning," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Dec. 2018, pp. 1–5, doi: 10.1109/NCG.2018.8593146.

[16] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, "Detection of cyberbullying using deep neural network," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2019, pp. 604–607, doi: 10.1109/ICACCS.2019.8728378.

[17] A. John, A. C. Glendenning, A. Marchant, P. Montgomery, A. Stewart, S. Wood, K. Lloyd, and K. Hawton, "Self-harm, suicidal behaviours, and cyberbullying in children and young people: Systematic review," *J. Med. Internet Res.*, vol. 20, no. 4, p. 9044, 2018, doi: 10.2196/jmir.9044.

[18] E. L. Backe, P. Lilleston, and J. McCleary-Sills, "Networked individuals, gendered violence: A literature review of cyberviolence," *Violence Gender*, vol. 5, no. 3, pp. 135–146, 2018, doi: 10.1089/vio.2017.0056.

[19] V. Vijayakumar, D. H. Prasad, and P. Adolf, "Multi-input deep learning algorithm for cyberbullying detection," *Int. J. Res. Eng. Appl. Manag.*, vol. 7, no. 5, 2021.

[20] D. A. Andrade-Segarra and G. A. Leon-Paredes, "Deep learning-based natural language processing methods comparison for presumptive detection of cyberbullying in social networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, 2021, Art. no. 0120592.

[21] S. S. Shylaja, A. Narayanan, A. Venugopal, and A. Prasad, "Recurrent neural network architectures with trained document embeddings for flagging cyber-aggressive comments on social media," in *Proc. Int. Conf. Adv. Comput. Commun. (ADCOM)*, 2018.

[22] S. Paul, S. Saha, and J. P. Singh, "COVID-19 and cyberbullying: Deep ensemble model to identify cyberbullying from code-switched languages during the pandemic," *Multimedia Tools Appl.*, vol. 82, pp. 8773–8789, Jan. 2022, doi: 10.1007/s11042-021-11601-9.

[23] M. Raj, S. Singh, K. Solanki, and R. Selvanambi, "An application to detect cyberbullying using machine learning and deep learning techniques," *Social Netw. Comput. Sci.*, vol. 3, no. 5, pp. 1–13, Jul. 2022, doi: 10.1007/s42979-022-01308-5.

[24] A. Dewani, M. A. Memon, and S. Bhatti, "Cyberbullying detection: Advanced preprocessing techniques & deep learning architecture for Roman Urdu data," *J. Big Data*, vol. 8, no. 1, pp. 1–20, Dec. 2021, doi: 10.1186/s40537-021-00550-7.

[25] A. Agarwal, A. S. Chivukula, M. H. Bhuyan, T. Jan, B. Narayan, and M. Prasad, "Identification and classification of cyberbullying posts: A recurrent neural network approach using under-sampling and class weighting," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2020, pp. 113–120.

[26] L. Anindyati, A. Purwarianti, and A. Nursanti, "Optimizing deep learning for detection cyberbullying text in Indonesian language," in *Proc. Int. Conf. Adv. Informatics: Concepts, Theory Appl. (ICAICTA)*, Sep. 2019, pp. 1–5, doi: 10.1109/ICAICTA.2019.8904108.

[27] C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, "Cyberbullying detection solutions based on deep learning architectures," *Multimedia Syst.*, vol. 29, pp. 1839–1852, Oct. 2020, doi: 10.1007/s00530-020-00701-5.

[28] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying detection using pre-trained BERT model," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 1096–1100, doi: 10.1109/ICESC48915.2020.9155700.