Multimedia Systems
https://doi.org/10.1007/s00530-020-00701-5

SPECIAL ISSUE PAPER

Cyberbullying detection solutions based on deep learning architectures

Celestine Iwendi[1] · Gautam Srivastava[2,3] · Suleman Khan[4] · Praveen Kumar Reddy Maddikunta[5]

Received: 1 July 2020 / Accepted: 24 September 2020

Abstract

Cyberbullying is disturbing and troubling online misconduct. It appears in various forms and is usually in a textual format in most social networks. Intelligent systems are necessary for automated detection of these incidents. Some of the recent experiments have tackled this issue with traditional machine learning models. Most of the models have been applied to one social network at a time. The latest research has seen different models based on deep learning algorithms make an impact on the detection of cyberbullying. These detection mechanisms have resulted in efficient identification of incidences while others have limitations of standard identification versions. This paper performs an empirical analysis to determine the effectiveness and performance of deep learning algorithms in detecting insults in Social Commentary. The following four deep learning models were used for experimental results, namely: Bidirectional Long Short-Term Memory (BLSTM), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN). Data pre-processing steps were followed that included text cleaning, tokenization, stemming, Lemmatization, and removal of stop words. After performing data pre-processing, clean textual data is passed to deep learning algorithms for prediction. The results show that the BLSTM model achieved high accuracy and $F1$-measure scores in comparison to RNN, LSTM, and GRU. Our in-depth results shown which deep learning models can be most effective against cyberbullying when directly compared with others and paves the way for future hybrid technologies that may be employed to combat this serious online issue.

Keywords Cyberbullying · Social media · Deep learning · NLP · Mining · Emotions

1 Introduction

The development of information and networking technology has created open online communication channels. Unfortunately, trolls have exploited this technology for cyber-attacks and threats. Statistics show that about 18% of Europe's children were affected either through people bullying or harassing them via the Internet and mobile communication. EU Kids Online Report of 2014 stated that nearly 20% of kids who are between the ages of 11 and 16 are vulnerable to cyberbullying [19]. Quantitative research [29] indicates cyber-victimization rates among adolescents ranging from 20 to 40%. All of these show how important it is to find an adequate, speedy, and tested approach to solving this online pandemic.

There is a need to consider and tackle cyber-bullying from various viewpoints including automatic detection and avoidance of these accidents. There are methods already developed that can mark as instances of bullying, including the engagement of [9] services that seek to help the victims

✉ Gautam Srivastava
srivastavag@brandonu.ca

Celestine Iwendi
celestine.iwendi@ieee.org

Suleman Khan
171518@students.au.edu.pk

Praveen Kumar Reddy Maddikunta
praveenkumarreddy@vit.ac.in

1 Department of Electronics, BCC of Central South University of Forestry and Technology, Changsha, China

2 Department of Math and Computer Science, Brandon University, Brandon, MB, Canada

3 Research Center for Interneural Computing, China Medical University, Taichung 40402, Taiwan, ROC

4 Air University Islamabad, Islamabad, Pakistan

5 Vellore Institute of Technology, Vellore, Tamil Nadu, India

Published online: 13 October 2020

[32]. Besides, most online channels that are widely used by adolescents have safe centers, such as YouTube Safety Center and Twitter Safety and Protection, which offer user assistance and track communications.

A jet age transformation of cyberbullying is now in force and has become very common with students seeing it as fun on cyberspace to harass their friends and enemies. The authors in [22] discussed the metamorphosis in the domain of electronics and how the influence has become negative, creating problems for the world. Furthermore, their research follows an extension of an original study intended to evaluate by experimental procedures the nature and extent of cyberbullying. They aim to add to the fact that there can be a systemic stoppage of communication by intersecting between communication and computers. Hence, providing a backdrop of which there can be continuity of their research.

In cyberspace, cyber-bullying takes place through several mediums, and it's mainly on social media, where youth and adults access almost all the time. A cyber-bullying research group surveyed between July and October 2016. High school students and the findings show that 34% of students had encountered cyber-bullying in their lifetime [3]. Given this, automatic and valid identification of cyber-bullying is necessary to resolve such issues. Researchers have suggested that cyber-bullying comes in various ways, such as embarrassment, stalking, coercion, exploitation, or domination of a designated victim [10]. All types can be summarized in text format when the words are explicit or implied. Explicit 11 expressions arise by using profane words with a negative emotion, where implicit expressions may come with ironic or cynical phrases that have no foul words. There has been a lot of research on explicit speech identification. Still, a lot of work is needed to solve the implicit language that makes detecting cyber-bullying on social media a challenging job.

Nevertheless, progress has been made into the detection of cyber-bullying using approaches to machine learning (ML) and deep learning (DL). However, much of the current work needs to be more developed to provide a reliable approach that incorporates clear and indirect factors into account. Analytical research aimed at evaluating the output of DL algorithms is discussed in this paper.

The contributions of this paper include:

1. Deep Bidirectional Long Short Term Memory (BLSTM) is used for prediction. We also used linguistic methods to evaluate our results. We used parts of speech to see which type of pattern is followed by normal and abusing tweets.
2. Text cleansing, tokenization, stemming, Lemmatization, and removal of stop words are performed as pre-processing steps.
3. Application of four deep learning models for the experimental results, namely: Bidirectional Long ShortTerm Memory (BLSTM), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN).
4. Carried out an empirical analysis to determine the effectiveness and performance of deep learning algorithms in detecting insults in Social Commentary.
5. Finally, a comparison between all the DL algorithm used. Our result shows that the BLSTM model achieved high accuracy and $F1$-measure scores in contrast to RNN LSTM and GRU in detecting insults in Social Commentary.

The rest of the paper is organized as follows: Sect. 2 is about related work, the proposed methodology is discussed in Sect. 3, Sect. 4 discusses experimental results and finally, Sect. 5 concludes the paper.

2 Related work

In recent times, there has been an increase in online activities by teens, especially on social networking sites, which have invariably exposed them to cyber-bullying. Comments containing abusive words affect the psychology of teens, demoralize them, and may lead them to depression or suicide. Two rules for feature extraction that was used to detect perceived negative and offensive comments often directed towards peers were presented by the authors [7]. Their combined hand-crafted features with the traditional feature extraction tend to increase the accuracy of detection of the system positively. Although current methods inspired by deep learning and machine learning have enhanced the accuracy of cyber-bullying detection, the fact remains that lack of good standard labeled datasets limits the advancement of this approach. Therefore, the authors [8] proposed a system where the dataset of the user comments for labeling is applied with crowdsourcing, capturing the real-time scenario of deliberately abusive words or kind words.

The authors in [28] argue that Cybersecurity experts should not be pushed away in the advent of AI-controlled cyberbullying. They must be allowed to continue doing their job and testing networks, just as doctors are still allowed to read the result of X-ray scans in situations where Human Intelligence is needed to control the Artificial intelligence. This is the goal of the next generation of artificial intelligence, known as AI 2.0. The idea of using soft computing methods for the detection of cyberbullying, especially in social media platforms, was studied by [18]. They compared their study with previous studies and came up with the idea that social media platforms should use a meta-analytic method in tackling cyberbullying detection. Using a method to identify, text classifying, and personalized text-based cyberstalking was created by [13]. It was an ethical framework and a way to detect text-based cyberstalking. They went further to focus on using other initiatives such as digital forensics to perform author identification.

The authors in [31] presented different stages and multiple technique systems that first uses crowdsourcing for post and hashtag annotation and subsequently uses machine-learning methods to identify extra posts for annotation. This is proper research as we can compare our latest results and that performed from this paper. They concluded that you could have an excellent performance of models if the dataset is trained with their approach. Meanwhile, according to [21, 26], their results from multiple variate logistic regression techniques show that physical violence is associated with peers that smoke or with another feature of carrying weapons before the cyberbullying enactment. They highlight the importance of ensuring positive and reliable monitoring by parents, teachers, or peers to improve cyberbullying prevention efforts. The author in [35] proposed a method that is capable of analyzing the hidden feature structure of cyberbullying evidence and acquire a vigorous and discriminative depiction of text. Finally, their results and approaches perform better than other baseline text depiction methods.

The authors in [25] used 22 studies and experiments to validate current practices on automatic cyberbullying detection. They finally ended with results indicating that cyberbullying is frequently distorted, creating the assumption that it is not a big deal after all. With the imbalance of datasets, it is difficult to have actual practical impartation of the consequences of cyberbullying. Rosa et al. used for their studies two data sets, which are intimidating trace and databases with Formspring. To build the prediction models, they implemented Support Vector Machines (SVM), Random Forest, and Logistical Regression. $F1$ score was used to test the outcomes of experiments, and the embedding achieves an $F1$ score of 0.45. This same approach was considered by [27]. The only difference is that their response grading system felt the ruthlessness of cyberbullying and gave suitable responses.

Rakib et al. first developed a word embedding application using Reddit, then followed by a cyberbullying identification model using the Kaggle dataset comprising of 6594 comments. Random Forest model was used to train the system. The prediction model obtained 0.90 Area under the curve (AUC) and 0.89 Precision. The drawback of this analysis, however, is that the sample is also imbalanced with cyberbullying texts consisting of only 25% [24]. While the authors [1, 11] repeated another related research with three real-world datasets, namely: Formspring, Twitter, and Wikipedia. They applied deep learning algorithms to construct the prediction models after the datasets were extracted with three word embedding structures, including random vector initialization, GloVe, and Sense-Specific Embedding Word (SSWE). They used over-sampling methods and maybe a limitation of this application and Reproducibility study from integrating other sources of information from the impact of gaining access to the social media profile.

Haidai et al. suggested a multilingual cyberbullying prevention system [14]. The authors strive to avoid cyberbullying attacks on the Arabic language. The experiment was performed on a real-time Arabic dataset from Arab countries. During this cycle, the authors used Dataiku DSS and WEKA, supporting Arabic. Naive Bayes and SVM Classifiers are used for prediction, achieving satisfactory results. Nevertheless, this research can be expanded by considering deep learning and increasing the dataset size. In [2], the researchers implemented a cyberbullying strategy by collecting 20,000 random tweets. Data pre-processing was applied to remove noisy and unwanted data. Such pre-processed data is divided into training and trained data. For training data, tweet classification was provided to mark tweets. Later, deep convolutional neural networks were used to classify a dataset. No encouraging experimental results were achieved. Research must be expanded by considering a large dataset and several languages. Similarly, the authors in [4] used deep convolutionary neural networks by considering the 69,874 tweets twitter dataset. Tweets were mapped to vectors through Glove's open-source word embedding. The experimental results showed that with deep convolutionary neural networks, the authors achieved 93.7% accuracy. However, detecting cyberbullying in chats containing Hindi and English together can further expand the research.

Wiki-Detox dataset was the main point of the research of [33]. They presented a classifier that can generate a result roughly as good as the 3 Human Workers in total, as calculated by ROC curve and Spearman correlation field. In terms of model construction, they looked at three dimensions: Architecture model (Logistic Regression vs Multi-Layer Perceptron), sort n-gram (word vs char) and sort of mark (one-hot vs empirical distribution). They then answer questions on identifying harassment using their classifier.

We used the work of [6] as a practical application based on Turkish contents since the detection of cyberbullying has been ignored. The authors designed eight different artificial neural network models that detected cyberbullying in Turkish social media. According to the evaluation results, they had 91% $F1$-measure score and a better performance than the experimented machine learning classifiers in their previous study. Another similar scenario is the work done by [23]. Their paper presented a solution for detecting and stopping cyberbullying with focus on content written in the Arabic language.

Finally, deep learning was optimized with the algorithm, generating a good parameter tuning. The authors in [17] used another scenario and proved the inefficiency in the classification of previous methods. The only limitation which we are now considering in our research is lack of regressive training of system that makes sure that cyberbullying is detected in real-time chats. It also created another channel where cyberbullying can be detected in chats that contains a mixture of different words in different languages. We can conclude after considering the stipulated approaches that our approach will solve most of the limitations of the past and present research in the detection of words used to harass, intimidate others while using the social media.

3 Methodology

The proposed methodology is depicted in Fig. 1. The steps applied in the methodology are listed below:

- Load the dataset from the Kaggle repository.
- Perform pre-processing of the dataset by doing text cleaning, tokenization, stemming, lemmatization and stop word removal.
- After cleaning the text, some linguistic approaches used to analyze the bad comments pattern.
- Then Split the dataset into training and testing data.
- Train the dataset by various deep learning algorithms.
- Evaluate the performance of the deep learning algorithms by using the testing dataset with several metrics.

3.1 Dataset

In this research, the Kaggle dataset has been used [5] for the detection of insult over social media platforms. The dataset used in this research will help us to solve the problem of the classification of a single class. The label is either 0, which means a neutral statement or one which means an offensive comment. In order words, we are using it as neutral irrespective that it does not belong to the class of insults. The first attribute to consider is the date the comment was made. This is often blank, which means it is not possible to get an exact timestamp. This material is based mainly on commentary in the English language, with some editing on different occasions.

3.2 Pre-processing

3.2.1 Text cleaning

When a text is received based on the implementation of Fig. 1, the data is scrutinized to a high degree of refinements applying and following the steps stipulated below: Stop word remover, tokenization, lower casing, sentence segmentation, and punctuation removal. These are the steps that were taken to have the data reduced to size, and thus, we also removed unwanted information that could be found in the data. In furtherance of this approach, we created a generic pre-processing that resulted in the removal of punctuation and also some non-letter characters from each document. Finally, the letter case of each document was lowered. The result from this approach gave us a sliced document text based on the n length with an n-gram word-based tokenizer.

3.2.2 Tokenization

Tokenization has been used in this process to address a scenario where a given text will be separated into smaller bits known as tokens. The following are also regarded as tokens. They include Words, numbers, and punctuation marks. In addition, another non-sensitive equivalent element replaced by a sensitive data element with no meaning or value. We assured that the tokenization method used was protected and tested using the best standards relevant to the safety of confidential data. The tokenization framework methodology we have used offers authority and APIs for obtaining tokens for

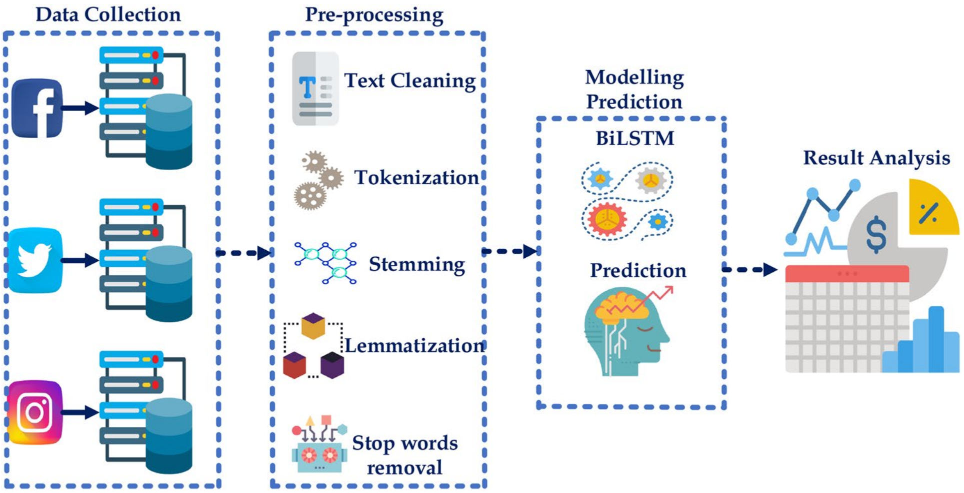


Fig. 1 Proposed model

data processing applications Where necessary and can be detokenized back to sensitive data.

3.2.3 Stemming

The next step after we had gone through the tokenization system is to transform the tokens into another standard format. Stemming, simply means, we can now change the words back to their form where we originally started from but now with a decrease in the number of words types and/or classes in the data. For example, we have used the words "Running," "Ran," and "Runner" was reduced to the word "run.". it shows that stemming can actually be used for make classification.

3.2.4 Lemmatization

Like stemming, the purpose of lemmatization is to minimize inflectional forms to a specific base form. Lemmatization does not necessarily break off inflections, rather than stemming. It just uses the bases of lexical information to achieve the right fundamental types of vocabulary.

3.2.5 Stopwords

This paper has used insignificant words as languages capable of creating noise as a useful feature when we are performing text classification. Such terms are called Stop words. We can see them used in sentences that assist in connecting our thinking while helping with the way the sentences are constructed. Articles, prepositions and conjunctions, and some pronouns, for example, are considered to stop words. Our method extracted common terms from the records, such as "*a, for, an, are, like, at, are, by, for, from, how, in, is, in, on, or, the, these, this too, was when, where, where, how, how, how,*" etc. Afterward, we store the documents being processed and prepared for the next step.

3.3 Application of long short term memory (LSTM)

RNN, which we know as a class of artificial neural networks with connections between nodes has some setbacks due to the excess number of network layers. It was a tough task, and we had to look at a recent study and discovered that the LSTM network is an answer to this challenge due to its chain structure similar to that of multiple neural network modules with RNN. Figure 2 represents the LSTM architecture, consisting of various gates, including 1st gate is the input gate, 2nd gate is the output gate, and it also has a forget gate that was used inside the LSTM model. We have used these gates in selecting how information is accepted and rejected across the network.

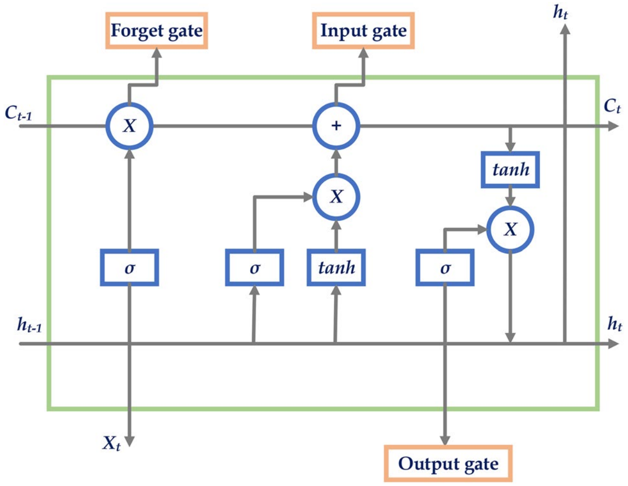


Fig. 2 Long short term memory architecture

When the activation function range Input from – 1 to 1 with a gate $i^{(t)}$ consisting of tan$h$, it made the current input to become $x^{(t)}$ with attributes $h^{(t-1)}$ and $C^{(t-1)}$. Inside forget gate $f^{(t)}$ there is tan$h$ and sigmoid function which is used as an activation function. It is interesting to note here that the forget gate makes the decision of the number of information to retain when it receives information from the previous output. For example, when we have a value to be 1, it means that the data was transferred to the network. But, if it is 0, it means the data will not be allowed to pass through the network. Note that the output gate $o^{(t)}$ also has sigmoid as another activation function with a range of -1 to 1. It shows that at any time mark, $i^{(t)}$, $o^{(t)}$, $f^{(t)}$ will be computed when we apply Eqs. (1), (2), and (3), respectively.

$$i^{(t)} = \sigma\left(W^i[C^{t-1}, h^{(t-1)}, x^{(t)}] + b^i\right), \tag{1}$$

$$o^{(t)} = \sigma\left(W^o[C^{t-1}, h^{(t-1)}, x^{(t)}] + b^i\right), \tag{2}$$

$$f^{(t)} = \sigma\left(W^f[C^{t-1}, h^{(t-1)}, x^{(t)}] + b^i\right). \tag{3}$$

What we have shown is different from the usual Traditional LSTM that works on bi-direction. This research we have used two LSTMs which includes; one LSTM for upward and downward scanning and the other LSTM used is for right and left scanning. We have imputed the second LSTM as a summation of the first LSTM. Therefore, our proposed LSTM utilizes double input gates, output gates and forget gates in comparison to the traditional LSTM known. This achievement gives a better accuracy. However, our proposed model has more computational complexity and cost in terms of performance.

3.4 Bidirectional long short term memory (BLSTM)

Bidirectional LSTM (BLSTM) model retains two separate input and forwards input states provided by two different LSTMs.The first LSTM is a regular sequence starting from the starting of the paragraph, while the second LSTM is a standard sequence, the series of inputs are fed in the opposite order. The concept behind the bi-directional network is to gather knowledge about the inputs around it. It typically knows more rapidly than a one-way approach, but it depends on the mission as shown in Fig. 3 that represents the stricture of used BLSTM model.

BLSTM mode consists of 200 neurons, 2nd layer has 400 neurons. We have 3 dense layers, 1st dense layer has 128 neurons, 2nd dense layer has 64 neurons, and 3rd dense layer has 32 neurons, respectively. We also used 3 dropout layers to avoid over-fitting.

3.4.1 Forget gate

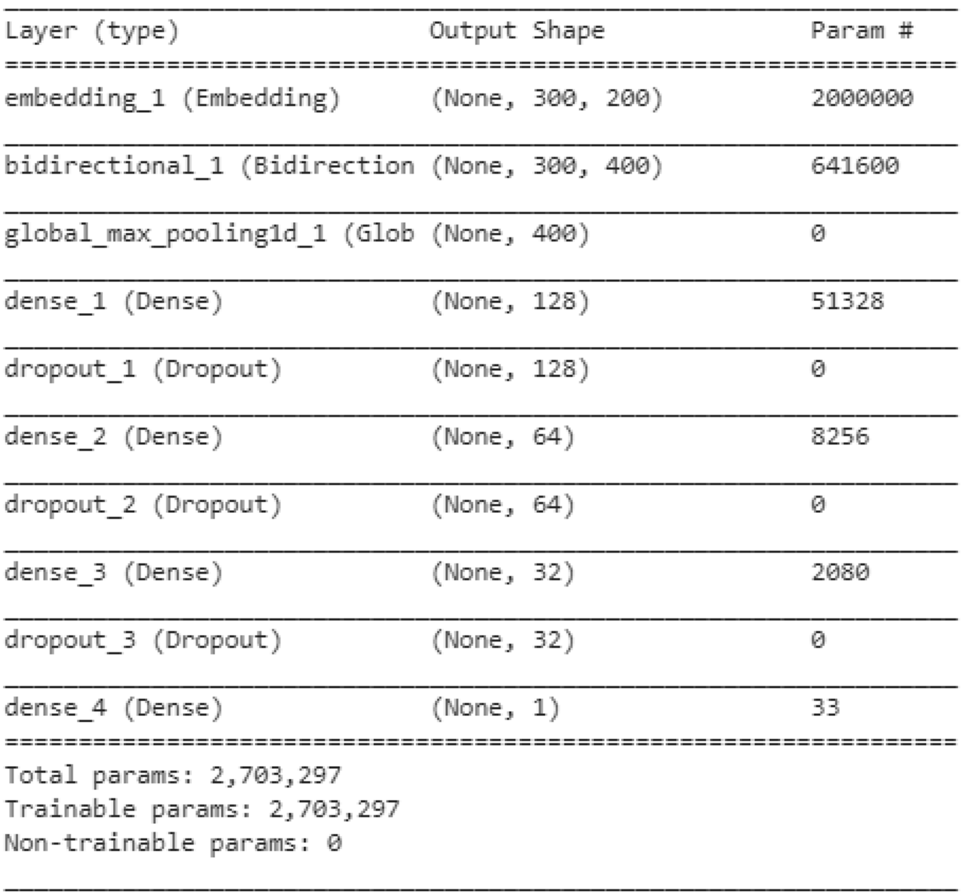
We have applied the Forget gate to be responsible for the way extraction is done with the cell-state information by multiplying a filter. This stage removes the information that we don't need to make the LSTM understand things or less critical data. This is important for the optimization of the LSTM network output.

$$f_t = \sigma\left(W_{\mathrm{f}} \cdot \left[h_{t-1}, x_t\right] + b_{\mathrm{f}}\right) \tag{4}$$

h_{t-1} is the hidden state of the previous cell or the last cell's output, and x_t is the input at that particular time step. We used the weight matrices to multiply the data that was given, and a bias is applied. Following this, the value is added with the sigmoid function and generates a vector corresponding to each number in the cell structure. The value varies from 0 to 1. Again, If '0' is the output given as the value in the cell state, the forgotten gate would want the cell state not to recognize the piece of knowledge. In the same way, a '1' means the lost gate will automatically recall the whole bit of knowledge. Finally, the vector output is multiplied to the cell state from the sigmoid function.

3.4.2 Input gate

The duty of the input gate is for the addition of cell state information. This was done by first involving a sigmoid function where it regulates which values are to be added to cell state.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 300, 200)          2000000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 300, 400)          641600
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 400)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               51328
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 2,703,297
Trainable params: 2,703,297
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3 Structure of BLSTM used

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{5}$$

$$\tilde{C}_t = \mathrm{tan}h\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \tag{6}$$

This is similar to the gate no more recognized in the network but acts as a filter for all $\left[h_{t-1}, x_t\right]$ information. It then generates a vector that will include all possible values applicable to the cell state (as interpreted from h_{t-1} and x_t). This is performed with the tan$h$ function, outputting values from -1 to $+1$. Finally, the value of the regulatory filter (the sigmoid gate) we have used is now multiplied to the vector generated (the tan$h$ function) and this information is then applied through additional operation to the cell status.

3.4.3 Output gate

The output gate acts as the selection center. Important cell state information is picked as inputs. A vector is created after applying the tan$h$ function to the cell state when the scaling the value of the ranges down to operate from -1 to $+1$.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{7}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{8}$$

We then used the values of h_{t-1} and t to make a filter that controls the values needed to be extracted from the vector generated with the filter created using a sigmoid feature. Finally, the value of this regulatory filter is multiplied to the vector generated using the tan$h$ function.

3.4.4 Recurrent neural networks

Due to the vanishing gradient problem, conventional Neural Networks (NN) do not provide us a satisfactory result when implemented on time series data. In 1982 JohnHopfield implemented RNN to address the above-mentioned subject matter. See Fig. 4 represents the structure of RNN model.

RNN is better with the trends of using NN learn over a time frame. RNN was used to forecast serial data such as actions in a video based on past events, voice audio, text events, etc. Figure 4 demonstrates the operating configuration for the RNN. In the figure, X_t Its weight vector stands for the hidden layer and represents the output layer weight vector; E_t Is the output layer Weight Vector, D_t denotes matrix for the input word. Timestamp on the hidden layer t is measured by using Equation (9).

$$X_t = \sigma\left(A \times D_t + C \times X_{t-1}\right), \tag{9}$$

$$E_t = \sigma(B \times X_t), \tag{10}$$

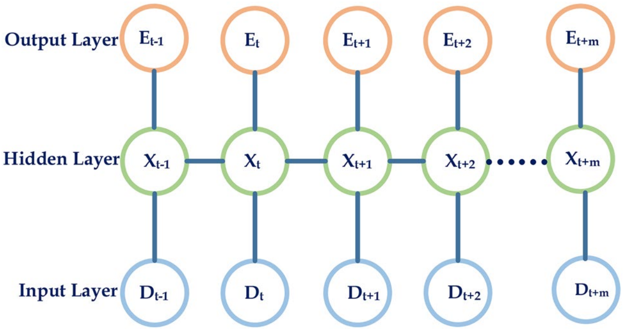


Fig. 4 Working model of RNN

where $\sigma(\cdot)$ the activation function is considered to be. The activation feature may be Sigmoid, Relu or Tanh. At any time mark t, the Concealed StateX_t is calculated by using Equation (9) with the required parameters and inputs.

3.5 Gated recurrent unit (GRU)

We have applied GRU as the latest variant of RNN designed to deal with short-term memory problems that are similar to LSTM. Note that GRU does not have a cell state and makes use of a hidden state to carry information. It consists of two gates: a reset gate r^t and an update gate z^t represented by Eqs. (11), (12). The update gate performs similar functions of the forget gate and an input gate of an LSTM with a responsibility to choose which information should be dropped or included. The reset gate determines the amount of the previous data be forgotten since GRU has fewer gates compared to LSTM, which speeds up the training process.

$$z^t = \sigma\left(w_{z^t}.x^t + U_{z^t}.h_{t-1} + b_{z^t}\right), \tag{11}$$

$$r^t = \sigma\left(w_{r^t}.x^t + U_{r^t}.h_{t-1} + b_{r^t}\right), \tag{12}$$

where z^t denotes update gate, $\sigma(\cdot)$ represents the sigmoid function, w, U and b: parameter matrices and vector, h_t denotes the output vector, x^t denotes the input vector.

4 Experiment results

Experimentation results are presented in this section following the idea from the authors in [15, 16, 20]. The experimentation is carried out using “Google Colab”, Google’s online Graphical Processing Unit (GPU). In this research we were equipped with Python 3.7 as our programming language, a good personal computer operating with a higher capacity Operating System and processor.

4.1 Performance metrics

This paper uses the below performance metrics in evaluating and projecting how our proposed BLSTM performed.

Confusion matrix We have used the confusion matrix to evaluate the performance of a classification model. This is a typical usage of the confusion matrix.

$$\text{Accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{\text{Total Instances}}. \tag{13}$$

$$\text{Precision} = \frac{\text{True Positive}}{(\text{Predicted Instances} = \text{True})}. \tag{14}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Actual number of instances as True}}. \tag{15}$$

$$F1\text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{ Recall}} \tag{16}$$

The word cloud we have implemented is an image collection of cyberbullying words used in a precise text or subject used in open online communication, in which the size of each word designates its occurrence or significance [12, 30, 34]. Therefore, from Figs. 5 and 6 we can see that

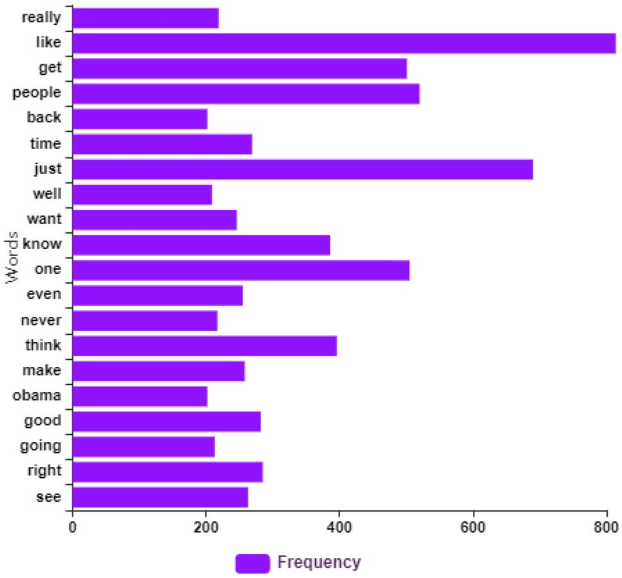
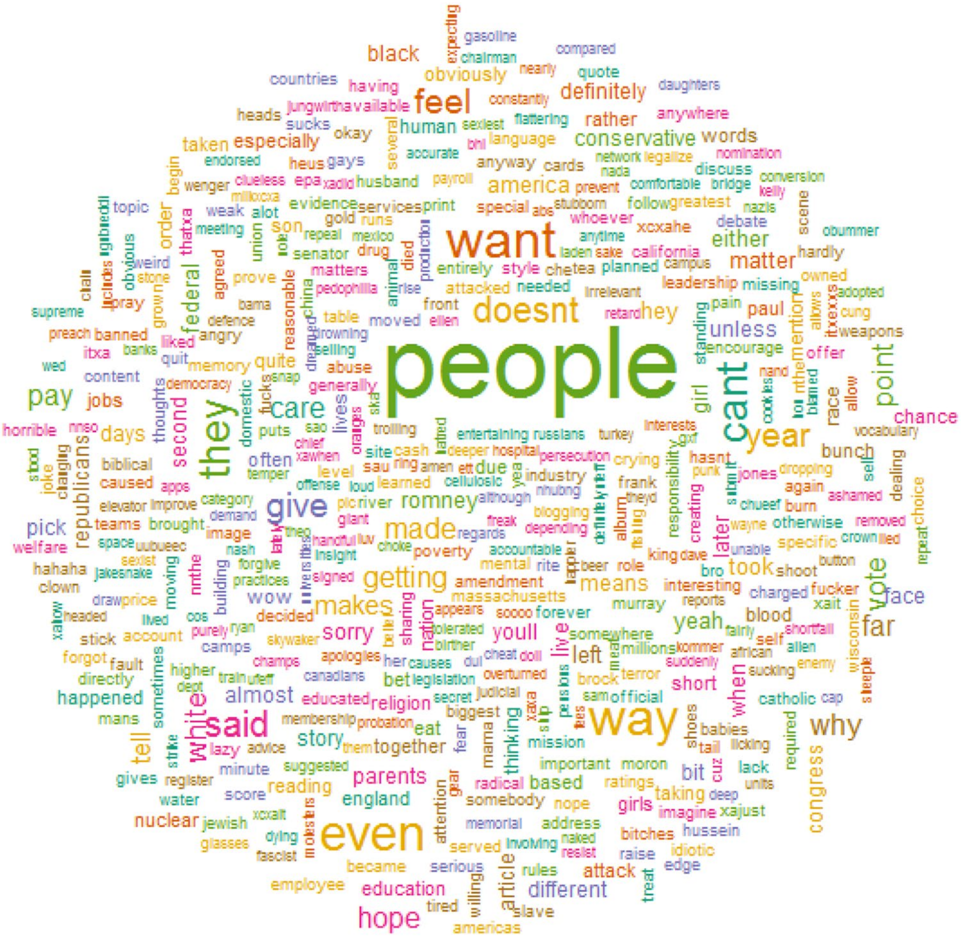


Fig. 6 Frequency bargraph for neutral words

Fig. 5 Wordcloud for neutral words



majority words used in neutral sentences are “people”, “like”,“just”,“make”,“now”,“right”,“can”,“one”, “think” these are majority words or most frequent words used in neutral sentences. Figure 7 depict that majority of words in neutral sentences are positive and its count is 4232, and 3888 words are negative. 1831 words contain angry words. Similarly, 1983 words contain fear in words. 1013 words contains sadness in words. Anticipations, disgust, joy and trust words are 2174, 1488, 1783, 2945 in sentences, respectively.

From Figs. 8 and 9, we can see that majority words used in neutral sentences are “people”,“shit”,“think”,“idiot”,“life ”,“little”,“bitch” ,“back”,“dumb” these most frequent words used in bad sentences.

Figure 10 depict that majority of words in bad sentences are negative and its count is 1930 and 947 words are positive. 703 words contain angry words. Similarly, 632 words contains fear in words. 704 words contains sadness in words. Anticipations, disgust, joy and trust words are 488, 943, 417, 641 in sentences, respectively (see Figs. 11, 12).

In this research, 4 deep learning models were used for the detection of cyberbullying. The results show that BLSTM outperformed other deep learning models when we consider accuracy, Precision, Recall and $F1$-Measure.

From Table 2 we can see that BLSTM testing accuracy is 82.18% while testing loss is 1.8 after 20 epoch. Similarly, GRU and RNN achieved 81.46% and 81.01% testing accuracy, respectively. Loss for GRU and RNN model is 1.9 and 1.5 , respectively. LSTM testing and loss scores are 80.86% and 2.1 , respectively.

Precision, Recall and $F1$-Measure scores for the normal class using BLSTM model are 86%, 91% and 88% , respectively as seen in Fig. 11. Similarly for Insult class Precision is 71%, Recall is 60% and $F1$-Measure is 65% , respectively. For GRU normal class Precision, Recall and $F1$-Measure scores are 86%, 89% and 87% , respectively. Similarly for Insult class Precision, Recall and $F1$-Measure scores are 68%, 62% and 65% , respectively as shown Table 1 and Fig. 11. Classifier accuracy scores are given in Fig. 12.

LSTM and RNN model Precision score for normal class is 85% each , respectively. Similarly, Recall and $F1$-Measure scores for both the models are 90% and 87% each, respectively.

Receiver Operating Characteristic (ROC) curve for BLSTM is represented in Fig. 13 and for GRU ROC curve is represented in Fig. 14. GRU area under the curve (AUC) has a score of 74.72% which is an increased for BLSTM by 2% and for BLSTM AUC score is 76.56%. Similarly, AUC score for LSTM and RNN are 73.30% as shown in Figs. 15 and 16, respectively. BLSTM AUC is 3% higher then LSTM and 5% from RNN algorithm. BLSTM on this dataset outperforms others in terms of Precision, Recall, $F1$-Measure and AUC.

LSTM model for insult class achieved 68% Precision, 80% Recall and 63% $F1$-Measure , respectively. Similarly, Precision, Recall and $F1$-Measure scores for RNN model insult class are 69%, 57% and 62% , respectively. In this research, we find what type of most frequent words are used to insult someone over the social media. We also find different emotions inside the text in both insulting text and in normal text.

From Table 3 we can see that for Bidirectional Long Short Term Memory 1735 tweets were detected correctly

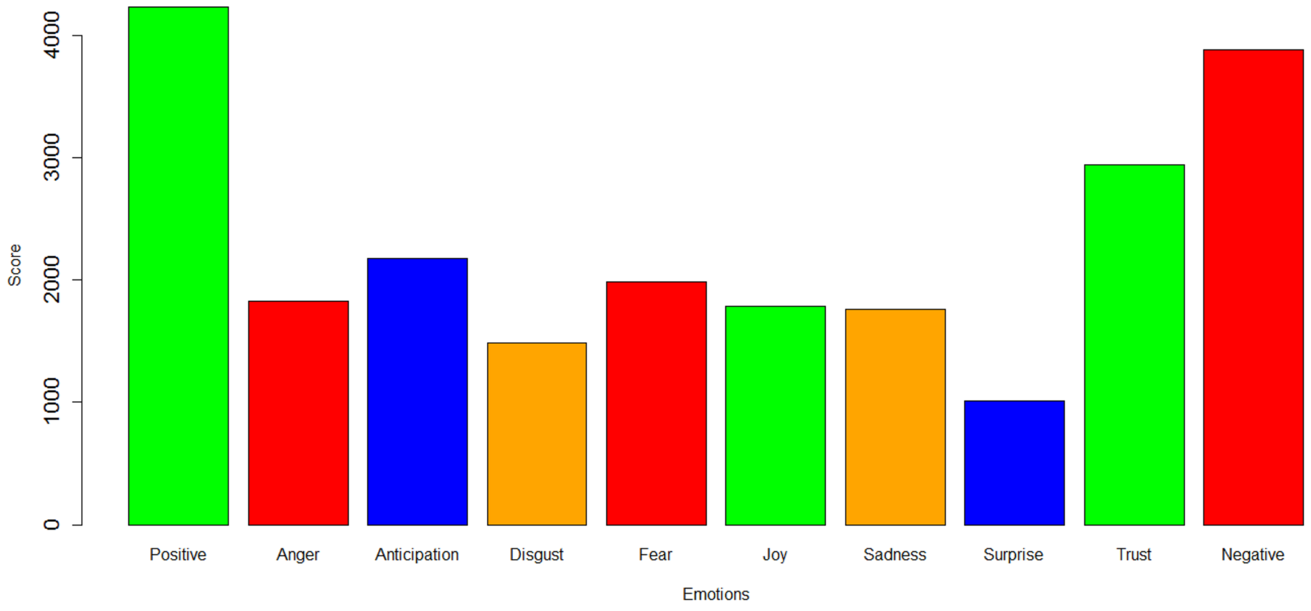


Fig. 7 Emotion mining neutral words

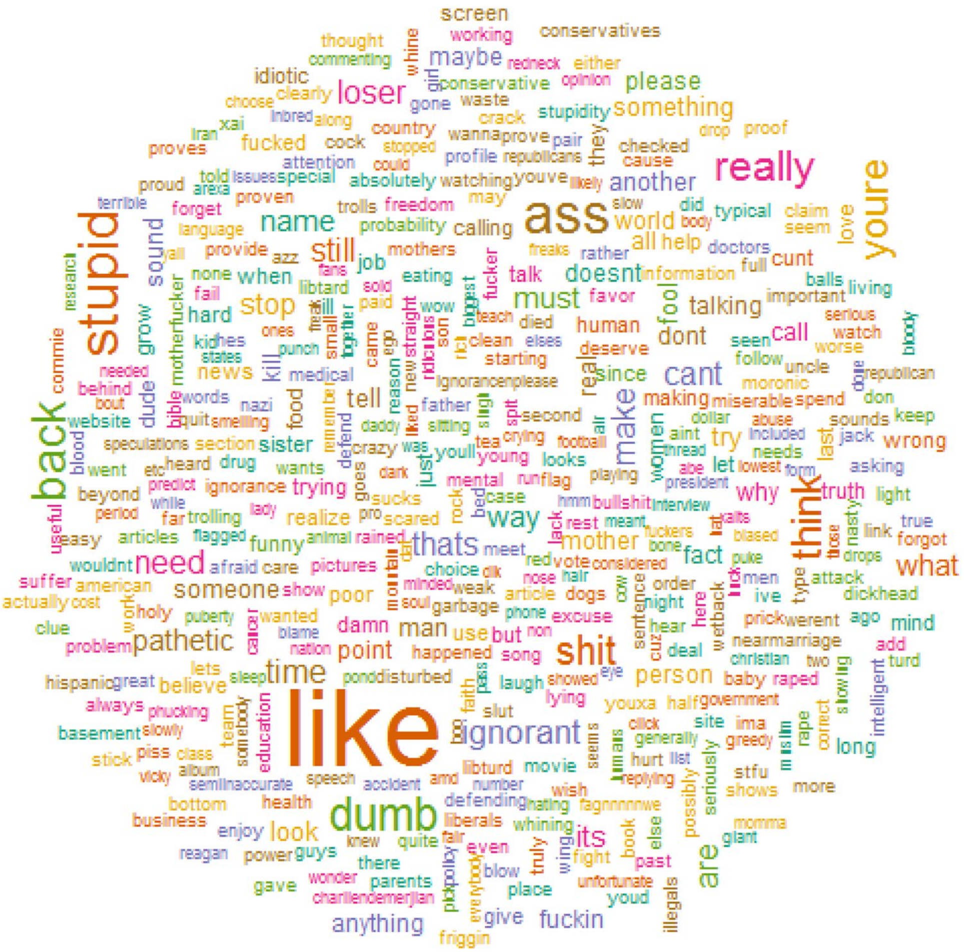


Fig. 8 Wordcloud for bad words

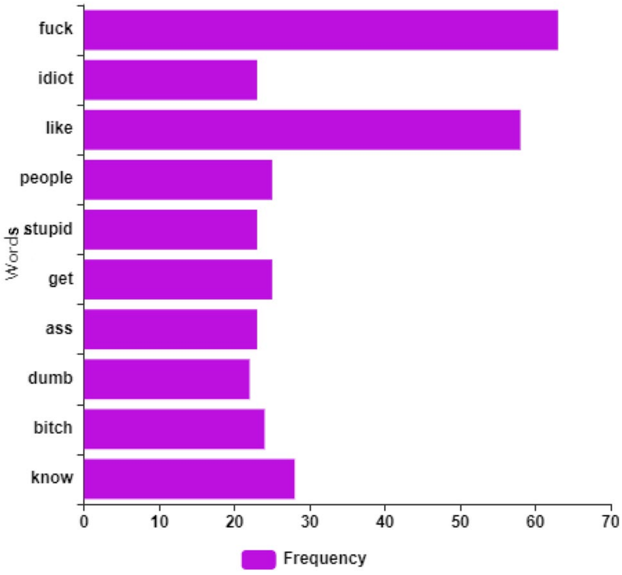


Fig. 9 Frequency bargraph for bad words

Table 1 Classification report for all models

Model	Labels	Precision	Recall	$F1$-Measure
BLSTM	Normal	86	91	88
	Insult	71	60	65
GRU	Normal	86	89	87
	Insult	68	62	65
LSTM	Normal	85	90	87
	Insult	68	80	63
RNN	Normal	85	90	87
	Insult	69	57	62

Table 2 Accuracy of all models

Model	Testing accuracy
BLSTM	82.18
GRU	81.46
LSTM	80.86
RNN	81.01

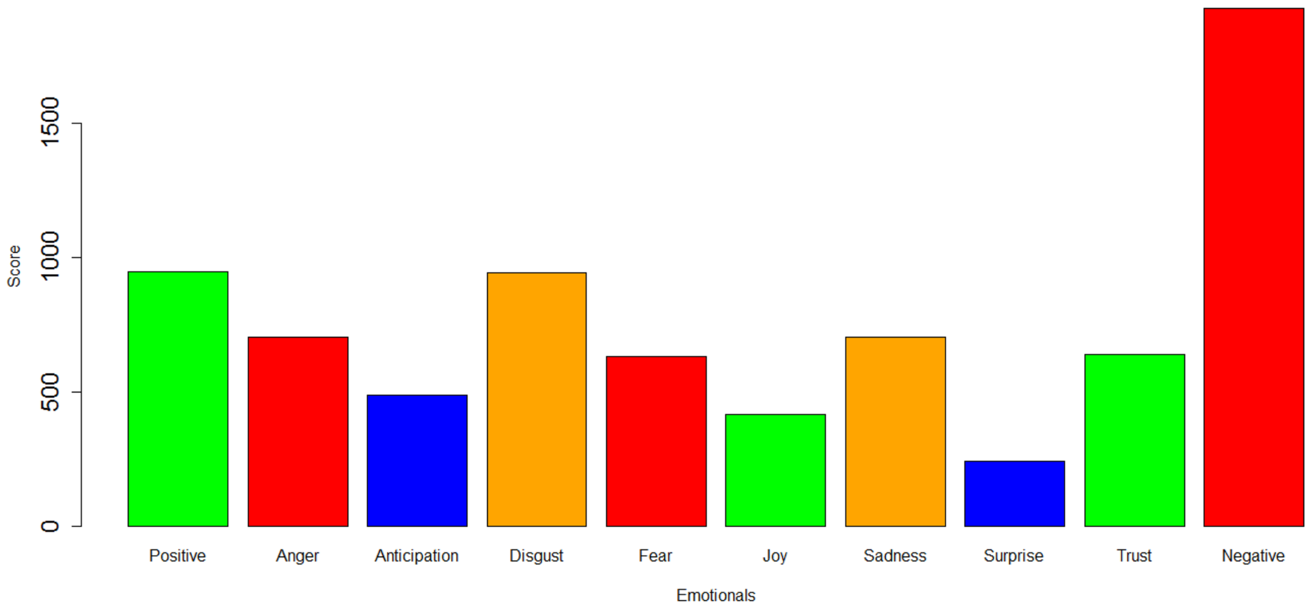


Fig. 10 Emotion mining bad words

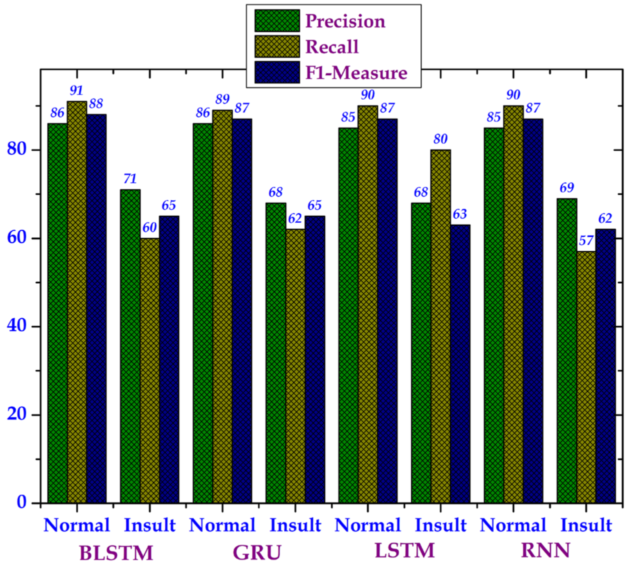


Fig. 11 Classification report

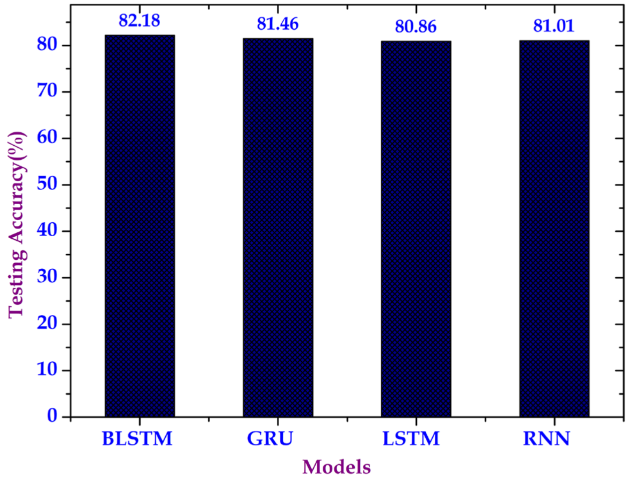


Fig. 12 Classifiers accuracy

as no abusing tweet while 176 tweets predicated as abusing tweets. For Long Short Term Memory true positive and true negative tweets are 1712 and 421 , respectively. Similarly, false positive and false positive tweets for LSTM is 199 and 277 , respectively. True positive rate tweets for Recurrent Neural Network and Gated Recurrent Unit are 1699 and 1722 , respectively. Similarly, true negative tweets for both GRU and RNN are 450 and 415 , respectively. For GRU false positive and false negative rates for tweets are 212 and 277, respectively. Similarly, false positive and false negative scores using RNN for tweets are 312 and 415, respectively.

Receiver Operating Characteristic (ROC) curve for BLSTM is represented in Fig. 13 and for GRU ROC curve is represented in Fig. 14. GRU area under the curve (AUC) has a score of 74.72% which is an increased for BLSTM by 2% and for BLSTM AUC score is 76.56%. Similarly, AUC score for LSTM and RNN are 73.30% as shown in Figs. 15 and 16 , respectively. BLSTM AUC is 3% higher then LSTM and 5% from RNN algorithm. BLSTM on this dataset outperforms others in terms of Precision, Recall, $F1$-Measure and AUC.

LSTM model for insult class achieved 68% Precision, 80% Recall and 63% $F1$-Measure , respectively. Similarly

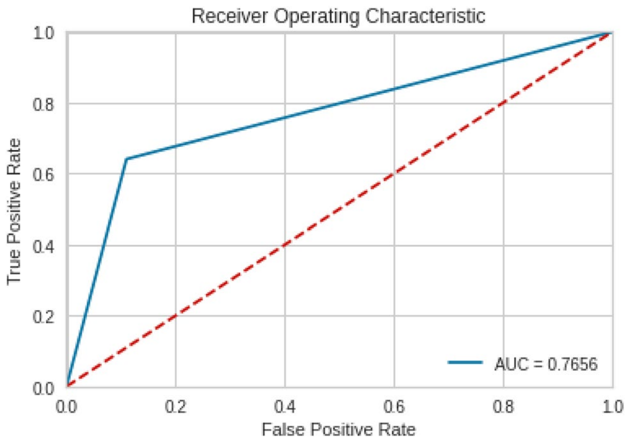


Fig. 13 BLSTM ROC curve

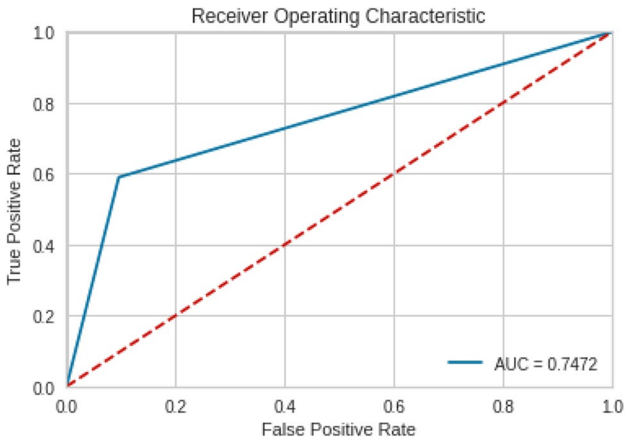


Fig. 14 GRU ROC curve

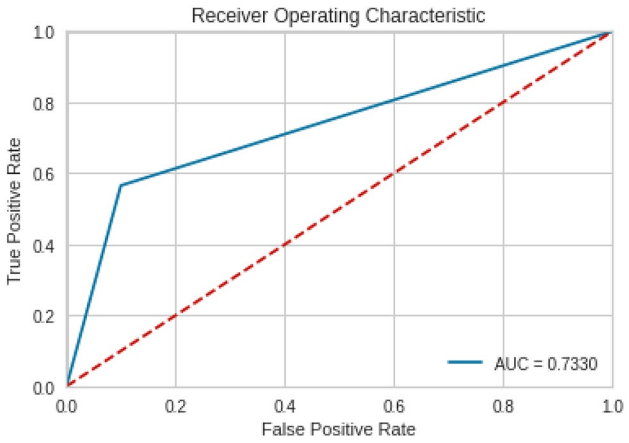


Fig. 15 LSTM ROC curve

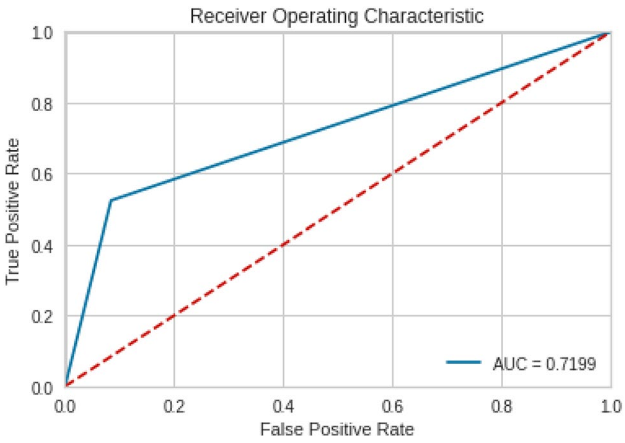


Fig. 16 RNN ROC Curve

Table 3 Confusion matrix

Classifiers	TP	FP	FN	TN
BLSTM	1735	176	294	433
GRU	1699	212	277	450
LSTM	1712	199	306	421
RNN	1722	189	312	415

Precision, Recall and $F1$-Measure scores for RNN model insult class are 69%, 57% and 62% , respectively. In this research, we find what type of most frequent words are used to insult someone over the social media. We also find different emotions inside the text in both insulting text and in normal text.

From Table 3, we can see that for Bidirectional Long Short Term Memory 1735 tweets were detected correctly as no abusing tweet while 176 tweets predicated as abusing tweets. For Long Short Term Memory true positive and true negative tweets are 1712 and 421 , respectively. Similarly false positive and false positive tweets for LSTM is 199 and 277 , respectively. True positive rate tweets for Recurrent Neural Network and Gated Recurrent Unit are 1699 and 1722 , respectively. Similarly, true negative tweets for both GRU and RNN are 450 and 415 , respectively. For GRU false positive and false negative rates for tweets are 212 and 277 , respectively. Similarly false positive and false negative scores using RNN for tweets are 312 and 415, respectively.

5 Conclusion and future work

The advent of information and networking technology has created the good, the bad, and the ugly in online communication responses. These responses are often abused and have caused irreparable emotional damage that most often lead to depression and suicide on innocent individuals when they were unable to speak out to get help from different agencies or family members. Meanwhile, some researchers have previously discussed this issue with traditional machine learning models, however most of these models built in these experiments can be applied to one social network at a time. In this paper, our novel methoodology is compared with the latest research on using deep learning-based models to make their way in the detection of cyberbullying incidents. Our proposed LSTM utilizes doubled input gates, output gates, and forget gates in comparison to the traditional LSTM in use. This achievement gives better accuracy. However, our proposed model has more computational complexity and cost in terms of performance. This paper takes a closer look at resolving the limitations of previous studies with better identification efficiency when directly compared to standard versions. Furthermore, our paper has performed an empirical analysis to determine the effectiveness and performance of deep learning algorithms in detecting insults in Social Commentary. Four deep learning models, namely RNN, LSTM, GRU, and BLSTM are used in the experiments. Data pre-processing steps are applied that include: text cleaning, tokenization, stemming, as well as lemmatization to remove and stop words in the communication chain from getting to gullible users. The data from the preprocessing step is later passed through clean textual data and directly into deep learning algorithms for prediction. We can conclude that the BLSTM model achieved high accuracy and $F1$-measure scores in comparison to RNN, LSTM, and GRU. In the future, we shall integrate our deep learning approach with automatic detection by tracking and object identification mechanism through the application of the next phase of artificial intelligence (AI) 2.0 interface to aid law enforcement agencies in the smart city to curb the menace of cyberbullying.

References

1. Agrawal, S., Awekar, A.: Deep learning for detecting cyberbullying across multiple social media platforms. In: European Conference on Information Retrieval, pp. 141–153. Springer (2018)
2. Al-Ajlan, M.A., Ykhlef, M.: Optimized twitter cyberbullying detection based on deep learning. In: 2018 21st Saudi Computer Society National Computer Conference (NCC), pp. 1–5. IEEE (2018)
3. Al-Hashedi, M., Soon, L.K., Goh, H.N.: Cyberbullying detection using deep learning and word embeddings: An empirical study. In: Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems, pp. 17–21 (2019)
4. Banerjee, V., Telavane, J., Gaikwad, P., Vartak, P.: Detection of cyberbullying using deep neural network. In: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 604–607. IEEE (2019)
5. Bhaskaran, J., Kamath, A., Paul, S.: DISCo: Detecting insults in social commentary. Stanford CS 229 Repository (2017)
6. Bozyiğit, A., Utku, S., Nasiboğlu, E.: Cyberbullying detection by using artificial neural network models. In: 2019 4th International Conference on Computer Science and Engineering (UBMK), pp. 520–524. IEEE (2019)
7. Chavan, V.S., Shylaja, S.: Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2354–2358. IEEE (2015)
8. Chen, H., Mckeever, S., Delany, S.J.: Presenting a labelled dataset for real-time detection of abusive user posts. In: Proceedings of the International Conference on Web Intelligence, pp. 884–890 (2017)
9. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 71–80. IEEE (2012)
10. Chisholm, J.F.: Review of the status of cyberbullying and cyberbullying prevention. J. Inf. Syst. Educ. **25**(1), 77 (2014)
11. Dadvar, M., Eckert, K.: Cyberbullying detection in social networks using deep learning based models; a reproducibility study. arXiv preprint arXiv:1812.08046 (2018)
12. Dwivedi, A.D., Malina, L., Dzurenda, P., Srivastava, G.: Optimized blockchain model for internet of things based healthcare applications. In: 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), pp. 135–139 (2019)
13. Frommholz, I., Al-Khateeb, H.M., Potthast, M., Ghasem, Z., Shukla, M., Short, E.: On textual analysis and machine learning for cyberstalking detection. Datenbank-Spektrum **16**(2), 127–135 (2016)
14. Haidar, B., Chamoun, M., Serhrouchni, A.: Multilingual cyberbullying detection system: Detecting cyberbullying in arabic content. In: 2017 1st Cyber Security in Networking Conference (CSNet), pp. 1–8. IEEE (2017)
15. Iwendi, C., Jalil, Z., Javed, A.R., Reddy, T., Kaluri, R., Srivastava, G., Jo, O.: Keysplitwatermark: zero watermarking algorithm for software protection against cyber-attacks. IEEE Access **8**, 72650–72660 (2020)
16. Javed, A.R., Sarwar, M.U., Khan, S., Iwendi, C., Mittal, M., Kumar, N.: Analyzing the effectiveness and contribution of each axis of tri-axial accelerometer sensor for accurate activity recognition. Sensors **20**(8), 2216 (2020)
17. Jeyasheeli, P.G., Selva, J.J.: An iot design for smart lighting in green buildings based on environmental factors. In: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1–5. IEEE (2017)
18. Kumar, A., Sachdeva, N.: Cyberbullying detection on social multimedia using soft computing techniques: a meta-analysis. Multimed. Tools Appl. **78**(17), 23973–24010 (2019)
19. Livingstone, S., Haddon, L., Hasebrink, U., Ólafsson, K., O'Neill, B., Smahel, D., Staksrud, E.: Eu kids online: Findings, methods, recommendations. LSE, London, EU Kids Online. http://lsedesignunit.com/EUKidsOnline (2014). Accessed May 2020
20. Mittal, M., Iwendi, C., Khan, S., Rehman Javed, A.: Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using levenberg-marquardt neural network and gated recurrent unit for intrusion detection system. Trans. Emerg. Telecommun. Technol. (2020). https://doi.org/10.1002/ett.3997
21. Paez, G.R.: Assessing predictors of cyberbullying perpetration among adolescents: the influence of individual factors, attachments, and prior victimization. Int. J. Bullying Prev. **2**, 149–159 (2020). https://doi.org/10.1007/s42380-019-00025-7

22. Patchin, J.W., Hinduja, S.: Bullies move beyond the schoolyard: a preliminary look at cyberbullying. Youth Viol. Juv. Just. **4**(2), 148–169 (2006)
23. Pawar, R., Raje, R.R.: Multilingual cyberbullying detection system. In: 2019 IEEE International Conference on Electro Information Technology (EIT), pp. 040–044. IEEE (2019)
24. Rakib, T.B.A., Soon, L.K.: Using the reddit corpus for cyberbully detection. In: Asian Conference on Intelligent Information and Database Systems, pp. 180–189. Springer (2018)
25. Rosa, H., Pereira, N., Ribeiro, R., Ferreira, P.C., Carvalho, J.P., Oliveira, S., Coheur, L., Paulino, P., Simão, A.V., Trancoso, I.: Automatic cyberbullying detection: a systematic review. Comput. Hum. Behav. **93**, 333–345 (2019)
26. Siriaraya, P., Zhang, Y., Wang, Y., Kawai, Y., Mittal, M., Jeszenszky, P., Jatowt, A.: Witnessing crime through tweets: A crime investigation tool based on social media. In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 568–571 (2019)
27. Sugandhi, R., Pande, A., Agrawal, A., Bhagat, H.: Automatic monitoring and prevention of cyberbullying. Int. J. Comput. Appl. **8**, 17–19 (2016)
28. Taddeo, M.: Three ethical challenges of applications of artificial intelligence in cybersecurity. Minds Mach. **29**(2), 187–191 (2019)
29. Tokunaga, R.S.: Following you home from school: a critical review and synthesis of research on cyberbullying victimization. Comput. Hum. Behav. **26**(3), 277–287 (2010)
30. Vallathan, G., John, A., Thirumalai, C., Mohan, S., Srivastava, G., Lin, J.C.W.: Suspicious activity detection using deep learning in secure assisted living iot environments. J. Supercomput. (2020). https://doi.org/10.1007/s11227-020-03387-8
31. Van Bruwaene, D., Huang, Q., Inkpen, D.: A multi-platform dataset for detecting cyberbullying in social media. Lang. Resour. Eval. (2020). https://doi.org/10.1007/s10579-020-09488-3
32. Van der Zwaan, J., Dignum, M., Jonker, C.: Simulating peer support for victims of cyberbullying. In: BNAIC 2010: 22rd Benelux Conference on Artificial Intelligence, Luxembourg, 25–26 October 2010. Citeseer (2010)
33. Wulczyn, E., Thain, N., Dixon, L.: Ex machina: Personal attacks seen at scale. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1391–1399 (2017)
34. Yazdinejad, A., HaddadPajouh, H., Dehghantanha, A., Parizi, R.M., Srivastava, G., Chen, M.Y.: Cryptocurrency malware hunting: A deep recurrent neural network approach. Appl, Soft Comput., 106630 (2020)
35. Zhao, R., Mao, K.: Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. IEEE Trans. Affect. Comput. **8**(3), 328–339 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.