Cybercrime Detection Using Live Sentiment Analysis

Balvinder Singh Gambhir, Jatin Habibkar, Anjesh Sohrot, and Rashmi Dhumal

Abstract Cyberbullying is a continual act that harasses, humiliates, threatens, or hassles people through electronic devices and online social networking Websites. Cyberbullying through the Internet is considered additionally more dangerous than any form of bullying done in the past, because it can probably amplify the humiliation to a vast online audience. Current models have a variety of issues which our proposed system try to address through our proposed work. Many of the other models use small, heterogeneous datasets, without a thorough evaluation of applicability. At the same time, many models yield small datasets that fail to capture the required complex social dynamics and impede direct comparison of progress. Our model uses real-time data from Twitter API which is preprocessed using regex. It is then fed to the LSTM neural network which will filter out negative tweets and also passed through a sentiment analyzer. After passing through these components, if the sentence is found to be negative in nature and contains an abusive word, then we classify it as an act of cyberbullying. In the coming years, we may see people trying to find different ways to harass each other on social media, so by an over proposed method, we have the model to detect sentiment analysis of sentences as well as we have the reference as to why the sentence is passing negative impressions to people using sentiment analyzer.

Keywords Cyberbullying · LSTM neural network · Twitter API · Sentiment analyzer · Stemmer · Machine learning · Web app

B. S. Gambhir (✉) · J. Habibkar · A. Sohrot · R. Dhumal
Ramrao Adik Institute of Technology, Mumbai, India
e-mail: balvindersi2@gmail.com

R. Dhumal
e-mail: rashmi.dhumal@rait.ac.in

D. Gupta et al. (eds.), *Pattern Recognition and Data Analysis with Applications*,
Lecture Notes in Electrical Engineering 888,
https://doi.org/10.1007/978-981-19-1520-8_32

1 Introduction

Cybercrime is a crime which uses digital technologies to do crime. There are many types of cyberbullying happening online today. We are focusing on detecting hate or bullying speech on social media platforms like Twitter. The proposed system focuses on creating a model that will be able to differentiate between hate/bullying text or normal text, and then, if the text is detected as a bullying text, we will show why that text is considered as a bullying text. The proposed system focuses on developing a Web application through which users can track topics in which they want to detect cyberbullying. The system monitors these topics for cyberbullying and then informs the user if a cyberbullying text is found. Then, the user can take the required actions. The proposed system will be used by people who monitor social media Websites like Twitter and Facebook. Through this application, they will be able to filter out bullying texts and will be able to take appropriate actions.

2 Literature Review

This section describes the methodology adopted for the literature review. This paper represents an exploration of the contributions that have already been made in the academic field.

Nurrahmi and Nurjanah discussed the cyberbullying detection for Indonesian tweets to identify cyberbullying text and actors in Twitter [1]. They proposed a system based on texts and credibility analysis of users and notify them about the harm of cyberbullying. They have applied SVM and KNN to learn and detect cyberbullying text on the data collected from Twitter. The data collected from Twitter are unlabeled, so the author designed a Web-based tool to classify it into two classes: cyberbullying and non-cyberbullying. The SVM shows better results than KNN and also categorized users into four categories based on credibility analysis. Foong and Oussalah used sentiment140 training data using Twitter database and proposed a method to improve classification using Naive Bayes [2]. Their accuracy is around 58.40%. Their main focus was on doing sentiment analysis on tweets related to movies. They provided negativity, positivity, and objectivity of a tweet using Naive Bayes and SVM. They implemented their system using NLTK and Python Twitter API. Sarlan et al. focused on discovering public opinion by performing sentiment analysis on real-time Twitter data [3]. They used Hadoop, Hive warehouse, and Apache Flume for storing and analyzing tweets data. After doing sentiment analysis on the tweets, the tweets were classified as positive, negative, and neutral, and these were used for decision making. The analyzed tweets were then plotted on histogram and bar chart. Bahrainian and Dengel used a deep learning model for detecting cyberbullying on various platforms like Twitter and Wikipedia [4]. They used many machine learning techniques and found out that CNN and BLSTM were best for detecting cyberbullying. One limitation of their model was that it takes too much time to analyze one

tweet so it could not be used on a real-time system. Yazğılı and Baykara mention how cybercrime is affecting physical/mental health of a person and which lead to suicidal tendencies [5]. They also tried to create a model using SVM, KNN to detect cybercrime. Hang and Dahlan created a dataset of words that are used in cyberbullying [6]. Their strategy is made up of several steps, namely understanding of cyberbullying exclusion principles, word list selection, recognition of a keyword, classes and subclasses identification of ontology and lexicon and lastly, cyberbullying detection. Bertot et al. have written about the effects of cyberbullying on various topics like political, technical, and crowdsourcing and how to overcome such a problem [7]. They also explain how privacy and social data of users are important in detecting cyberbullying. Banerjee et al. used word vectors with CNN to detect cyberbullying tweets [8]. Their accuracy is 81.6% on Twitter tweets. They also suggested making parents track their kids' social media activity. They focused on finding cyberbullying tweets in Arabic language [9]. They used a dataset of abused words that were used to detect cyberbullying tweets. Their system allowed users to add their own abuse words that should be considered while marking a tweet as cyberbullying or not.

3 Limitation of Existing System

- The proposed system needs human labeled data to train a model to detect various types of harassment.
- Less advanced algorithms like SVM, decision tree, etc., are not providing good results, and advanced algorithms take more time to predict the results.
- As we have to keep our model up to date with the changing the way people harass each other, so we have to keep training the model on new labeled datasets.
- Not able to detect spammers from different accounts harassing the same person.
- Unsupervised learning can be useful but how can we be so sure about the result as it may lead to biased prediction.

4 Methodology

4.1 Proposed Work

This proposal is aimed at development of an application system through which the user enters keywords which is then passed to Twitter API which will then fetch all tweets related to those keywords, and these tweets are then passed to our model Fig. 1. The main objective of the project is the development of an application system through which the users can monitor cyberbullying on a particular topic. Real-time analysis of tweets is done, and then, the sentiment of each tweet is calculated. The sentiment can be positive, neutral, or negative. The main functions include

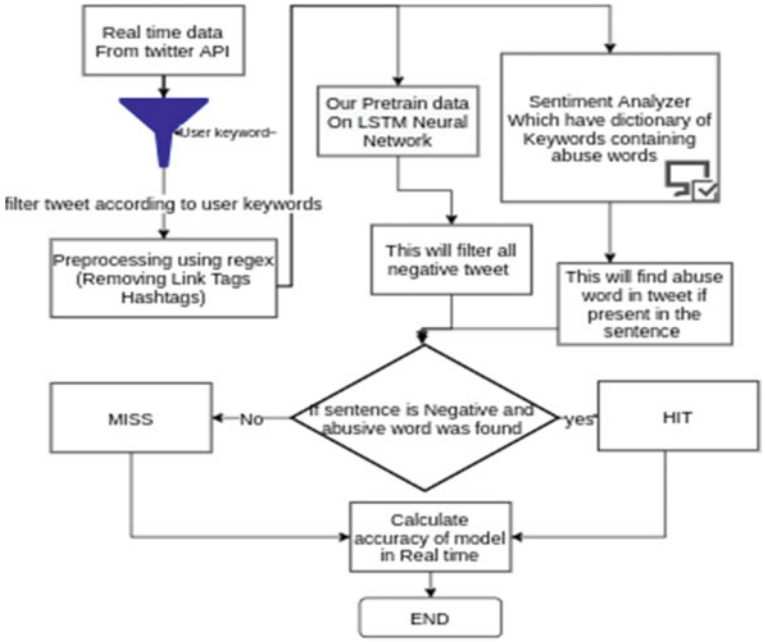

Fig. 1 Proposed system workflow

- Detecting negative sentence tweets
- Detecting harassing tweets
- Give feedback on various tweets to improve the ML model.
- Real-time model accuracy on testing data.
- Creating word-cloud in our Web application to understand filtered tweets in real time.
- Sentiment analysis as well as extracting harassing words from tweet help us to understand people's thoughts in real time.

Proposed System Workflow Explanation

As shown in Fig. 1, the user will supply keywords that it wants to filter the tweets on. These topics will be sent to our backend which will then fetch the related tweets from Twitter. The fetched tweets are preprocessed and then is passed to our classifier which will classify them as positive or negative. After the classifier gets the result, we need to analyze if the tweet is harassing or not, so we used sentiment analyzer to predict the accuracy of the model in real time. This analyzer has dictionary which contains harassing words. This software will detect/underline words in sentence if the tweet contains any kind of harassing word. As analyzer is our reference to calculate accuracy of model in real time so we come to a conditional statement where we check if our model predicts a negative statement as well as it contains any abuse word, then we will consider it as HIT. If the text is negative, and it does not contain any abusive

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None,300,300)	87171600
dropout_4 (Dropout)	(None,300,300)	0
lstm_4 (LSTM)	(None,100)	160400
dense_4 (Dense)	(None,1)	101

Total params: 87,332,101 Trainable params: 160,501 Non-trainable params: 87,171,600

Fig. 2 Neural network architecture

words, it will be a MISS. After this process, we will calculate to total number of HIT and MISS and calculate the accuracy of model in real time using this formula, e.g., 1.

$$\text{HIT}/(\text{HIT} + \text{MISS}) \tag{1}$$

4.2 Classifier Implementation

Preprocessing the data: As data are fetched with the help of Twitter API, it contains hyper link, numbers, stop words. So we remove all these useless features.

Word to vector: As computers do not understand words, so we converted words to vectors using NLTK software. It contain size of vocab dictionary ie30520 W2VSIZE = 300 weights = initially embedding matrix by random float number SEQUENCELENGTH = 300 same as W2VSIZE.

Create tokenizer: After converting all the words to vectors, we tokenize whole words so that all words are understood by a computer.

Creating neural network: After creating tokenized words, the word is passed to the (LSTM) neural network. Fig. 2 refers to model architecture.

4.3 LSTM Networks

LSTM Networks: Long short-term memory networks (LSTM) are a special kind of RNN that are capable of learning from long-term dependencies. They work well with many types of problems and are widely used. LSTMs are designed to avoid the

long-term dependency problem. They are able to remember information for a long duration of time. The key to LSTMs is the cell state. LSTM has the ability to remove or add information to the cell state, through structured gates. Gates are made through sigmoid neural network and pointwise multiplication. The sigmoid layer has output between zero and one which can be used to know how much information should be transferred. A value of 0 means that no information should be passed through, while value of 1 means every information should be passed through. We are passing preprocessed text to our LSTM model which then classifies the text as negative or positive.

- Sentence is preprocessed and pass to lemmatization
- Lemmatizing convert all words to lemma words
- Convert all unique words to tokens
- Convert words to vector–score every word and averaging the score.

Sentiment analyzer is a rule-based model for sentiment analysis.

There are over 9000 features that are rated from extremely negative to extremely positive. −4 is used for extremely negative, and 4 is used for extremely positive. 0 is considered as neutral. Our model kept features that had a mean rating not equal to 0, and whose standard deviation was less than 2.5. So after removing these features, 7500 features were left. For example, okay has a positive score 0.9, good has 1.9, and great has 3.1, whereas the score for the word horrible is −2.5. We are passing the preprocessed text to the sentiment analyzer for getting abuse words. The sentiment analyzer score is calculated by adding all of the scores for each term in the lexicon and then normalizing the result between −1 and 1.

5 Data Description and Data Cleaning

We have scraped 1.6 million tweets by using Twitter. The tweets have been classified as negative and positive. These tweets are then used to detect sentiment of new tweets. We have used the following fields:

1. ids: tweet ids that are randomly created
2. Target: polarity of the tweet. It has the following values Neg, Neu, and positive.
3. Users: username of the user (balvinderzuser).
4. Dates: date of the tweet (Sat May 21 10:13:44 UTC 2022).
5. Texts: content of tweets like hello there.

6 Experimental Design

As shown in Fig. 3, the proposed system has a Web application (developed in vue) and a backend (developed in flask). Initially, the user will select three 3 topics that he wants to detect cybercrime on. These topics will be sent to the backend which will

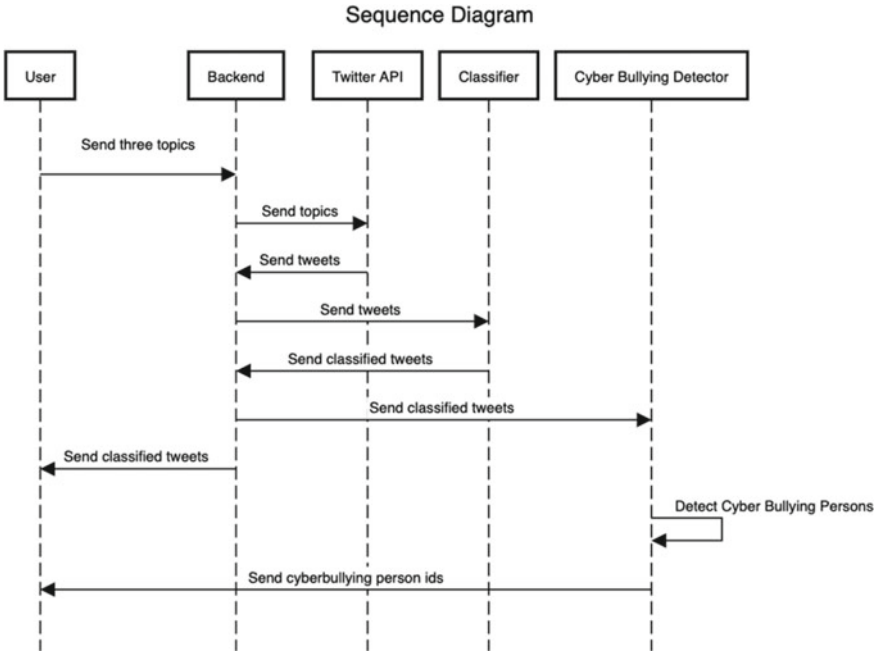

Fig. 3 Sequence diagram of application

then fetch the related tweets from Twitter. The fetched tweets will then be passed to our classifier which will classify them as positive or negative. After the classifier gets the result, we need to analyze if the tweet is harassing or not, so we use sentiment analyzer to predict if the tweet contains harassing words or not. This analyzer has a dictionary which contains harassing words. This software will detect/underline words in sentences if the tweet contains any kind of harassing word.

In Fig. 4, we have considered an example from the live tweets. "@RahulGandhi is a shameful idiot @NationalistCol". The tweet is first preprocessed (RahulGandhi is a shameful idiot nationalistcol). After preprocessing, the tweet is tokenized. Then, it is converted into a word to vector model which is then passed to our trained model. The trained model classifies the tweet as positive or negative. The tweet is also passed through NLTK sentiment analyzer. In this case, our model classifies the tweet as negative, and the NLTK sentiment analyzer also found harassing words ("shameful", "idiot"). So it is a hit.

6.1 Result and Analysis

The proposed system focuses on designing a Web-based app that will monitor the Twitter Website for detecting cyberbullying tweets and also justifying as in why the

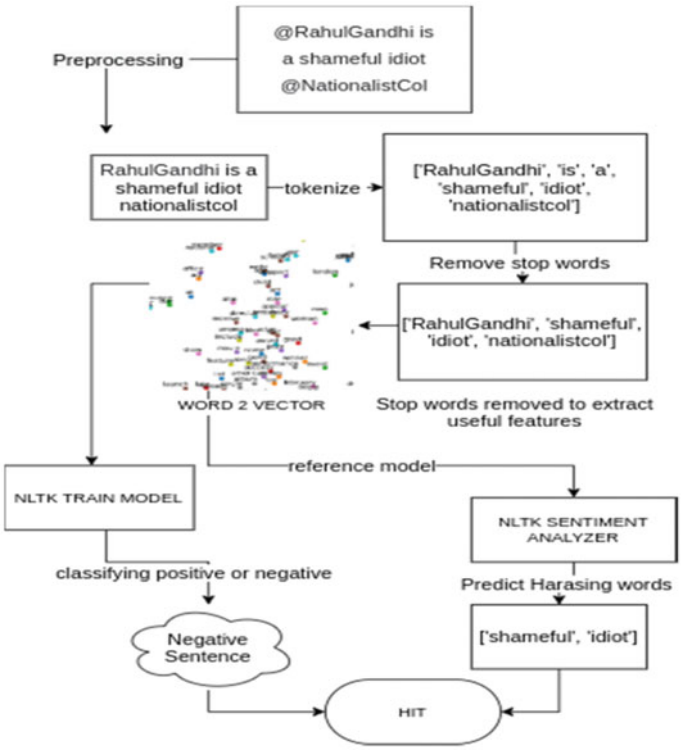

Fig. 4 Text example

tweet is harassing. The model is able to filter out negative tweets in real time using the LSTM model with an accuracy of around 70–80%. We are using sentiment analyzer to detect harassing words that will act as a reference to get the accuracy of our model in real time (Fig. 5).

Reference to Fig. 6, as we are working on real-time data, we are not considering if our model predicted positive and it contain a harassing word as most of the people on Twitter is talking about same topic, and this may lead to biased dataset containing only positive sentences. Reference to Fig. 5, negative sentences which do not contain harassing words according to sentiment analyzer.

Example 1: "Why oh why did I read the comments. The level of idiocy is just … I cannot even".

So our model may have identified idiocy as harassing, but the sentiment analyzer may not have identified it as harassing because of two reason: (1) We have currently set sentiment analyzer to greater than 0.0, and we can increase it to get more predictions which contain less harassing words. (2) It is dictionary may not contain that word.

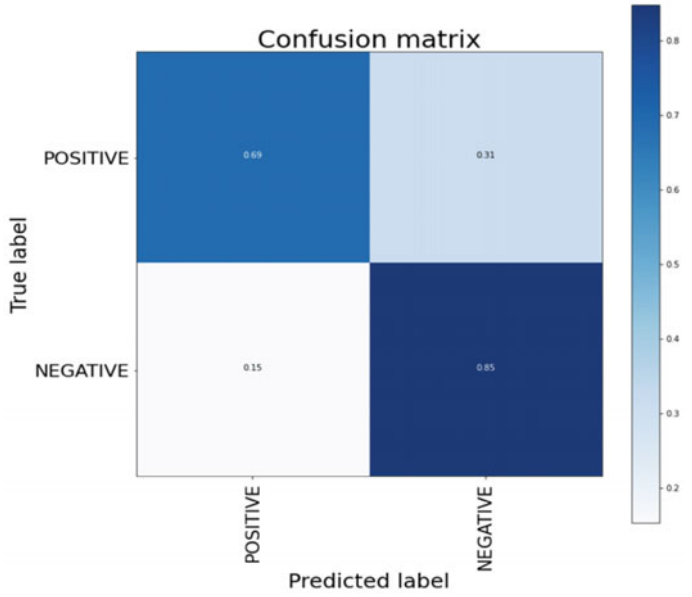

Fig. 5 Confusion matrix of testing data (0.69 | 0.31 | 0.15 | 0.85)

Training Phase
Training data : 1280000 Sentences
Testing data : 320000 Sentences
Total words TOKENS 290572

ACCURACY On training phase: 81.3 %

Testing Phase on real Data
Topics considered ["disha ravi","farmer protest","trump"]

```
count of our model
negative  200
actual count of harasing words in sentences
sentence count (TP):  163

In [8]: accuracy=len(sentNeg)/len(countneg)
        print("Accuracy is : ",accuracy*100,"%")

        Accuracy is :  81.5 %
```

Fig. 6 Accuracy on 200 new real-time data

	precision	recall	f1-score
NEGATIVE	0.82	0.69	0.75
POSITIVE	0.73	0.85	0.79
accuracy			0.77
macro avg	0.77	0.77	0.77
weighted avg	0.77	0.77	0.77

Fig.7 F-score of training and testing data

6.2 Precision and Recall

Reference to Fig. 7, calculating the f-score of our deep learning model which was trained on 12.80 k training and 320 k testing dataset has and (if you ask what proportion of positive identifications was actually correct?) precision of 82% on positive dataset and 73% on negative dataset. (If you ask what proportion of actual positives was identified correctly?) Recall, also called as sensitivity, our model has classified the tweets with a recall of 69% on negative tweets and 85% on positive tweets. F1 score passes on the harmony between the precision and the recall the formula to calculate f1 score is 2*((precision*recall)/(precision + recall)).

7 Conclusion and Future Work

By our proposed work, we have detected harassing tweets on real-time system also shown real-time accuracy of our system. We have also justified, if the tweet is negative or not using custom sentiment analyzer. As in real life, we cannot just rely on neural networks or we cannot just rely only on dictionaries, so we have to create such methodology and techniques to solve this problem with minimum resources. In future work, we can also detect users who are harassing other users and give those users details to concerned authorities.

References

1. Nurrahmi, H., Nurjanah, D.: Indonesian twitter cyberbullying detection using text classification and user credibility. In: 2018 International Conference on Information and Communications Technology (ICOIACT), pp. 543–548. IEEE, Indonesia (2018)

2. Foong, Y.J., Oussalah, M.: Cyberbullying system detection and analysis. In: European Intelligence and Security Informatics Conference (EISIC), pp. 40–46. IEEE, Athens (2017). https://doi.org/10.1109/EISIC.2017.43
3. Sarlan, A., Nadam, C., Basri, S.: Twitter sentiment analysis. In: Proceedings of the 6th International Conference on Information Technology and Multimedia, pp. 212–216. Putrajaya (2014). https://doi.org/10.1109/ICIMU.2014.7066632
4. Bahrainian, S., Dengel, A.: Sentiment analysis and summarization of twitter data. In: 16th International Conference on Computational Science and Engineering, pp. 227–234. IEEE, Sydney (2013). https://doi.org/10.1109/CSE.2013.44
5. Yazğılı, E., Baykara, M.: Cyberbullying and detection methods. In: 1st International Informatics and Software Engineering Conference (UBMYK), pp. 1–5. IEEE, Turkey (2019). https://doi.org/10.1109/UBMYK48245.2019.8965514
6. Hang, O.C., Dahlan, H.M.: Cyberbullying lexicon for social media. In: 6th International Conference on Research and Innovation in Information Systems (ICRIIS), pp. 1–6. IEEE, Malaysia (2019). https://doi.org/10.1109/ICRIIS48246.2019.9073679
7. Bertot, J.C., Jaeger, P.Y., Hansen, D.: The impact of polices on government social media usage: issues, challenges, and recommendations. Gov. Inf. Q. **29**(1), 30–40 (2012)
8. Banerjee, V., Telavane, J., Gaikwad, P., Vartak, P.: Detection of cyberbullying using deep neural network. In: 5th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 604–607. IEEE, India (2019). https://doi.org/10.1109/ICACCS.2019.8728378
9. Mouheb, D., Abushamleh, M.H., Abushamleh, M.H., Aghbari, Z.A., Kamel, I.: Real-time detection of cyberbullying in Arabic twitter streams. In: 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS). IEEE, Spain (2019). https://doi.org/10.1109/NTMS.2019.8763808