I.J. Modern Education and Computer Science, **2025, 2, 123-134**
Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.02.06

Toxicity Detection Using TextBlob Sentiment Analysis for Location-Centered Tweets

Varun Mishra*
Department of Computer Engineering & Applications, GLA University, Mathura, Uttar Pradesh, 281406, India
E-mail: varun.mishra97@gmail.com
ORCID ID: https://orcid.org/0000-0001-7048-5611
*Corresponding Author

Tejaswita Garg
School of Studies in Computer Science & Applications, Jiwaji University Gwalior, Madhya Pradesh, 474011, India
E-mail: tejaswitagarg@gmail.com
ORCID ID: https://orcid.org/0009-0005-5097-4326

Received: 16 July, 2024; Revised: 03 September, 2024; Accepted: 17 October, 2024; Published: 08 April, 2025

Abstract: The toxic comment detection over the internet through social networking posts found hatred comments and apply certain limitations to stop the negative impact of that information in our society. In order to perform sentiment analysis, NLP text classification approach is very effective. In this paper, we design a specific algorithm using Convolution Neural Network (CNN) approach and perform TextBlob sentiment analysis to evaluate the polarity and subjectivity analysis of posted tweets or comments. This paper can also filter the tweets collected over different locations formed Twitter dataset and then model is evaluated in terms of accuracy, precision, recall and f1-score as calculated results of 0.984, 0.887, 0.905 and 0.895 respectively for the analysis of toxic/non-toxic comment identification. Hence, our algorithm utilized NLTK and TextBlob libraries and suggests that the analyzed post can be recommended to the others or not.

Index Terms: Toxic, TextBlob, Sentiment Analysis, Convolutional Neural Network, Machine Learning

1. Introduction

Web technology has made name for itself on a global scale by enabling individuals to get engaged so that people can express their opinions online on any topic they want others to know about. On the internet, we can find people with different political, religious, and ethnic back grounds. Social media gives people from different countries an online forum to communicate their opinions, cultures, and civilizations. Online interaction can lead to a variety of effects on people's lives, both positive & negative. The positive outcomes are that individuals can speak with one another around the world despite being thousands of meters apart, while negative outcomes are poisonous on global scale. Comment section is degenerating into abuse, insults and threats [1].

In order to expose antisocial activity on social networks, it is crucial to identify suspicious online reviews or tweets [2]. Critics or other influencers can hurt online users and even threaten democracy in various nations with their aggressive 1ehavior. Trolls are active in disseminating false hoods at important events like elections or referendums, which highlights the serious problem [3]. The goal is to provide an efficient model for troll identification in online environments that combine such efforts with numerous web-based system that are now making significant efforts to drive trolls out of business [4].

The importance of the work also lies in the identification of various sentimental analysis and learning techniques in order to choose the useful technique to develop detection models. As a result, the main purpose of this work is to compare and propose various methods for creating an algorithm for toxic detection in social networks [5]. This comparison should provide a response to the question of whether sentiment analysis or machine learning techniques should be utilized to create this detection model, as well as which techniques should be employed specifically for textual data that is derived from the structure of web debates.

For the purpose of identifying and preventing suspicious reviewers, web portals are also working to develop automated solutions. Trolls frequently conceal harmful messages, making it harder to spot them. Toxic post masking is a strategy used to write toxic communications in a way that makes it difficult for automated algorithms to identify them, for instance, by not using harsh or nasty terms in posts that are offensive in context. Masking can be done by

purposefully using grammatical faults or by swapping letters inside specific words. This also necessitates the development of incredibly extensive, emotive, and meticulously crafted lexicons [6].

The use of social media platforms as a communication tool has drastically grown in popularity especially in youngsters but it has pros and cons. Due to the major interference of those platforms in our daily life we are strongly influenced the kind of comments, tweets, images which we shared on those platforms and most of the times those kind of things spreads negativity in our mind as for the last ten years. So, to calculate the kind of toxicity due to those comments we have chosen the location-centred tweet analysis over different-different locations.

Major contributions of the proposed work are summarized as follows:

- Firstly, we collect all the tweets by using the Twitter source dataset from different-different locations.
- After that, select all the negative comments into specific category using CNN as machine learning approach.
- To perform sentiment analysis by using TextBlob library of python programming language and finds its polarity and subjectivity to predict analysis scores.
- The proposed model gives a major contribution to predict toxicity of the analyzed tweets by using the proposed algorithm and flowchart for the analysis process

Although there is presently no clear definition for this term toxic, it is frequently used to describe the words that contain offensive language, threats, highly unfavorable or bad attitude toward a user or a condition, taunts, or other rude remarks that could embarrass or offend a reader. Such communications are more frequently referred to as hatespeech if they are racist, sexist, religious, or of another kind who foster a bad behavior against a common social group.

Abusive language is cruel language that uses vulgarity, hateful speech, and other offensive words [7]. However, a lot of scholars called the harsh speech the abuse speech. Fig. 1 describes the relationship between various types of toxic or HateSpeech concepts shown below.

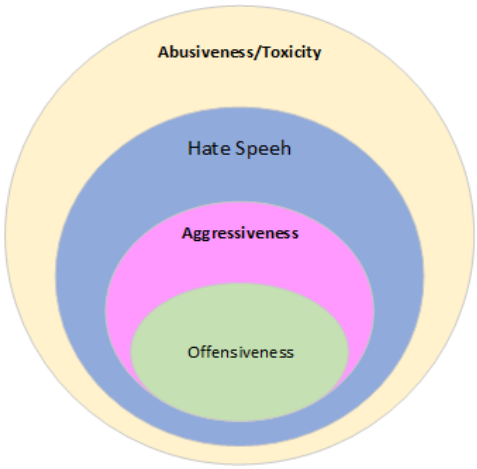

Fig. 1. Relationship between several topics in Hate Speech.

This paper is organized as follows: The introduction for the toxicity and toxic comment detection techniques are discussed briefly in Section 1. In Section 2, methods adopted by researchers through literature review are then discussed. Section 3 discusses the methodology adopted in this study are explained. The proposed framework includes data preparation; pre-processing methods and chosen learning classifiers utilized to perform sentiment analysis over Twitter dataset for different locations along with proposed algorithm for evaluated findings are described in Section 4 and discussions are covered in Section 5. Then after, Section 6 briefs the conclusion of this study.

2. Background Study

This section discusses the methods used by the researchers and gives comparison over the suggested parameters highlights in this study. In the discipline of NLP, text representation and classification are extremely essential. The goal of toxic comment classification, an NLP problem related to sentiment analysis is to further classify unfavorable comments into hazardous and non-toxic categories (reviews that merely show a bad attitude). Deep learning and machine learning models have seen increased adoption in recent years for the classification of hate speech or toxic comments.

Prior to gathering relevant review papers (Google scholar5 and ACM6), they carefully chose the highly occurred keywords to search up relevant results in the database. Since the idea of hate speech is new and has lately gained popularity, necessary to look at other pertinent terms related to specific sorts of hate speech (For instance, homophobia, sexism, racism, and cyberbullying).

Some HateSpeech keywords are selected for hate identification along with deep learning and language keywords as shown in Table 1.

Table 1. Search query includes HateSpeech, deep learning and language keywords.

HateSpeech Keywords	Deep Learning Keywords	Language keywords
HateSpeech Detection Sexism Detection Cyberbullying Detection Abusive Language Detection	Deep Learning HateSpeech CNN HateSpeech Deep Learning or CNN Cyberbullying RNN HateSpeech	Chinese, Arabic, Bengali, Russian, Hindi, Portuguese, Urdu, Indonesian, German, French, Japanese, Marathi, Tamil, Spanish, Korean and more.

Here, various approaches are based on the classification of hazardous comments. It implies that even while toxicity detection has significantly improved due to deep learning methodologies CNNs, RNNs, ANNs, and LSTMs, there is still chance for high accuracy in the detection modules' accuracy. This table shows how the performance of the method has been enhanced by the use of word embeddings. Various authors have not assessed the ROC-AUC score to see whether their method is usable. The models could not accurately distinguish between the classes, according to the methods that are currently in use as measured by the ROC-AUC score.

The author of this study suggested three methods for enhancing data: adding unique phrases, using a random mask, and substituting synonyms to address the issue of class disparity. Additionally, in order to identify toxicity in user-generated content, the authors developed an ensemble modeling approach with CNN, Bidirectional LSTM, and GRU [8].

In this study [9], authors investigated different deep learning models for the classification of toxic comments, both with and without pre-trained word embeddings, including CNN and LSTM. According to this study, CNN performs better than the pre-trained word embedding with GloVe.

In this study [10], authors examined the effectiveness of CNN model using sentiment analysis methods. According to their research, CNN improves the work of identification of toxic comments.

In this study [11], authors have used the classification models logit, I Bayes, Support Vector Machines, FastText-BiLSTM, and Extreme gradient boosting method as XGBoost to show how raw comments may be transformed. The reviewer concludes that without any change, this model produce a quite good classification score.

This study [12] has provided a multi-pronged ensemble methodology to address the challenges of hazardous comment classification. This model works the best in classes with low instance counts and substantial data variance.

The authors of this study [13] compared the deep neural network to look for negative sentiment data where proposed model excels in the job of overlaying multi-label toxic emotion detection.

For stance identification, an author has suggested a two-channel GRU convolutional unit. According to the findings, the SVM (support vector machine) approach has improved by 15:6%. In conclusion, the multichannel deep learning models were employed by the researchers at the time to solve the single label and multiclass problems. In order to detect multiple labels of harmful categories, we suggest a multichannel convolutional bidirectional gated recurrent unit [14].

This research has classified toxicity in the comments using LR, CNN, LSTM, and CNN-LSTM. They claimed that CNN-LSTM provided the best accuracy, with a score of 98.20% [15].

In this study [16], authors have utilized three classification algorithms, including LR, RF, and Gradient Boosting while taking into account the TF-IDF features. The best accuracy of 97.20% was provided by logistic regression out of the three techniques used.

This study has used base classifier algorithms with character n-grams and words or phrases unigrams that computed their efficiency over three different dataset collected from different social media platforms. LSTM, Bi-LSTM, and CNN have also been used, along with GloVe, SSWE, and random word embedding. Transfer learning has also been used to determine whether or not a model developed for one dataset may be used for another [17].

In this study [18], researchers have used the J48, Jrip and SVM with various kernels along with the KNN, NB, RF, and CNN with one and two hidden layers have all been used to detect cyber-bullying. They claimed that CNN outperformed all other algorithms when it had two hidden layers.

This study has employed various classification techniques like sentiment analysis and regression to detect bogus news [19].

To identify fake news, authors said that the primary research revealed that proper nouns are used more frequently in fake news than other nouns. Therefore, we may access the ratio of proper nouns to other nouns and assess the model in accordance with that in order to determine the credibility of one post. However, issue with false news detection algorithms will always exist because it's challenging to attain perfect certainly without fact verification [20].

This study discusses the different issues with online hate detection and presented false positives (FP) and false negatives (FN). FP happen when a non-toxic comment is recognized by the model having hostile content, and FN happen when the model is unable to recognize hateful messages and treats them as non-threatening. Some problems can also arise from the subjective nature of the offensive comments in an article. In addition to these, polysemy, which is when a single term in an article, has multiple meaning, can also cause confusion and prevent the model from performing to expectations [21].

In order to identify toxic comment detection, the current study focus on machine learning CNN model that is evaluated over tweets gathered from several locations on Twitter.

3. Materials and Methods

In this proposed work, we have implemented a deep learning algorithm that performs sentiment analysis using CNN. The work is defined to determine the toxicity of content posted on online social networking sites chosen here as Twitter. Toxic content classification is carried out using various sentiment analysis approaches and goes through a sequence of operations as text pre-processing, feature extraction and selection and then toxic comment categorization is done over the selected posted comment from dataset and defined its category that can be considered as positive, negative or neutral.

3.1. Data Preparation

The first stage in the detection of toxic comments is data augmentation. Online networks are frequently used to collect datasets (Facebook, YouTube, Twitter, etc.). Preprocessing step is carried out based on the structure and quality of the dataset. Typically, this entails filtering and normalization of textual inputs, which include, among other things, tokenization, stop word elimination, misspelled correction, remove noise, stemming and lemmatization. We'll also see that the dataset might be given to us right away, eliminating the need for collecting. The training and testing portions of the dataset should be separated during data preparation for the next machine learning stage [22].

We have collected a dataset from Twitter source that contained around 16634 comments for testing and training of the data. In this work, we have introduced a feature based on location selection posts and split our data into 80:20 training and testing sets. Hence, this model is trained on balanced datasets to decrease bias and increase accuracy. Among seven different cities, twitter posts are collected and prepared a dataset for toxicity identification. According to these assessments, there exists tweets word analysis based on the seven cities location including Aligarh, Amritsar, Ghaziabad, Gurugram, Moradabad, Noida and Panipat about the content posted on the Twitter networking site. A screenshot of the dataset collected from Twitter is shown in Table 2 below.

Table 2. Dataset generated using Twitter posts.

Unnamed:0	Date	content
0	2022-11-01 15:46:59 +00:00	BIG: 3 Hybrid Terrorist arrested with 10kg IED..
1	2022-11-01 15:41:16 +00:00	Requested urgent intervention of Union Ministe…
2	2022-11-01 15:27:01 +00:00	@SpareSidekick "I am sorry but I am not in moo..
3	2022-11-01 15:25:08 +00:00	@DrTejaswini07 Attack the views. Not the person.
4	2022-11-01 15:20:43 +00:00	Scientist have their eyes on several Deltacro….
5	2022-11-01 15:17:40 +00:00	@BBC Breaking Police explore motives of bomber..
6	2022-11-01 15:16:29 +00:00	@BBC Breaking Police urged to treat Dover attack..
7	2022-11-01 14:53:25 +00:00	@AamAadmiParty Ankit Tyagi, the reporter in vi..
…	…	….

3.2. Text Preprocessing

In this text pre-processing phase, data pre processing is done using various natural language processing tasks. In the proposed algorithm, function pre-process text takes an input posts and perform clean operation that removes URL's, stop words, punctuations, emojis and converted text to lower case characters. After that natural language pre-processing operations such as stemming and lemmatization is carried out over posted content such as posts mentioned a statement including 'likes', 'likely' and 'liked' that all can be converted to 'like' after stemming and get it stored.

For lemmatization is to be performed, all such words including 'better', 'nice' and 'fine' will all be converted to 'good' and gets stored. This shows that particular word is split into tokens through tokenization, words are paired down to their stems through stemming and lexical knowledge bases are used in lemmatization to determine the proper word bases. With these operations, dataset tweets stored in column 'content' to get cleaned text and stored in new column of the dataset named as 'clean' as shown in Table 3.

Table 3. Dataset posts are cleaned and saved in new column named 'clean'.

Unnamed:0	Date	content	clean
0	2022-11-01 15:46:59 +00:00	BIG: 3 Hybrid Terrorist arrested with 10kg IED..	big 3 hybrid terrorist arrested 10kg I plan attack
1	2022-11-01 15:41:16 +00:00	Requested urgent intervention of Union Ministe…	request urgent intervention of union minister
2	2022-11-01 15:27:01 +00:00	@SpareSidekick "I am sorry but I am not in moo..	Sparesidekick sorry not mood carol food attack
3	2022-11-01 15:25:08 +00:00	@DrTejaswini07 Attack the views. Not the person.	Drtejaswini07 attack view not person
4	2022-11-01 15:20:43 +00:00	Scientist have their eyes on several Deltacro….	Scientist eye several deltacron

3.3. Proposed Framework

The proposed architecture of our modeling approach is then discussed in Fig. 2. It performs TextBlob analysis over twitter source data that is being cleaned with our proposed algorithm mentioned in Figure 3.10 and then passed on to perform sentiment analysis with generated scores of polarity and subjectivity. Convolution neural network machine learning approach is then predicted the sentiment toxic classification scores using the achieved input from obtained TextBlob analysis scores as shown in Fig. 3.

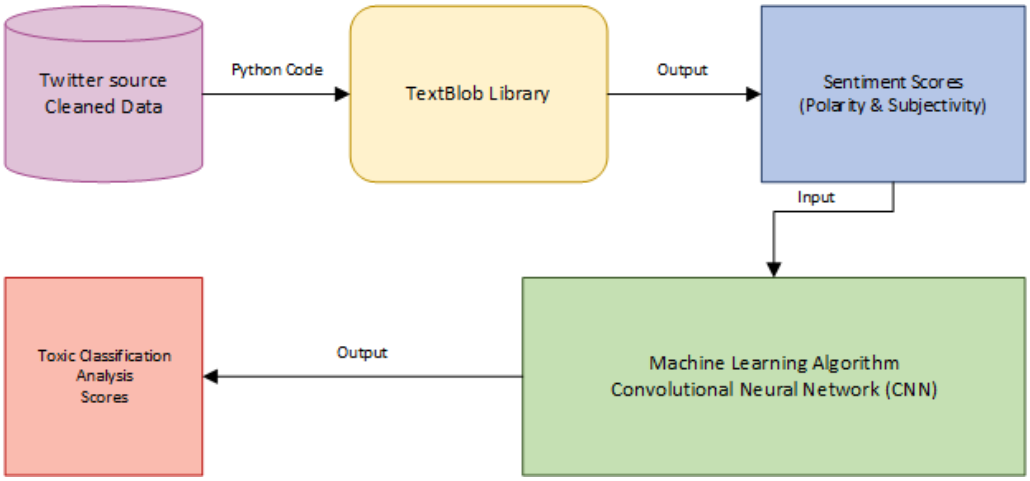

Fig. 2. Proposed architecture for toxic comment classification using CNN.

The detailed Flowchart of our work is proposed in Fig. 3 shown below.

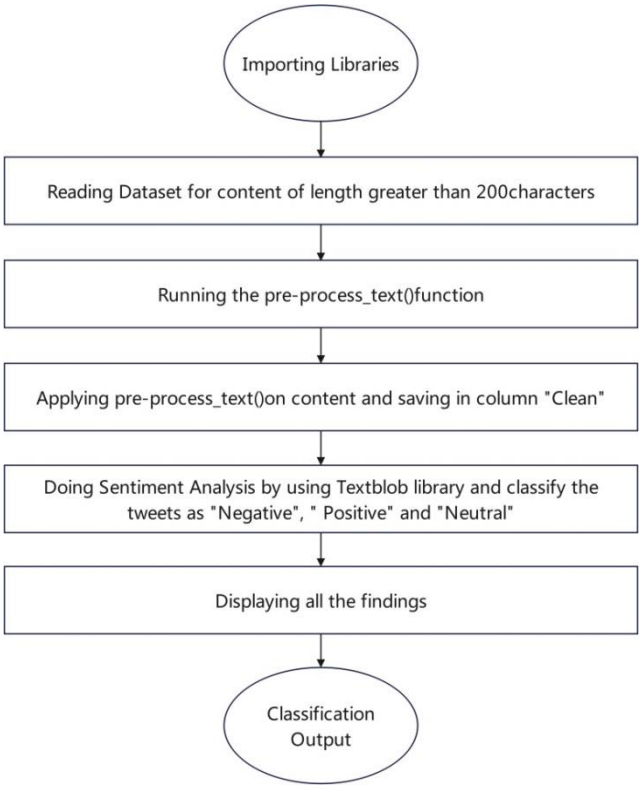

Fig. 3. Proposed flowchart for Toxicity detection.

3.4. Convolutional Neural Network

The proposed model is evaluated using CNN that can directly extract textual features from the input data. To perform predictive tasks, CNN consists of three primary layers: a convolution layer, a pooling layer, and a dense layer. One dimensional convolution layer over the concatenated embeddings using count vectorizer for each input tweet texts makes up the CNN model that we implemented. With a kernel size of three, the convolutional layer contains a set number of filters. The output layer is the following layer; which is completely connected layer with fixed number units.

To determine the ideal settings that would provide us with the highest accuracy, we will go into more detail about our hyperparameter tweaking [23]. The architecture of CNN model is shown in Fig. 4 below.

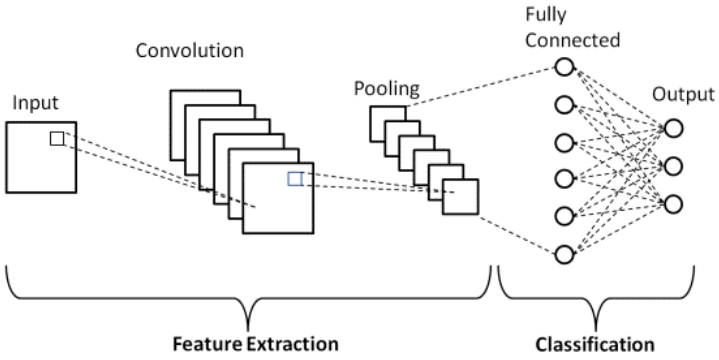

Fig. 4. The CNN architecture for the proposed model.

Assume that our convolutional layer comes after some PxP square neuron layer. The output of our convolutional layer will have a size of $(P-q+1)*(P-q+1)$ if we apply a $q*q$ filter ω. We must add up the contributions from the cells in the preceding layer, weighed by the filter components in order to calculate the pre-nonlinearity intake to some unit x_{mn}^{r} in our layer shown in equation 1:

$$x_{mn}^{r} = \sum_{a=0}^{q-1}\sum_{b=0}^{q-1}\omega_{ab}\, y_{(m+a)(n+b)^{*}}^{r-1} \tag{1}$$

Just a convolution, this can be expressed as:

$$conv2\ (x, \omega, 'valid')$$

Next, the nonlinearity of the convolutional layer is applied in equation 2:

$$y_{mn}^{r} = \sigma(\, x_{mn}^{r}) \tag{2}$$

The max-pooling layers don't learn on their own and are rather basic. Take a given $z*z$ region and output its greatest value, which is a single value. For example, if their input layer is a $P*P$ layer, they will output a $\frac{P}{z}*\frac{P}{z}$ layer since the max function reduces each $z*z$ block to a single value [24].

3.5. Text Classification

Deep Learning has effectively resolved various classification challenges, demonstrating its dependability in the field of NLP. In the language categorization and modeling, CNN is able to efficiently conveying meaningful representations of sentences. Since one-dimensional CNN seems to be very capable of reducing the features of defined set portions over the entire collected data and performs well for NLP issues, it is used in this study to design the classification of articles into a variety of themes in the dataset. The CNN architecture includes an input layer, convolutional layers, maximum pooling, and fully linked layers.

CNNs' primary advantages are its ability to offer an effective dense network that effectively handles the tasks like identification and prediction. While BERT model can undoubtedly be used for high accuracy but CNN models are used with huge datasets and to shorten the training period if the model needs to be retrained on a daily basis.

The proposed algorithm for performing text classification to get clean text analysis with our approach is defined in **Algorithm 1** shown below. First, data is gathered from Github source and split into training and testing sets in the ratio of 80:20. Then, preprocess_text function outputs cleaned text that is being performed by sentence lowercase, removing URL, punctuations and stop words and it gets stored in an array. After stemming and lemmatization process, cleaned text fed into TextBlob library to calculate its polarity and subjectivity scores. Now, the training data is fed to CNN model to evaluate dataset to obtain its metric calculation scores that represents in terms of Accuracy, Precision, Recall and F1-Scores.

Algorithm 1

Input: Tweets data source collected over various locations on Twitter.
Output: Accuracy, Precision, Recall and F1-Score of Classified Posts.
Step1. Collection of tweets from Github dataset source. Split data into training and testing sets.
Step2. Function Preprocess_text gets a tweet as input which we need to be cleaned before applying the sentiment analysis model.
Step3. Converting the tweet to lowercase.
Step4. Removing all the URL from the tweet.
Step5. Removing all the punctuation and stop word from the tweet.
Step6. After removing all the stop word we are storing all words in an array called fil_words.
Step7. Performing stemming on the fil_words array which convert each word as an example – *likes likely liked* will all be converted to *like*.
Step8. Storing words in stem_words after Stemming.
Step9. Now performing lemmatization on stem_words doing this we convert all the similar word to a common word example – *better nice fine* will all be converted to *good.*
Step10. Returning the cleaned tweet without any URL, punctuation and useless word which will help the analysis.
Step11: Fed the cleaned text into TextBlob library to calculate its polarity and subjectivity scores.
Step12: Train the data using CNN model and then using test data evaluate the proposed model.
Step13: Generate the toxic classification scores for analysis.
Step14: Evaluate the metric calculation in terms of accuracy, precision, recall and F1-score as desired output.
Step15: END.

3.6. TextBlob Sentiment Analysis

A Python library for NLP tasks is called TextBlob. Natural Language Toolkit (NLTK) was a tool that TextBlob actively employed to complete its tasks. The NLTK library gives users easy access to numerous lexical resources while enabling them to do classification, segmentation, and a range of other tasks. TextBlob is a straightforward library that provides intricate textual analysis and processing [25].

The polarity and subjectivity of a statement are returned by TextBlob.

- The polarity scale is [-1,1], where -1 standing for a bad emotion and 1 for favorable one. Negative or bad language shifts the polarity. Semantic labels in TextBlob facilitate detailed analysis. Emoticons, exclamation points, etc. are examples.
- The range of subjectivity is [0, 1]. Subjectivity measures how much factual information and subjective opinion are present in the text. The text contains opinion instead of actual facts due to text's heightened subjectivity.
- One other setting for TextBlob is intensity. TextBlob uses the "intensity" to determine subjectivity. Whether a phrase or term changes the next word depends on its intensity. Adverbs are employed as modifiers in English, such as "extremely good."

A small analysis was carried out with Python programming language and performed TextBlob sentiment analysis over Twitter API collected posts for seven different locations and after performing text pre-processing with our proposed algorithm and thus analysis is done using such code defined in Fig. 5 shown below.

```
from textblob import TextBlob
def sentiment_analysis(tweet):
 def getSubjectivity(text):
   return TextBlob(text).sentiment.subjectivity

 #Create a function to get the polarity
 def getPolarity(text):
   return TextBlob(text).sentiment.polarity

 #Create two new columns 'Subjectivity' & 'Polarity'
 tweet['TextBlob_Subjectivity'] = tweet['clean'].apply(getSubjectivity)
 tweet['TextBlob_Polarity'] = tweet['clean'].apply(getPolarity)
 def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'
 tweet ['TextBlob_Analysis'] = tweet  ['TextBlob_Polarity'].apply(getAnalysis )
 return tweet

dfnew=sentiment_analysis(df)
```

Fig. 5. TextBlob sentiment analysis with our proposed approach.

As, TextBlob analysis can parse the content using just few lines of code and is created with the help of NLTK and Pattern. It is also very simple to use. TextBlob carries out a variety of operations on textual data which includes the extraction of noun phrases, spelling checks, sentiment analysis, categorization etc. It offers features like language identification and translation that are supported by Google Translate.

4. Results

Based on seven selected location across the country, it is determined using the dataset and then cleaned dataset is introduced that also calculate the text polarity and subjectivity of the dataset. After that, TextBlob analysis is carried out on the basis of sentiment score considered as positive, negative or neutral.

The highly occurred word or phrases set is generated and shown as bar graphs represented below in screenshots provided for two locations.

4.1. Location: Aligarh

In Fig. 6 explains three categories as neutral, positive and negative tweets count of chosen location "Aligarh" through Textblob sentiment analysis. It evaluates and count the neutral tweets as 160, negative tweets as 98 and positive tweet count as 95.

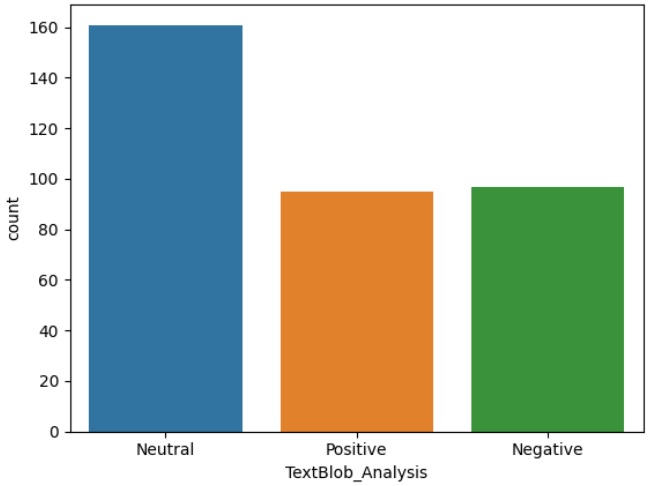

Fig. 6. Number of Aligarh tweets correctly classified as Positive, Negative and Neutral.

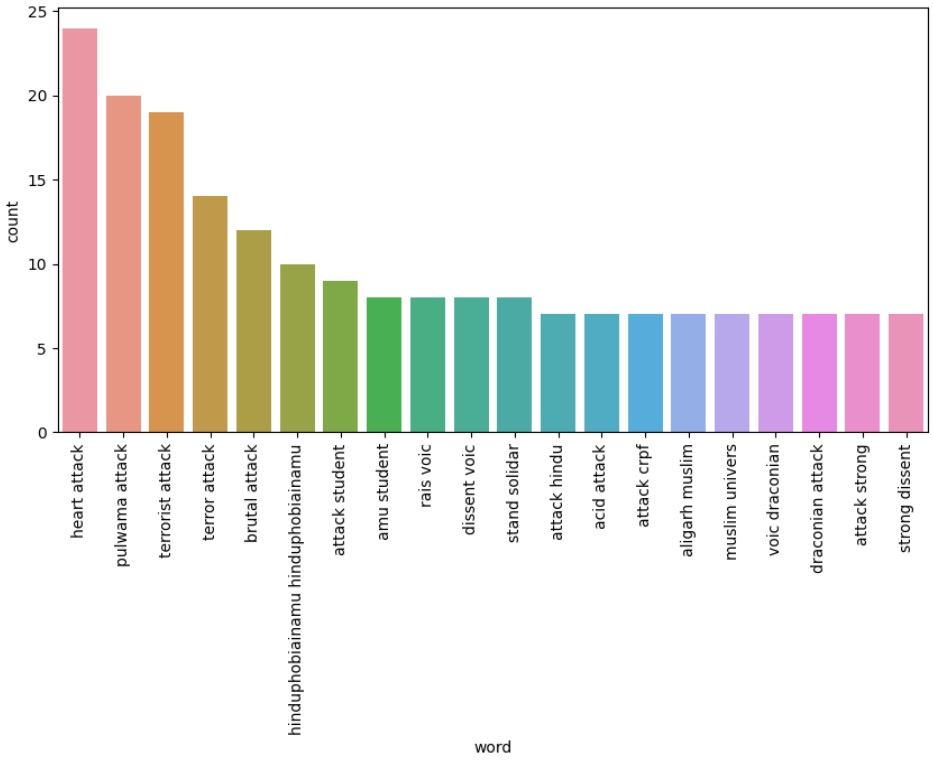

Fig. 7. Word count of high occurrence phrases of tweets from location Aligarh.

Fig. 7 describes the word or phrases count which spreads toxicity like heart attack, pulwama attack, terrorists attack, raise voice and more.

4.2. Location: Gurugram

Fig. 8 explains three categories as neutral, positive and negative tweets count of chosen location "Gurugram" through TextBlob sentiment analysis. It found that the 138 counts for positive, 113 count for neutral and 117 counts for negative tweets.

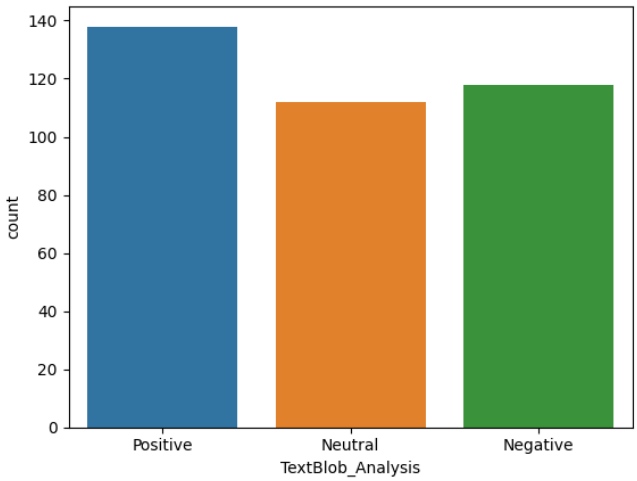

Fig. 8. Number of Gurugram tweets correctly classified as Positive, Negative and Neutral.

Fig. 9 gives the count for high occurrence word or phrases count like heart attack, boat attack, terrorist attack, blacksea and more.

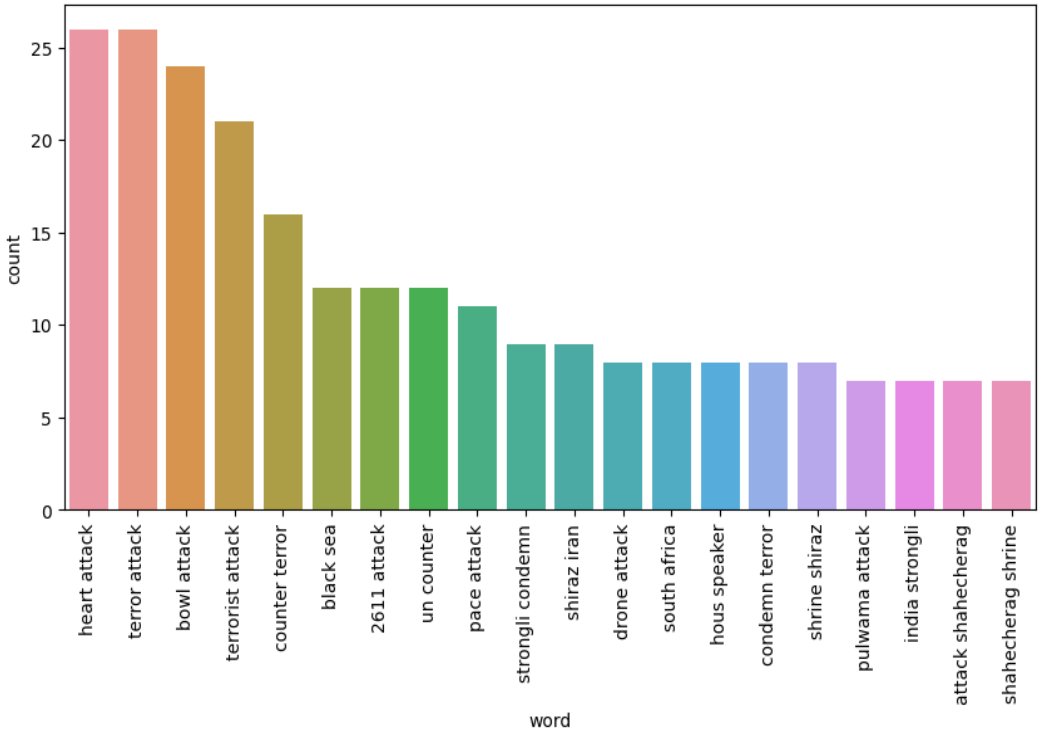

Fig. 9. Word count of high occurrence phrases of tweets from location Gurugram.

In the experimental setup and analysis, it is found that proposed model correctly classifies the tweets by highlighting the toxic words or phrases for various locations as represented in Fig 7 and 9 above. The metrics calculations that we have calculated as shown in Table 4. The final result can be shown through pie chart that gives comparison between data points or categories. This shows the overall outcome of experimental work that out of 16634 tweets as a corpus, 8475 were positive, 5748 were neutral and 2411 were negative i.e. represented in Fig. 10 below.

Table 4. Metric calculation scores for the proposed model.

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
CNN	0.984	0.887	0.905	0.895

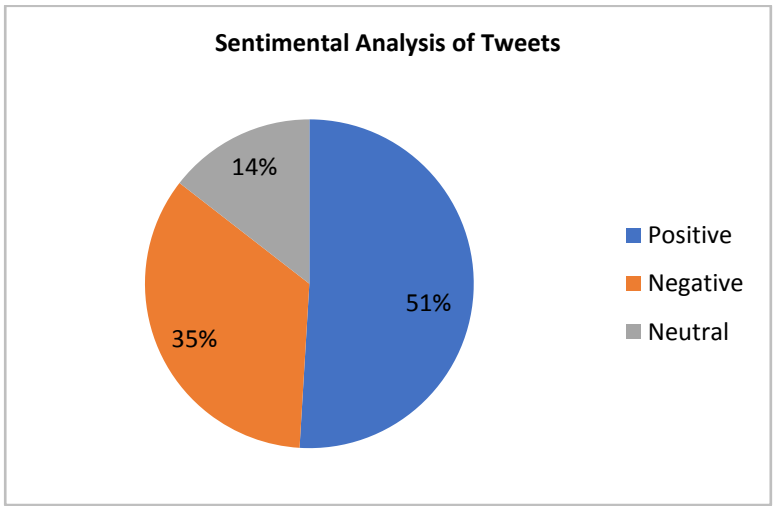

Fig. 10. Visualization of result.

5. Discussions

This paper is based on the location based tweet analysis, clean text analysis using proposed algorithm that classify tweets as Positive, Neutral and Negative along with the highly occurred words or phrases term in the collected document of tweets. In this manner, the possibility of finding such highly toxic word or phrase in the posted text can be easily detected using Convolutional neural network as machine learning approach. The location of tweets in existing approach will provide great benefit to the developers by reducing the amount of efforts; otherwise large efforts have to invest to develop same things.

In this paper, we have preprocessed dataset using clean text analysis and perform sentiment analysis through TextBlob library that calculated for polarity and subjectivity of particular tweet with generated scores. The text then fed to count vectorizer to convert it into proper embeddings for feature generation and selection. The CNN model is then applied and performs the metric calculation in terms of accuracy, precision, recall and f1-score with calculated results as 0.984, 0.887, 0.905 and 0.895 respectively.

In general, we have analyzed 16634 tweets, the plots for various locations encountered positive, negative and neutral tweets for toxicity detection. TextBlob and NLP are great methods to tackle this problem when attempting to use sentiment analysis and enter into the data realm.These are simple yet effective strategies that facilitate understanding of difficult language.This study set out to determine the most straightforward and approachable method for finding out what people thought about any given article, campaign, product, character, or industry.The steps to accomplish the aforementioned goal are suggested in this paper. A special technique was used to capture the data that is contained in the CSV file. After cleaning and transformation, the TextBlob tool was used to measure the polarities in this data. To ascertain the true sentiment of the tweet, the TextBlob program looked at the tweet's scores. It is also challenging to gather data at several locations for performing location based analysis. This paper can categorize for the cluster of negative tweets and inform to the concern authority of those identified locations from which those tweets have generated to prevent the occurrences of hatred unwanted events such as riots which will spread the negativity in our society because specially in the festival season such type of unwanted events occur because of those negative tweets but we are not aware the exact location. So here in this paper we also performed the location-based analysis of toxic/non-toxic comment identification.

6. Conclusion

In industrial and research academia, an efficient model algorithm for toxic/non-toxic comment identification have always been an important consideration in interaction for any social media platform. This work presents an algorithm and flowchart using CNN for detecting toxic content classification over given dataset stream. It can be beneficial to sentimental analysis for detection of toxic/non-toxic comments through which a social media platform company can take subsequent decision for handling the positive and negative impact on the society and in addition to that we can also filter the tweets according to the given locations as mentioned in the dataset in result section and on the basis of this location based analysis. As far as future work is concerned, the proposed algorithm detects the toxicity of tweeted

materials such as text which also considered as a limitation of the proposed work, but we can extend this work to check the toxicity of images, audio or any other multimedia contents over the social media platforms and detect their exact location from where that kind of toxicity generated. Using CNN model and proposed algorithm, images can be processed to perform sentimental analysis. Furthermore, we can extend this work over various locations to perform location-centred tweets sentiment analysis.

References

[1] Chakrabarty N., 2020. A Machine Learning Approach to Comment Toxicity Classification. In: Das A., Nayak J., Naik B., Pati S., Pelusi D. (eds) Computational Intelligence in Pattern Recognition. Advances in Intelligent Systems and Computing, Springer, Singapore, 183-193. https://doi.org/10.1007/978-981-13-9042-5_16.

[2] Tóth, Z., Mrad, M., Itani, O.S., Luo, J. and Liu, M.J., 2022. B2B eWOM on Alibaba: Signaling through online reviews in platform-based social exchange. Industrial Marketing Management, 104, pp.226-240.

[3] Mourad, A., Srour, A., Harmanani, H., Jenainati, C. and Arafeh, M., 2020. Critical impact of social networks infodemic on defeating coronavirus COVID-19 pandemic: Twitter-based study and research directions. IEEE Transactions on Network and Service Management, 17(4), pp.2145-2155.

[4] Abbasi, A., Javed, A.R., Iqbal, F., Kryvinska, N. and Jalil, Z., 2022. Deep learning for religious and continent-based toxic content detection and classification. Scientific Reports, 12(1), p.17478

[5] Machova, K.; Mach, M.; Vasilko, M. Comparison of Machine Learning and Sentiment Analysis in Detection of Suspicious Online Reviewers on Different Type of Data. Sensors 2022, 22, 155. https://doi.org/10.3390/s22010155.

[6] Bhatt, U.; Iyyani, D.; Jani, K.; Mali, S. Troll-Detection Systems Limitations of Troll Detective Systems and AI/ML Anti-trolling Solution. In Proceedings of the 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 6–8 April 2018; pp. 1–6.

[7] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y., 2016. Abusive language detection in online user content, in: Proceedings of the 25th international conference on world wide web, pp. 145–153. . International World Wide Web Conferences Steering Committee.

[8] M. Ibrahim, M. Torki, N. El-Makky, Imbalanced toxic comments classification using data augmentation and deep learning, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, IEEE, pp. 875–878.

[9] M. Anand, R. Eswari, Classification of abusive comments in social media using deep learning. in: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, IEEE, pp. 974–977.

[10] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," Proceedings of the 10th Hellenic Conference on Artificial Intelligence, 2018. [Online]. Available: https://doi.org/10.1145/3200947.3208069]

[11] Mohammad, Fahim. 2018. Is preprocessing of text really worth your time for online comment classification? arXiv:1806.02908.

[12] Van Aken, Betty , Julian Risch, Ralf Krestel, and Alexander L. öser. 2018. Challenges for toxic comment classification: An in-depth error analysis. arXiv:1809.07572.

[13] H.H. Saeed, K. Shahzad, F. Kamiran, Overlapping toxic sentiment classification using deep neural architectures, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018, IEEE, pp. 1361–1366.

[14] Li, Siyuan. 2018. Application of recurrent neural networks in toxic comment classification. Ph.D. thesis, UCLA.

[15] Saif, Mujahed A., Alexander N. Medvedev, Maxim A. Medvedev, and Todorka Atanasova. 2018. Classification of online toxic comments using the logistic regression and neural networks models. In AIP conference proceedings, vol. 2048, 060011. AIP Publishing LLC.

[16] Ravi, Pallam, Greeshma S Hari Narayana Batta, and Shaik Yaseen. 2019. Toxic comment classification. International Journal of Trend in Scientific Research andDevelopment (IJTSRD).

[17] Agrawal, Sweta, and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In European conference on information retrieval, 141–153. Springer, arXiv:1809.07572.

[18] Ptaszynski,Michal, Juuso Kalevi Kristian Eronen, and FumitoMasui. 2017. Learning deep on cyberbullying is always better than brute force. In LaCATODA@ IJCAI, 3–10.

[19] Ray Oshikawa, J. Q. (2020). A Survey on Natural Language Processing for Fake News Detection. arXiv:1811.00770v2 [cs.CL], 8.

[20] Marina Danchovsky Ibrishimova, K. F. (2020). A Machine Learning Approach to Fake News Detection Using NLP. researchgate.

[21] Joni Salminen, M. H. g. (2020). Developing an online hate classifier for multiple social media platforms. Huma centric computing and information sciences.

[22] Jahan, M.S. and Oussalah, M., 2021. A systematic review of hate speech automatic detection using natural language processing. arXiv preprint arXiv:2106.00742.

[23] Wang, K., Yang, J. and Wu, H., 2021. A survey of toxic comment classification methods. arXiv preprint arXiv:2112.06412.

[24] Andrew Gibiansky, 2014. Convolutional Neural Networks. https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/

[25] Palavar, Y. E., Çelik, Z., Albayrak, A., Özçelik, E., & Erdil, M. (2022). Analysis of the Covid-19 Process in Terms of Health Managers. In 2022 2nd International Conference on Computing and Machine Intelligence (ICMI) (pp. 1-5). IEEE.

Authors' Profiles

Varun Mishra is currently working as an Assistant Professor in GLA University Mathura, UP, He has completed his PhD in Computer Science in 2023. He has completed his Master of Technology in Computer Science from SATI, Vidisha in 2013. He has more than 13 years teaching experience as an Assistant Professor. He is a life member of professional societies like IEEE, IAENG etc. and published more than 40 articles in National and International Scopus conferences, journals and book chapters.

Tejaswita Garg is PhD research scholar at Jiwaji University, Gwalior, Madhya Pradesh. She has completed her Master of Technology in Computer Science from Banasthali Vidyapeeth, Rajasthan in 2013. She has published her research papers in Scopus Indexed Journals and Conferences. She is life member of IAENG professional society of Computer Science. She is interested in research areas including Machine Learning, Artificial Intelligence and Data Analytics.

How to cite this paper: Varun Mishra, Tejaswita Garg, "Toxicity Detection Using TextBlob Sentiment Analysis for Location-Centered Tweets", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.2, pp. 123-134, 2025. DOI:10.5815/ijmecs.2025.02.06