

Semantic-Based Classification of Toxic Comments Using Ensemble Learning

Thilagavathy A^{1*}, Deepa R¹, Lalitha S.D¹, Naveen Raju D¹, Ramya R¹, Ramya M¹, Rithika L¹, Karpoora Sundari K²

¹Department of Computer Science and Engineering, R.M.K Engineering College, Tamil Nadu, India.

²Department of Electronics and Communication Engineering, K. Ramakrishnan College of Technology, Trichy - 621112, Tamil Nadu, India

Abstract. A social media is rapidly expanding, and its anonymity feature completely supports free speech. Hate speech directed at anyone or any group because of their ethnicity, clan, religion, national or cultural their heritage, sex, disability, gender orientation, or other characteristics is a violation of their authority. Seriously encourages violence or hate crimes and causes social unrest by undermining peace, trustworthiness, and human rights, among other things. Identifying toxic remarks in social media conversation is a critical but difficult job. There are several difficulties in detecting toxic text remarks using a suitable and particular social media dataset and its high-performance, selected classifier. People nowadays share messages not only in person, but also in online settings such as social networking sites and online groups. As a result, all social media sites and apps, as well as all current communities in the digital world, require an identification and prevention system. Finding toxic social media remarks has proven critical for content screening. The identifying blocker in such a system would need to notice any bad online behavior and alert the prophylactic blocker to take appropriate action. The purpose of this research was to assess each text and find various kinds of toxicities such as profanity, threats, name-calling, and identity-based hatred. Jigsaw's designed Wikipedia remark collection is used for this.

1 INTRODUCTION

Social media is frequently used to exchange many forms of information. Individuals frequently use it to convey their opinions and ideas. Though social networking is extremely fast, transparent, accessible, and access is easy, it is also extremely vulnerable owing to its exponential expansion. It becomes a vehicle for lawbreakers to disseminate various forms of hatred or prejudiced speech toward another group. Hate speech is a discourse that may be extremely detrimental to a person's or group's emotions and may add to violence or insensitivity, demonstrating illogical and inhuman actions. The increasing number of users of social media has resulted in a rise in statements of hatred, which is a felony. It is clear that hateful comments as well as racist violence are connected; it's additionally clear that offences are related. As the issue of sexist remarks becomes more prevalent, many initiatives are being undertaken at the government level. Networking and internet platforms have expanded exponentially in recent years. People can now use these tools to express themselves and their views, as well as to converse with others. Disagreements are unavoidable in such a circumstance due to differences in thought. However, these debates frequently degenerate into nasty social media fights, with one side using insulting language referred to as toxic comments. These venomous comments can be menacing, obscene, insulting, or driven by hatred of one's own identity. As a consequence, there is a danger of slander and harassment. As an outcome, some people cease expressing their opinions or seeking different points of view, which actually results in an unpleasant and unfair discussion. As a consequence, platforms, and communities struggle to facilitate equitable debate and are frequently forced to either limit user comments or disintegrate by completely closing down user comments. Jigsaw and Google created the Conversation AI team to create tools and methods for fostering healthy conversation. Using the Perspective API, they have also developed publicly available models on Comment Toxicity. Nevertheless, these models are prone to inaccuracies and do not provide users with the option of specifying the type of toxicity they want.

Toxic Remark Reduction in Online Conversation calls for a more secure and adaptive framework. This approach examines any message (a text or email or a remark on a social network that might be harmful or harmless) and determines the kind of toxicity present within it. Just harmful, seriously toxic, vulgar, menace, offense, and identity-based hatred are the many sorts of toxicity. This approach solves the disadvantage of the Aspect API framework, which presented all sorts of toxicity in the comment. Toxic remark spotting on social media has proven to be critical for content filtering. According to

* Corresponding author: atv.cse@rmkec.ac.in

the French Minister of Education, 18% of French pupils will be victims of online abuse by 2021. At the same time, the quantity of posts on these networks has been growing. In 12 years, the number of tweets per day has grown tenfold to 500 million currently. This demonstrates that the rapid and focused detection of toxic remarks on social networks has become a critical issue in assuring societal harmony. Therefore, this can only be done by automating online moderation. People misuse the chat system and transmit toxic messages since the advent of online platforms and simple access to the internet communities with not so familiar names. It is tricky to detect such potentially dangerous interactions that may contain insults, obscene phrases, and threats. Text matching is the most conventional and extensively used way of detecting such harmful content. The program compares the text to an available dictionary and determines whether or not the information is in the library. This strategy was immediately abused since individuals naturally desire to avoid censorship by inventing new terms that have the exact meaning but written differently. Another common technique is to shorten the term as well as phrase, which likewise throws off a lexicon toxic text detection algorithm. Because new techniques for countering lexicon dangerous text systems are constantly being devised by internet users on a daily basis, the process of creating such a dictionary quickly becomes repetitive, and it's straightforward that an improved discussion prediction system is required to fix this issue. Presumably, the most cutting-edge approach in computer science would be used to tackle this problem. Given the enormous rise in processing capacity in consumer devices, conventional machine learning approaches, which demand considerably more computational capacity than lexical comparison methods but provide possibly better outcomes, become a possibility. To ensure a clean and welcoming place for online conversation, social media censors must detect toxic content on social networking sites. To determine if a message or post is harmful or not, online network admins frequently go through the entire comment or post. But, with several lengthy postings, administrators want aid in locating harmful phrases in every post to determine if a comment is harmful or non-toxic rather than reading the entire post.

2 LITERATURE REVIEW

The primary disadvantages include Overheads in communication and computation are high. Solutions are ineffectual. The method is time-consuming. This paper offers a retrogression vector voting classifier (RVVC) ensemble technique to identify harmful remarks on social media sites. Under soft voting conditions, the ensemble combines the logistic retrogression with the support vector classifier. Numerous trials are run on the unbalanced and balanced datasets to evaluate the proposed approach's performance. To investigate their felicity, two point-birth techniques, TF- IDF and BoW are utilized [1]. RVVC surpasses all other individual models when TF- IDF characteristics are utilized with the SMOTE balanced dataset, achieving a delicacy of 0.97. This research suggests a way to prevent spam commentary on YouTube, which has recently seen massive growth. It looked at similar studies on YouTube spam comment cords and ran rocket tests with six different machine literacy methods. The results revealed that the ESM- S model performed well in four evaluation measures, while the ensemble model performed better in vids in colorful orders. Unborn exploration is expected to include TF- IDF or deep literacy methods to improve performance [2]. The main disadvantage is that those that have not been thoroughly investigated can easily lead to an error. This study describes the DComment multi-input neural network architecture for autonomous comment evaluation [3]. It assets the characters in the source and remarks using the Bi-directional long short term memory and graded GloVe models, and then concatenates the vectors as input to the MLP classifier.

DComment beatsprior approach and numerous types of deep convolutional neural networks in terms of generalization ability over data from multiple sources, according to experimental results. The key drawbacks, in this case, are that it is unsuitable for large-scale applications. Has major design challenges. Solutions have been proved ineffectual. This article examines the most current breakthroughs in the field of autonomous code remark creation [4]. It addresses four different forms of computerized assessment metrics, notably BLEU, METEOR, ROUGE, and CIDER, along with the most often used metrics in human review from three different perspectives: natural language features, response substance, and remark efficiency. Lastly, it discusses future research potential for investigating neural networks, combining several techniques, and dynamic switching in cross settings. prone to mistakes Complexity, inaccuracy, and insufficiency are all present [5]. To improve the semantic extraction of online remarks, this research proposes a novel Transformer-based memory network (TF-MN). The TF-MN model models the Weibo sentiment analysis of the Q&A task using the memory network model, and experimental findings demonstrate that it outperforms the state-of-the-art model. M. Jiang et al. will investigate how to use the capsule network for sentiment analysis of web comments in the future. This research offers TBLC-attention, a mongrel deep neural network model for Chinese textbook motion recognition. It entails acquiring and preprocessing the Chinese corpus, mapping the preprocessed textbook into word vectors, employing a Bi-directional Long Short- Term Memory network (Bi-LSTM) with the attention medium to acquire environment semantic features, employing Convolutional Neural Network (CNN) to gain original semantic features, and inputting the final point vectors into the bracket sub caste [6]. The findings suggest that the model can recognize semantic point information to the greatest extent possible, although deep literacy is an effective method for evaluating and recycling medical textbooks. When parameterized by millions of associated variables, this results in a loss of function that is exceedingly uncontrollable, resulting in data inaccuracies. This article highlights the current sentiment analysis research's usage of artificial

intelligence technologies to deconstruct sentiment and its activities. It presents a novel word representation method based on BiLSTM that incorporates emotional information directly into the classic TF IDF technique to build weighted word vectors [7-8]. The suggested sentiment classification system is compared against RNN, CNN, LSTM, and NB sentiment analysis styles. According to the experimental results, the suggested sentiment logical system has improved perfection, recall, and F1 score. A novel word vector representation scheme based on the improved length and weight computation is offered. The biggest disadvantages are the complexity of its real-time implementation and the poor performance of the application. This paper presents an integrated convolutional neural Network (CNN) and long short-term memory (LSTM) learning model for executing several tasks in multi-scale opinion grouping (MTL-MSCNN-LSTM). The model manages global and local properties of various message scales to present and depict phrases, and it enhances encoder quality while also improving state-of-mind classification results [9]. The preliminary findings demonstrate that the proposed model's opinion categorization results outperform the current state of the handicraft approaches. The effect of onslaught and remarks with varied emotional orientations on video attractiveness, in addition to their variations, is investigated in this study [10-11]. It discovers that a bombardment of favorable comments has a good effect on video popularity, however, the emotional intensity of positive remarks has a negative effect. The study's findings have significant managerial implications for video websites. Further study can improve emotional categorization and take into account the influence of video attributes. This study investigates the current application of artificial intelligence technology in sentiment analysis and research, as well as its operations [12]. It proposes a new BiLSTM-based word representation method that adds sentiment information into the standard TF IDF technique and generates weighted word vectors. The proposed sentiment analysis system is compared to Recurrent Neural Network, Convolutional Neural Network, Long Short Term Memory, and Naïve Bayes sentiment analysis approaches, and the testing results show that the proposed sentiment reasoning system outperforms them in terms of perfection, recall, and F1 score. It is proposed a new word vector representation strategy based on enhanced length and weight computation [13-14]. This project received a poor rating because it is unable to fulfill contemporary network business expectations in Fig.1.

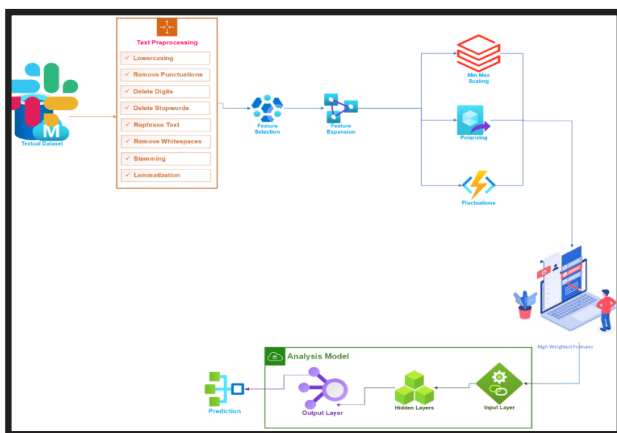


Fig.1. Architecture Diagram

3 Methodology

By using the given dataset and three feature extraction methods: TF-IDF Vectorize, Logistic Regression, and Random Forest Algorithm. Based on the Wikipedia information we developed the toxic remark categorization.

Description: Wikipedia toxicity subtypes

Text: English Wikipedia discussion page remarks archive

Label: Determination of the level of toxicity of the dataset's remarks.

1: Known to have a toxic characteristic.

0: Not a characteristic associated with that toxicity trait.

3.1 Text & data preprocessing

Information pre-processing could be considered a subset of information mining, which entails transforming raw data into a more logical structure. Crude knowledge is frequently contradictory or fragmented, and it usually includes numerous errors.

Checking for missing values, looking for category values, splitting the dataset into preparation and test groups, and lastly doing highlight scaling to limit the range of factors are all part of the information preprocessing. The preprocessing of information is a method used in information extraction to turn unprocessed information into a usable and efficient framework.

One-shot encoding is used to convert explanatory data such as shop category. The selection should then be adjusted for data balance. A dataset that has numbers that are not "0" within a given time is considered positive; otherwise, it is considered negative.

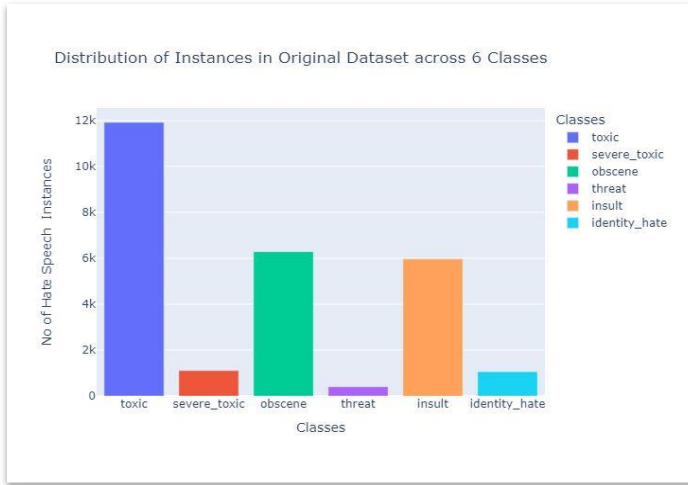


Fig.2. Dataset Distribution

We can extract stop words and words from the dataset using the Natural Language Toolkit tool and remove all stopwords such as "a," "the," and "is," among others. After the stop words have been removed. We can lowercase all of the phrases to make them appear more usual. Using the is null function across the initial training and test data, we found no missing entries in Fig.2. The cleansed data is then loaded into the Pandas data structure as training and test data. Because remarks on social media sites, such as Wikipedia, frequently contain irregular English, the use of special characters in lieu of characters (e.g., @UserName), and the use of digits to indicate letters, the text data was normalized. (e.g., G00d). To manage this, we use Regular expressions to perform word normalization. Then, for the training and testing data, we can create and save a new CSV file. To build a deep learning engine from text-based input, the information must first be converted into a machine-readable format in Table 1.

Table 1. Dataset Value after Processing

Total no. of Comments	134625 comments
No. of toxic comments	11917 comments
No. of severe toxic comments	1105 comments
No. of obscene comments	6284 comments
No. of threat comments	400 comments
No. of insult comments	5966 comments
No. of hate comments	1058 comments
Predicted table	51759 comments

3.2 Vector space model

The first step in the feature extraction procedure is to collect the important features. To identify a critical phrase in a document, the sentences must be represented as vectors or scored. To describe the text for this job, some characteristics are used as attributes. The most common characteristics for calculating a sentence's value and showing the extent to which it corresponds to a summary. To properly display the source papers, text summarization methods were already utilized. Text classification approaches in NLP include translating words to statistics so that machines could understand & decode language sequences. In general, these strategies create a connection in between selected sentence and the paper's context word.

Word-based vector space models incorporate phrases in a vector space where terms with comparable definitions are mapped close to each other. Words that share syntactic or grammatical relationships will be depicted by vectors of comparable size and put in close access to one another through word embedding rather than using lexical-based grammar parsers or extra resources. We use TF-IDF, Logistic Regression, and the Random Forest Algorithm to convert string remarks to numerical formulas for the converted machine-readable format code. Vectorizer is a famous method for converting text into understandable numerical forms. It is used to extract characteristics from text sequences based on frequency. We assume that the frequency of repetition of a term increases its significance in the given text in table.2.

Table 2. Precision values

Precision Value		
	Logistic Regression	Random Forest
Toxic	95.15	99.88
Severe Toxic	99.22	99.95
Obscene	96.96	99.91
Threat	99.79	99.99
Insult	97.12	99.9
Identity Hate	99.34	99.97

The frequency of a word is normalized based on the content of the text, and this is referred to as term frequency. Following the TF-IDF process, the information is subjected to Linear Regression and Random Forest to determine the precision of both processes. The One Vs Rest classifier fits one classifier per class, with all other classes treated as positives, and then aggregates the findings to make a final forecast. We use One Vs Rest classifier for multi-class classification where we have to predict multiple values for input. A binary classifier is trained and then it is subject to One Vs Rest classifier which trains one classifier against all other classes. The first stage is to use a One Vs Rest classifier to conduct multi-class classification. This is helpful when the dataset contains more than two groups. The Logistic Regression classification is the pipeline's second stage. It is used to calculate the likelihood of a binary occurrence using one or more predictor factors. The'sag solver, which stands for Stochastic Average Gradient descent, is used to answer the optimization issue. This solver works well with big datasets and converges more quicker than other solvers. The 'n jobs' option is set to -1, indicating that the solver will use all available jobs. We can use the pipeline to combine numerous stages into a unique object and execute them successively. The One Vs Rest classifier is used to tackle the multi-class classification issue in this instance, followed by the Logistic Regression classifier to make the final prediction.

The logistic regression classifier findings are limited to rows where the expected values for all six classes that match the real values, and the actual value for "toxic" is higher than 0. In other words, it selects only the rows where the logistic regression classifier correctly predicted that a comment is toxic, and also correctly predicted the absence or presence of the other five types of toxicity. This enables a more precise assessment of the classifier's performance in identifying toxic comments in Fig.3. The Logistic Regression Accuracy Score is 90.39.

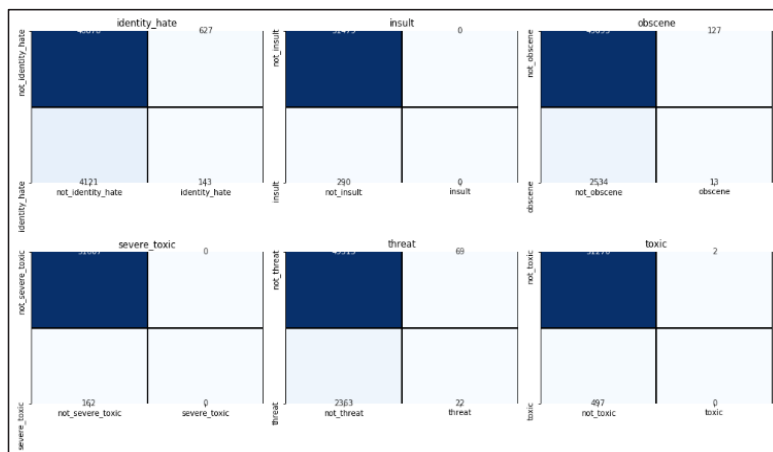


Fig. 3. Confusion Matrix- Logistic Regression

The same One Vs Rest classifier is used for Random Forest Classification. The first stage is to use an One Vs Rest classifier to conduct multi-class classification. This is helpful when the dataset contains more than two groups. The Random Forest Classification is the pipeline's second stage. It enables learning classification and regression in this stage. The Random Forest classifier contains a large number of decision trees which helps to train data in different subsets then the predictions from each decision tree are combined for the Final prediction. The Random Forest classifier is used in this pipeline as the binary classifier that is learned for each class in the One Vs Rest classifier phase. Because it is resistant to overfitting, can handle both category and numerical features, and can record complicated relationships between features, the Random Forest classifier is an excellent option for this issue. We can manage multi-class classification issues and obtain excellent accuracy in forecasting class labels for a given input by using a pipeline with One Vs Rest classifier and Random Forest classifiers.

Pipeline (steps = [('ovr', One Vs Rest classifier (estimator = Random forest classifier()))])

Then we create a data frame of the random forest classifier's predicted values, concatenate it with the original test data frame, and filter the resulting data frame to include only rows where the predicted values for all six classes match the actual values, and where the actual value for "toxic" is greater than 0. In other words, it chooses only the rows where the random forest classifier correctly predicted that a comment is toxic, as well as whether or not the other five types of toxicity are present. This allows for a more precise evaluation of the classifier's performance in identifying toxic substances in Fig.4. The Random Forest Classifier Accuracy is 94.26.

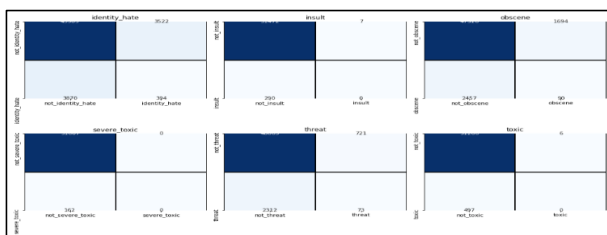


Fig. 4. Confusion Matrix – Random Forest

Tokenizer class in Keras that transforms a collection of strings of text into a set of sequences. Each number in the series corresponds to a specific term in the Tokenizer's vocabulary. The tokenizer object has been fitted on the training data, so it has already learned a vocabulary of words from the training set. `tokenizer.Texts to Sequences(X test)` takes the list of test comments and converts each comment into a sequence of integers that represents the words in the comment according to the vocabulary learned during training. The resulting output is a list of sequences of integers, where each sequence represents a single comment. This is required as the input to the LSTM model during testing.

3.3 Model Training and Prediction

The Neural Network is made up of a memory cell module that can acquire data properties in the time domain. It has three additional units: an input gate, forgetting gate, and an affair gate, which separately regulate the input, update, and affair of information. For ordering and modelling activities, LSTM is used.

Glove which is known as Global Vectors for Word Representation is a popular word embedding method that converts every word in a collection into a high-dimensional vector. Glove is used to insert the credentials. This anchoring method is used to initialize the neural network. Load the pre-trained word embeddings from "glove.6B.50d.txt" and build an embedding matrix for the words in our vocabulary. First, it defines a function `get_coefs()` which is used to extract the word and its corresponding vector representation from each line of the embedding file. The lines of the file are read in and processed one at a time using list comprehension and the resulting pairs are stored in a dictionary called `embeddings_index`. Next, all embs are created as a numpy array by stacking the vector representations from the `embeddings_index`. The embeddings' mean and standard deviation are then calculated and saved in `emb_mean` and `emb_std`.

We build a Keras LSTM model for text categorization that includes an embedding layer, a bidirectional LSTM layer, a dense layer, and an output layer. The output layer is composed of six neurons, each representing a binary classification assignment to one of six toxic remark groups. In layer of the output, the activation function is sigmoid, which produces integers between 0 and 1. The LSTM model is taught to predict whether a specific comment is harmful or not based on the patterns and correlations discovered in the text data. The LSTM model is built with the binary cross-entropy as its loss function, the optimizer is 'adam,' which is a popular stochastic optimization algorithm that adapts the learning rate based on the gradient of the loss function, and the metrics are accuracy, which is collected from training and testing data. To identify

the loss ratio, the model is taught repeatedly to eliminate the loss ratio and better forecast the toxicity level. Because the data is trained for a specific group of epochs, the loss ratio is gradually reduced in the graph. To extract the predicted value from the learnt data, the Keras prediction approach is now employed. The forecast function is built using the tokenized data. We want to build a pandas Data Frame from an array of predicted values, and you want to give the data frame unique column titles. The data frame that results can then be used for additional research, display, or comparison with other data in Fig.5.

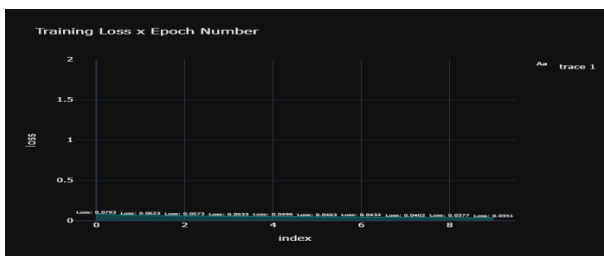


Fig. 5. Training Loss

After executing the prediction function, the prediction function and test results are concatenated. With the help of the trained deep learning model, this function determines whether a comment includes hate speech or not after receiving it as input. The training data's tokens are first applied to the remark using the same tokenizer. The tokenized remark is then padded to a fixed length before being fed into the deep learning model. Next, the tokenized and padded comment's output class probabilities are predicted using the predict technique. In the results variable, the estimated probabilities are kept. The predicted target column is then stored in the text variable, along with whether the predicted probability is higher than 0.5, which serves as the cutoff point for determining whether the comment is valid.

4. Results & discussion

The initiative concentrates on detecting the toxicity of comments on non-social media sites using the Keras model. The model, which was trained using an evolutionary learning approach and accessible data prediction, is designed to detect hate speech and toxic remarks which may be categorized as name calling, cyberbullying, and aggressive behaviors. The model's accuracy was assessed, and it was discovered to be more accurate than current models accessible online. The Keras model used in this research analyses text data and classifies remarks as toxic or non-toxic using LSTM networks. Furthermore, the project employs the Term Frequency-Inverse Document Frequency (TF-IDF) approach, which is a statistical method for determining the significance of a word in a document. This technique assists the model in identifying and classifying the most pertinent words in a remark. The project's results indicate that the Keras model obtained an accuracy of 95.3%, which is considerably better than the currently online models. Furthermore, the project assessed the performance of other categorization algorithms, such as Random Forest and Logistics Regression, and discovered that they performed worse than the Keras model in terms of precision. This project's conversation subject could be the significance of spotting toxic comments and hate speech on social media platforms in Fig.6. Overall, this project emphasizes the significance of creating effective tools for spotting toxic comments and hate speech on social media platforms, and it offers a hopeful solution that can be improved and refined further in the future in Fig.7.

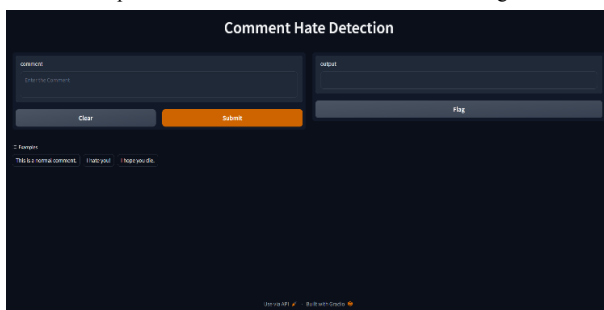


Fig. 6. Output without Comments

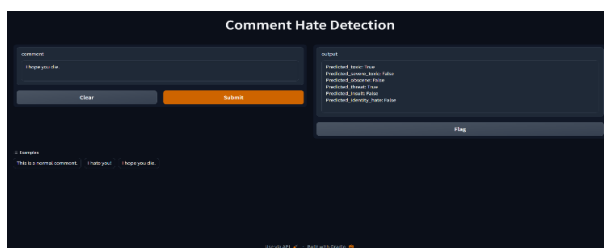


Fig. 7. Output with Comments

5. Conclusion

Due to the significance of online engaging interactions among users, both the business and academic groups have made several attempts in recent years to develop a successful approach for online harmful comment prediction, but these efforts are still in their early stages. This article examines an innovative text classification technique that makes use of word representations and the Long short-term memory network concept. To detect toxic amounts in different phrases, we used a method based on online comments, and all of the models worked well. Long short-term memory networks with Word embedding layers, in particular, achieve the greatest accuracy, and Word embedding layers function more effectively altogether.

REFERENCES

- 1 V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model," in *IEEE Access*, **9** (2021).
- 2 H. Oh, "A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model," in *IEEE Access*, **9** (2021).
- 3 D. Wang, Y. Guo, W. Dong, Z. Wang, H. Liu and S. Li, "Deep Code-Comment Understanding and Assessment," in *IEEE Access*, **7** (2019).
- 4 X. Song, H. Sun, X. Wang and J. Yan, "A Survey of Automatic Generation of Source Code Comments: Algorithms and Techniques," in *IEEE Access*, **7** (2019).
- 5 M. Jiang, J. Wu, X. Shi, and M. Zhang, "Transformer-Based Memory Network for Sentiment Analysis of Web Comments," in *IEEE Access*, **7** (2019).
- 6 Jin, Qibing & Xue, Xingrong & Peng, Wenjuan & Cai, Wu & Zhang, Yuming & Zhang, Ling. TBLC-rAttention: A Deep Neural Network Model for Recognizing the Emotional Tendency of Chinese Medical Comment. *IEEE Access*. (2020).
- 7 G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," in *IEEE Access*, **7** (2019).
- 8 N. Jin, J. Wu, X. Ma, K. Yan, and Y. Mo, "Multi-Task Learning Model Based on Multi-Scale CNN and LSTM for Sentiment Classification," in *IEEE Access*, **8** (2020).
- 9 Wang, Fudong & Fu, Xiayu & Sun, Zengyao. A Comparative Analysis of the Impact of Barrage and Comments on Video Popularity. *IEEE Access* (2021).
- 10 G. Tian, Q. Wang, Y. Zhao, L. Guo, Z. Sun and L. Lv, "Smart Contract Classification with a Bi-LSTM Based Approach," in *IEEE Access*, **8** (2020).
- 11 P Matheswaran, C Navaneethan, S Meenatchi, S Ananthi, K Janaki, A Manjunathan. *Image Privacy in Social Network Using Invisible Watermarking Techniques*. *Annals of the Romanian Society for Cell Biology*, **25(5)** 319-327 (2021)
- 12 Rajagopal Sudarmani, Kanagaraj Venusamy, Sathish Sivaraman, Poongodi Jayaraman, Kannadhasan Suriyan, Manjunathan Alagarsamy. *Machine to machine communication enabled internet of things: a review*. *International Journal of Reconfigurable and Embedded Systems*, **11(2)**, 126-134 (2022).
- 13 Kannadhasan Suriyan, Nagarajan Ramaingam, Sudarmani Rajagopal, Jeevitha Sakkarai, Balakumar Asokan, Manjunathan Alagarsamy. *Performance analysis of peak signal-to-noise ratio and multipath source routing using different denoising method*. *Bulletin of Electrical Engineering and Informatics*, **11(1)**, 286–292 (2022).
- 14 Chitra Raju, Sudarmani Rajagopal, Kanagaraj Venusamy, Kannadhasan Suriyan, Manjunathan Alagarsamy. *SDSFLF: fault localization framework for optical communication using software digital switching network*. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, **12 (1)**, 113-124 (2023)