Analysis of Tweets for Cyberbullying Detection

Shipra Anil Mathur
Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka, India - 575025
Email: mathurshipra33@gmail.com

Shivam Isarka
Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka, India - 575025
Email: shivam1221agrawal@gmail.com

Bhuvaneswar Dharmasivam
Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka, India - 575025
Email: bhuvaneswar5d@gmail.com

Jaidhar C. D.
Department of Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka, India - 575025
Email: jaidharcd@nitk.edu.in

2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC) | 979-8-3503-0071-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSCCC58608.2023.10176416

***Abstract*—Cyberbullying takes place online on gadgets like smartphones and computers. Cyberbullying can occur through social media platforms. This paper presents a real-time cyberbullying detection system for Twitter using Natural Language Processing (NLP) and Machine Learning (ML). The system is trained on a dataset of cyberbullying tweets using several ML algorithms and their performance is compared. Random Forest was found to provide the best results after tuning. To achieve real-time analysis, Selenium was used to scrape tweets from a given Twitter account and store the timestamp of the already checked tweets. Additionally, an image captioning model was employed to generate descriptions for images posted on the account and compare them with user-written captions to filter out spam tweets. The proposed work aims to prevent cyberbullying and provides a valuable tool for online platforms to detect and remove harmful content. The results of this study have shown that the selection of appropriate ML algorithms and preprocessing techniques significantly impact the performance of cyberbullying detection on Twitter. Our model sheds light on the appropriateness of different ML algorithms for the detection of cyberbullying.**

***Index Terms*—Twitter, Cyberbullying, ML, Image Captioning**

I. Introduction

The rise in social media usage has brought numerous benefits, including excellent connectivity and instantaneous information sharing. However, it has also given rise to a new form of harassment known as cyberbullying. Cyberbullying is using technology to harass, intimidate, or humiliate others online.

To combat cyberbullying, there has been an increasing interest in developing automated systems for real-time detection and prevention of harmful content on social media platforms. Machine Learning (ML) algorithms and Natural Language Processing (NLP) techniques have been used to detect cyberbullying, hate speech and other online abuse. This study presents a real-time cyberbullying detector that uses NLP and ML on the social media platform Twitter. We trained our ML model on a dataset of cyberbullying tweets and compared the performance of several algorithms to identify the most effective one. Our system can detect cyberbullying in real-time by scraping tweets from a given Twitter account using Selenium. We avoid checking the same tweets multiple times by storing their timestamps. The research results showed that the machine learning-based solution effectively detected and classified instances of cyberbullying with a high accuracy. The solution successfully identified cases of cyberbullying in real-time, making it a valuable tool in the fight against this issue on social media platforms. Additionally, we used an image captioning model to generate descriptions for images posted on the account and compared them with user-written captions to filter out spam tweets. Our proposed system provides a helpful tool for online platforms to detect and remove harmful content and prevent cyberbullying.

This paper's contribution is three-fold: first, it demonstrates the effectiveness of various ML algorithms and preprocessing techniques for cyberbullying detection on Twitter. Second, it presents a practical system for real-time cyberbullying detection. Third, using image captioning helps to filter out spam tweets. The proposed method offers a promising direction for future research in this field and has the potential to contribute to the development of a safer and healthier online community.

II. Literature Survey

M. M. Islam et al. [1] proposed a technique to detect cyberbullying messages on social media using NLP and ML. The authors studied four machine learning algorithms—SVM, Naive Bayes, Decision Trees and Random Forest—and ran tests on two datasets derived from various comments, tweets and posts on Twitter and Facebook. They used Bag-of-Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF) for analyzing the performance. The results show that SVM outperforms the other machine learning methods and the TF-IDF feature offers superior accuracy than BoW. The paper provides insights into detecting cyberbullying on social media, a significant issue affecting many people worldwide.

A BoW method was suggested as a supervised machine learning algorithm to identify a sentence's sentiment and contextual characteristics. This algorithm's accuracy rate is only 61.9%. A project named Ruminati [2] used SVM to identify cyberbullying in YouTube comments. It included social characteristics to combine detection with common sense reasoning. It enhanced the accuracy to 66.7% by using probabilistic modelling. It is built on a deep learning network using the transformer approach to detect cyberbullying. It uses a single linear layer of a neural network for classification. A Formspring dataset is the smaller of the two datasets used, whereas the other dataset is larger (Wikipedia dataset). For the latter, the model offered more accurate and trustworthy results.

Zhang et al. [6] utilized a Pronunciation-based Convolution Neural Network (PCNN) to overcome the problem of noise and bullying data sparsity. Twitter sent 13,313 messages and Formspring sent out 13,000 messages. The accuracy of the Twitter dataset wasn't calculated since it needed to be more balanced. Although they achieved 56% precision, 78% recall and 96% accuracy, their dataset was unbalanced, it produced false results and decreased the precision score to 56%.

Image captioning and similarity analysis have been studied in recent years, particularly in the field of computer vision and NLP. The use of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks, has been suggested as one method for captioning images. Similarity analysis has also been studied in regard to retrieving documents and answering questions. In the specific context of social media analysis, image captioning and similarity analysis have been used for various applications, such as spam detection and content analysis.

III. Proposed Methodology

This section outlines the methodology proposed to detect cyberbullying and conduct real-time Twitter account analysis. The proposed model's overall architecture is shown in Fig. 1.

A. Data Collection

The dataset used to train the classifier is taken from Kaggle, which provides more than 47000 tweets categorized into the various forms of cyberbullying and non-cyberbullying. The data has been balanced to ensure that there are about 8000 of each class. Data is imported into a pandas data frame after data collection to continue the preprocessing, extraction, classification, evaluation and real-time analysis processes.

B. Preprocessing

Fig. 2 shows the preprocessing steps used for the tweets.

1) Standardizing the case: Change the capital letters to lowercase characters and make them standard in order to speed up the subsequent process while maintaining consistency.

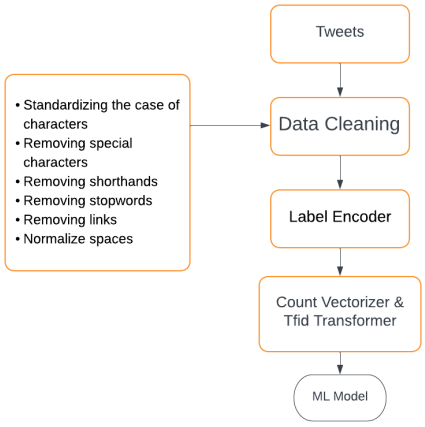

Fig. 1. Preproceesing

2) Special Characters Removal: The Twitter data could include emoticons, Links, mentions, duplicate or irrelevant tweets, retweets and special characters. Using regular expressions, preprocess the data to eliminate these undesirable components and also split the tweet text into individual words or tokens
3) Shorthands Removal: Removing the shorthands like "shouldn't" and "Would've" of words and expanding them to their complete form like "should not" and "would have" to standardize the texts.
4) Stopwords Removal: Remove common words like "a", "an", "the", etc., that don't carry much meaning or discriminate between classes. Using NLTK's stopwords library attribute to filter out the stopwords.
5) Links Removal: Removing links to other web pages or tweets from the text as they don't add any value to the tweet's inherent text.
6) Normalizing spaces: Normalizing the spaces between the different words in the tweets and making sure that each word is separated by a space from another.
7) Label Encoding: Due to the fact that our dataset contains various text-based categories for different types of cyberbullying. To make the labels machine-readable, we encode them using numbers through the process of label encoding where the different labels are assigned different numbers.
8) Count Vectorization and TF-IDF transformation: After Label Encoding is done, Count Vectorizers are employed to convert the tweet's text into a vector based on the frequency with which each word appears across the entire message. For the purposes of experiments, a word can serve as a representation of a cyberbully if it occurs more frequently in those tweets than in typical tweets. The TF-IDF is calculated using Equation (1).
where W_{ij} is the word's TF-IDF score, N_{ij} is how many times the word appears in the document 'd', while the total number of word appearances in the document 'd' serves as the denominator. The percentage of a document

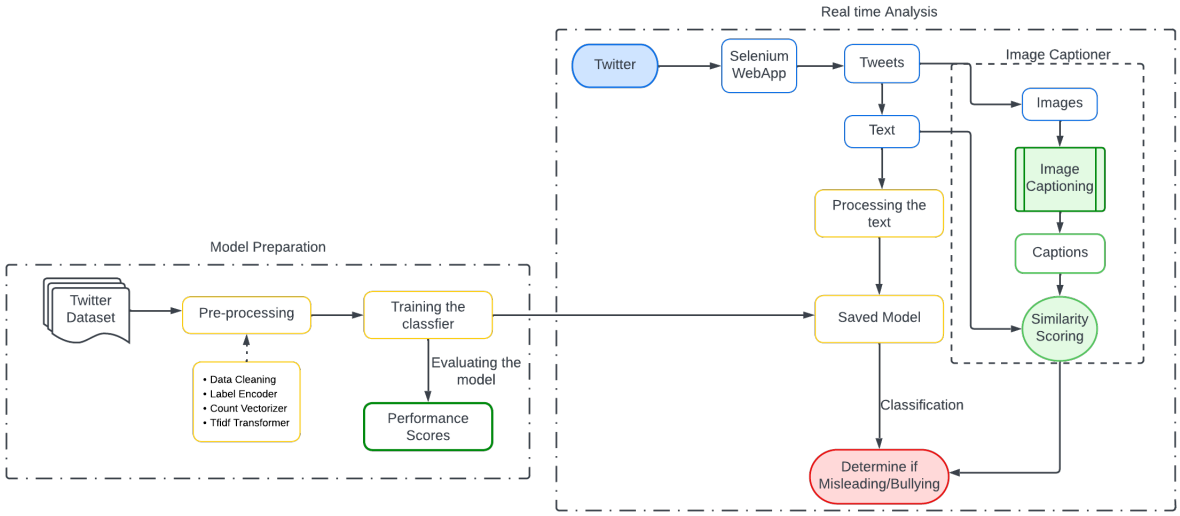

Fig. 2. Proposed Architecture

that contains the term t_j is S_{ij}. We use $1 + Sij$ as the denominator because the term might not be in the corpus and the numerator will be 0.

$$W_{ij} = \frac{N_{ij}}{\sum N_{kj}} \times log \frac{D}{1 + Sij} \quad (1)$$

After these preprocessing steps, the data frame can be passed on to the classifier for training and validation.

C. Training the Classifier

In this stage of classification, different machine learning classifiers like Random forest, AdaBoost and Gradient Boosting were used and their accuracy scores were compared. Based on that, the one with the highest accuracy for further real-time analysis of the tweets was used.

1) Random Forest Classifier: It is a multiple decision tree classifier. Each tree provides a classification. The class with a majority vote is given as the output. This classifier is a supervised learning method. A model with accurate outcomes based on numerous choices combined with trees that produce the desired result. Instead of using a single decision tree, RF uses predictions from all of the trees that are formed. A majority vote of the individual trees then decides the final result. It was trained using the preprocessed dataset with different parameters like the number of estimators and criteria for information such as the Gini Index, Entropy and etc. and checked which combination fetches us the best accuracy for the test dataset. Fig. 3 shows the working of a Random Forest.
2) Gradient Boosting Classifier: Gradient boosting is predicated on the idea that fusing the best next model with the prior model will reduce overall prediction error. Setting a goal result for this next model in order to reduce error is an important concept. The desired result for each data case is determined by how modifications to the predictions for that case impact the overall prediction error.
3) AdaBoost Classifier: To perform better than individual classifiers, it integrates other weak classifiers or machine learning techniques. Among the boosting family, the AdaBoosting model performs better and is frequently applied to classification issues. The AdaBoosting model computes the loss value using an exponential loss function, which allows it to choose the optimal classification procedure at each iteration.

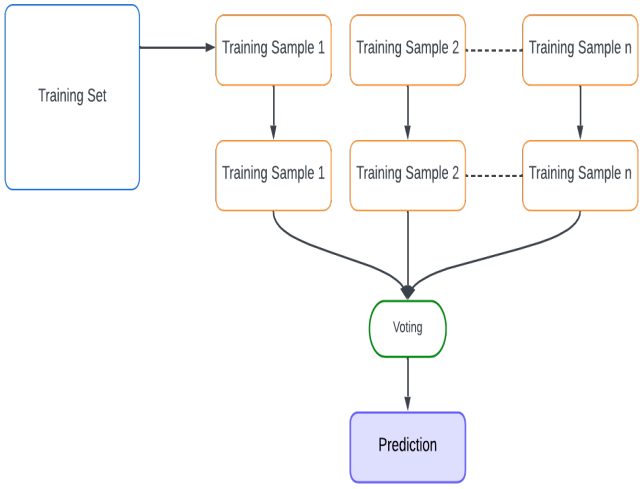

Fig. 3. Random Forest

D. Real-Time Tweet Collection and Classification

Selenium Webdriver was used to automate the process of extraction of tweets. The entire program is written in Python programming language. The initial steps of logging in by

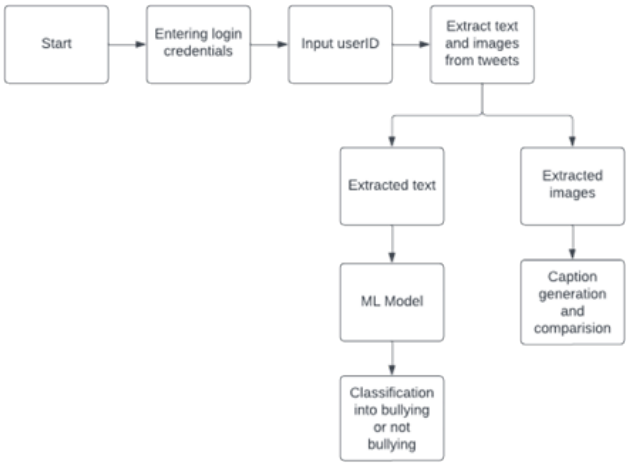

Fig. 4. Tweet Retrieval

entering the username and password are automated by the program. In order to determine which profile will be evaluated, the program collects the user's input. After the username is entered, the web page is directed to the required profile. After reaching the profile, the text of tweets and links of images posted is collected in a CSV file. The CSV file containing text lists is then passed to the ML model. The extracted tweets are then classified as bullying or not by the model that was trained using a sample of about 47000 tweets from Twitter. The output from the ML model is sent back to the Selenium program. It was noticed that when the user tests for a profile more than once, the same tweets are extracted again and again. To avoid this, another CSV file was created, which saves the DateTime of the latest tweet checked on that profile. If a profile is checked repeatedly, the DateTime of the latest tweet checked is updated accordingly. If a new profile which has not been checked before is tested, the username and DateTime of the latest tweet checked are added to the CSV file. Fig. 4 shows how the tweets are collected in real-time and classified.

E. Caption Generation and Similarity scoring

The images pulled by the selenium driver were imported and then passed to the image captioning module provided by the Salesforce-LAVIS library, the returned captions thus returned by the model were stored along with the tweet ID's of the images. Then the original caption of the image and the one generated by the model were matched and the BLEU similarity score was calculated. Based on the aforementioned score it was determined whether the tweet is misleading or not. Fig. 5 shows the working of the caption generation and similarity scoring.

IV. Results and Analysis

As the proposed model is a multitasking framework which performs three primary tasks, Cyberbullying detection using machine learning, caption generation for images in tweets and real-time analysis of Twitter accounts we require an

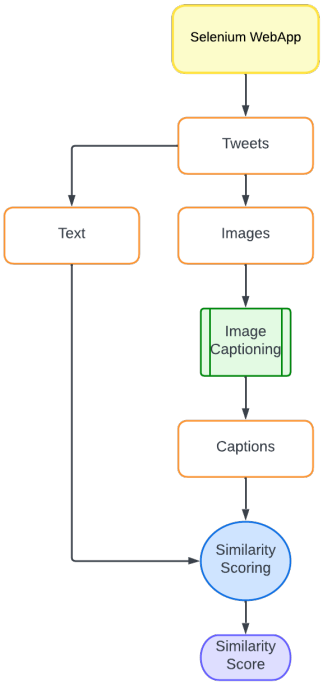

Fig. 5. Caption Generation and Similarity Scoring

extensive dataset for training the classifier. Experiments were conducted with a publicly available tweet dataset from Kaggle. Furthermore, to increase the accuracy and robustness of the classifier, experiments were conducted with different preprocessing techniques and parameters for the classifier. The final output of the work is displayed in Fig. 12.

A. Dataset

This dataset contains more than 47000 tweets labelled according to the class of cyberbullying Age, Ethnicity; Gender; Religion; Other types of cyberbullying; Not cyberbullying. To ensure that there are about 8000 of each class, the data has been balanced. This dataset is then encoded to make classification and training the model easier. The dataset description has been tabulated in Table I.

TABLE I
Twitter Dataset Description

CATEGORY	DATA
Age	8000
Gender	8000
Religion	8000
Ethnicity	8000
Others	8000
Non Bullying	8000

B. Tuning of Random Forest Classifier

After the data was preprocessed and vectorized, the RF classifier was trained. Different parameters were used for the algorithm to see which one yields the best accuracy

as well as has the most optimal test accuracy to training time ratio. From experimental results with different combinations, as shown in Table II, it was observed that using the RF classifier with 50 base estimators and Gini Index for information criterion yielded the best overall result. The experiments resulted in a 6% boost in accuracy, indicating that these preprocessing methods and hyperparameter tuning were effective in improving the classifier's performance. These findings highlight the importance of careful preprocessing and tuning of machine learning models to achieve optimal results in real-time analysis. The comparison between the accuracy scores of the different ML models could be seen in Table II and the performance matrices can be seen in Table III.

TABLE II
ACCURACY SCORES OF DIFFERENT ALGORITHMS

Model	Accuracy
Random Forest with default parameters	80.75%
RF with Count-Vectorizer and TF-IDF in Preprocessing and 10 estimators and entropy for information gain	90.66%
RF with Count-Vectorizer and TF-IDF in Preprocessing and 10 estimators and Gini index for information gain	91.36%
RF with Count-Vectorizer and TF-IDF in Preprocessing and 25 estimators and Gini index for information gain	93.44%
RF with Count-Vectorizer and TF-IDF in Preprocessing and 50 estimators and Gini index for information gain	94.06%

TABLE III
PERFORMANCE MATRICES FOR OUR TUNED RANDOM FOREST CLASSIFIER

Accuracy	94.06%
Precision	94.01%
Recall	94.24%

C. Comparison of other classifiers

Other Machine Learning-based classifiers namely AdaBoost and Gradient Boost were also tried. but none of the aforementioned algorithms achieved a higher accuracy score than the tuned Random Forest classifier.

Although the performance of the Random Forest classifier with default parameters and using the BoW method for preprocessing the tweets wasn't better than the other classifier as Table IV shows. But, after some testing, it was discovered that Random Forest is considerably simpler to tune to give better accuracy than gradient boost and AdaBoost since it had typically only three parameters: number of trees and number of features to be selected at each node and the information gain criterion. Furthermore, using a random sampling of the data, each tree in a Random Forest is trained independently. The model is more robust than a single decision tree because of this randomness and it is unlikely to cause overfitting. The accuracy of the different algorithms tried can be seen in Table IV.

D. Real-Time Tweet Collection of Tweets

The program begins by logging into Twitter by inputting login information. The user is then prompted for a username for which the user wants to test for cyberbullying which is shown in Fig. 6. The website then navigates to the user profile

TABLE IV
ACCURACY SCORES OF DIFFERENT ALGORITHMS

Model	Accuracy
Vanilla Random Forest	80.75%
AdaBoost	90.12%
Gradient Boost	83.09%
Tuned Random Forest	94.06%

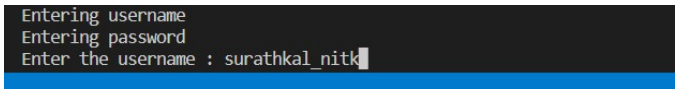

Fig. 6. Accessing the Twitter Page that needs to detect Cyberbullying

for the entered username which can be seen in Fig. 7. The tweets are extracted in real-time by the program. The extracted tweets print the text and link to the images posted and are then passed on to the ML and image captioning programs as shown in Fig. 9. The output from the ML model is then printed along

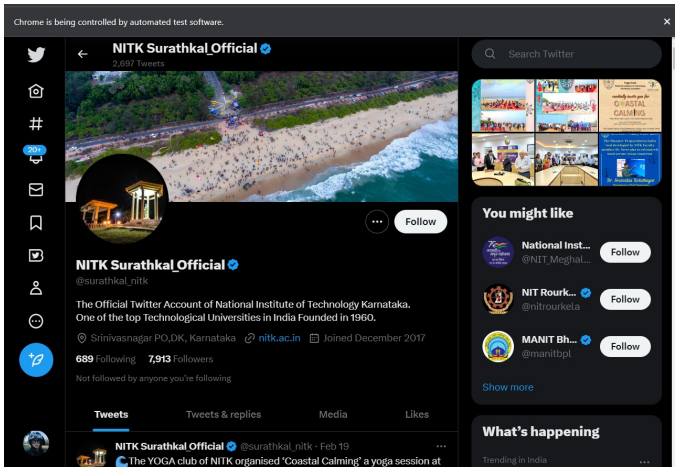

Fig. 7. Extracting Tweets from Twitter in Real Time

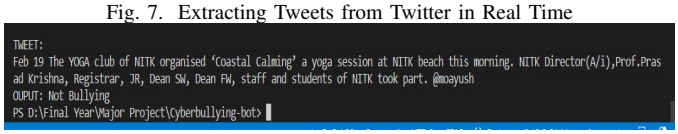

Fig. 8. Classification of Tweets

with the content of the tweets which is shown in Fig. 8.

To avoid repetitive testing of the same tweets, the data regarding the last testing of a profile was stored. Here the DateTime of the latest tweet check was stored in a CSV file. So, whenever a visit is made to any profile, the DateTime stored is checked and the tweets which were posted after that were extracted. After that, the CSV file is updated according to the latest tweet tested. For profiles not mentioned in the list, we check the profile and add them to the CSV file. This can be seen in Fig. 10.

E. Caption Generation and Similarity

The images pulled by the selenium web app were then passed to the Image Captioning Model using the Salesforce-Lavis library and the captions for each image were stored

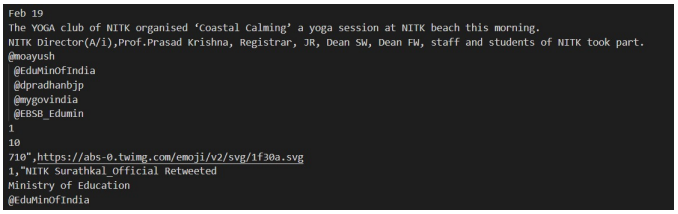

Fig. 9. Extracted Tweets

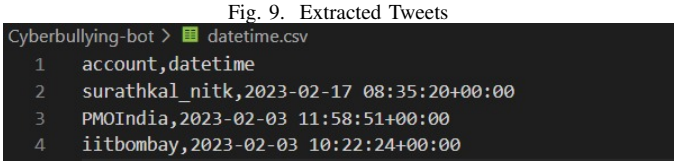

Fig. 10. Timestamp of the tweets checked

against its unique ID. Once the captions for each image is generated, we matched them with the original captions of the images provided by the user and using the BLEU similarity index We determined how similar the two captions were. The more similar the two captions are, the better the score and the less likely it is that the tweet is misleading and vice versa. For the tweets that do not have any caption, the model suggests a caption. This can be seen in Fig. 11.

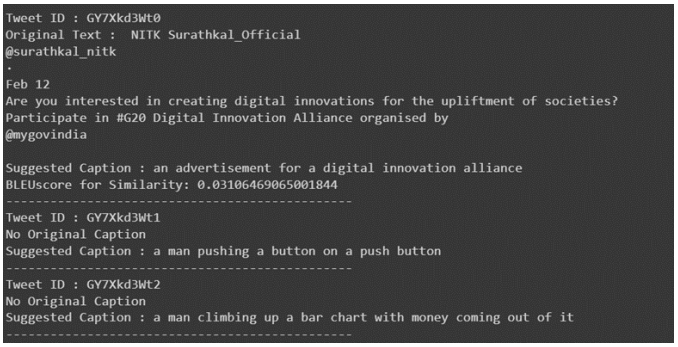

Fig. 11. Caption Generation

V. Conclusion and Future Work

Cyberbullying is recognized on a global basis as a serious problem with negative impacts on a person's health and society. There is significant scientific merit in developing a reliable cyberbullying detection model. The proposed work in this paper aimed at detecting cyberbullying in real-time tweets using machine learning techniques. The research results showed that the proposed models, which utilized Gradient Boost, AdaBoost and Random Forest could achieve a high level of accuracy in detecting cyberbullying on tweets. This work also showed that comparing images and text within tweets effectively identified spam tweets. This research lays the groundwork for future investigations on the identification of cyberbullying on social media sites. It can be expanded in a number of areas. For example, the model can be fine-tuned to improve its performance in detecting subtle forms of cyberbullying, such as sarcasm and irony.

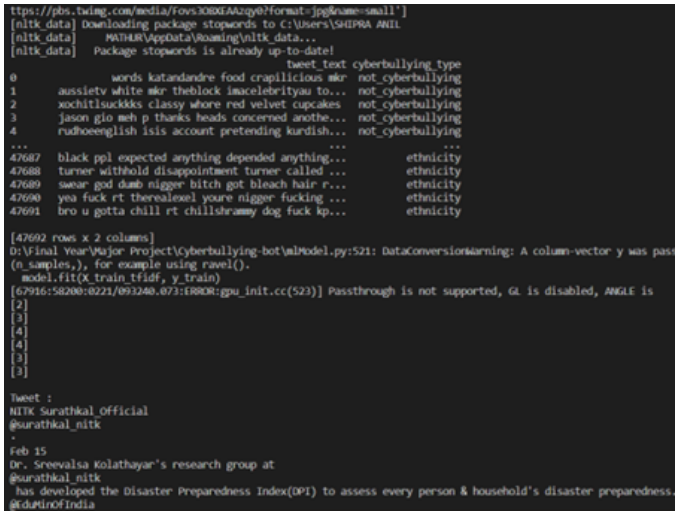

Fig. 12. Final Output

References

[1] M. M. Islam, M. A. Uddin, L. Islam, A. Akter, S. Sharmin and U. K. Acharjee, "Cyberbullying Detection on Social Networks Using Machine Learning Approaches," in 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2020, pp. 1-6, doi: 10.1109/CSDE50874.2020.9411601. https://ieeexplore.ieee.org/document/9411601

[2] Ruminati, "Tackling Cyberbullying with Computational Empathy," MIT https://www.media.mit.edu/projects/ruminati-tackling-cyberbullying-with-computational-empathy/overview/

[3] J. Yadav, D. Kumar and D. Chauhan, "Cyberbullying Detection using Pre-Trained BERT Model," in International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 1096-1100, doi: 10.1109/ICESC48915.2020.9155700. https://ieeexplore.ieee.org/document/9155700

[4] S. Agrawal and A. Awekar, "Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms," 2018.

[5] B. Sri Nandhini and J. I. Sheeba, "Cyberbullying Detection and Classification Using Information Retrieval Algorithm," in Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering amp; Technology (ICARCSET 2015) (ICARCSET '15). Association for Computing Machinery, New York, NY, USA, Article 20, 1–5, 2015. https://doi.org/10.1145/2743065.2743085

[6] X. Zhang et al., "Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network," in 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 2016, pp. 740-745, 2016, doi: 10.1109/ICMLA.2016.0132. https://ieeexplore.ieee.org/document/7838236

[7] K. Reynolds, A. Kontostathis and L. Edwards, "Using Machine Learning to Detect Cyberbullying," in 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI, USA, 2011, pp. 241-244, doi: 10.1109/ICMLA.2011.152.https://ieeexplore.ieee.org/document/6147681

[8] K. Wang, Q. Xiong, C. Wu, M. Gao and Y. Yu, "Multi-modal cyberbullying detection on social networks," in 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206663.

[9] B. Jang, M. Kim, G. Harerimana, S. Kang, J.W. Kim, "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism," Applied Sciences, 2020, 10(17):5841.

[10] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 3156-3164, doi: 10.1109/CVPR.2015.7298935.