Journal of Information Technology and Digital World (ISSN: 2582-418X)
www.irojournals.com/itdw/

Cyberbully Retarder System using Machine Learning

Abinaya SP[1], Suji N[2], Kaviya Sree M[3]

[1]Assistant Professor, Information Technology, Velammal Engineering College, Tamil Nadu, India.

[2,3]Student, Information Technology, Velammal Engineering College, Tamil Nadu, India.

E-mail: [1]abipalani0610@gmail.com, [2]sujichan2002@gmail.com, [3]kaviyasree288@gmail.com

Abstract

Covid-19 has switched almost every facet of life to online mode. Therefore, parents are forced to buy gadgets for their children for learning purposes. As a result, cyberbullying has also increased. Nowadays, youngsters get bullied online while using social media and playing online games. Everyday nearly thousands of users deal with bullying related to body shame, facial appearance, behavior, racism, sexual harassment, and other kinds of online bullying. To prevent this harassment, Machine learning algorithms are used to automatically detect the use of abusive words used by the bullies, and the developers will be notified if any type of abusive words are found and the necessary action can be taken. Moreover, a message will be sent if there is any abusive content in the chat. Therefore, the proposed method is efficient in identifying a cyber bullying activity on social media. This system will undoubtedly be useful as many students create social media accounts to keep track of their school life. Now that everything is online, this system proves beneficial in preventing cyberbullying.

Keywords: Cyberbully Detection, Machine Learning, Natural Language Processing.

1. Introduction

Social media has gained very much importance in one's life in the recent years. It is useful to know what is happening in the world. The number of social media users has increased dramatically following the coronavirus pandemic. The increased use of social media has also

DOI: https://doi.org/10.36548/jitdw.2023.2.008
Received: 01-05-2023. received in revised form: 01-06-2023, accepted: 16-06-2023, published: 26-06-2023

raised concerns about increasing online cyber bullying. Most of the existing systems can only detect whether abusive content is present or not using the available dataset, and cannot detect the abusive content from real time data. Also, the system does not detect sarcastic text as cyber bullying. The research is still ongoing, with the researchers aiming to achieve higher accuracy.

This research has proposed a solution for finding the abusive and offensive words. The language used in the code is python. By using clear indentation, its design philosophy places a strong emphasis on code readability. The object-oriented approach and language's structure are made to make it easier for programmers to develop logical code for both big and small projects. Python uses garbage collection and has dynamic typing. It supports a number of programming paradigms, including functional, object-oriented, and structured programming (particularly procedural). Apart from analyzing the data, this model automatically sends an alert notification to the registered email ID, with the bully's number, if any abusive content is found.

2. Related Works

Andrea Pereraa et al. described a method for automated cyberbullying detection and prevention that uses supervised machine learning, including support vector machines and logistic regression, along with contextual embedding models like BERT to capture evolving language patterns. However, the solution does not detect sarcastic text as cyber bullying [1]. Seunghyun kim et al., proposed a human-centered systematic assessment for automatic cyberbullying identification from the last ten years. The work highlighted the need to incorporate human-centeredness in future research to develop detecting technologies that are more useful, practical, and tailored to the various demands and circumstances of stakeholders [2].

The research [3] discussed the growth of cyberbullying on user-generated platforms and the need for tools for automated cyberbullying detection. The findings demonstrate that BERT works better than cutting-edge cyberbullying detection methods and initialized deep learning models using "slang-based word embeddings" perform better than models initialized with conventional word embeddings [3]. The study [4] offered the DEA-RNN hybrid deep learning model to identify Twitter cyberbullying. DEA-RNN outperformed conventional algorithms such as SVM, Multinomial Naive Bayes, Random Forests, and Bi-LSTM" in all scenarios [4]. LSTM and 1D-CNN are used in cyberbully detection.

The study by Sourodip Ghosh et al., projected a technique that uses a comparative analysis of 1D-CNN with LSTM architecture using text semantics to find terms that denote bullying. The following formula was used: $CE = -C\sum itilog(si)$. This model proved that in terms of accuracy, BiLSTM performs better than other models (0.9745). This model cannot be used to detect cyber bullying from real-time data [5-6].

3. Proposed Work

3.1 Overview

3.1.1 Existing System

Most of the existing systems provide only sentiment analysis and classify the words as good or bad. These systems cannot prevent online bullying. They can only detect whether there is abusive content or not, using the available dataset. Hence, they cannot be used to detect the abusive content from real time data.

Drawbacks of the Existing System

- Model can perform Natural Language Processing (NLP) on the existing data, and not on real time data.
- Model takes more time to train.
- Model cannot prevent online bullying and can only check whether abusive content is present or not.

3.1.2 Proposed System

This model not only provides detailed analysis of the past data, but can also identify abusive content from real time data. It can classify the chat dataset into 3 classes namely offensive language, hate speech, and normal text classes. Apart from analyzing the data, this model automatically sends an alert notification to the registered email ID if any abusive content is found. The bully's number is also known through the email alert sent. If the text is normal, no alert notification will be sent.

Advantages of the Proposed System

- Model can provide detailed sentiment analysis of real time data.

● Model has an accuracy of 90%.

● Model sends a trigger mail along with the bully's number if abusive content is found.

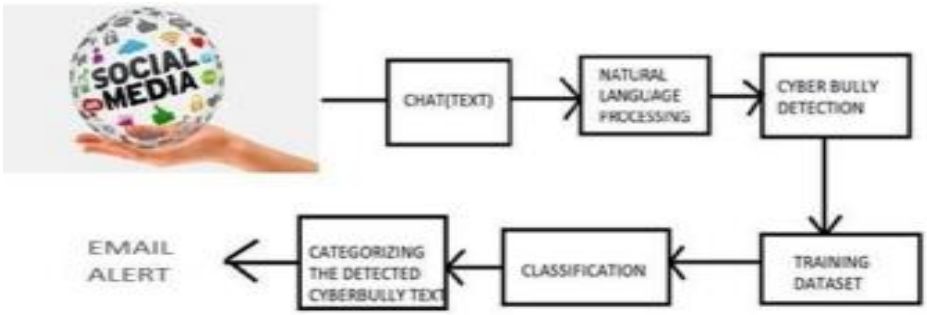


Figure 1. System Architecture Diagram

3.2 Methodology

3.2.1 Modules

- Data Collection
- Data Cleaning and Preprocessing
- Count Vectorization
- Algorithm Implementation
- Trained Model Generation
- Connecting with WhatsApp Web

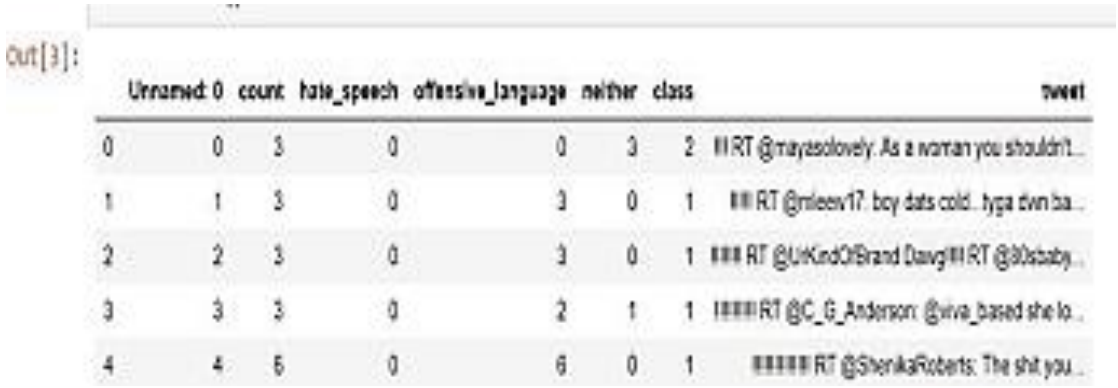
Data Collection

The real time data is used to classify whether the data is abusive or not. The dataset includes a set of random tweets of around 24,000 lines of conversation in csv format downloaded from Kaggle website. The dataset is in the csv format and hence it is easier to provide data analysis and Natural Language Processing as data from csv files are organized as rows and columns and are labelled properly.

Figure 2. Cyberbully Dataset Sample

Data Cleaning and Preprocessing

The dataset is then properly organized and transformed into the 3 classes namely hate speech, offensive language, and normal classes. Data cleaning also removes the unnecessary words and punctuation mark.



Out[3]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	tweet
0	0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't...
1	1	3	0	3	0	1	!!! RT @mleew17: boy dats cold... tyga dwn ba...
2	2	3	0	3	0	1	!!! RT @UrKindOfBrand Dawg!!! RT @80sbaby...
3	3	3	0	2	1	1	!!!!! RT @C_G_Anderson: @viva_based she lo...
4	4	6	0	6	0	1	!!!!!! RT @ShenikaRoberts: The shit you...

Figure 3. Data Cleaning and Preprocessing

After removing the unwanted punctuations, visualization and NLP are performed. The model is built using python3 and Anaconda and also the dataset is imported to the model building environment.

Data Visualization

The NLP text is then visualized in the form of bar graphs. The 3 classes are named as class 0, class 1, and class 2. The x axis denotes the class and y axis denotes the count or number of appearances of the identified word.

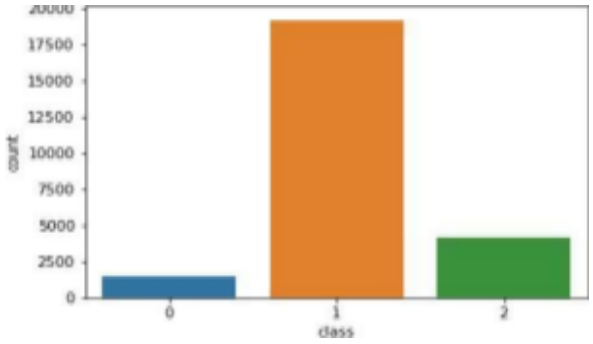


Figure 4. Data Visualization Chart

Count Vectorization

The count of each appearance of texts is noted down in Count Vectorization using a library called count vectorizer.

Extract Feature With CountVectorizer

```
In [11]: # Extract Feature With CountVectorizer
         import pandas as pd
         import pickle
         from sklearn.feature_extraction.text import CountVectorizer
         cv = CountVectorizer()
         X = cv.fit_transform(X)  # Fit the Data

In [12]: pickle.dump(cv, open("models/vectorizer_1.pickle", "wb"))
```

Figure 5. Count Vectorization

Algorithm Implementation

6 algorithms- Decision tree, Naive Bayes classifier, Support Vector Machine, Logistic Regression, K-Nearest Neighbour, and Random Forest techniques, have been implemented. Logistic Regression and SVM algorithms achieved 90% accuracy. The texts were classified, and the words were organized according to their classes- namely hate speech, offensive words, and normal text.

Logistic Regression Evaluation

```
In [30]: # Confusion matrix
         from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(Y_test, Y_pred_logreg)
         cm

Out[30]: array([[  66,  195,   25],
                [  44, 3681,  113],
                [   8,   89,  736]], dtype=int64)

In [31]: # classification report
         from sklearn.metrics import classification_report
         print(classification_report(Y_test, Y_pred_logreg))
```

	precision	recall	f1-score	support
0	0.56	0.23	0.33	286
1	0.93	0.96	0.94	3838
2	0.84	0.88	0.86	833
accuracy			0.90	4957
macro avg	0.78	0.69	0.71	4957
weighted avg	0.89	0.90	0.89	4957

Figure 6. Logistic Regression

Support Vector Machine Evaluation

```
In [28]: # Confusion matrix
         from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(Y_test, Y_pred_svc)
         cm

Out[28]: array([[  84,  184,   18],
                [  95, 3645,   98],
                [  19,   83,  731]], dtype=int64)

In [29]: # Classification report
         from sklearn.metrics import classification_report
         print(classification_report(Y_test, Y_pred_svc))
```

	precision	recall	f1-score	support
0	0.42	0.29	0.35	286
1	0.93	0.95	0.94	3838
2	0.86	0.88	0.87	833
accuracy			0.90	4957
macro avg	0.74	0.71	0.72	4957
weighted avg	0.89	0.90	0.89	4957

Figure 7. Support Vector Machine

Trained Model Generation

The dataset contains 24000 lines of conversation classified as hate speech, offensive language, and normal classes. These 3 classes are trained using SVM and Logistic Regression algorithms [14-21] to classify the words into the particular classes.

Connecting with Whatsapp Web

The software then automatically opens WhatsApp web and access can be given to the software to use WhatsApp by connecting the mobile WhatsApp to WhatsApp web using the app's QR code. By giving this access, the connected WhatsApp's chats can be read.

3.3 Requirements

The hardware specifications of the proposed model which lay as the corner stone for the software designing, are shown in the table 1 below.

Hardware Requirements

Table 1. Component Specification

COMPONENT	SPECIFICATION
PROCESSOR	Intel Core i5.
RAM	8 GB MINIMUM
GPU	INTEGRATED GRAPHICS
MONITOR	15" COLOR
HARD DISK	10 GB
PROCESSOR SPEED	MINIMUM 500MHZ

Software Requirements

The software requirements of the proposed design are listed below.

- Python 3.8
- Jupyter Notebook (Anaconda)

4. Results and Discussion

Training Data

To train a machine learning model, more data is required for a better performing model. Hence some techniques like data cleaning, pre-processing and count vectorization must be performed, so that it will be easier to perform Natural Language Processing.

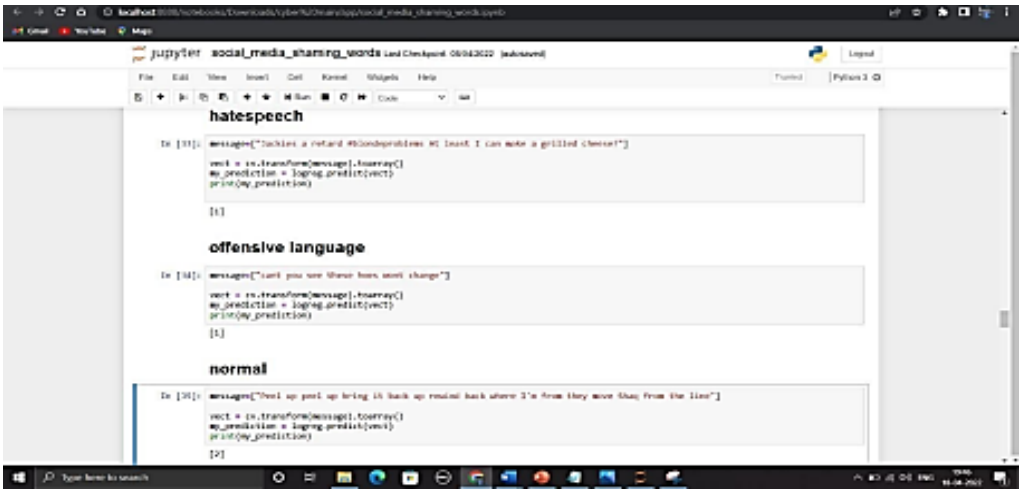


Figure 8. Trained Model Output

Final Output

The trained model is then deployed to perform NLP of real time data. The output will be as follows:

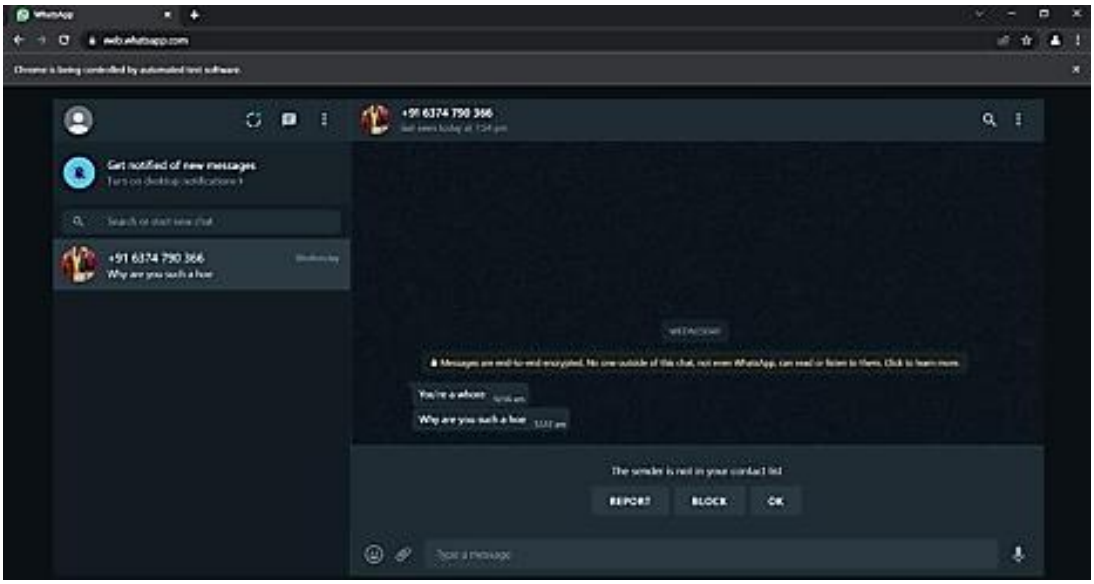


Figure 9. WhatsApp Web

If a bad word is present, then an Email alert is sent to the registered mail id.

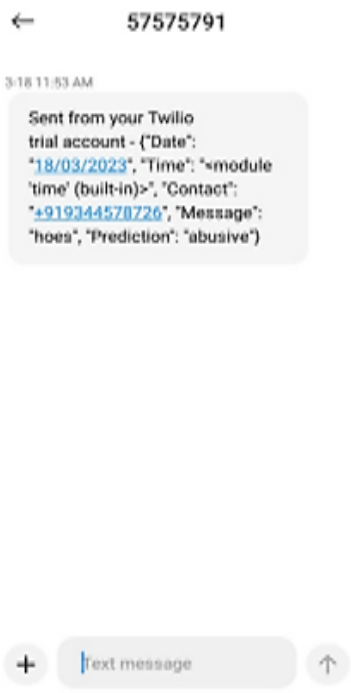


Figure 10. Message Alert

5. Conclusion

Cyberbullying is a serious issue. Due to the many hazardous implications, it has on the victim, it needs to be addressed seriously. Additionally, it upsets a person's state of mind. After

being cyberbullied, many people are known to experience despair. The proposed software successfully detected the bad words from WhatsApp web, and sent the number of the bully to the registered email ID. Parents can easily monitor their child's activity using this software and hence it will safeguard children from being harassed and bullied in WhatsApp. Cyberbullying is becoming very common nowadays with the excess use of social media platforms. Hence it is necessary to take precautions to protect youngsters from bullies.

References

[1] Balakrishna.V, Khan.S, and Arabnia.H.R (2020) "Improving cyberbullying detection using Twitter user's psychological features and machine learning", Computers & Security, vol. 90, p. 101710.

[2] Bandeh Ali Talpur, Declan O'Sullivan (2020) "Cyberbullying severity detection: A machine learning approach".

[3] Dawei Yin, Zhenzhen Xue, Liangjie Hon (2009) "Detection of Harassment on Web 2.0".

[4] Kelly Reynolds, April Kontostathis, Lynne Edwards (2011) "Using Machine Learning to Detect Cyberbullying".

[5] Manowarul Islam, Ashraf Uddin, Linta Islam, Uzzal.K.Acharjee, (2020) "Cyberbullying Detection on Social Networks Using Machine Learning Approaches" 10.1109/CSDE50874.2020.9411601.

[6] Massachusetts Institute of Technology (MIT) (2018) "Ruminati to detect harmful YouTube comments" vol. 3.

[7] Michael Fire et al, Roy Goldschmidt, Yuval Elovici (2014) "Online Social Networks: Threats and Solutions".

[8] Noviantho, Sani Muhamad Isa (2018) "Cyberbullying Classification using Text Mining" 10.1109/ICICOS.2017.8276369.

[9] P.K.Roy, Asis Kumar Tripathy, Tapan Kumar Das, Xiao-Zhi Gao (2020) "Framework for Hate Speech Detection Using Deep Convolutional Neural Network".

[10] Rachel Trana, Christopher E. Gomez, Rachel Adler (2021) "Fighting Cyberbullying: An Analysis of Algorithms Used to Detect Harassing Text Found on YouTube" 10.1007/978-3-030-51328-3.

[11] Rasel, Risul Islam & Sultana, Nasrin, Akhter, Sharna & Meesad, Phayung (2018) Sourodip Ghosh, Aunkit Chaki, Ankit Kudeshia (2021) "Cyberbully Detection Using 1D-CNN And LSTM".

[12] Sudhanshu, Rahul Ramesh Dalvi, Aparna Halbe (2020) "Detecting A Twitter Cyberbullying Using Machine Learning".

[13] Swetha Agarwal and Amit Awekar (2018) "Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms" LNISA, volume 10772.

[14] Thor Aleksander Buan and Raghavendra Ramachandra (2020), "Automated Cyberbullying Detection in Social Media Using an SVM Activated Stacked Convolution LSTM Network".

[15] Nalayini C.M., Gayathri, T. (2022). A Comparative Analysis of Standard Classifiers with CHDTC to Detect Credit Card Fraudulent Transactions. In: Sivasubramanian, A., Shastry, in Electrical Engineering, vol 792. Springer, Singapore. https://doi.org/10.1007/978-981-16-4625-6_99.

[16] C.M. Nalayini, Dr. Jeevaa Katiravan, "Detection of DDoS Attack using Machine Learning Algorithms", Journal of Emerging Technologies and Innovative Research, Volume 9, Issue 7, July 2022.

Author's Biography

S.P.Abinaya is an Assistant Professor currently working in Velammal Engineering College Chennai. She has experience as a programming faculty in Robotics, completed full stack development course and have done some projects in web development domain. Her area of specialization is Data Structure.

M.Kaviya sree is an Engineering student who is pursuing under graduation in Velammal engineering college. She is from the Information technology department. She has done internships and also participated in symposiums. She has done projects based on web

development. She got placed in Aspiring systems and mindtree and is currently doing internship at Aspiring Systems.

N.Suji is an Engineering student who is pursuing under graduation in Velammal engineering college. She is from the Information technology department. She has done internships and also participated in symposiums. She has done projects based on Data Science and Web Development. She got placed in Purchasing Power, Accenture, and CTS, and is currently doing an internship at Purchasing Power as a BI Developer.