IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

CYBERBULLYING DETECTION ON SOCIAL MEDIA USING MACHINE LEARNING

[1] Rayapureddy Sivani, [2] Sindhu Sravya Mangalapalli, [3] Sirisha Devada, [4] Tushara Kodi, [5]Preethi Sandilya Kakarla, [6]M. Sion Kumari

[1,2,3,4,5]Students, [6]Project Guide,

Computer Science and Systems Engineering (CS&SE)
Andhra University College Of Engineering For Women (AUCEW), Visakhapatnam, Andhra Pradesh, India

Abstract: Cyberbullying is a pervasive problem in the digital age that can lead to serious mental and physical harm for its victims. This is a very important and timely research topic, given the increasing prevalence of online abuse and bullying on social media. The use of natural language processing and machine learning to detect abusive messages could be an effective solution to mitigate the negative impact of online harassment. By developing a reliable and accurate technique to detect bullying text, we can potentially prevent or intervene in cases of cyberbullying before they escalate into more serious consequences. This paper proposes an approach for detecting cyberbullying on social media using the naive Bayes algorithm. The proposed method involves the use of natural language processing techniques to preprocess and extract relevant features from textual data, such as social media comments. These features are then used to train a multinomial naive Bayes classifier to classify comments as either cyber bullying or non-cyberbullying. Overall, this research has the potential to make a significant positive impact on society by addressing a critical issue in the digital age.

IndexTerms **– Cyberbullying, Machine learning, Multinomial Naïve Bayes, Natural Language Processing (NLP)**

I.INTRODUCTION

Cyberbullying is a serious problem that affects many individuals on social media platforms. It can cause depression, anxiety, and even suicide in some cases. Therefore, it is important to take measures to prevent cyberbullying and protect individuals from its harmful effects. Machine learning techniques have been used to detect and prevent cyberbullying on these platforms. In this project, we aim to develop a machine learning-based approach to detect and prevent cyberbullying on social media platforms. The approach used involves training a model on a dataset of labeled instances of cyberbullying and non-cyberbullying messages. We use natural language processing (NLP) techniques to preprocess the text data and extract relevant features. The extracted features will be used to train a classification model, which will be able to classify new messages as either cyber bullying or non-cyberbullying.
The results of this project will help to improve the safety and security of social media platforms and protect users from cyberbullying.

II. NEED OF THE STUDY.

Prevalence of cyberbullying: With the increasing use of technology and social media, cyberbullying has become a significant issue that affects individuals, especially children, and adolescents. According to a recent study, around 34% of students reported being victims of cyberbullying, which indicates the need for effective cyberbullying detection mechanisms.
Psychological impact: Cyberbullying can cause severe psychological harm to victims, leading to anxiety, depression, and even suicidal ideation. Early detection and prevention of cyberbullying can help mitigate the psychological impact and protect individuals from further harm.
Legal and ethical considerations: Cyberbullying is illegal and considered a violation of human rights. Therefore, detecting and preventing cyberbullying is not only necessary from an ethical perspective but also a legal one.
Technology advancements: The increasing use of technology and social media platforms has made cyberbullying more prevalent and sophisticated. Therefore, there is a need for advanced and effective detection mechanisms to address this issue.

Data and Sources of Data:

Data is one of the most important aspects of machine learning. The quality and quantity of data available for training and testing can have a significant impact on the performance of a machine-learning model. In addition, the type of data used (numerical, categorical, time-series, etc.) and the source of the data (databases, spreadsheets, APIs, etc.) can also affect the performance of a machine learning model. It's important to carefully select and preprocess the data before using it to train a machine learning model to ensure that the model can accurately learn patterns and relationships within the data. Here labeled data was used which is a type of data used in supervised learning where each example in the dataset has an associated target variable or label.
The dataset used was obtained from:

III. RESEARCH METHODOLOGY

Data collection:
Collect data from various sources, such as social media platforms or online forums, where cyberbullying is prevalent. The data should include both cyberbullying instances and non-bullying instances for comparison.

Data pre-processing:
Pre-process the collected data by cleaning and filtering out irrelevant information. This step may involve removing stop words, tokenizing, stemming, and lemmatization, among other techniques.

Feature extraction:
Extract relevant features from the pre-processed data that can be used to train the Naive Bayes classifier. Some features commonly used for cyberbullying detection include lexical features (e.g., profanity, aggression), syntactic features (e.g., sentence structure, grammatical errors), and semantic features (e.g., sentiment analysis, emotion detection).

Training the Naive Bayes classifier:
Use the pre-processed and feature-extracted data to train a Naive Bayes classifier. This involves splitting the data into training and testing sets, fitting the Naive Bayes model to the training set, and evaluating its performance on the testing set.

Model evaluation:
Evaluate the performance of the Naive Bayes classifier using metrics such as accuracy, precision, recall, and F1 score. These metrics can help determine the effectiveness of the classifier in detecting cyberbullying instances.

Validation and optimization:
Validate the Naive Bayes classifier using cross-validation techniques and optimize the model's hyperparameters to improve its performance. This step may involve experimenting with different feature sets, adjusting the smoothing parameter, or using other techniques to improve the model's accuracy.

Deployment:
Once the Naive Bayes classifier has been trained and validated, it can be deployed in real-world settings to detect cyberbullying instances automatically.

IV. SYSTEM ANALYSIS

HARDWARE REQUIREMENTS:
1. SYSTEM: i3 or above
2. HARD DISK: 20 GB or more
3. RAM: 8 GB or more

FUNCTIONAL REQUIREMENTS:
Load the data: Load the data from a text file that contains comments taken from social media.
Prepare the data: Split the data into training and test sets, preprocess the email text data using TfidfVectorizer to convert it into a numerical format, and vectorize it.
Train the model: Use Multinomial Naive Bayes to train the model using the training data.
Test the model: Test the accuracy of the model using the test data and calculate the F1 score using sklearn metrics.f1_score.
Save the model: Save the trained model for later use.
Predict: Use the trained model to predict whether a new comment is a bully or a non-bully comment.

NON-FUNCTIONAL REQUIREMENTS:
1. Performance
2. Scalability
3. Reliability
4. Security
5. Usability
6. Accuracy
7. Adaptability
8. Ethical considerations
9. Cost

WORKING MODEL:
The code loads data from a file containing comments which are taken from social media. It then splits the data into training and test sets, vectorizes the text using TfidfVectorizer, and trains a Naive Bayes classifier on the training set. The trained model is then saved to disk. The accuracy of the classifier is measured on the test set, and the F1 score is calculated using sklearn metrics. Finally, the user is prompted to input a comment/ message to classify using the trained model. The predicted output of the message is displayed on the console.

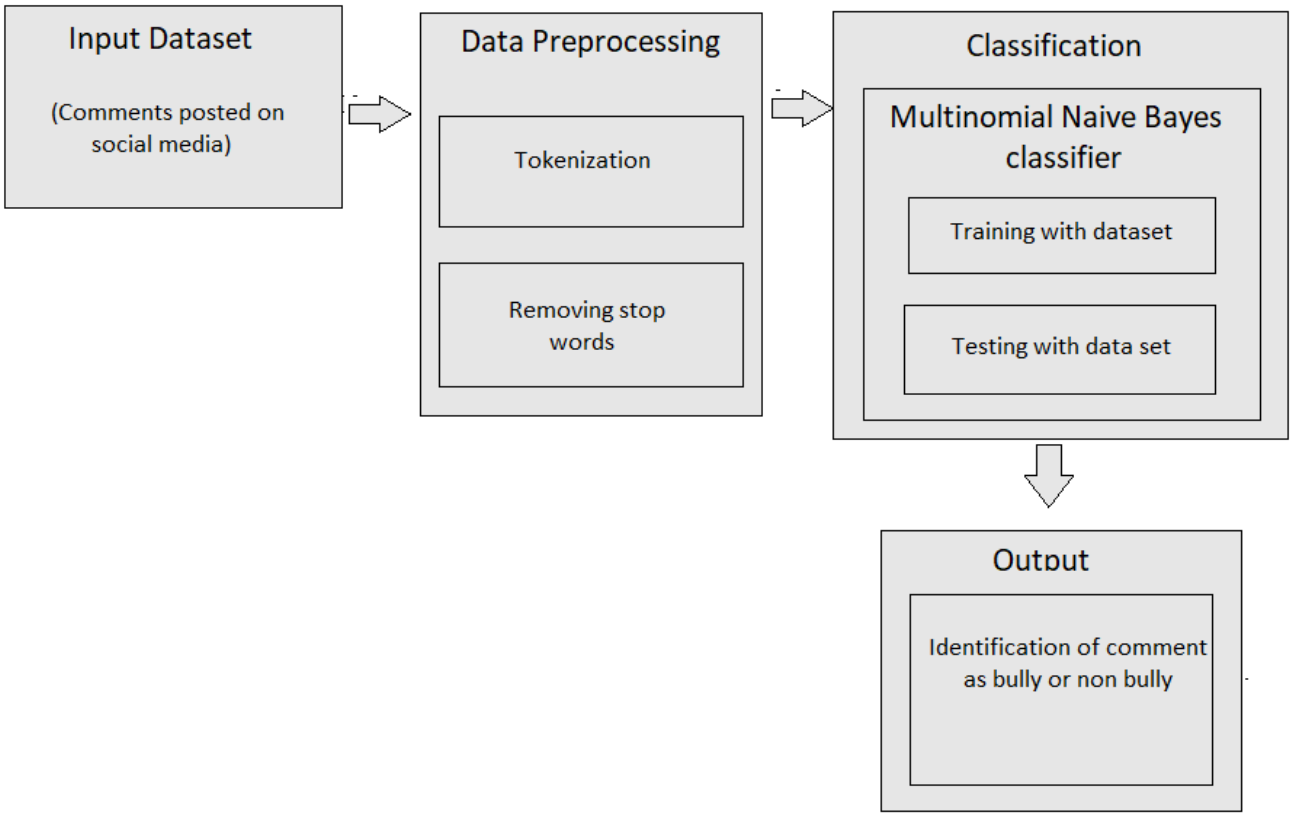


Fig: Flowchart of working model

V. ALGORITHMS AND LIBRARIES:

Multinomial Naïve Bayes:

Multinomial Naïve Bayes is a classification algorithm used in machine learning that is part of the Naive Bayes family of algorithms. It is a probabilistic algorithm that is commonly used for text classification tasks. The MultinomialNB() algorithm works by calculating the probabilities of each possible outcome (or class) given a set of input features. It does this by modeling the frequency distribution of the features within each class using a multinomial distribution. It then uses Bayes' theorem to calculate the posterior probability of each class given the input features and predicts the class with the highest probability.

Time:

In Python, the time module provides various time-related functions that allow you to work with time values and perform various operations related to time.

Sklearn:

Scikit-learn, commonly abbreviated as sklearn, is a Python library for machine learning built on top of NumPy, SciPy, and matplotlib. It provides a wide range of algorithms for supervised and unsupervised learning, including classification, regression, clustering, and dimensionality reduction.

Pickle:

In Python, the pickle module provides a way to serialize and deserialize Python objects. Serialization is the process of converting an object into a byte stream, which can be stored in a file or transmitted over a network. Deserialization is the process of converting the byte stream back into an object.

Numpy:

NumPy (Numerical Python) is a Python package that provides support for multidimensional arrays and matrices, as well as a large collection of mathematical functions to operate on these arrays. NumPy is widely used in scientific computing, data analysis, and machine learning, among other fields.

PySimpleGUI:

PySimpleGUI is a Python library for creating Graphical User Interfaces (GUIs) quickly and easily. It is designed to be simple and intuitive, making it accessible to users with little or no programming experience.

PySimpleGUI provides a high-level interface for creating GUIs, which means that users do not need to have a deep understanding of the underlying Tkinter, Qt, or WxPython libraries. Instead, PySimpleGUI uses a simpler and more consistent API that is easy to learn and use.PySimpleGUI also provides a wide range of pre-built widgets, such as buttons, checkboxes, sliders, and text boxes, which can be easily added to a GUI.

VI. SYSTEM TESTING:

System testing is an important aspect of cyberbullying detection, as it involves testing the entire system as a whole to ensure that it is functioning properly and meeting its requirements. In the context of cyberbullying detection, system testing involves evaluating the performance of the entire system, including the machine learning models, data preprocessing, feature extraction, and user interface.

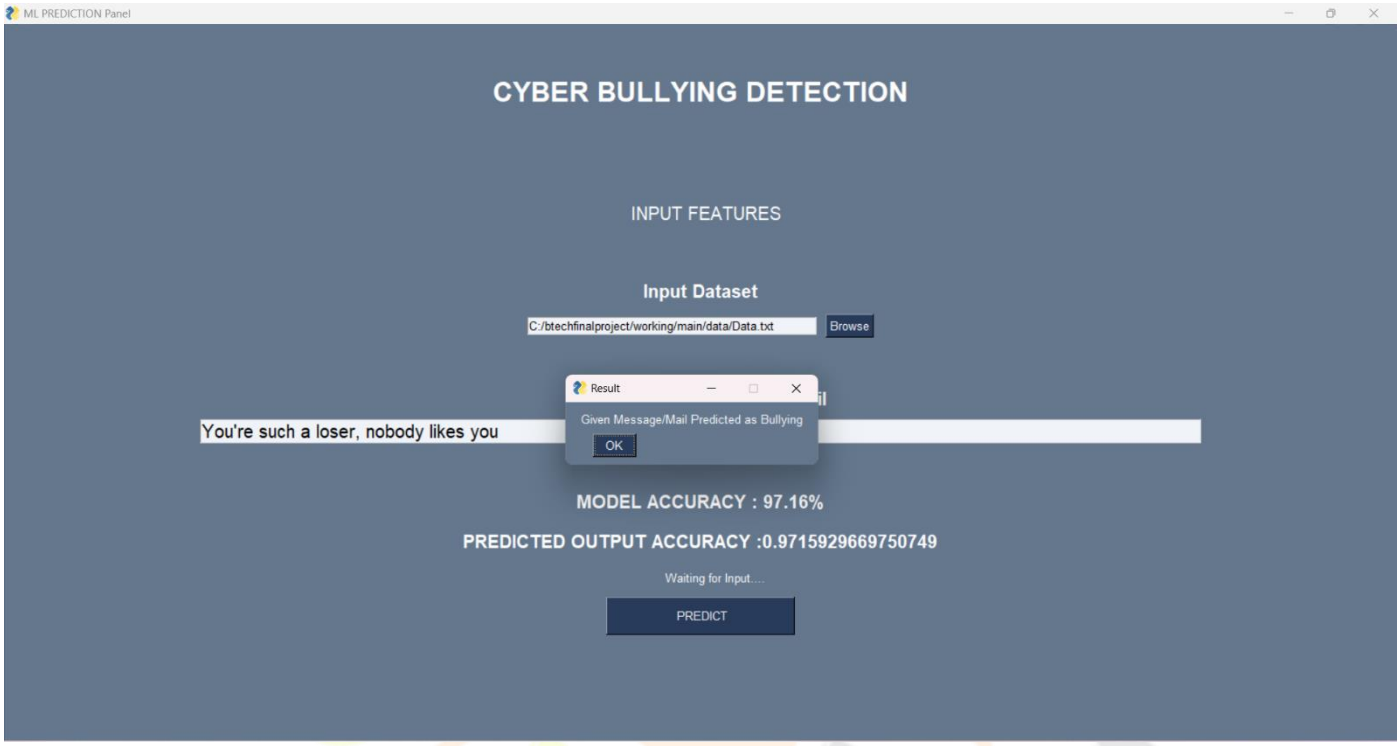
Testing with real-world data: It is important to test the system with real-world data, rather than just synthetic or simulated data, in order to ensure that the system can handle the complexity and variability of real-world scenarios.

Testing for false positives and false negatives: It is important to test the system for both false positives (i.e. instances where the system flags something as cyberbullying when it is not) and false negatives (i.e. instances where the system misses instances of cyberbullying).

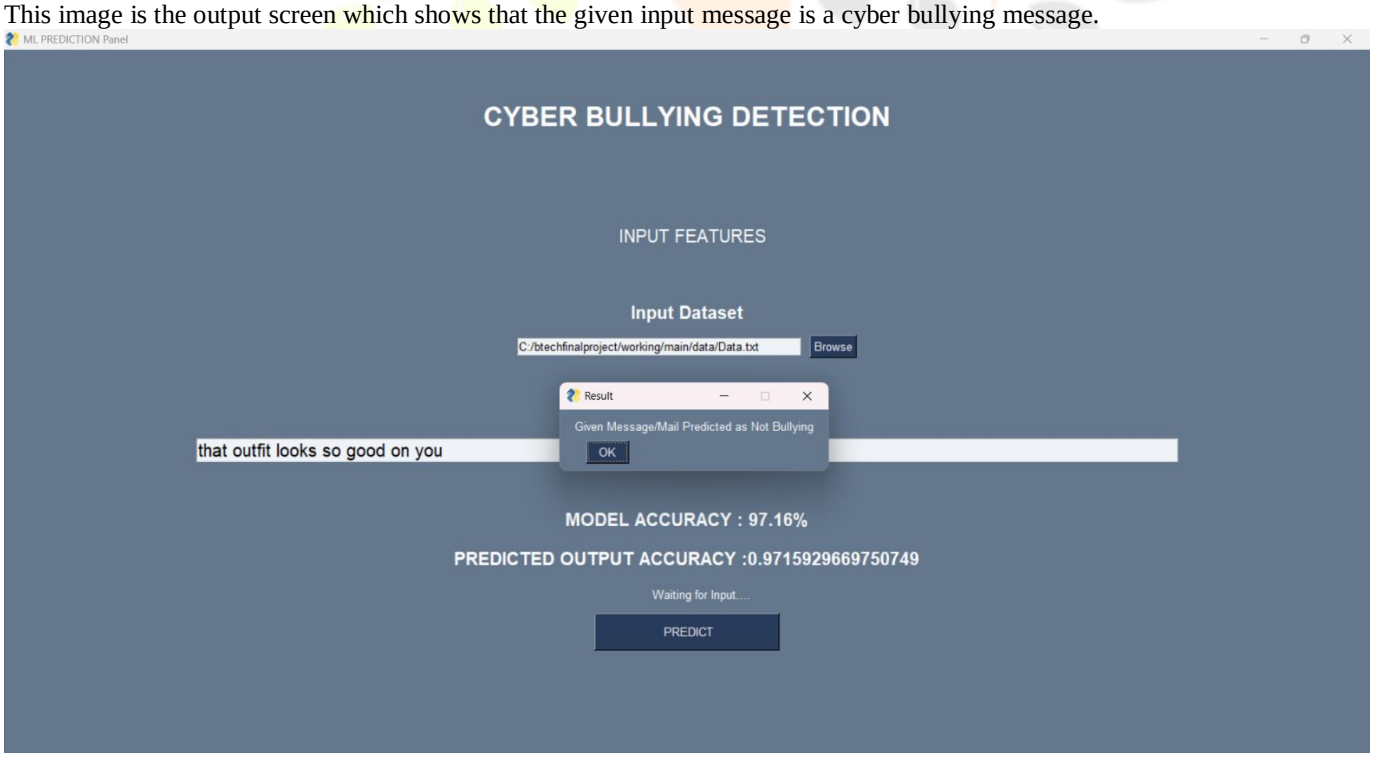
Testing for scalability: As the amount of data and users increases, it is important to test the system for scalability and performance, to ensure that it can handle the increased load.

Testing the user interface: It is important to test the user interface to ensure that it is user-friendly and easy to use, and that it provides appropriate feedback to users.

VII. RESULTS AND DISCUSSION:



This image is the output screen which shows that the given input message is a cyber bullying message.



This image is the output screen which shows that the given input message is not a cyberbullying message.

The classifier accuracy is printed as a percentage with two decimal places using the format() function. The accuracy is calculated using the score() method of the classifier object, which takes in the transformed test features and labels and returns the accuracy of the classifier on the test data.

The output accuracy is predicted using the f1-score.The f1-score () method takes the true labels and the predicted labels as inputs and returns the f1-score of the classifier on the test data.

VIII. CONCLUSION:

In conclusion, cyberbullying is a serious problem on social media that can have negative impacts on mental health and well-being. Machine learning algorithms can be used to automatically detect instances of cyberbullying in social media data, which can help to prevent and mitigate its harmful effects.

In this project, we used a dataset of social media posts labeled as either cyber bullying or non-cyberbullying to train a Multinomial Naive Bayes classifier. We used the scikit-learn library in Python to pre-process the data, extract features, and train the classifier. We then evaluated the performance of the classifier on a separate test dataset using accuracy and F1-score metrics.

We also developed a graphical user interface (GUI) using the PySimpleGUI library to allow users to input social media posts and get real-time predictions of whether or not they contain instances of cyberbullying.

Overall, our results show that machine learning algorithms can be effective in detecting cyberbullying on social media and that this approach can be implemented in a user-friendly way using a GUI. However, further research is needed to improve the accuracy and reliability of these algorithms, and to ensure that they are used in an ethical and responsible manner.

IX. ACKNOWLEDGEMENT:

Acknowledging the persistent leadership and commitment of those who made our project possible is vital for the satisfaction that comes with a successful completion. We are deeply grateful to our college administration and project advisor, Sion Kumari ma'am, for providing us with the essential tools to realize our project. The professionalism and internal support from our project guide were instrumental in turning our project into a success, and we appreciate their contributions. We want to express our gratitude to all those who supported us in any capacity to help us reach our current position. Finally, we extend our heartfelt appreciation to our family for their unwavering support and assistance throughout our academic journey, continually motivating us to complete our project.

X. REFERENCES:

[1] Varun Jain, Vishant Kumar, Vivek Pal, Dinesh Kumar Vishwakarma, “Detection of Cyberbullying on Social Media Using Machine learning”, IEEE Conference, 2021.

[2] Manowarul Islam, Md A shraf Uddin, Linta Islam, Arnisha Akhter, Selina Sharmin, Uzzal K. Acharjee,"Cyberbullying Detection on Social Networks Using Machine Learning Approaches", 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), April 2021.

[3] R. Zhang, L. Liu, Y. Chen, and C. Wang, "Cyberbullying Detection on Social Media: A Survey," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 16, no. 2, pp. 1-26, 2020.

[4] R. K. Sahoo, S. Sahoo, S. K. Rath, and S. Mishra, "Cyberbullying Detection on Social Media using Machine Learning: A Review," in Proceedings of the International Conference on Computing, Communication and Signal Processing (ICCCSP), Chennai, India, 2018, pp. 427-431.

[5] John Hani Mounir, Mohamed Nashaat, Mostafaa Ahmed, Zeyad Emad, "Social Media Cyberbullying Detection using Machine Learning", International Journal of Advanced Computer Science and Applications, January 2019.

[6] G. H. Loo, A. E. R. Shamsuddin, and A. N. Zainuddin, "Cyberbullying Detection on Social Media Using Machine Learning Techniques: A Review," in Proceedings of the International Conference on Computational Science and Technology (ICCST), Kota Kinabalu, Malaysia, 2019, pp. 1-7.

[7] A. Khurana, N. Bhatia, and A. Aggarwal, "Cyberbullying Detection on Social Media using Machine Learning Techniques: A Review," in Proceedings of the International Conference on Advances in Computing and Communication Engineering (ICACCE), Jaipur, India, 2020, pp. 1-7.

[8] Arnisha Akhter, Uzzal K. Acharjee, Md. Masbaul Alam Polash,” Cyber Bullying Detection and Classification using Multinomial Naïve Bayes and Fuzzy Logic” International Journal of Mathematical Sciences and Computing, A Machine Learning Based Approach to Detect Cyber Bullying using Fuzzy logic, November 2019.

[9] K. Reynolds, A. Kontostathis, and L. Edwards, “Using machine learning to detect cyberbullying,” in 2011 10th International Conference on Machine learning and applications and workshops, vol. 2. IEEE, 2011, pp. 241–244.

[10] DataTurks. (2018, July 12). Tweets Dataset for Detection of Cyber-Trolls. Retrieved November 07, 2020, from https://www.kaggle.com/dataturks/dataset-for-detection-ofcybertrolls?select=Dataset+for+Detection+of+Cyber-Trolls.json