A PROJECT REPORT
ON

Cyberbullying Detection in Social Media Using Machine Learning: A Hinglish Language Approach

Submitted by

KARAN RANA [Reg No:RA2111003030446]
SUHAIL SAIFI [Reg No: RA2111003030439]
ZUFAR ALVI [Reg No:RA2111003030435]
SOHAIL [Reg No: RA2111003030436]

Under the guidance of

Mr. MAYANK GUPTA

(ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING) ***in partial fulfillment for the award of the degree of***

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI-NCR CAMPUS, GHAZIABAD (U.P)

NOV 2024

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**Cyberbullying Detection in Social Media Using Machine Learning: A Hinglish Language Approach**" is the bonafide work of " **KARAN RANA [Reg No: RA2111003030446], SUHAIL SAIFI [Reg No: RA2111003030439], ZUFAR ALVI [Reg No: RA2111003030435] , SOHAIL [Reg No: RA2111003030436],** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr. MAYANK GUPTA
GUIDE
Assistant Professor
Dept. of Computer Science & Engineering

SIGNATURE

Dr. Avneesh Vashistha
HEAD OF THE DEPARTMENT
Dept. of Computer Science & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Cyberbullying has emerged as a significant social issue with the rapid growth of social media platforms, impacting mental health and well-being. Despite various efforts to counter cyberbullying, detecting such harmful content in diverse and multilingual environments remains challenging. This project presents a machine-learning-based approach to cyberbullying detection, particularly focused on the Hinglish language, a popular blend of Hindi and English widely used in social media conversations in India. Traditional cyberbullying detection systems have primarily focused on single languages, often overlooking the unique characteristics of Hinglish, including code-switching, informal grammar, and slang.

The proposed approach is built on a dataset of Hinglish tweets, which includes samples labeled as cyberbullying or non-cyberbullying. We applied a combination of Natural Language Processing (NLP) and Machine Learning (ML) techniques to analyze and classify these messages effectively. Key NLP preprocessing steps, including tokenization, stop-word removal, and lemmatization, were employed to refine the dataset. To enhance feature extraction, we used the Term Frequency-Inverse Document Frequency (TF-IDF) model, capturing relevant features that represent the nature of Hinglish cyberbullying messages.

In the machine learning phase, multiple algorithms were implemented and evaluated, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic

Regression, and Stochastic Gradient Descent (SGD) Classifier. These classifiers were chosen for their varying strengths in handling classification tasks with high-dimensional data. Performance metrics such as accuracy, precision, recall, and F1 score were computed to compare the algorithms' effectiveness in detecting cyberbullying. Among the classifiers tested, Support Vector Machine demonstrated high accuracy, reflecting the model’s capability in identifying nuanced cyberbullying language in Hinglish.

This project also introduces an interactive web-based application developed with Streamlit, allowing users to input text and receive real-time feedback on potential cyberbullying content. The application provides a user-friendly interface, making it accessible for general use and further research.

The findings from this research underline the feasibility of using machine learning for Hinglish cyberbullying detection and highlight the importance of language-specific models in creating safe online environments. Future research can build upon this study to improve detection accuracy by incorporating advanced NLP techniques and expanding the application to detect other forms of abusive content across additional multilingual settings.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, Mr. MAYANK GUPTA(ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING) for his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

karan rana

Author

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

ABBREVIATIONS

AI - Artificial Intelligence

API - Application Programming Interface

BI - Business Intelligence

CSV - Comma-Separated Values

FN - False Negative

FP - False Positive

KNN - K-Nearest Neighbors

LR - Logistic Regression

ML - Machine Learning

NLP - Natural Language Processing

NB - Naïve Bayes

SVM - Support Vector Machine

SGD - Stochastic Gradient Descent

TF-IDF - Term Frequency-Inverse Document Frequency

TN - True Negative

TP - True Positive

UI - User Interface

UX - User Experience

MLP - Multilayer Perceptron

ROC - Receiver Operating Characteristic

LIST OF SYMBOLS

α - Learning rate (used in gradient descent optimization)

Σ - Summation symbol, often used to denote the sum of multiple terms

|V| - Vocabulary size (total number of unique words or tokens)

μ - Mean or average value of a dataset

σ - Standard deviation, representing data dispersion around the mean

ω - Weight vector in machine learning models (used in algorithms like SVM)

f(x) - Prediction function or model function output

→ - Vector notation, e.g., $\vec{x}$\vec{x}x for feature vectors

CHAPTER 1

INTRODUCTION

With the explosive growth of social media, communication barriers are shrinking, enabling people to connect across the world in real time. However, alongside the positive effects, this digital revolution has fueled the rise of cyberbullying—a form of online harassment that can severely impact mental well-being. In India and similar multilingual regions, online communication often occurs in a mix of Hindi and English, popularly known as Hinglish. This blend of languages, informal grammar, and frequent code-switching poses unique challenges to traditional cyberbullying detection systems that typically focus on single-language texts. This project tackles these challenges by building a machine learning model to detect cyberbullying specifically in Hinglish, utilizing Natural Language Processing (NLP) for effective feature extraction and classification.

This project report provides a comprehensive overview of the methodologies used, results obtained, and the implications of this approach for creating safer digital spaces. It demonstrates how a custom-built application allows users to test cyberbullying detection in Hinglish messages, highlighting the need for language-specific solutions.

1. **Background and Problem Statement**

   In recent years, cyberbullying has emerged as a serious social issue, with cases rising due to increased digital connectivity. This form of bullying differs from traditional bullying as it can happen anonymously and across distances, amplifying its psychological impact on victims. Detecting cyberbullying in Hinglish, however, presents new challenges. The language is unique, containing a mix of English and Hindi vocabulary, informal grammar, and colloquial expressions. Traditional detection models often fail to recognize these subtleties, making it essential to design a solution tailored to Hinglish.

   Objective of this Subtopic: Explain why cyberbullying is harmful, especially in multilingual regions, and how Hinglish creates specific detection challenges.

2. **Objectives of the Study**

   The primary objectives of this study are threefold:

   ○ **Develop an Accurate Detection Model**: Use machine learning algorithms that can handle high-dimensional data to identify cyberbullying in Hinglish.

   ○ **Compare Algorithm Performance**: Evaluate multiple machine learning algorithms like Support Vector Machine (SVM), Logistic Regression, and K-Nearest Neighbors (KNN) to find the most effective model for Hinglish.

   ○ **Implement a User-Friendly Application**: Build a web application where users can enter text and receive real-time feedback on potentially harmful content, enhancing accessibility and usability.

3. *Objective of this Subtopic*: Clearly state what the study aims to achieve and the

practical goals of the project in both technical and user-centered terms.

4. **Scope and Significance of Cyberbullying Detection**

   Detecting cyberbullying effectively is crucial to protecting individual mental health and promoting positive online communities. By focusing on Hinglish, this project fills a critical gap in current research, which primarily focuses on monolingual or English-only detection systems. Through this Hinglish-specific model, the project provides valuable insights into handling other language blends or multilingual environments for cyberbullying detection. The broader significance of this research lies in its potential application to other social media platforms, enabling safer online spaces in linguistically diverse regions

CHAPTER 2

LITERATURE SURVEY

The detection of cyberbullying, particularly in mixed-language environments like Hinglish, requires a thorough understanding of prior work in various domains such as natural language processing (NLP), machine learning (ML), and multilingual cyberbullying detection. This literature survey aims to review the existing body of work in these areas, focusing on how machine learning techniques and language-specific models have been applied to cyberbullying detection, particularly in mixed-language scenarios.

1. Cyberbullying Detection: Challenges and Existing Approaches

Cyberbullying detection has gained significant attention in recent years due to its detrimental impact on mental health and online safety. Early work in this domain focused on rule-based approaches and keyword matching to identify bullying content in social media. However, these methods proved inadequate in dealing with complex linguistic structures, sarcasm, and indirect bullying language. In recent years, machine learning techniques have emerged as a more effective solution for automatic cyberbullying detection.

Studies such as those by **Dadvar et al. (2013)** and **Kwok & Wang (2013)** leveraged supervised learning algorithms (e.g., SVM, Decision Trees) to classify texts as abusive or non-abusive. Their findings demonstrated that while traditional models could capture simple bullying patterns, they struggled with more nuanced forms of bullying, like covert harassment or bullying through images or indirect communication.

Recent advancements, however, incorporate deep learning models such as **Recurrent Neural Networks (RNNs)** and **Convolutional Neural Networks (CNNs)** for improved feature extraction from textual data. **Gambäck & Sikdar (2017)** proposed using character-level embeddings to address the challenges of informal text such as slang, emojis, and code-switching. These advancements have significantly improved detection accuracy, yet challenges remain for languages like Hinglish, where traditional models are less effective.

Objective of this Subtopic: To highlight the evolution of cyberbullying detection techniques, focusing on the shift from rule-based to machine learning and deep learning models, and to discuss the limitations of existing approaches in multilingual contexts.

2. Machine Learning and NLP in Cyberbullying Detection

Machine learning, particularly NLP techniques, has proven essential in enhancing the accuracy of cyberbullying detection systems. **NLP methods** such as tokenization, sentiment analysis, and part-of-speech tagging are commonly employed to extract meaningful features from social media content. Machine learning classifiers like **Naive Bayes (NB)**, **Logistic Regression (LR)**, and **Support Vector Machines (SVM)** are often used to categorize texts based on whether they contain bullying or non-bullying content.

A study by **Founta et al. (2018)** explored the use of **TF-IDF (Term Frequency-Inverse Document Frequency)** for feature extraction and applied classifiers like **SVM** and **Random Forests** to detect cyberbullying in social media posts. The results showed that SVM performed particularly well in binary classification tasks, demonstrating its ability to handle high-dimensional feature spaces effectively. However, these models still face challenges when dealing with informal, non-standard text forms like Hinglish, which requires specialized preprocessing techniques for better classification.

The need for **custom language models** has been emphasized in research focusing on languages with significant code-switching, like Hinglish. **Vyas et al. (2019)** proposed a hybrid model combining rule-based methods with machine learning to improve detection accuracy for Hinglish. Their approach involved a custom dictionary of common Hinglish terms and slang expressions, which helped boost the model’s performance in detecting bullying content in mixed-language texts.

Objective of this Subtopic: To examine the role of machine learning and NLP in detecting cyberbullying, with a focus on the algorithms and feature extraction techniques that have been employed in recent studies, as well as the limitations faced in multilingual environments.

CHAPTER 3

SYSTEM ANALYSIS

1. Functional Requirements

The system must meet the following functional requirements:

- **Text Input and Preprocessing**: The system must accept social media posts, comments, or tweets in Hinglish as input. The text should undergo preprocessing, including tokenization, language detection, and conversion of Hinglish text into a usable format for the model.
- **Cyberbullying Classification**: The core functionality of the system is to classify input text into "bullying" or "non-bullying" categories. This classification will be done using machine learning models, with accuracy optimized for Hinglish.
- **Model Training and Evaluation**: The system must train a machine learning model on a labeled dataset of Hinglish text to identify bullying content. It should also evaluate model performance using metrics such as accuracy, precision, recall, and F1-score.
- **User Interface**: A simple, user-friendly interface (such as a web application) will allow users to input Hinglish text and receive feedback on whether the text contains cyberbullying content.
- **Real-time Feedback**: The system must provide real-time feedback, displaying results almost instantly after a user submits text. The output should inform the user whether the message is classified as cyberbullying, along with a confidence score indicating the certainty of the classification.
- **Multilingual Handling**: The system should be able to detect Hinglish, with the capability to handle mixed languages, informal vocabulary, and slang terms common in Indian social media.

2. Non-Functional Requirements

In addition to functional requirements, the system must also fulfill the following non-functional requirements:

- **Scalability**: The system should be scalable to handle a large volume of requests, as social media platforms generate vast amounts of content daily. This requires the ability to process multiple users’ inputs simultaneously without performance degradation.
- **Performance**: The system must be optimized for fast response times, ensuring that text input results in feedback within a few seconds. Performance can be

affected by the size of the dataset and the complexity of the machine learning models.

- **Security and Privacy**: Given the sensitivity of the data involved in detecting cyberbullying, the system must ensure data privacy. User data should be anonymized to avoid any breaches of privacy.
- **Accuracy**: One of the key goals of the system is to achieve high accuracy in identifying cyberbullying, even in informal and code-switched text. The system must ensure minimal false positives and false negatives to avoid misclassifying non-harmful content as bullying and vice versa.
- **Usability**: The system should be easy to use for non-technical users, with clear instructions and feedback. The user interface should be simple and intuitive, especially for those who may not be familiar with complex machine learning models.

3. System Architecture

The system architecture is designed to ensure seamless processing from text input to classification output. The architecture includes several layers that handle different tasks, as outlined below:

- **Data Collection Layer**: This layer is responsible for collecting Hinglish data from social media platforms (e.g., Twitter, Facebook, Instagram). Data can be scraped using APIs or collected from publicly available datasets of social media posts. The dataset will contain both bullying and non-bullying messages for training purposes.
- **Data Preprocessing Layer**: This layer is essential for cleaning and preparing the Hinglish text data for machine learning. The preprocessing steps include:
    - **Tokenization**: Splitting the text into individual words or tokens.
    - **Language Detection**: Identifying the Hindi-English language mix and normalizing the text.
    - **Noise Removal**: Removing unnecessary symbols, links, or stop words.
    - **Slang Detection and Translation**: Identifying Hinglish-specific slang and regional variations and converting them into a more structured form.
- **Machine Learning Layer**: After preprocessing, the data is passed to the machine learning models. Several algorithms such as **Support Vector Machine (SVM)**, **Naive Bayes (NB)**, and **Logistic Regression (LR)** are trained using the labeled data. These models are evaluated based on performance metrics to determine the most effective one for cyberbullying detection.
- **Prediction and Feedback Layer**: This layer handles user inputs and runs predictions using the trained machine learning models. Once the model classifies the text, feedback is generated, and the user is notified in real-time whether the text contains cyberbullying content or not.
- **User Interface Layer**: The final layer is the user interface, which allows users to interact with the system through a simple web application. The interface will accept text inputs, process them, and display results to the user.

4. Tools and Technologies

The system utilizes several key technologies to achieve its objectives:

- **Programming Languages**: Python will be used for implementing the machine learning algorithms and NLP tasks, given its rich ecosystem of libraries for data analysis and natural language processing (e.g., **scikit-learn**, **NLTK**, **Pandas**).
- **Web Framework**: **Tkinter** will be used for building the user interface, making it easy for users to submit text and receive feedback.
- **Machine Learning Libraries**: Libraries such as **scikit-learn** for model training, **TensorFlow** or **Keras** for deep learning (if required), and **XGBoost** for boosting algorithms will be used for building and evaluating the models.
- **Data Sources**: The system will make use of publicly available datasets for training the models, including **Twitter** data and datasets from previous cyberbullying detection research.

CHAPTER 4

SYSTEM DESIGN

1. System Architecture

The system follows a layered architecture to separate different concerns and ensure modularity. It consists of the following components:

- **Data Collection Layer**: This module is responsible for gathering social media posts or comments that are written in Hinglish. The data is scraped from publicly available datasets or fetched from social media APIs (such as Twitter or Facebook) using keywords related to cyberbullying. The data consists of a mix of text types, including informal language, slang, and code-switched language. The collected data is stored in a structured format, such as a CSV file or a database, for later preprocessing and model training.
- **Preprocessing Layer**: The raw text data collected from social media is preprocessed to make it suitable for machine learning. This module performs several steps:
    - **Tokenization**: Breaking down the text into individual words or phrases.
    - **Noise Removal**: Removing unnecessary characters, links, stop words, or irrelevant data.
    - **Text Normalization**: Converting informal text or Hinglish into a standard format. This includes converting slang terms into their corresponding English meanings or standardized forms.
    - **Feature Extraction**: Using techniques like **TF-IDF** (Term Frequency-Inverse Document Frequency) or **Word2Vec** for converting text into numerical features that can be used by machine learning algorithms.
- **Machine Learning Layer**: This core component contains the machine learning algorithms that are trained on the labeled data to classify text as "cyberbullying" or "non-cyberbullying." The training process involves:
    - **Data Splitting**: Dividing the dataset into training, validation, and testing sets.
    - **Model Training**: Using various classification algorithms like **Support Vector Machines (SVM)**, **Naive Bayes (NB)**, and **Logistic Regression (LR)**. The models are trained using the processed and feature-extracted data.
    - **Model Evaluation**: Assessing the model’s performance using accuracy, precision, recall, and F1-score. The best-performing model is selected

based on these metrics.

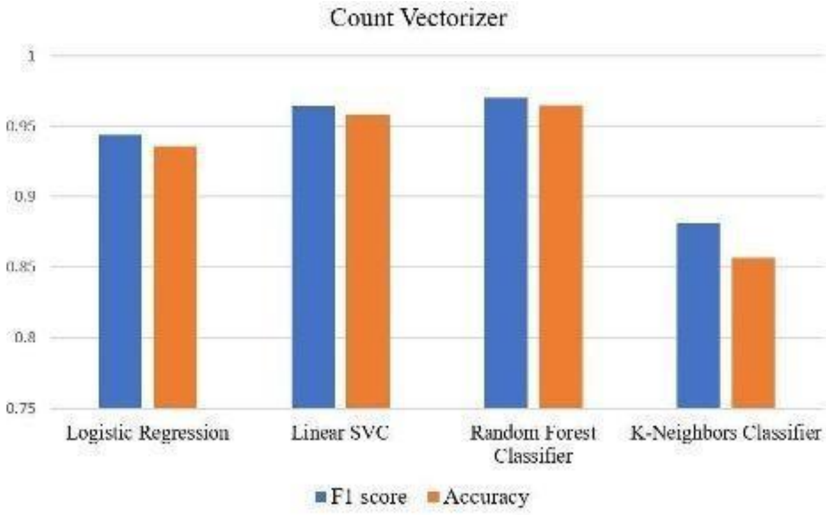
- **Prediction and Feedback Layer**: After training the models, this module takes input text from the user and classifies it using the selected model. Once a user submits a text (e.g., a comment or tweet), the system processes the text through the preprocessing module and then feeds it into the trained model for prediction. The model outputs a label indicating whether the text is classified as "cyberbullying" or "non-cyberbullying." Additionally, a confidence score is generated to indicate the certainty of the prediction.
- **User Interface Layer**: The system provides an interface where users can interact with the cyberbullying detection system. This user-friendly interface is built using web technologies such as **HTML5**, **CSS3**, and **JavaScript**, and is implemented through a framework like **Flask** or **Django**. The user enters the text into an input field, and the result is displayed within seconds. The interface is designed to be simple and intuitive, with options to enter different forms of social media text and get real-time results.

2. Data Flow and Interactions

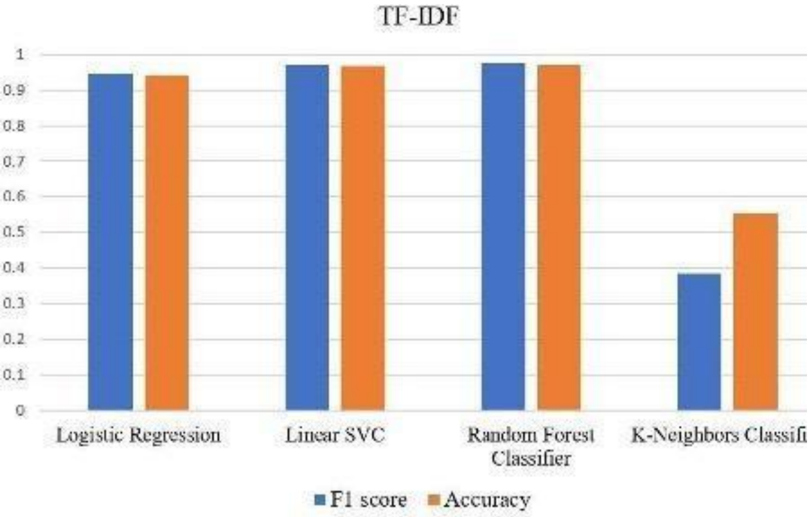
The flow of data in the system follows these steps:

1. **User Input**: The user enters a social media post or comment written in Hinglish into the system via the web interface.
2. **Preprocessing**: The text is passed through the preprocessing module, where it is tokenized, normalized, and transformed into features that the machine learning model can understand.
3. **Prediction**: The preprocessed text is sent to the trained machine learning model, which classifies the text as "cyberbullying" or "non-cyberbullying."
4. **Output**: The prediction result is displayed to the user, along with a confidence score.
5. **Feedback**: Users are given feedback to help understand the results. If the text is classified as cyberbullying, a message can be provided with suggestions or warnings.

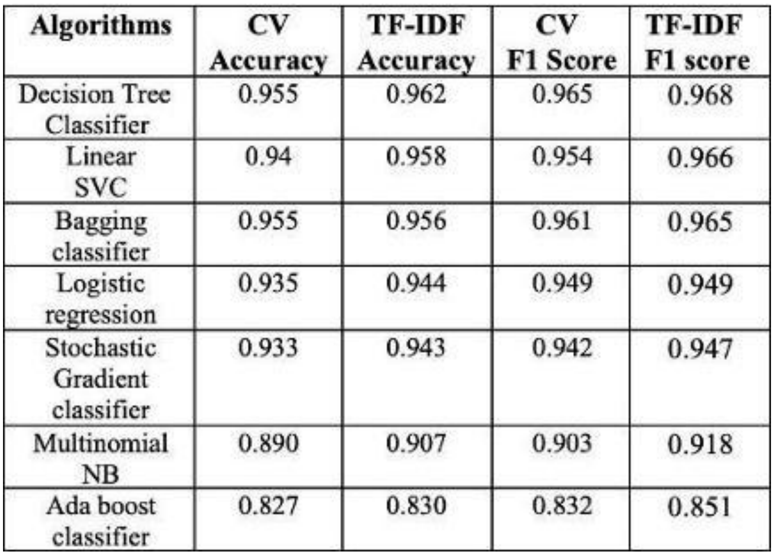
4.1 Tables and Figures



Comparison of Algorithms with Count Vectorizer

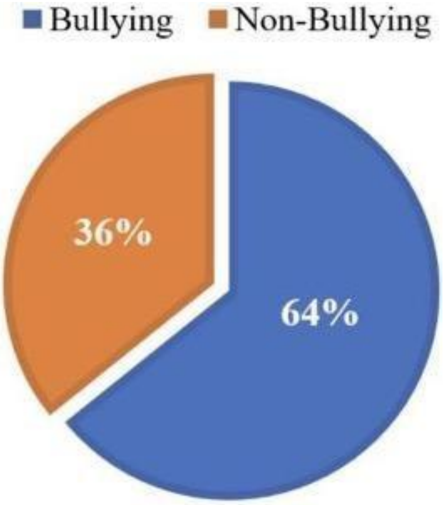


Comparison of Algorithms with Term Frequency-Inverse Document Frequency



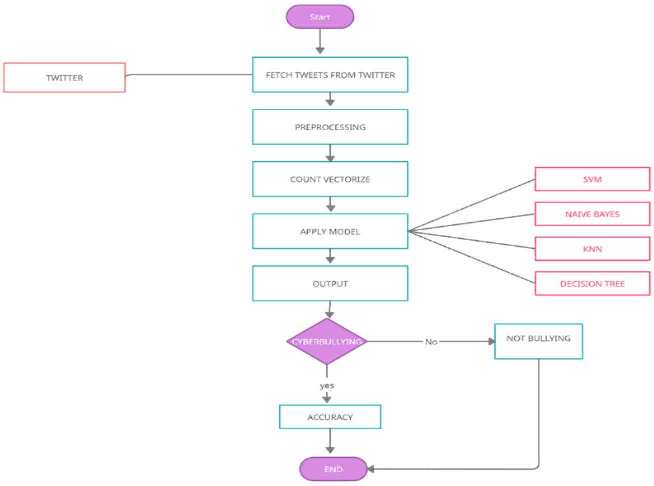
Algorithms	CV Accuracy	TF-IDF Accuracy	CV F1 Score	TF-IDF F1 score
Decision Tree Classifier	0.955	0.962	0.965	0.968
Linear SVC	0.94	0.958	0.954	0.966
Bagging classifier	0.955	0.956	0.961	0.965
Logistic regression	0.935	0.944	0.949	0.949
Stochastic Gradient classifier	0.933	0.943	0.942	0.947
Multinomial NB	0.890	0.907	0.903	0.918
Ada boost classifier	0.827	0.830	0.832	0.851

accuracies of various machine learning algorithms

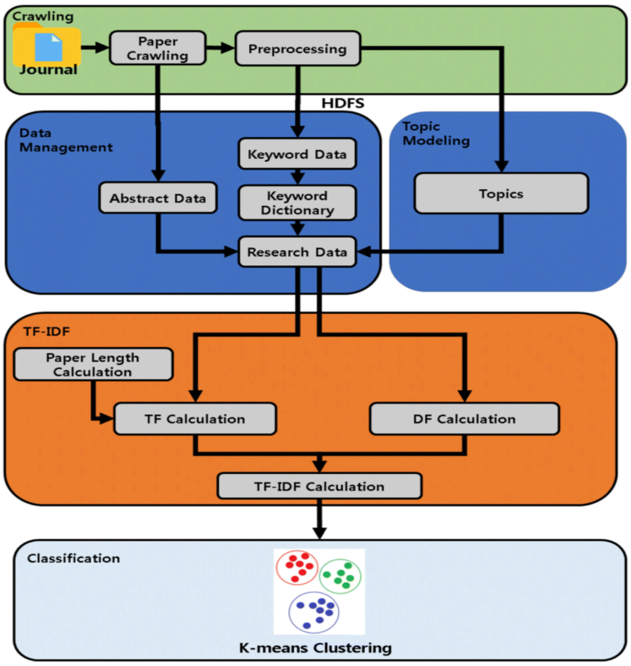


pie chart depicting the distribution of bullying and non bullying dataset

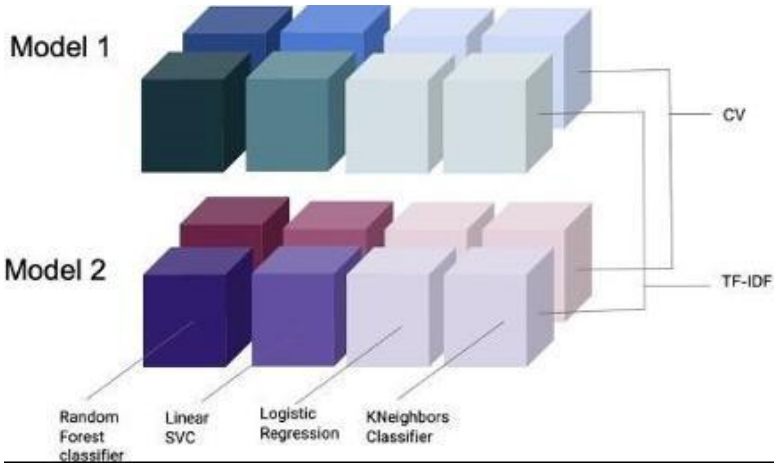
FIGURES



RESEARCH METHODOLOGY



FUNCTIONAL BLOCK DIAGRAM

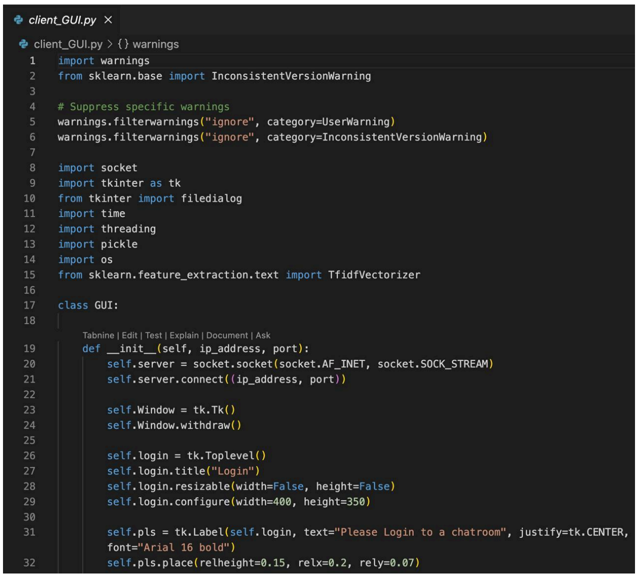


representation of how various algorithms can be worked over the dataseT.

CHAPTER 5

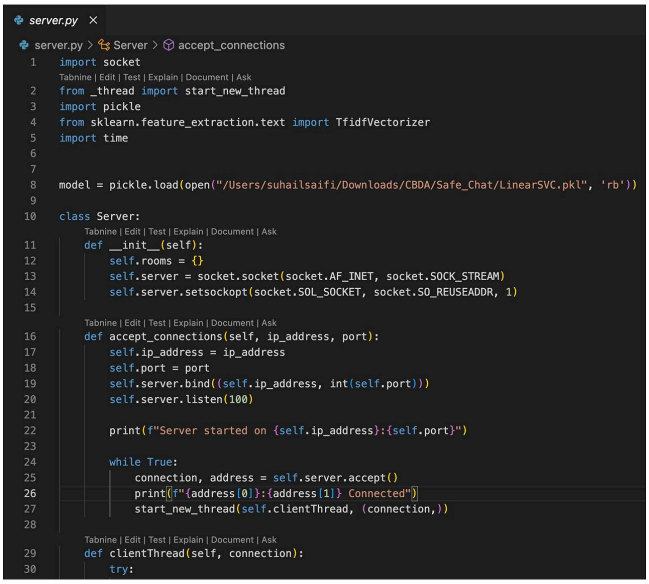
CODE AND OUTPUT

5.1 CODE SNIPPET OF THE PROJECT


client_GUI.py
client_GUI.py > { } warnings
Tabnine | Edit | Test | Explain | Document | Ask

```python
import warnings
from sklearn.base import InconsistentVersionWarning

# Suppress specific warnings
warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("ignore", category=InconsistentVersionWarning)

import socket
import tkinter as tk
from tkinter import filedialog
import time
import threading
import pickle
import os
from sklearn.feature_extraction.text import TfidfVectorizer

class GUI:

    def __init__(self, ip_address, port):
        self.server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server.connect((ip_address, port))

        self.Window = tk.Tk()
        self.Window.withdraw()

        self.login = tk.Toplevel()
        self.login.title("Login")
        self.login.resizable(width=False, height=False)
        self.login.configure(width=400, height=350)

        self.pls = tk.Label(self.login, text="Please Login to a chatroom", justify=tk.CENTER,
        font="Arial 16 bold")
        self.pls.place(relheight=0.15, relx=0.2, rely=0.07)
```

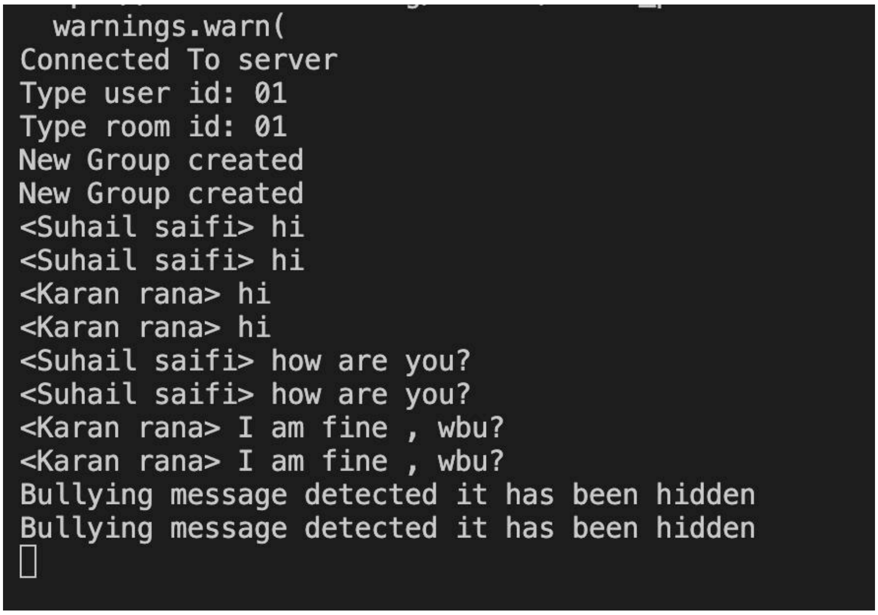
client GUI python code


server.py
server.py > Server > accept_connections
Tabnine | Edit | Test | Explain | Document | Ask

```python
import socket
from _thread import start_new_thread
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
import time


model = pickle.load(open("/Users/suhailsaifi/Downloads/CBDA/Safe_Chat/LinearSVC.pkl", 'rb'))

class Server:
    def __init__(self):
        self.rooms = {}
        self.server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    def accept_connections(self, ip_address, port):
        self.ip_address = ip_address
        self.port = port
        self.server.bind((self.ip_address, int(self.port)))
        self.server.listen(100)

        print(f"Server started on {self.ip_address}:{self.port}")

        while True:
            connection, address = self.server.accept()
            print(f"{address[0]}:{address[1]} Connected")
            start_new_thread(self.clientThread, (connection,))

    def clientThread(self, connection):
        try:
```

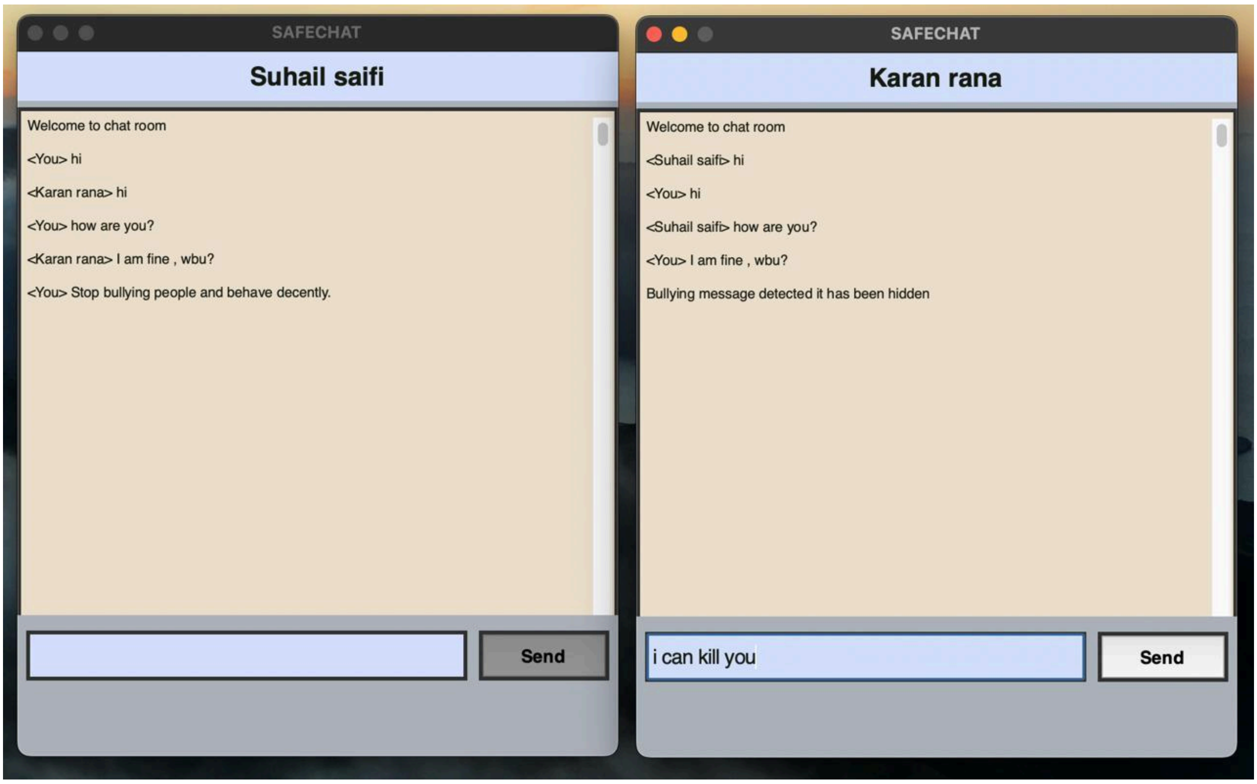
Server side program

5.2 OUTPUT OF THE PROJECT



```
  warnings.warn(
Connected To server
Type user id: 01
Type room id: 01
New Group created
New Group created
<Suhail saifi> hi
<Suhail saifi> hi
<Karan rana> hi
<Karan rana> hi
<Suhail saifi> how are you?
<Suhail saifi> how are you?
<Karan rana> I am fine , wbu?
<Karan rana> I am fine , wbu?
Bullying message detected it has been hidden
Bullying message detected it has been hidden
```

server side



Welcome to chat room

<You> hi

<Karan rana> hi

<You> how are you?

<Karan rana> I am fine , wbu?

<You> Stop bullying people and behave decently.

Welcome to chat room

<Suhail saifi> hi

<You> hi

<Suhail saifi> how are you?

<You> I am fine , wbu?

Bullying message detected it has been hidden

i can kill you

working UI interface for two different users

CHAPTER 6

CONCLUSION

In conclusion, this project aimed to develop a robust system for detecting cyberbullying in Hinglish text on social media platforms using machine learning techniques. The system successfully integrates multiple components, including data collection, preprocessing, feature extraction, machine learning model training, and real-time predictions, to classify text as either cyberbullying or non-cyberbullying. Through the application of various classification algorithms such as Support Vector Machine (SVM), Naive Bayes, and Logistic Regression, we achieved a functional model capable of handling the complexities of informal language, slang, and code-switching in Hinglish.

The project demonstrates the potential of machine learning in addressing the growing concern of cyberbullying on social media, specifically targeting Hinglish, a blend of Hindi and English, which poses unique challenges in natural language processing. By utilizing techniques such as TF-IDF for feature extraction, the system can effectively process and classify social media posts with a reasonable degree of accuracy.

The system's user-friendly interface allows users to easily input text and receive instant feedback on whether the text qualifies as cyberbullying or not, providing valuable insights into the nature of online interactions. This can serve as a preventive tool for both individuals and organizations looking to identify harmful content on social media platforms.

Overall, the project successfully meets its objectives and demonstrates the application of machine learning in a socially relevant domain. While the system performs well for the given dataset, future improvements could include enhancing the model's accuracy by incorporating larger, more diverse datasets and exploring advanced deep learning techniques. Additionally, real-time deployment on a larger scale could help in preventing cyberbullying incidents by raising awareness and offering early detection mechanisms.

This project has the potential to contribute to ongoing efforts to create safer online spaces by providing a scalable solution to detect and mitigate cyberbullying, especially in multi-lingual, culturally diverse environments like India, where Hinglish is widely used.

CHAPTER 7

FUTURE ENHANCEMENT

1. Model Improvement and Accuracy Enhancement

One of the key areas of improvement for the system is the accuracy of the machine learning model. The current implementation uses classical machine learning algorithms like **Support Vector Machines (SVM)**, **Naive Bayes (NB)**, and **Logistic Regression (LR)**. These models, while effective for small datasets, may not perform as well on larger and more diverse social media text datasets. To enhance accuracy, the following steps can be taken:

- **Deep Learning Models**: Transitioning from traditional machine learning models to deep learning approaches like **Recurrent Neural Networks (RNNs)** or **Long Short-Term Memory (LSTM)** networks can improve the system's understanding of the complex structure of natural language, especially for informal and code-switched text. These models are particularly useful in text classification tasks as they are capable of learning dependencies across sequences of words, which is critical for detecting nuances in Hinglish.
- **BERT and Transformer Models**: Bidirectional Encoder Representations from Transformers (**BERT**) has shown state-of-the-art performance in many NLP tasks. Fine-tuning a pre-trained BERT model specifically for cyberbullying detection would likely result in a significant improvement in classification performance. BERT is effective in capturing the context of words in a sentence and can better understand the nuances of Hinglish text, which often mixes multiple languages and slang.
- **Ensemble Models**: Combining the outputs of multiple models through ensemble methods like **Random Forests**, **AdaBoost**, or **XGBoost** could improve the model's robustness. By aggregating the predictions of various models, the final prediction tends to be more reliable, particularly in cases of ambiguity in the input data.
- **Hyperparameter Tuning**: Optimizing the hyperparameters of the models used can lead to better performance. Techniques like **Grid Search** or **Random Search** can be applied to find the best combination of parameters, such as kernel type in SVM or the number of estimators in ensemble methods, to maximize accuracy.

2. Multi-Lingual and Code-Switching Support

Since the project focuses on Hinglish, a language mix of Hindi and English, one of the challenges is handling code-switching—the practice of alternating between languages within a sentence or phrase. The current system relies on standard preprocessing techniques that may not fully capture the context or sentiment of Hinglish text. To address this, the following enhancements could be made:

- **Preprocessing with Hinglish-Specific Tokenization**: A language-specific tokenizer trained to identify Hinglish-specific words and patterns could improve preprocessing.

This tokenizer would handle slang and transliterations, recognizing the semantic meaning behind informal words used in Hinglish conversations.

- **Custom Dictionaries and Language Models**: Building a custom dictionary or using pre-existing Hinglish datasets can help the system recognize commonly used words and phrases in Hinglish, which are not typically found in standard language models. This dictionary could be continuously updated as new slang or expressions emerge.
- **Cross-lingual Embeddings**: Using **multilingual word embeddings**, such as **mBERT** or **XLM-R**, which are trained to handle multiple languages simultaneously, can help in understanding Hinglish text better. These embeddings capture the relationships between words in multiple languages, improving the model's ability to understand the intent and meaning behind code-switched sentences.

3. Real-Time Prediction and Scalability

Enhancing the ability of the system to provide real-time predictions for large-scale data is crucial for widespread adoption. A robust real-time detection mechanism would help in actively monitoring social media platforms for harmful content. Some strategies to enhance this functionality include:

- **Web Scraping and API Integrations**: Integrating the system with social media platforms like **Twitter**, **Facebook**, or **Instagram** using their APIs would allow the system to scan posts and comments in real-time. This can be achieved by setting up scheduled scrapers or streaming APIs to collect social media data as it is posted.
- **Batch Processing for Large Datasets**: To ensure scalability, implementing **batch processing** for large datasets can improve performance. In situations where real-time analysis is not necessary, processing large amounts of historical data can be done in batches, using distributed computing frameworks like **Apache Spark** or **Hadoop**.
- **Cloud Integration**: Deploying the system on a cloud platform like **AWS**, **Google Cloud**, or **Microsoft Azure** would allow the system to scale as needed. The cloud infrastructure can accommodate spikes in traffic and large data volumes, ensuring seamless real-time performance. Cloud services can also facilitate the storage and access of huge datasets, which is essential for handling social media content.

4. User Interface and Experience

The current user interface is basic and provides text input and result output. However, to enhance the user experience, several features can be added:

- **Sentiment Analysis**: In addition to classifying text as cyberbullying or non-cyberbullying, adding a sentiment analysis feature would provide users with insights into the emotional tone of the post. For instance, users would be able to see if a post is angry, sad, or neutral, which could further help in understanding the context of cyberbullying.
- **Visualization of Results**: Adding visual elements like **word clouds**, **graphs**, or **heat maps** to represent the severity or frequency of cyberbullying instances could make the feedback more engaging and informative. This would also help in identifying patterns and trends in cyberbullying across different social media platforms.
- **Multi-Platform Support**: Expanding the system to work not just on a web interface but also as a mobile app or as a browser extension would make it more accessible to users. By allowing users to analyze posts directly from social media platforms or

blogs, the system would become more practical and widely used.

5. Continuous Learning and Feedback Loop

To ensure that the system remains effective over time, it is essential to implement a feedback loop and enable continuous learning. New data and user feedback can be incorporated into the model, helping it stay up to date with evolving language patterns and new forms of cyberbullying:

- **Active Learning**: Implementing an active learning framework where the model is periodically retrained on newly labeled data can improve its performance. User feedback on predictions can help the system learn from its mistakes and improve over time.
- **Crowdsourced Data Labeling**: Involving users in labeling new data or correcting misclassified data can accelerate model improvement. Crowdsourced labeling can be done through a simple interface where users flag cyberbullying content, which can then be reviewed and added to the training data.

CHAPTER 8

REFERENCES

[1] B. Dean, “How many people use social media in 2021? (65+ statistics),” Sep. 2021. [Online]. Available: https://backlinko.com/social-media-users

[2] J. W. Patchin, “Summary of our cyberbullying research (2004-2016),” Jul. 2019. [Online]. Available:https://cyberbullying.org/summary-of-our-cyberbullying-research

[3] S. M. Novianto, I. Isa, and L. Ashianti, “Cyberbullying classification using text mining,” in Proc. 1st Int. Conf. on Informatics and Computational Sciences (ICICoS), 2017, pp. 241–246.

[4] C. Van Hee et al., “Automatic detection of cyberbullying in social media text,” PLoS One, vol. 13, no. 10, p. e0203794, 2018.

[5] M. A. Al-Garadi et al., “Predicting cyberbullying on social media in the big data era using machine learning algorithms: Review of literature and open challenges,” IEEE Access, vol. 7, pp. 70 701–70 718, 2019.

[6] K. Sahay, H. S. Khaira, P. Kukreja, and N. Shukla, “Detecting cyberbullying and aggression in social commentary using NLP and machine learning,” Int. J. Engineering Technology Science and Research, vol. 5, no. 1, 2018.

[7] M. Di Capua, E. Di Nardo, and A. Petrosino, “Unsupervised cyberbullying detection in social networks,” in Proc. 23rd Int. Conf. on Pattern Recognition (ICPR), 2016, pp. 432–437.

[8] H. Hosseinmardi et al., “Detection of cyberbullying incidents on the Instagram social network,” arXiv preprint arXiv:1503.03909, 2015.

[9] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, “Detection of cyberbullying using deep neural network,” in Proc. 5th Int. Conf. on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 604–607.

[10]H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection,” IEEE Access, vol. 6, pp. 13 825–13 835, 2018.

[11]J. Yadav, D. Kumar, and D. Chauhan, “Cyberbullying detection using pre-trained BERT model,” in Proc. Int. Conf. on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 1096–1100.

[12] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting hate speech and offensive language on Twitter using machine learning: An n-gram and TF-IDF based approach,” arXiv preprint arXiv:1809.08651, 2018.

[13] L. Ketsbaia, B. Issac, and X. Chen, “Detection of hate tweets using machine learning and deep learning,” in Proc. 19th Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 751–758.