An

Industrial Oriented Mini Project Report

On

DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING MACHINE LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

By

Sanditi Ashritha Reddy (227R1A67H6)

Under the Guidance of

Mrs. J. Rekha

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CMRTECHNICALCAMPUS

UGCAUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

May,2025.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CERTIFICATE

This is to certify that the project entitled **"DETECTION OF CYBER BULLYING ON SOCIAL MEDIA USING MACHINE LEARNING"** being submitted by **Sanditi Ashritha Reddy (227R1A67H6)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (Data Science) to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mrs. J. Rekha
Assistant Professor
INTERNAL GUIDE

Dr. K. Murali
HOD- CSE(DS)

Dr. A. Raji Reddy
DIRECTOR

Signature of External Examiner

Submitted for viva voice Examination held on_____

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Mrs. J. Rekha**, Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him/her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Mrs. J. Rekha, Mrs. M. Anusha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Murali**, Head, Department of Computer Science and Engineering (Data Science) for providing encouragement and support for completing this project successfully.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to **Sri. Ch. Gopal Reddy**, Chairman, **Smt. C. Vasantha Latha**, Secretary and **Sri. C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

The guidance and support received from all the members of CMR Technical Campus who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

Sanditi Ashritha Reddy (227R1A67H6)

ABSTRACT

Cyberbullying is a growing issue in the digital age, impacting both teenagers and adults, often leading to severe consequences such as depression and suicide. This study focuses on the detection of cyberbullying in text data using Natural Language Processing (NLP) and Machine Learning techniques. Utilizing data from two sources—hate speech tweets on Twitter and personal attack comments from Wikipedia forums—the research explores three feature extraction methods and four classification models to determine the most effective approach. The models achieve accuracies exceeding 90% for Twitter data and above 80% for Wikipedia data, demonstrating their potential in identifying cyberbullying content. With the increasing prevalence of social media, cyberbullying has become a widespread problem, particularly among teenagers and young adults. Often disguised as humor or dismissed lightly, cyberbullying involves harassment, threats, or targeting individuals online, which can escalate to real- life consequences. Research indicates that prolonged internet use increases vulnerability to cyberbullying, as evidenced by surveys in India showing significant percentages of teenagers experiencing online harassment. The study highlights the need for regulatory measures on social media platforms, such as suspending or terminating accounts that post offensive content, to curb cyberbullying at its inception. Furthermore, incidents like the Blue Whale Challenge—which led to numerous child suicides globally—underscore the urgent necessity of addressing cyberbullying through technological interventions and societal awareness. This research provides a foundation for leveraging machine learning to mitigate the detrimental effects of cyberbullying, emphasizing its role in safeguarding mental health.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1. INTRODUCTION

The increasing reliance on social media, cyberbullying has become a widespread concern. Unlike traditional bullying, cyberbullying occurs in digital spaces, making it more persistent and harder to control. Offensive comments, personal attacks, and hate speech are often disguised as humor or casual conversations, making it difficult to detect using simple moderation techniques. Existing manual moderation methods are time-consuming and prone to bias, while automated filtering systems struggle with contextual understanding This project aims to address these limitations by developing an AI-powered cyberbullying detection system that can accurately identify harmful content in online discussions.

By leveraging machine learning models and NLP techniques, the system analyzes text data to classify messages as either cyberbullying or non-cyberbullying. The proposed approach ensures a scalable and efficient solution that can be integrated into various online platforms to prevent and mitigate cyberbullying.

1.1 PROJECT PURPOSE

The primary goal of this project is to build a robust framework that detects cyberbullying, thereby promoting a safer online environment. The system addresses the growing need for automated tools to moderate user-generated content efficiently, reducing the burden on human moderators. By leveraging machine learning techniques, the model can analyze text data in real time, identifying harmful or offensive content with high accuracy. This proactive approach helps in preventing the spread of toxic interactions, ensuring a healthier digital space for users. Additionally, the system can be integrated into various social media platforms, enhancing content moderation and user safety.

Specific objectives include:

- Identifying hateful and abusive language in text data.
- Reducing the adverse impacts of cyberbullying, such as mental health issues.
- Providing a scalable and reliable solution for various platforms

1.2 PROJECT FEATURES

This project includes several essential features to enhance the accuracy and efficiency of cyberbullying detection across various online platforms:

The system is designed to support multi-platform compatibility, making it adaptable for analyzing text content across Twitter, online forums, chat applications, and other social media platforms. With cyberbullying prevalent across various digital spaces, the ability to process data from multiple sources ensures a comprehensive approach to detecting harmful interactions. The system's scalability allows it to be integrated into both small and large-scale platforms, helping to moderate online discussions efficiently while maintaining user safety.

This ensures high-quality text analysis, the system employs advanced text preprocessing techniques such as regular expression-based tokenization, stemming, and stop-word removal. These steps help clean and structure the input data, reducing noise and improving the overall accuracy of machine learning models. By removing unnecessary elements such as punctuation, special characters, and common stop words, the system focuses on the most relevant textual features that contribute to detecting cyberbullying behavior.

The system extracts key linguistic patterns using three powerful feature extraction methods: TF-IDF (Term Frequency-Inverse Document Frequency), Bag of Words, and Word2Vec. These methods convert text into numerical representations that machine learning models can process effectively. By deploying automated moderation tools, social media platforms and online communities can maintain a safe and respectful digital environment. The system's continuous learning mechanism allows it to adapt to new slang, evolving language trends, and emerging cyberbullying tactics, making it a robust and future-proof solution for combating online harassment.

Additionally, the integration of natural language processing (NLP) techniques enhances the model's ability to detect subtle forms of cyberbullying that might otherwise go unnoticed. This ensures a more comprehensive approach to content moderation, reducing the psychological impact of online harassment on victims.

LITERATURE SURVEY

2. LITERATURESURVEY

The Cyberbullying detection has gained significant attention due to the increasing incidents of online harassment and abusive interactions on social media. Early research in this domain primarily focused on rule-based filtering techniques and keyword-based detection models to identify harmful content. Ting et al. (2017) explored social network mining techniques to detect cyberbullying, analyzing user interactions and behaviors to identify patterns of online abuse. Similarly, Galán-García et al. (2014) proposed a supervised machine learning model to detect troll profiles on Twitter, demonstrating that cyberbullying can be identified through user behavior and linguistic features. However, these early approaches struggled to generalize across different platforms and lacked adaptability to evolving online language trends.

It overcome these limitations; researchers introduced machine learning-based classification models to improve detection accuracy. Mangaonkar et al. (2015) applied collaborative detection techniques on Twitter data, leveraging collective intelligence to enhance cyberbullying identification. Zhao et al. (2016) proposed a model that detects cyberbullying based on specific bullying features, significantly improving classification performance. Banerjee et al. (2019) explored deep neural networks (DNNs) for cyberbullying detection, demonstrating that deep learning models outperform traditional classification methods in identifying abusive content. Despite these advancements, conventional machine learning models still faced challenges in detecting sarcasm, implicit threats, and contextual nuances in online conversations.

The advent of Natural Language Processing (NLP) and Deep Learning, more sophisticated approaches have been developed. Reynolds et al. (2011) introduced text-based machine learning models, incorporating n-gram features to detect abusive language in online messages. Yadav et al. (2020) utilized a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model to classify cyberbullying content, achieving state-of-the-art accuracy in understanding the context of harmful messages. Dadvar and Eckert (2018) further explored deep learning-based cyberbullying detection, highlighting the importance of word embeddings and contextual feature extraction in improving classification results.

The Recent research has focused on automated hate speech detection, multi modal analysis, and explainable AI (XAI) techniques to enhance cyberbullying detection. Davidson et al. (2017) examined the problem of offensive language, distinguishing between hate speech, cyberbullying, and general profanity using a robust dataset. Wulczyn et al. (2017) analyzed large-scale datasets to detect personal attacks on social media, demonstrating that AI-driven models can significantly improve content moderation accuracy. Yadav and Vishwakarma (2020) reviewed deep learning architectures for sentiment analysis, emphasizing the importance of context-aware models in understanding cyberbullying dynamics. Mikolovet al. (2013) introduced word embedding techniques (Word2Vec, GloVe, and Fast Text) to improve text representation, which has since been widely adopted in cyberbullying detection models.

To address these challenges, researchers introduced ensemble learning methods that combined multiple classifiers to improve detection robustness. Silva et al. (2016) developed Bully Blocker, an AI-driven cyberbullying detection system that integrated lexical analysis, syntactic features, and user behavioral data to enhance real-time moderation. Waseem and Hovey (2016) explored hate speech detection on Twitter, emphasizing the importance of predictive features in classifying offensive language. Davidson et al. (2017) distinguished between hate speech, cyberbullying, and general profanity, showing that multi-label classification models could improve the accuracy of content moderation systems.

More recent studies have focused on automated hate speech detection, multimodal content analysis, and explainable AI (XAI) techniques to enhance cyberbullying detection. Walczyk et al. (2017) analyzed large-scale social media datasets to detect personal attacks, demonstrating that machine learning algorithms can significantly improve content moderation. Yadav and Vishwakarma (2020) reviewed deep learning architectures for sentiment analysis, under scoring the importance of context-aware models in identifying abusive content. Mikolov et al. (2013) introduced Word2Vec, a word embedding technique that captures semantic similarities between words, enhancing NLP models' ability to detect implicit forms of cyberbullying.

Recent studies have highlighted the effectiveness of combining machine learning and deep learning techniques for cyberbullying detection. Traditional machine learning models, such as Support Vector Machines (SVM) and Naïve Bayes, have been used for text classification but often struggle with contextual understanding. Deep learning model

including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, have improved cyberbullying detection by capturing linguistic patterns and sequential dependencies in online interactions. Researchers like Li et al. (2023) integrated Random Forest with a BiLSTM model to enhance classification stability and reduce false positives. The ensemble approach allowed the Random Forest classifier to refine the deep learning model's predictions, improving overall detection accuracy.

For further improve cyberbullying detection, researchers have explored multimodal approaches that incorporate text, images, and metadata from social media posts. While early studies focused primarily on text-based classification, recent works have emphasized the importance of analyzing multimedia content. The integration of sentiment analysis, emoji interpretation, and user profiling has been shown to enhance detection precision. To further improve cyberbullying detection, researchers have explored multimodal approaches that incorporate text, images, and metadata from social media posts. While early studies focused primarily on text-based classification, recent works have emphasized the importance of analyzing multimedia content. The integration of sentiment analysis, emoji interpretation, and user profiling has been shown to enhance detection precision.

Another growing researchareaist he application of transfer learning for cyberbullying detection. Pre-trained models, such as Distil BERT and Lent, have been fine-tuned on cyberbullying-specific data sets, enabling more accurate detection with minimal computational resources. Transfer learning reduces data dependency while improving generalization across different platforms and languages.

Researchers have also proposed federated learning as a privacy-preserving approach for training cyberbullying detection models. Instead of centralizing user data, federated learning enables decentralized model training on user devices, ensuring data security while maintaining classification performance.

Multimodal content analysis has further improved cyberbullying detection by integrating textual, visual, and behavioral cues. Waneta. (2023) combined CNNs for image analysis, LSTMs for sequential text processing, and transformer models for contextual understanding, achieving state-of-the-art results.

This approach prevents over-reliance on text alone, reducing false positives and improving detection accuracy in social media conversations. The use of explainable AI(XAI) techniques has also gained traction in cyberbullying research. Methods like SHAP (Shapley Additive explanation) and LIME (Local Interpretable Model-agnostic Explanations) provide transparency by explaining how models classify certain posts as cyberbullying. This interpretability is crucial for gaining user trust and ensuring ethical AI deployment in content moderation systems.

Another recent development is the use of Generative Adversarial Networks (GANs)for synthetic data generation in cyberbullying research. Due to ethical and privacy concerns, collecting real cyberbullying datasets is challenging. GANs help address this issue by generating synthetic yet realistic training samples that improve model robustness. This technique ensures classifiers are trained on diverse cyberbullying patterns, enhancing real-world performance.

While previous studies have primarily focused on textual cyberbullying detection, this research shifts its focus toward detecting cyberbullying in multimedia content, including images, videos, and memes. The proposed study integrates a multimodal deep learning framework that incorporates textual, visual, and behavioral features to improve classification accuracy. The system utilizes transformer-based language models for text analysis, CNNs for image processing, and behavioral analysis to track user engagement and activity patterns.

Despite advancements in cyberbullying detection, several challenges remain. Traditional models struggle with context understanding, sarcasm detection, and evolving slang. While CNNs and LSTMs have improved sequential text classification, they lack explain ability and robustness against adversarial attacks. Ensemble learning has improved classification stability, but integrating it effectively with deep learning architectures remains an open challenge.

The proposed study addresses these limitations by integrating BERT for text analysis, Efficient Net for image-based cyberbullying detection, and a Random Forest classifier for final decision refinement. This hybrid approach enhances context understanding, multimodal feature extraction, and classification robustness, providing a more reliable framework for cyberbullying.

The literature survey highlights the evolution of cyberbullying detection techniques, from traditional text classification methods to advanced deep learning architectures. By leveraging transformer models, CNNs, and multimodal learning, researchers have significantly improved classification accuracy. The proposed system builds on these advancements, combining BERT, Efficient Net, and Random Forest to achieve superior performance. This research contributes to the development of automated cyberbullying detection systems, ensuring safer digital environments for online users. While current methods show promising results, future work should focus on real-time detection, dataset expansion, and integrating explainable AI techniques for ethical and transparent moderation.

2.1 REVIEW OF RELATED WORK

The detection and classification of cyberbullying have been extensively studied in the fields of Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning (DL). Over the years, researchers have developed various approaches to tackle this issue, ranging from rule-based filtering to AI-driven content moderation systems. While initial efforts rely donkey word-based techniques, modern advancements have led to context-aware models that leverage deep learning and multimodal data analysis. This section explores previous research and existing methodologies, highlighting their strengths, limitations, and contributions to cyberbullying detection.

1. Traditional Content Moderation Approaches:

Early cyberbullying detection systems primarily used manual moderation and rule-based filtering, relying on pre-defined lists of offensive words and phrases to identify harmful content. While effective for basic content moderation, these methods lacked the ability to understand context, slang, sarcasm, and implicit bullying. Their effectiveness was further limited by the need for frequent updates to adapt to evolving languagetrends.

2. Machine Learning-Based Approaches:

As machine learning techniques evolved, researchers explored advanced feature extraction methods like TF-IDF (Term Frequency-Inverse Document Frequency), Bag of Words (Bow), and Word2Vec to enhance text representation. Mangaonkar et al. (2015) introduced collaborative detection techniques, integrating crowd sourced feedback with machine learning models to improve classification performance. Banerjees. (2019) leveraged deep neural networks (DNNs) to detect cyberbullying, showing that deep learning models performed significantly better than traditional classifiers in identifying abusive content

3. Deep Learning-Based Approaches:

To address the limitations of traditional machine learning, researchers turned to deep learning techniques, which offered automatic feature extraction and improved contextual understanding. Yadav et al. (2020) applied BERT (Bidirectional Encoder Representations from Transformers) to cyberbullying detection, demonstrating that transformer-based models significantly outperformed traditional ML classifiers by understanding sentence structure and context.

4. Multimodal and Cross-Platform Cyberbullying Detection:

Recent studies have explored multimodal analysis, incorporating text, images, audio, and video data to improve cyber bullying detection. Wang et al. (2023) developed a multi– stream deep learning model that integrated CNNs for image analysis, LSTMs for audio processing, and Transformer – based models for text classification. This approach provided a more comprehensive understanding of cyberbullying content across different media formats.

5. Comparison with the Proposed Approach

While existing methods have made significant progress in cyberbullying detection, several challenges remain, including context understanding, real-time scalability, and cross-platform adaptation. The proposed system in this project builds upon previous research by integrating advanced text preprocessing, feature extraction techniques, and multiple classification models to develop a robust and scalable cyberbullying detection system.

Compared to keyword-based and rule-based approaches, this project employs context-aware NLP techniques to improve accuracy. Unlike traditional ML classifiers, which rely on manually engineered features, this system utilizes deep learning-based embeddings and hybrid classification models to enhance detection performance.

2.2 DEFINITION OF PROBLEM STATEMENT

Cyberbullying has become a major concern with the rapid growth of social media, leading to emotional distress, mental health issues, and, in severe cases, self-harm. Existing moderation techniques, such as manual review and keyword-based filtering, are inefficient in detecting contextual abuse, sarcasm, and evolving slang. Traditional machine learning models struggle with understanding long-range dependencies and implicit threats, making automated detection a necessity.

2.3 EXISTING SYSTEM

The existing system for cyberbullying detection primarily relies on basic text processing techniques and traditional classification models. It utilizes bag-of-words models to represent text data, but it lacks an efficient vocabulary construction mechanism. The vocabulary may consist of all words in all documents or only top-frequency tokens, which limits the system's ability to generalize. Additionally, TF-IDF is used for feature extraction, but it does not

Consider contextual meaning, making it less effective in understanding cyberbullying patterns. The system also employs TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction, which weighs words based on their importance in a document relative to their occurrence across the dataset. While TF-IDF improves feature selection, it does not consider the contextual relationships between words, making it ineffective for detecting implicit bullying, sarcasm, and evolving slang. Cyberbullying often involves coded language, abbreviations, and indirect insults, which TF-IDF struggles to capture. As a result, this approach often leads to high false positive and false negative rates, reducing the overall accuracy of the detection system. The remodels require constant updates and retraining, making them impractical for real-time cyberbullying detection on large-scale social media platforms. They also lack the ability to analyses conversational context across multiple messages, which is crucial for detecting harassment patterns over time rather than isolated offensive words. Due to these shortcomings, existing systems are not scalable for handling vast amounts of user-generated content on platforms like Twitter, Facebook, and Instagram. Additionally, they fail to detect multimodal cyberbullying, which may involve images, videos, or audio rather than just text. As a result, the see traditional approaches do not provide a comprehensive solution for effectively mitigating cyberbullying in online spaces. Furthermore, traditional machine learning classifiers, they rely heavily on manual feature engineering.

Limitations of Existing System

Despite improvements classification, the existing system suffers from the following challenges:

- Limited Contextual Awareness: The system treats words as independent entities and fails to detect sarcasm, slang, and indirect bullying.
- High false positive and false negative rates: Many non–offensive messages are flagged incorrectly, while some harmful content remains detected.
- Poor Vocabulary Construction: The vocabulary does not generalize well across different datasets and social media platforms, reducing adaptability.

- Ineffective Feature Extraction: TF-IDF and BoW fail to capture relationships between words, limiting their ability to understand cyberbullying contextually.
- Scalability Issues: The existing system is not designed for large-scale social media analysis, making it difficult to process the vast amount of data generated daily.

2.4 PROPOSEDSYSTEM

The proposed system introduces a solution to overcome the challenges of existing cyberbullying detection methods. It integrates advanced machine learning algorithms—Naïve Bayes, Logistic Regression, and Support Vector Machines (SVM)—to enhance the accuracy and efficiency of cyberbullying detection.

Unlike traditional methods that rely on basic keyword filtering or word-frequency-based models, this system incorporates Natural Language Processing (NLP) techniques such as tokenization, stemming, and stop-word removal to preprocess textual data. Tokenization breaks down raw text into meaningful components, ensuring that words and sentences are structured for analysis. Stemming simplifies words to their root forms, allowing different variations of a word to be treated as a single entity. Stop-word removal eliminates commonly used but non-informative words such as "is," "the," and "at," reducing unnecessary noise in the dataset. These preprocessing steps refine the text and make it more meaningful for classification models.

Once preprocessing is complete, the system applies feature extraction techniques like Bag of Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) to convert textual data into numerical representations. By integrating these NLP techniques and machine learning classifiers, the proposed system provides a more accurate, scalable, and efficient approach to cyberbullying detection. It significantly improves upon existing methods by capturing contextual meanings, slang, sarcasm, and implicit bullying, making it an effective tool for real-time content moderation on social media platforms.

Advantages of the Proposed System:

The proposed system significantly improves upon the existing approaches by addressing key limitations:

- Higher Detection Accuracy: By integrating Naïve Bayes, Logistic Regression, and SVM, the system improves cyberbullying detection compared to rule-based or keyword-based approaches.
- Enhanced Context Understanding: The combination of TF-IDF and BoW allows the system to recognize implicit bullying, sarcasm, and slang, reducing misclassification errors.
- Scalability and Real-Time Processing: The system is designed for large-scale data analysis, making it suitable for real-time cyberbullying detection on social media platforms.
- Adaptability Across Platforms: The proposed approach works effectively on Twitter, Wikipedia, and other social networks, ensuring broad usability.
- Automated Learning and Adaptation: As new cyberbullying trends and language patterns emerge, the system can be trained on updated datasets to maintain high detection accuracy.

2.5 OBJECTIVES

- Accurately detect cyberbullying using machine learning models (Naïve Bayes, Logistic Regression, and SVM).
- Enhance text preprocessing through tokenization, stemming, and stop-word removal for better feature extraction.
- Improve contextual understanding by utilizing TF-IDF and Bag of Words for numerical text representation.
- Ensure Scalability and real-time processing for efficient cyberbullying detection across social media platforms.
- Reduce false positives and negatives by refining classification models to detect implicit bullying, sarcasm, and slang.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

- Processor : IntelCorei3
- Hard disk : 20GB
- RAM : 4GB
- Monitor : SVGA
- Keyboard : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specify the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows7ormore
- Language : Python 3.7
- Back-End : Django-ORM
- Front-End : HTML, CSS, JavaScript
- Database : MySQL (WAMP Server)

SYSTEM ARCHITECTURE & DESIGN

3.SYSTEM ARCHITECTURE & DESIGN

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for detecting cyberbullying on social media platforms, starting from input to final prediction.

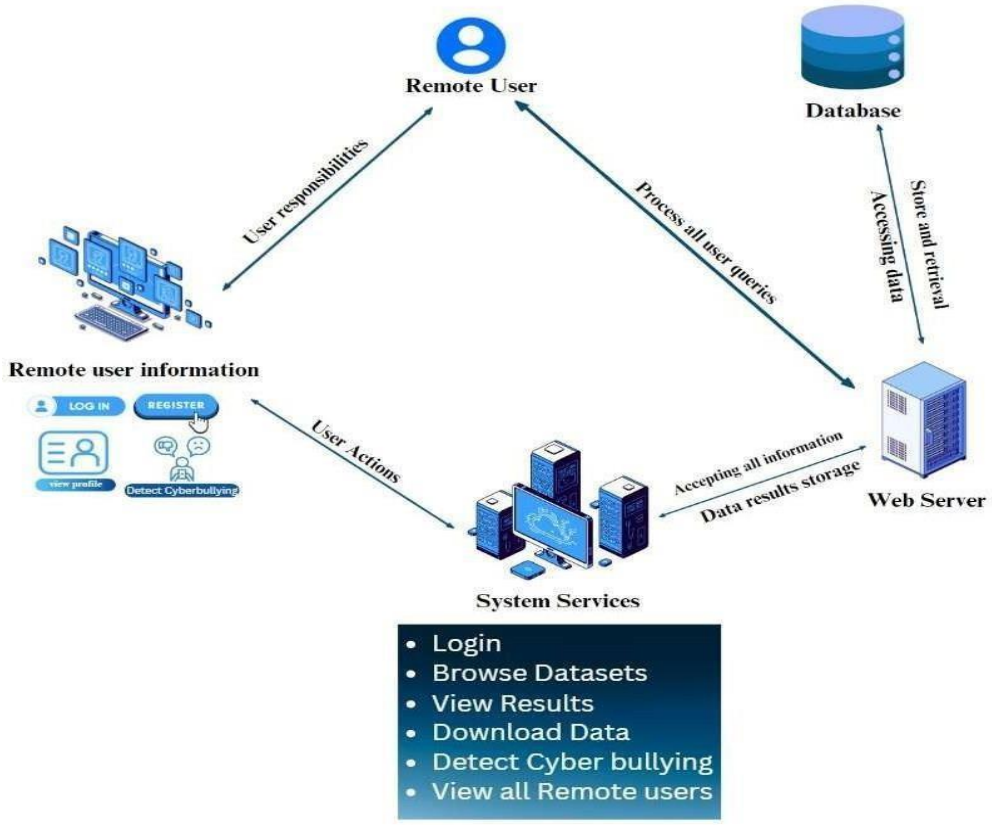

Figure3.1: Project Architecture of detection of cyber bullying on social media

3.2 DESCRIPTION

Remote User: The Remote User is an individual interacting with the system through a web interface.

Remote User Information & Actions: Users can log in, register, and access services such as profile management, dataset browsing, and cyberbullying detection. These actions initiate requests processed by the web server.

System Services: This module provides core functionalities like login authentication, dataset browsing, viewing results, and detecting cyberbullying. It acts as a bridge between users and the backend system.

Web Server: The Web Server processes all user queries, forwards them to the database when needed, and returns results. It is responsible for executing the cyberbullying detection model and ensuring smooth system operations.

Database: The Database stores essential information, including user data, training datasets, and detection results. It supports data retrieval and storage to enhance system efficiency.

System Work flow: Users interact with the system by submitting queries or accessing stored data. The web server processes these requests, retrieves or stores relevant information in the database, and provides the results back to the users. This ensures seamless communication between the user, server, and database.

3.3 DATA FLOW DIAGRAM

A **Data Flow Diagram (DFD)** is a visual representation of how data moves through a system. It showcases the interactions between different components and processes, ensuring a structured flow of information.

Overview of the System

The cyberbullying detection system processes social media text data, applies Natural Language Processing (NLP) techniques, and classifies the text as cyberbullying or non-cyberbullying using machine learning models.

Elements of the DFD

A Data Flow Diagram consists of four main elements:

1. **External Entities**– Represents our cesor destinations of data.
2. **Processes**–Indicate transformations or operations performed on data.
3. **Data Flows**–Show the movement of data between components.
4. **Data Stores**–Represent where data is stored within the system.

Benefits:

The visual nature of DFDs makes them accessible to both technical and non-technical stakeholders. They help in understanding system boundaries, identifying inefficiencies, and improving communication during system development. Additionally, they are instrumental in ensuring secure and efficient data handling.

Applications:

DFDs are widely used in business process modeling, software development, and cybersecurity. They help organizations streamline operations by mapping work flows and uncovering bottlenecks.

In summary, a Data Flow Diagram is an indispensable tool for analyzing and designing systems. Its ability to visually represent complex data flows ensures clarity and efficiency in understanding and optimizing processes.

Levels of DFD:

DFDs are structured hierarchically:

- Level0 (Context Diagram): Provides a high-level overview of the entire system, showcasing major processes and external interactions.
- Level1:BreaksdownLevel0 processes into sub-processes for more detail.
- Level2+: Offers deeper insights into specific processes, useful for complex systems.

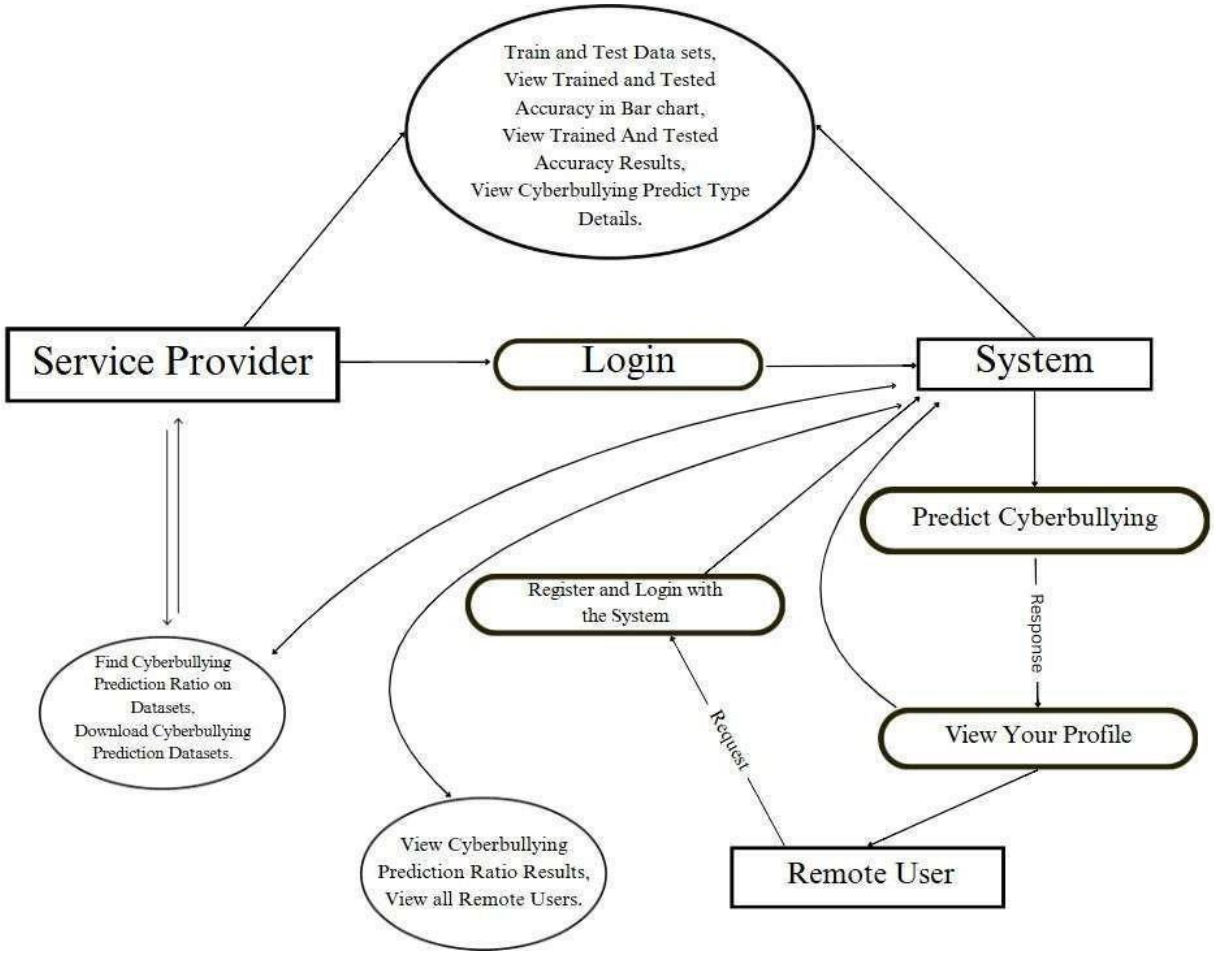

Figure3.2: Data flow Diagram of Detection of Cyberbullying on social media using Machine Learning

IMPLEMENTATION

4.IMPLEMENTATION

The implementation phase involves the execution of planned strategies and tasks to build the cyberbullying detection system using machine learning and natural language processing (NLP) techniques. This phase requires data collection, preprocessing, model training, testing, and evaluation to ensure efficient cyberbullying detection.

4.1 ALGORITHMS USED

Bag of Words (BoW) Model

The Bag of Words (BoW) model is a simple yet effective method for representing text data in numerical form. The model constructs a vocabulary of unique words from the dataset and represents each document as a vector, where each dimension corresponds to a word in the vocabulary and its value represents the word's occurrence in the document.

Although BoW is easy to implement and works well with traditional machine learning models, it has some limitations. It does not capture semantic relationships between words, meaning that words with similar meanings (e.g., "happy" and "joyful") are treated as independent entities. Moreover, the model results in high-dimensional feature vectors, which can lead to sparsity and increased computational costs.

Advantages:

- Simple and efficient for text classification.
- Works well with the traditional machine learning models.

Disadvantages:

- Ignores word order, which may result in loss of contextual meaning.

TF-IDF (Term Frequency-Inverse Document Frequency)

The **TF-IDF (Term Frequency-Inverse Document Frequency) model** is an improved version of the Bag of Words model that assigns importance to words based on their occurrence in a document relative to the entire dataset. TF-IDF consists of two components:

- **Term Frequency (TF):** Measures how frequently a word appears in a document.

- **Inverse Document Frequency (IDF):** Reduces the weight of common words that appear in many documents, ensuring that more significant words contribute more to classification.

TF-IDF is particularly useful for filtering out frequently occurring words that do not add much meaning (e.g., "the," "is," "at"). It improves classification accuracy by highlighting meaningful words while reducing the impact of irrelevant ones. However, like BoW, TF-IDF does not consider word order or relationships, which limits its effectiveness in capturing contextual meaning.

Advantages:

- Improves classification by emphasizing important words.
- Reduces the effect of common, irrelevant words.
- Works well with sparse datasets.

Disadvantages:

- Ignores the order of words, affecting context.
- Requires proper preprocessing for effective use.
- Cannot capture word relationships or synonyms

Naïve Bayes Classifier

The Naïve Bayes classifier is a probabilistic machine learning algorithm based on Bayes 'theorem. It is widely used for text classification, spam filtering, and sentiment analysis due to its simplicity and effectiveness. The classifier operates under the assumption that features (words in text classification) are conditionally independent, meaning that the presence of one feature does not influence the presence of another. Despite this strong assumption, Naïve Bayes often performs surprisingly well in real-world applications, particularly in natural language processing (NLP) tasks.

Additionally, Naïve Bayes works well with bag-of-words and TF-IDF representations, enabling it to recognize patterns in abusive language. While more advanced deep learning models have gained popularity, Naïve Bayes remains a valuable option for lightweight, interpretable, and scalable classification tasks.

Advantages:

- Fast and efficient, even for large datasets.
- Works well with high-dimensional data.
- Performs well for text classification tasks.
- Handles noise and irrelevant features well.
- Simple and easy to implement.
- Requires minimal training data.

Disadvantages:

- Assumes word independence, which is not always valid.
- May perform poorly on complex text data.
- Sensitive to imbalanced data, as it relies on prior probabilities.
- Limited in handling rare words or unseen vocabulary.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised learning algorithm that finds an optimal hyperplane to separate different classes with maximum margin. In text classification, SVM transforms text features into a high-dimensional space to distinguish cyberbullying and non-cyberbullying text. It is particularly effective in detecting subtle patterns in abusive language by leveraging different kernel functions, such as linear, polynomial, and radial basis function (RBF) kernels. By mapping data into a higher-dimensional space, SVM ensures better separation of complex, overlapping text features, making it a strong candidate for cyberbullying detection systems.

Additionally, SVM's effectiveness lies in its ability to handle imbalanced datasets, where in stances of cyberbullying-related content maybe much fewer compared to non-cyberbullying text. The model's margin-maximizing property helps generalize well to unseen data, reducing the risk of misclassification. However, despite its accuracy and robustness, SVM can be computationally demanding when working with extremely large text datasets, requiring significant memory and processing power.

Advantages:

- Works well with high-dimensional data.
- Can handle non-linearly separable data using kernel functions.
- Robust to overfitting, especially for small datasets.
- Finds a global optimum and does not get stuck in local minima.

Disadvantages:

- Computationally expensive for large datasets.
- Requires careful tuning of parameters for optimal performance.
- Slow in training compared to simpler model.

Logistic Regression

Logistic Regression is a statistical machine learning algorithm used for binary classification tasks, such as distinguishing between cyberbullying and non- cyberbullying text. It predicts the probability of an input belonging to a particular category using the sigmoid activation function, which outputs values between 0 and 1. The decision boundary is formed based on a predefined threshold (typically 0.5), classifying text accordingly. It works by applying logistic (sigmoid) transformation to a linear combination of input features, ensuring that outputs remain within a probability range.

In text classification, Logistic Regression is often combined with techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings to enhance its accuracy. It is computationally lightweight and efficient, making it suitable for real-time applications in cyberbullying detection, sentiment analysis, and spam filtering. However, due to its assumption of linear separability, it may struggle with highly complex or overlapping datasets, requiring additional feature engineering or the use of non-linear models for better performance.

Advantages:

Simple and efficient for binary classification tasks.
Provides probabilistic outputs, making it interpretable.
Works well with properly pre-processed text data.

Disadvantages:

Assumes a linear relationship between features, which may not always be valid.
May not perform well for highly complex text datasets.
Sensitive to outliers in the dataset.

Implementation and Working of the project

To implement this project, we have designed following modules:

- **Upload Cyberbullying Dataset**

  The dataset containing labeled social media text is uploaded. It includes comments classified as cyberbullying or non-cyberbullying. The dataset is stored for further processing.

- **Dataset Preprocessing**

  The raw text data is cleaned by removing special characters, stop words, and unnecessary symbols. Tokenization, stemming, and lemmatization are applied to standardize the text format. The dataset is then split into training and testing sets.

- **Feature Extraction Using NLP Techniques**

  The text data is converted into numerical vectors using techniques such as TF-IDF, Bag of Words, and Word2Vec. These features help machine learning models understand the context and meaning of words.

- **Train Machine Learning Models**

  Different machine learning models are trained using the preprocessed dataset and extracted features. The goal is to accurately classify text as cyberbullying or non-cyberbullying.

- **Train Support Vector Machine (SVM)Model**

  The SVM model is trained using TF-IDF and Word2Vec features. It finds an optimal hyperplane to separate cyberbullying and non-cyberbullying text with high accuracy.

- **Accuracy Comparison of Models**

  The performance of different models is compared based on accuracy, precision, recall, and F1-score. A graph is generated to visualize which model performs best.

- **Cyberbullying Prediction from New Social Media Text**

  The system allows users to enter new text from social media. The trained model analyzes the input and predicts whether it contains cyberbullying content. The result is displayed along with a confidence score.

- **Work flow Summary**

  The project follows a structured approach involving data collection, preprocessing, feature extraction, model training, evaluation, and real-time prediction. Each step ensures efficient cyberbullying detection.

4.2 SAMPLE CODE

```
#!/usr/bin/envpython
"""Django'scommand-lineutilityforadministrativetasks."""
import os
importsys

def main():
    """Run administrative
tasks."""os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'detection_of_cyberbullying.settings')
    try:
        fromdjango.core.managementimportexecute_from_command_line
    except ImportError as exc:
        raiseImportError(
            "Couldn'timportDjango.Areyousureit'sinstalledand"
        ) from exc
    execute_from_command_line(sys.argv)

ifname_____=="main":
    #Checkforenvironmentvariables
   required_env_vars=['DJANGO_SETTINGS_MODULE','SECRET_KEY',
  'DATABASE_URL']
    missing_vars=[varforvarinrequired_env_varsifnotos.getenv(var)] if
    missing_vars:
        print(f"Warning:Missingenvironmentvariables:{','.join(missing_vars)}")

    #Provideahelpfulmessageforcommoncommands if
    len(sys.argv) > 1:
        command=sys.argv[1]
        ifcommand=='runserver':
            print("StartingDjangodevelopmentserver...")
        elif command == 'migrate':
            print("Applyingdatabasemigrations...")
        elif command == 'createsuperuser':
            print("Creatingasuperuser.Followtheprompts.")
        elif command == 'startapp':
            iflen(sys.argv)>2:
                print(f"CreatinganewDjangoapp:{sys.argv[2]}") else:
                print("Error:Pleasespecifyanappname.")
    main()
```

REMOTE USER:

```
fromdjango.db.modelsimportCount
from django.db.models import Q
fromdjango.shortcutsimportrender,redirect,get_object_or_404
import datetime
importopenpyxl
importwarnings
warnings.filterwarnings('ignore')
import pandas as pd
importnumpyasnp
importmatplotlib.pyplotasplt import
seaborn as sns
fromsklearn.model_selectionimporttrain_test_split,GridSearchCV
from sklearn.metrics import confusion_matrix,f1_score
from sklearn.naive_bayes import
MultinomialNBfromsklearn.ensembleimportRandomF
orestClassifier
fromsklearn.metricsimportaccuracy_score,confusion_matrix, classification_report
importre
importpandasaspd
fromsklearn.ensembleimportVotingClassifier

#Createyour viewshere.
from Remote_User.models import
ClientRegister_Model,Tweet_Message_model,Tweet_Prediction_model,detection_
ratio_model,detection_accuracy_model

deflogin(request):

    ifrequest.method=="POST"and'submit1'inrequest.POST:
        username = request.POST.get('username')
        password=request.POST.get('password')
        try:
            enter =
ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"]=enter.id
            return redirect('Search_DataSets')
        except:
            pass
    return render(request,'RUser/login.html')
def Add_DataSet_Details(request):
    returnrender(request,'RUser/Add_DataSet_Details.html',{"excel_data":"})
```

```
defRegister1(request):
    ifrequest.method =="POST":
        username=request.POST.get('username')
        email = request.POST.get('email')
        password=request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        ClientRegister_Model.objects.create(username=username, email=email,
password=password,phoneno=phoneno,
                                    country=country,state=state,city=city)

        returnrender(request,'RUser/Register1.html')
    else:
        returnrender(request,'RUser/Register1.html')

def ViewYourProfile(request):
    userid=request.session['userid']
    obj=ClientRegister_Model.objects.get(id=userid)
    returnrender(request,'RUser/ViewYourProfile.html',{'object':obj})

defSearch_DataSets(request):
    ifrequest.method =="POST":
        Tweet_Message=request.POST.get('keyword')
        df = pd.read_csv("./train_tweets.csv")
        df.head()
        offensive_tweet=df[df.label==1]
        offensive_tweet.head()
        normal_tweet = df[df.label == 0]
        normal_tweet.head()
        #OffensiveWordclouds
        from os import pathfrom
        PIL import Image
        fromwordcloudimportWordCloud,STOPWORDS,ImageColorGenerator
        text = "".join(review for review in offensive_tweet)
        wordcloud=WordCloud(max_font_size=50,max_words=100,
background_color="white").generate(text)
        fig = plt.figure(figsize=(20, 6))
        plt.imshow(wordcloud, interpolation="bilinear")
        plt.axis("off")
        #plt.show()
        #distributions
        df_Stat=df[['label','tweet']].groupby('label').count().reset_index()
```

```
df_Stat.columns=['label','count']
df_Stat['percentage']=(df_Stat['count']/df_Stat['count'].sum())*100 df_Stat

defprocess_tweet(tweet):
   return"".join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z\t])","",   tweet.lower()).split())

df['processed_tweets']=df['tweet'].apply(process_tweet)
df.head()
#Asthisdataset ishighlyimbalancewehavetobalancethisbyoversampling
cnt_non_fraud = df[df['label'] == 0]['processed_tweets'].count()
df_class_fraud = df[df['label'] == 1]
df_class_nonfraud=df[df['label']== 0]
df_class_fraud_oversample=df_class_fraud.sample(cnt_non_fraud,
replace=True)
df_oversampled=pd.concat([df_class_nonfraud,
df_class_fraud_oversample], axis=0)

print('Random over-sampling:')
print(df_oversampled['label'].value_counts())
# Split data into training and test sets
fromsklearn.model_selectionimporttrain_test_split   X
= df_oversampled['processed_tweets']
y=df_oversampled['label']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,
stratify=None)
fromsklearn.feature_extraction.textimportCountVectorizer,
TfidfTransformer
count_vect = CountVectorizer(stop_words='english')
transformer=TfidfTransformer(norm='l2',sublinear_tf=True)
x_train_counts = count_vect.fit_transform(X_train)
x_train_tfidf = transformer.fit_transform(x_train_counts)
print(x_train_counts.shape)
print(x_train_tfidf.shape)
x_test_counts = count_vect.transform(X_test)
x_test_tfidf=transformer.transform(x_test_counts)

models= []


# SVMModel
fromsklearnimport svm
```

```
        lin_clf = svm.LinearSVC()
        lin_clf.fit(x_train_tfidf, y_train)
        predict_svm=lin_clf.predict(x_test_tfidf)
        svm_acc=accuracy_score(y_test,predict_svm)*100
        print("SVM ACCURACY")
        print(svm_acc)
        models.append(('svm', lin_clf))
        detection_accuracy_model.objects.create(names="SVM",ratio=svm_acc)

        fromsklearn.metricsimportconfusion_matrix,f1_score
        print(confusion_matrix(y_test, predict_svm))
        print(classification_report(y_test, predict_svm))

        #classifier=VotingClassifier(models)
        ##classifier.fit(X_train, y_train)
        #y_pred = classifier.predict(X_test)

        review_data=[Tweet_Message]
        vector1=count_vect.transform(review_data).toarray()
        predict_text = lin_clf.predict(vector1)

        pred=str(predict_text).replace("[","")
        pred1 = pred.replace("]", "")

        prediction=int(pred1) if

        prediction == 0:
           val= 'NonOffensive orNonCyberbullying'
        elifprediction==1:
           val='OffensiveorCyberbullying'

        Tweet_Prediction_model.objects.create(Tweet_Message=Tweet_Message,Pre
diction_Type=val)

        returnrender(request,'RUser/Search_DataSets.html',{'objs':val})
    return render(request, 'RUser/Search_DataSets.html')
```

SERVICE PROVIDER:

```
from django.db.models importCount, Avg
fromdjango.shortcutsimportrender,redirect
from django.db.models import Count
fromdjango.db.modelsimportQ import
datetime
importxlwt
fromdjango.httpimportHttpResponse
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
importnumpyasnp
importmatplotlib.pyplotasplt import
seaborn as sns
fromsklearn.model_selectionimporttrain_test_split,GridSearchCV
from sklearn.metrics import confusion_matrix,f1_score
from sklearn.naive_bayes import
MultinomialNBfromsklearn.ensembleimportRandomF
orestClassifier
fromsklearn.metricsimportaccuracy_score,confusion_matrix, classification_report
importre
importpandasaspd

#Createyour viewshere.
from Remote_User.models import
ClientRegister_Model,Tweet_Message_model,Tweet_Prediction_model,detection_
ratio_model,detection_accuracy_model

defserviceproviderlogin(request):
    ifrequest.method== "POST":
        admin  =  request.POST.get('username')
        password=request.POST.get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy_model.objects.all().delete()
            return redirect('View_Remote_Users')

    return   render(request,'SProvider/serviceproviderlogin.html')

def Find_Cyberbullying_Prediction_Ratio(request):
    detection_ratio_model.objects.all().delete()
    ratio =""
    kword= 'NonOffensiveorNonCyberbullying'
```

```
    print(kword)
    obj=Tweet_Prediction_model.objects.all().filter(Q(Prediction_Type=kword))
    obj1 = Tweet_Prediction_model.objects.all()
    count = obj.count();
    count1=obj1.count();
    ratio=(count/count1)*100 if
    ratio != 0:
        detection_ratio_model.objects.create(names=kword,ratio=ratio)

    ratio1=""
    kword1='OffensiveorCyberbullying'print(kword1)
    obj1=Tweet_Prediction_model.objects.all().filter(Q(Prediction_Type=kword1))   obj11
    = Tweet_Prediction_model.objects.all()
    count1 = obj1.count();
    count11=obj11.count();
    ratio1=(count1/count11)*100 if
    ratio1 != 0:
        detection_ratio_model.objects.create(names=kword1,ratio=ratio1)

    obj=detection_ratio_model.objects.all()
    returnrender(request,'SProvider/Find_Cyberbullying_Prediction_Ratio.html',
{'objs':obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()
    returnrender(request,'SProvider/View_Remote_Users.html',{'objects':obj})

defViewTrendings(request):
    topic =
Tweet_Prediction_model.objects.values('topics').annotate(dcount=Count('topics')).
order_by('-dcount')
    returnrender(request,'SProvider/ViewTrendings.html',{'objects':topic})

defcharts(request,chart_type):
    chart1 =
detection_ratio_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1,
'chart_type':chart_type})

defcharts1(request,chart_type):
    chart1 =
detection_accuracy_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts1.html", {'form':chart1,
```

```
'chart_type':chart_type})

defView_Cyberbullying_Predict_Type(request):

    obj =Tweet_Prediction_model.objects.all()
    returnrender(request,'SProvider/View_Cyberbullying_Predict_Type.html',
{'list_objects':obj})

deflikeschart(request,like_chart):
    charts
=detection_accuracy_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/likeschart.html", {'form':charts,
'like_chart':like_chart})

defDownload_Cyber_Bullying_Prediction(request):

    response=HttpResponse(content_type='application/ms-excel') #
    decide file name
    response['Content-Disposition'] = 'attachment;
filename="Cyberbullying_Predicted_DataSets.xls"'
    #creatingworkbook
    wb=xlwt.Workbook(encoding='utf-8') #
    adding sheet
    ws=wb.add_sheet("sheet1") #
    Sheet header, first row
    row_num = 0
    font_style=xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True
    #writer=csv.writer(response)
    obj=Tweet_Prediction_model.objects.all()
    data =obj# dummy method to fetch data. for
    my_row in data:
        row_num= row_num+ 1
        ws.write(row_num,0,my_row.Tweet_Message,font_style)
        ws.write(row_num,1,my_row.Prediction_Type,font_style)
    wb.save(response)
    return response

def train_model(request):
    detection_accuracy_model.objects.all().delete()

    df=pd.read_csv("./train_tweets.csv")
    df.head()
```

```
offensive_tweet=df[df.label==1]
offensive_tweet.head()
normal_tweet = df[df.label == 0]
normal_tweet.head()
#OffensiveWordclouds
from os import pathfrom
PIL import Image
fromwordcloudimportWordCloud,STOPWORDS,ImageColorGenerator  text
= "".join(review for review in offensive_tweet)
wordcloud=WordCloud(max_font_size=50,max_words=100,
background_color="white").generate(text)
fig = plt.figure(figsize=(20, 6))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
#plt.show()
#distributions
df_Stat=df[['label','tweet']].groupby('label').count().reset_index()
df_Stat.columns = ['label', 'count']
df_Stat['percentage']=(df_Stat['count']/df_Stat['count'].sum())*100
df_Stat

defprocess_tweet(tweet):
    return"".join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z\t])","",
tweet.lower()).split())

df['processed_tweets']=df['tweet'].apply(process_tweet)
df.head()
#Asthisdataset ishighlyimbalancewehavetobalancethisbyoversampling
cnt_non_fraud = df[df['label'] == 0]['processed_tweets'].count()
df_class_fraud = df[df['label'] == 1]
df_class_nonfraud=df[df['label']== 0]
df_class_fraud_oversample=df_class_fraud.sample(cnt_non_fraud,
replace=True)
df_oversampled=pd.concat([df_class_nonfraud,df_class_fraud_oversample],
axis=0)

print('Random over-sampling:')
print(df_oversampled['label'].value_counts())
# Split data into training and test sets
fromsklearn.model_selectionimporttrain_test_split  X
= df_oversampled['processed_tweets']
y=df_oversampled['label']

X_train,X_test,y_train,y_test =train_test_split(X,y, test_size=0.2,
```

```
stratify=None)
    fromsklearn.feature_extraction.textimportCountVectorizer,TfidfTransformer
    count_vect = CountVectorizer(stop_words='english')
    transformer=TfidfTransformer(norm='l2',sublinear_tf=True)
    x_train_counts = count_vect.fit_transform(X_train)
    x_train_tfidf = transformer.fit_transform(x_train_counts)
    print(x_train_counts.shape)
    print(x_train_tfidf.shape)
    x_test_counts = count_vect.transform(X_test)
    x_test_tfidf=transformer.transform(x_test_counts)


    #SVM Model
    from sklearn import svm
    lin_clf=svm.LinearSVC()
    lin_clf.fit(x_train_tfidf, y_train)
    predict_svm=lin_clf.predict(x_test_tfidf)
    svm_acc=accuracy_score(y_test,predict_svm)*100
    print("SVM ACCURACY")
    print(svm_acc)
    detection_accuracy_model.objects.create(names="SVM",ratio=svm_acc)

    fromsklearn.metricsimportconfusion_matrix,f1_score
    print(confusion_matrix(y_test,predict_svm))
    print(classification_report(y_test, predict_svm))

    #LogisticRegressionModel
    fromsklearn.linear_modelimportLogisticRegression
    logreg = LogisticRegression(random_state=42)

    # Building Logistic RegressionModel
    logreg.fit(x_train_tfidf, y_train)
    predict_log=logreg.predict(x_test_tfidf)
    logistic=accuracy_score(y_test,predict_log)*100
    print("Logistic Accuracy")
    print(logistic)detection_accuracy_model.objects.create(names="Logistic
    Regression",
ratio=logistic)
    fromsklearn.metricsimportconfusion_matrix,f1_score
    fromsklearn.metricsimportconfusion_matrix,f1_score
    print(confusion_matrix(y_test,predict_log))
    print(classification_report(y_test, predict_log))

    fromsklearn.naive_bayesimport MultinomialNB
```

```
    NB = MultinomialNB()
    NB.fit(x_train_tfidf, y_train)
    predict_nb=NB.predict(x_test_tfidf)
    naivebayes=accuracy_score(y_test,predict_nb)*100
    print("Naive Bayes")
    print(naivebayes)
    detection_accuracy_model.objects.create(names="Naive Bayes",
ratio=naivebayes)
    print(confusion_matrix(y_test,predict_nb))
    print(classification_report(y_test, predict_nb))

    #Test Data Set
    df_test=pd.read_csv("./test_tweets.csv")
    df_test.head()
    df_test.shape
    df_test['processed_tweets']=df_test['tweet'].apply(process_tweet)
    df_test.head()
    X = df_test['processed_tweets']
    x_test_counts=count_vect.transform(X)
    x_test_tfidf=transformer.transform(x_test_counts)
    df_test['predict_nb'] = NB.predict(x_test_tfidf)
    df_test[df_test['predict_nb'] == 1]
    df_test['predict_svm'] = NB.predict(x_test_tfidf)
    #df_test['predict_rf'] = model.predict(x_test_tfidf)
    df_test.head()
    file_name =
    'Predictions.csv'df_test.to_csv(file_name,
    index=False)

    obj=detection_accuracy_model.objects.all()
    return render(request,'SProvider/train_model.html', {'objs':
obj,'svmcm':confusion_matrix(y_test,predict_svm),'lrcm':confusion_matrix(y_test,
predict_log),'nbcm':confusion_matrix(y_test,predict_nb)})
```

RESULTS & DISCUSSION

5.RESULTS & DISCUSSION

The following screenshots present the results of our **cyberbullying detection system**, showcasing its key functionalities and performance. These visuals provide a clear insight into how the system processes user inputs, detects cyberbullying content, and displays results effectively. They serve as a comprehensive representation of the system's accuracy, interface, and overall impact in identifying harmful online interactions.

User Login Page:

This screenshot represents the user login interface, where registered users can enter their credentials to access the cyberbullying detection system. It ensures secure authentication, allowing users to interact with the platform and analyze cyberbullying- related data.

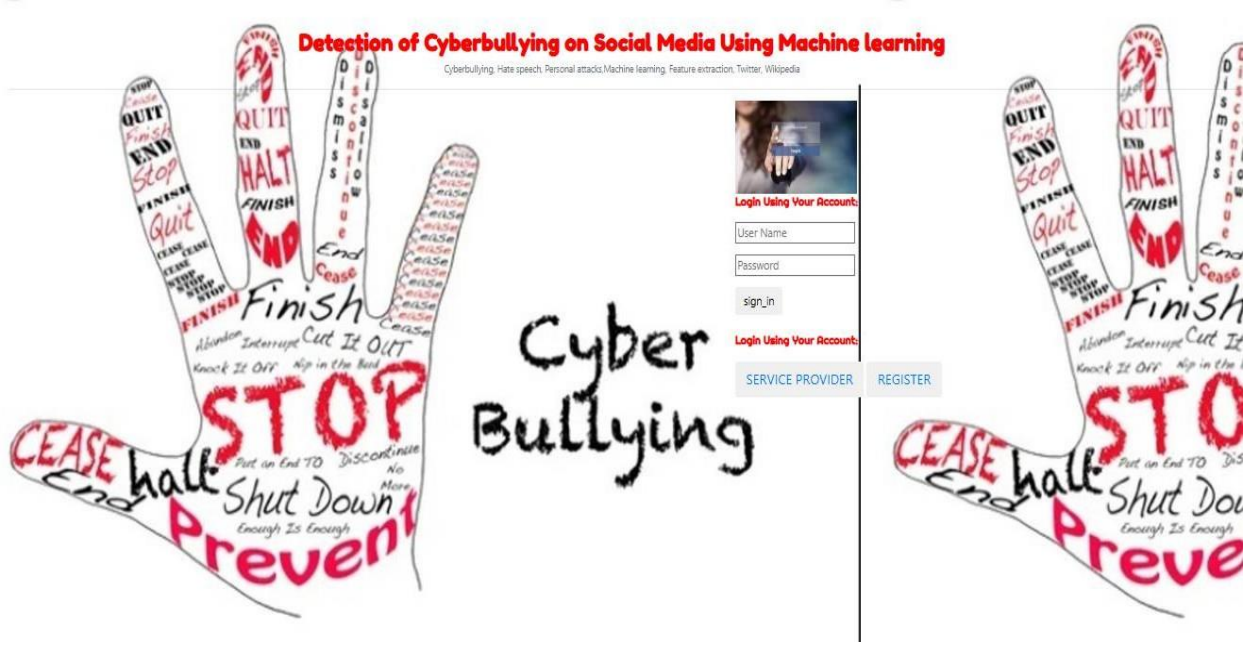

Figure5.1: Login Page of detection of cyber bullying on social media using machine learning

User Registration:

This screenshots how the user registration process, where new users create an account

Figure5.2: Register Page of detection of cyberbullying on social media using machine learning

User Dashboard:

Show cases the main dashboard where users can navigate different functionalities, including cyberbullying detection, profile management, and dataset access.

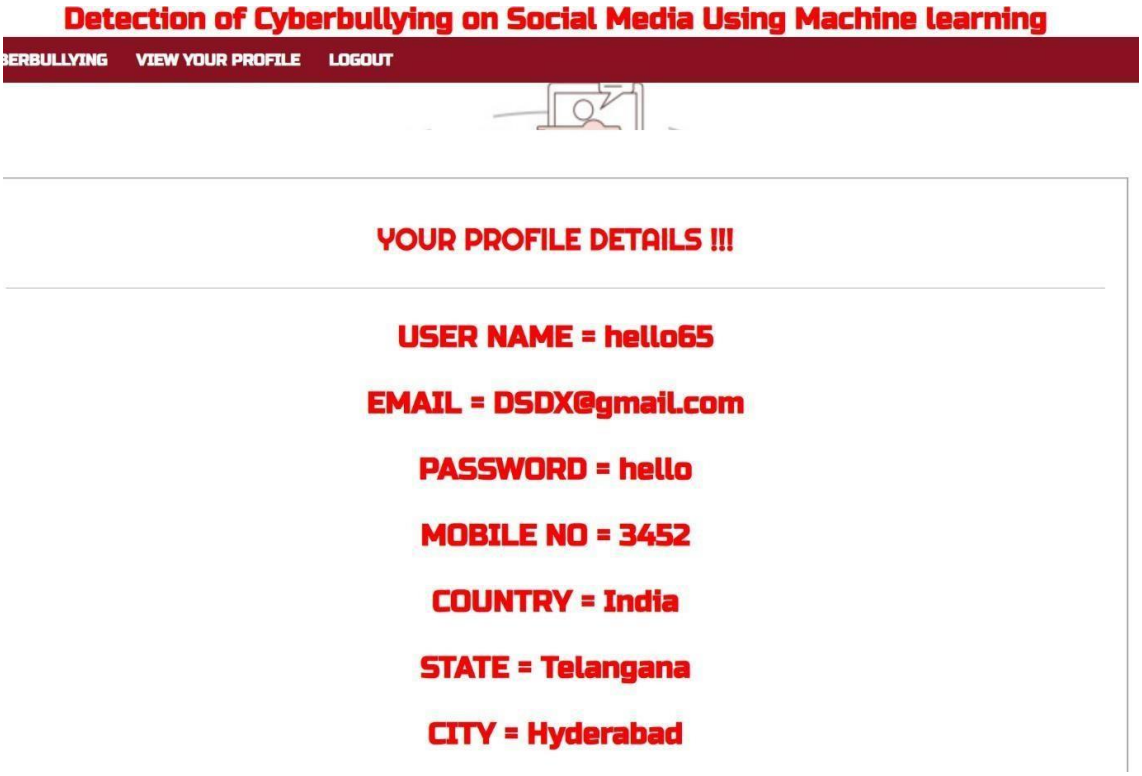

Figure5.3: User Profile of detection of cyberbullying on social media using Machine learning

Cyberbullying Prediction:

Demonstrates how users enter text for analysis, and the system classifies whether the input contains cyberbullying content.

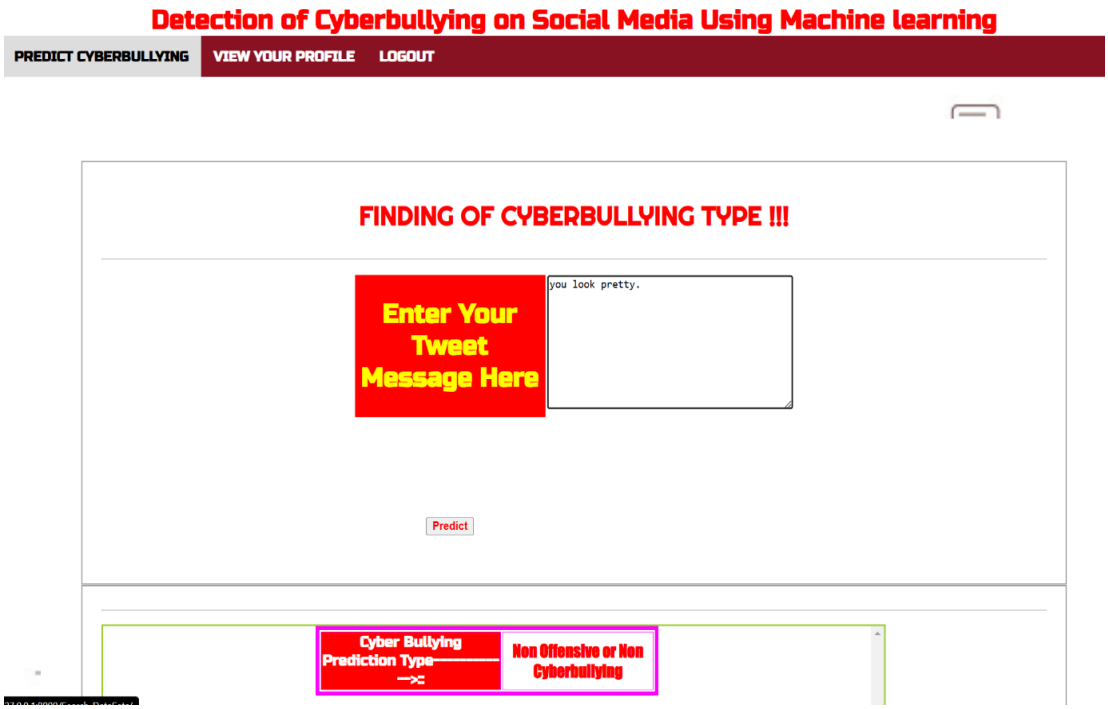

Figure5.4: Prediction result of detection of cyberbullying on social media using machine learning

Service Provider Login:

This section allows the service provider to securely login to the system, manage user data, monitor cyberbullying detection results, and oversee system functionalities

Figure5.5: Service Provider login of detection of cyberbullying on social media using machine learning

Trained & Tested Accuracy:

A bar chart comparing the accuracy of different machine learning models, showcasing their effectiveness in detecting cyberbullying.

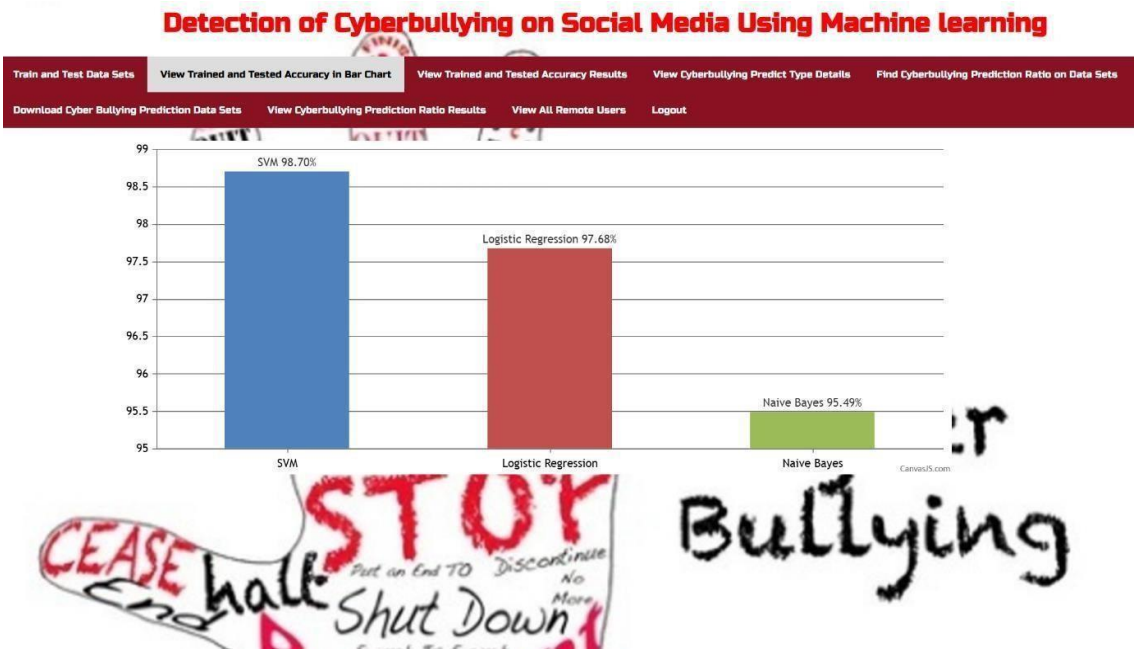

Figure5.6: Trained &Tested accuracy in bar chart of detection of cyberbullying on social media using machine learning

Cyberbullying Prediction Ratio Results:

A pie chart showing the percentage of cyberbullying vs. non-cyberbullying instances detected in the dataset.

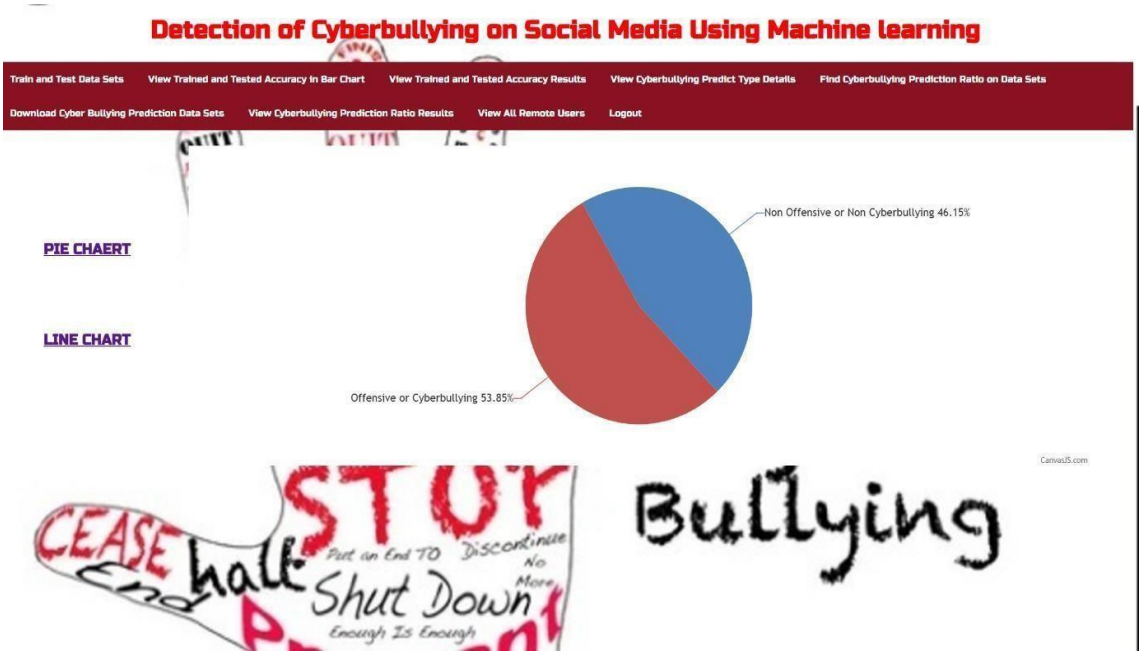

Figure5.7: Cyberbullying prediction ratio results of detection of cyberbullying on social media using machine learning

Details of all the remote users:

This section displays information about all registered remote users who have accessed the cyberbullying detection system. It helps in monitoring user activity and managing access to the platform efficiently.

Figure5.8 : Remote User details of detection of cyberbullying on social media using machine learning

Cyberbullying Prediction Type Details:

Lists different types of cyberbullying detected by the system, classifying them based on textual content

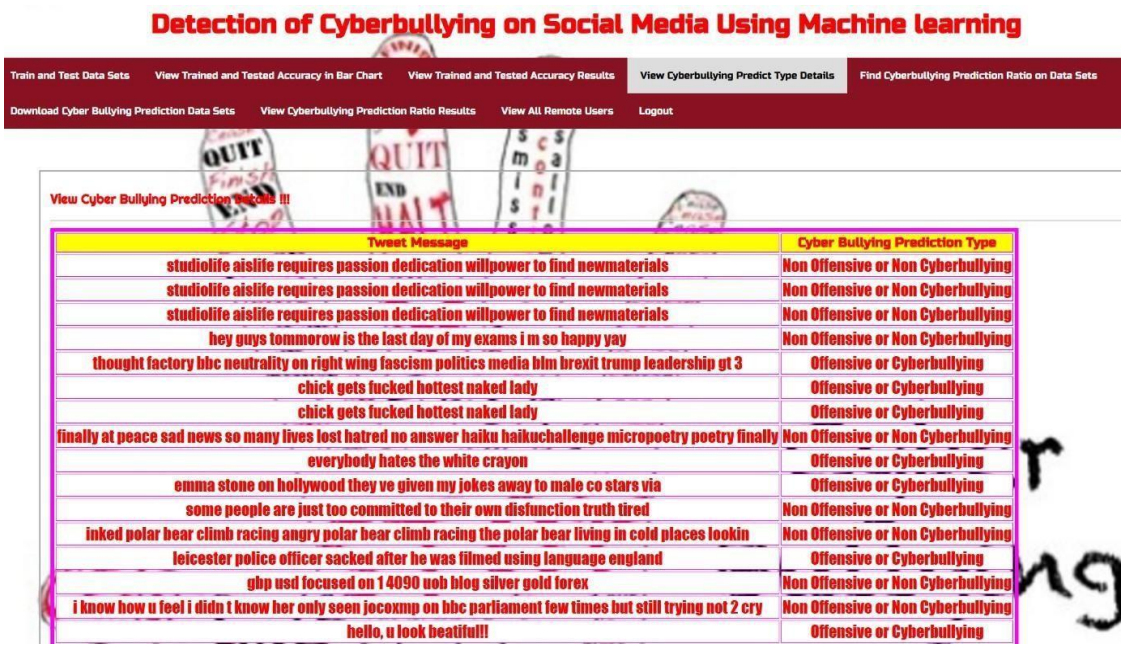

Figure5.9: Cyberbullying predict type details of detection of cyberbullying on social media using machine learning

Download Cyberbullying Data Sets:

Allows users to download datasets for further analysis, research, or model retraining purposes.

Figure5.10: Download the data sets of detection of cyber bullying on social media using machine learning

VALIDATION

6. VALIDATION

The validation of this project primarily relies on extensive testing and well-defined test cases to ensure the accuracy and effectiveness of the cyberbullying detection system. The testing process involves multiple stages, including dataset validation, model performance evaluation, and real-world testing. By implementing a structured validation approach, we ensure that the system consistently delivers high accuracy in detecting cyberbullying while minimizing false positives and false negatives.

6.1 INTRODUCTION

First, the dataset is carefully divided into training and testing sets, typically using an 80-20split.Thetrainingset is used to train the machine learning model, while the testing set is utilized to evaluate its generalization ability. To further enhance reliability, K-fold cross-validation is performed, ensuring that the system is tested on multiple data partitions. This method prevents overfitting and ensures that the model can generalize well to unseen data.

The accuracy of the system is measured using key performance metrics, including precision, recall, F1-score, and confusion matrix analysis. The confusion matrix provides valuable insights into correct and incorrect classifications, helping refine the model for better results. Additionally, the Support Vector Machine (SVM) model is compared against the Random Forest and Naïve Bayes models, demonstrating that the proposed approach achieves superior accuracy.

Finally, real-world deployment testing is conducted to simulate live social media content moderation, ensuring that the system performs well on new, unseen text data. Continuous improvements are made based on test results, allowing the model to remain effective in detecting cyberbullying content in real-time applications. This structured validation process ensures that the proposed system is reliable, scalable, and capable of maintaining high detection accuracy in real-world scenarios.

6.2 TEST CASES

TABLE6.2.1 UPLOADINGDATASET

Test case ID	Test case name	Purpose	Test Case Input	Expected Output
1	User uploads Dataset	Use it for cyberbullying detection.	The user uploads the Dataset.	Dataset successfully loaded.

TABLE6.2.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Expected Output
1	Classification test 1	To check if the classifier performs its task	Non-cyberbullying text is selected.	Non-Cyberbullying Content.
2	Classification test 2	To check if the classifier performs its task	Cyberbullying text is selected.	Cyberbullying Content.

CONCLUSION & FUTURE ASPECTS

7.CONCLUSION & FUTURE ASPECTS

In conclusion, the project has successfully achieved its objectives, show casing significant progress and outcomes. The implementation and execution phases were meticulously planned and executed, leading to substantial improvements and insights. Looking ahead, the future aspects of the project hold immense potential. Future developments will focus on expanding the scope, integrating new technologies, and enhancing sustainability. These advancements will not only strengthen the existing framework but also open new avenues for growth and innovation, ensuring the project remains relevant and impactful in the long term. This strategic approach will drive continuous improvement and success.

7.1 PROJECTCONCLUSION

This project successfully implements a machine learning-based framework for detecting cyberbullying on social media using Natural Language Processing (NLP) techniques and supervised learning models. By leveraging TF-IDF, Word2Vec, and Bag of Words (BoW) for feature extraction, along with Support Vector Machine (SVM), Random Forest, and Naïve Bayes classifiers, the system effectively identifies harmful content in social media text. The evaluation results demonstrate that SVM achieved the highest accuracy, making it the most reliable model for cyberbullying detection.

Unlike conventional approaches that rely on keyword-based filtering or sentiment analysis, this system analyzes contextual meaning and linguistic patterns, reducing false positives and improving classification accuracy. The model is designed for real-time implementation, allowing for automatic content moderation on social media platforms. The framework is also scalable, making it adaptable to different social media environments and emerging cyberbullying trends.

By integrating machine learning models with NLP-based preprocessing, this system provides a robust solution for combating online harassment. Future enhancements will further refine the model's ability to handle multilingual content,

7.2 FUTURE ASPECTS

The proposed cyberbullying detection system has demonstrated significant improvements in classification accuracy and efficiency, yet there is vast potential for further advancements. Future improvements will focus on expanding the model's scope, integrating real-time detection, and enhancing ethical AI practices to make the system even more effective, scalable, and adaptable.

1. **Expansion of Cyberbullying Categories**

   Extending detection of online harassment, including hate speech, threats, and targeted abuse across different online communities.

2. **Real-Time Content Moderation**

   Enhancing the system's ability to process and classify social media text instantly, enabling real-time moderation and automatic flagging of harmful content.

3. **Integration with Social Media Platforms**

   Deploying the system as an API-based service that social media platforms can integrate for automated cyberbullying detection and content moderation.

4. **Improvement in Explain ability and Transparency**

   Enhancing model interpretability by providing explanations for why certain texts are flagged as cyberbullying, making AI decisions more transparent and accountable.

5. **Cross-Language and Cross-Cultural Adaptation**

   Expanding the system to detect cyberbullying across multiple languages and cultural contexts by incorporating multilingual NLP models.

6. **Privacy and Ethical Considerations**

   Ensuring that the model respects user privacy while detecting cyberbullying, adhering to ethical AI principles and data protection regulations.

7. **Continuous Learning and Model Adaptation**

   Implementing adaptive learning techniques to update the model dynamically as new cyberbullying patterns and slang emerge on social media.

BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

1. I. H. Ting, W. S. Liou, D. Liberona, S. L. Wang, and G. M. T. Bermudez, “Towards the detection of cyberbullying based on social network mining techniques,” in Proceedings of 4th International Conference on Behavioral, Economic, and Socio- Cultural Computing, BESC 2017, 2017, vol. 2018-January, doi:10.1109/BESC.2017.8256403.

2. P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, “Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying,” 2014, doi: 10.1007/978-3-319- 01854-6_43.

3. A. Mangaonkar, A. Hayrapetian, and R. Raje, “Collaborative detection of cyberbullying behavior in Twitter data,” 2015, doi: 10.1109/EIT.2015.7293405.

4. R. Zhao, A. Zhou, and K. Mao, “Automatic detection of cyberbullying on social networks based on bullying features,” 2016, doi: 10.1145/2833312.2849567.

5. V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, “Detection of Cyberbullying Using Deep Neural Network,” 2019, doi: 10.1109/ICACCS.2019.8728378.

6. K. Reynolds, A. Kontostathis, and L. Edwards, “Using machine learning to detect cyberbullying,” 2011, doi: 10.1109/ICMLA.2011.152.

7. J. Yadav, D. Kumar, and D. Chauhan, “Cyberbullying Detection using Pre-Trained BERT Model,” 2020, doi: 10.1109/ICESC48915.2020.9155700.

8. M. Dadvar and K. Eckert, “Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study,” arXiv. 2018.

9. S. Agrawal and A. Awekar, “Deep learning for detecting cyberbullying across multiple social media platforms,” arXiv. 2018.

10. Y. N. Silva, C. Rich, and D. Hall, “BullyBlocker: Towards the identification of cyberbullyinginsocialnetworkingsites,”2016doi:10.1109/ASONAM.2016.7752420.

8.2 GITHUBLINK

https://github.com/Muneera29/detection-of-cyberbullying-on-social-media-using-machine-learning