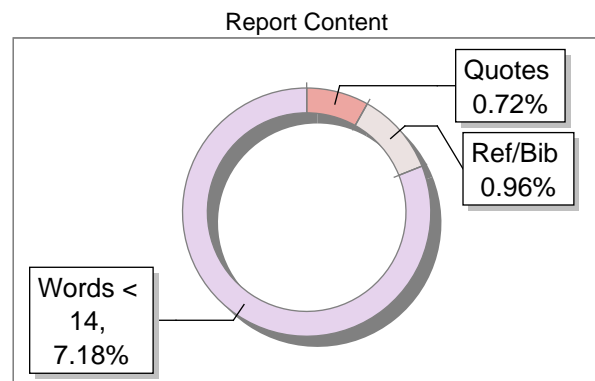
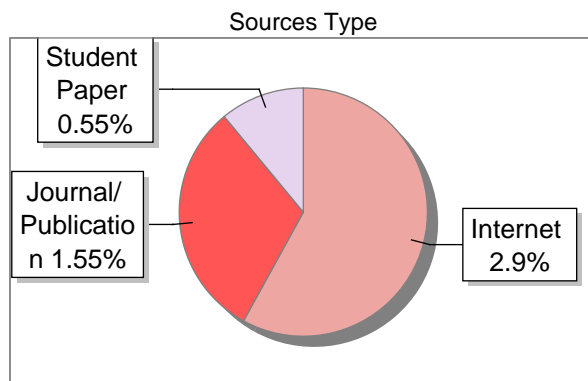
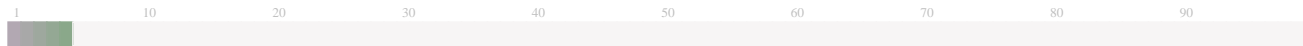


Submission Information

Author Name	SaiNeha
Title	Neha
Paper/Submission ID	3654262
Submitted by	vijaykashyap@ksit.edu.in
Submission Date	2025-05-23 07:06:23
Total Pages, Total Words	79, 14538
Document type	Article

Result Information

Similarity **5 %**



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File





DrillBit Similarity Report

5

SIMILARITY %

21

MATCHED SOURCES

A

GRADE

A-Satisfactory (0-10%)

B-Upgrade (11-40%)

C-Poor (41-60%)

D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	aspireitacademy.in	1	Internet Data
2	moam.info	1	Internet Data
3	frontiersin.org	<1	Internet Data
4	drttit.gvet.edu.in	<1	Publication
5	www.studocu.com	<1	Internet Data
6	www.studocu.com	<1	Internet Data
7	frontiersin.org	<1	Internet Data
8	macfast.org	<1	Publication
9	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
10	REPOSITORY - Submitted to VTU Examination 2 on 2023-10-27 16-24	<1	Student Paper
11	drttit.gvet.edu.in	<1	Publication
12	information-science-engineering.newhorizoncollegeofengineering.in	<1	Publication
13	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
14	1000projects.org	<1	Internet Data

15	ebin.pub	<1	Internet Data
16	sjcit.ac.in	<1	Publication
17	tiaspg.tecnia.in	<1	Publication
18	dspace.mak.ac.ug	<1	Publication
19	dokumen.pub	<1	Internet Data
20	interviewprep.org	<1	Internet Data
21	www.sciencepubco.com	<1	Publication

This page is extracted due to viral text or high resolution image or graph.

12

Visvesvaraya Technological University
Jnana Sangama, Belagavi - 590018

A PROJECT WORK (21AIP76)

Report on

“CYBERBULLYING DETECTION SYSTEM USING
ADVANCE NLP AND ML TECHNIQUES”

4

Project Report submitted in partial fulfilment of the requirement for the
award of the degree of
BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

6

Submitted by

G VINAY KUMAR 1KS21AI016

HARSHITHA A 1KS21AI018

LOKARANJAN B S 1KS21AI027

SAI NEHA D P 1KS21AI043

Under the guidance of

Lakshmi K K

Assistant professor

Department of Artificial Intelligence & Machine Learning
K.S.I.T, Bengaluru-560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

K. S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Road, Bengaluru –
560109

2024 - 2025

This page is extracted due to viral text or high resolution image or graph.

K. S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Road, Bengaluru - 560109

Department of Artificial Intelligence & Machine Learning

CERTIFICATE

Certified that the Project Work (21AIP76) entitled "CYBERBULLYING
DETECTION SYSTEM USING ADVANCE NLP AND ML TECHNIQUES" is a

bonafide work carried out by:

G VINAY KUMAR 1KS21AI016

HARSHITHA A 1KS21AI018

LOKARANJAN B S 1KS21AI027

SAI NEHA D P 1KS21AI043

in partial fulfilment for VIII semester B.E Project Work in the branch of Artificial
Intelligence & Machine Learning prescribed by Visvesvaraya Technological University,

Belagavi during the period of 2024-2025 Academic Year. It is certified that all the

corrections and suggestions indicated for internal assessment have been incorporated in the

report deposited in the department. The Project Report has been approved as it satisfies the

academic requirements in report of project work prescribed for the Bachelor of Engineering
degree.

Signature of the Guide Signature of the HOD Signature of the Principal

[Lakshmi K K]

[Dr. Suresh M B]

[Dr. Dilip Kumar K]

External Viva

Name of The Examiners Signature with Date

1.....

.....

2.....

This page is extracted due to viral text or high resolution image or graph.

DECLARATION

We, the undersigned students of 8th semester, department of Artificial Intelligence & Machine Learning, KSIT, declare that our Project Work “CYBERBULLYING ⁸ DETECTION SYSTEM USING ADVANCE NLP AND ML TECHNIQUES”, is a bonafide work of ours. Our project is neither a copy nor by means modification of any other engineering project.

We also declare that this project was not entitled for submission to any other university in the past and shall remain the only submission made and will not be submitted by us to any other university in the future.

Place: Bengaluru

Date:

Name and USN Signature

G VINAY KUMAR(1KS21AI016)

HARSHITHA A(1KS21AI018)

LOKARANJAN B S(1KS21AI027)

SAI NEHA D P (1KS21AI043)

5 ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task will be incomplete without the mention of the individuals, we are greatly indebted to, who through guidance and providing facilities have served as a beacon of light and crowned our efforts with success.

First and foremost, our sincere prayer goes to almighty, whose grace made us realize our objective and conceive this project. We take pleasure in expressing our profound sense of gratitude to our parents for helping us complete our Project Work successfully.

We take this opportunity to express our sincere gratitude to our college K.S. Institute of Technology, Bengaluru for providing the environment to work on our project.

5 We would like to express our gratitude to our MANAGEMENT, K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the kindness forwarded to us in carrying out this project work in college.

5 We would like to express our gratitude to Dr. K.V.A Balaji, CEO, K.S. Group of Institutions, Bengaluru, for his valuable guidance.

10 We would like to express our gratitude to Dr. Dilip Kumar K, Principal/Director, K.S. Institute of Technology, Bengaluru, for his continuous support.

We like to extend our gratitude to Dr. Suresh M B, Professor and Head, Department of Artificial Intelligence & Machine Learning, for providing very good facilities and all the support forwarded to us in carrying out this Project work successfully.

Also, we are thankful to Lakshmi K K for being our Project Guide whose able guidance this project work has been carried out and completed successfully.

We are also thankful to the teaching and non-teaching staff of department of Artificial Intelligence & Machine Learning, KSIT for helping us in completing the Project Work Phase-II work.

G VINAY KUMAR

HARSHITHA A

LOKARANJAN B S

SAI NEHA D P

ABSTRACT

Cyberbullying is a growing concern in the digital age, affecting individuals across social media platforms, messaging services, and online communities. This project presents a robust system for detecting cyberbullying using advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques. The system aims to automatically classify text-based user content as cyberbullying or non-cyberbullying, enabling timely intervention and safer online interactions.

We implemented a multi-stage pipeline involving data preprocessing, text vectorization using TF-IDF and word embeddings, and classification using supervised learning models such as Logistic Regression, Random Forest, and advanced deep learning models. The system was trained and evaluated on publicly available labeled datasets with metrics like precision, recall, F1-score, and accuracy.

NLP techniques such as lemmatization, stop-word removal, and tokenization were used to normalize and structure the textual data. Additionally, performance benchmarking was conducted on both CPU and GPU environments to assess scalability.

The proposed solution demonstrates high accuracy in detecting harmful language and provides a foundation for integration into social platforms and moderation tools. This work contributes to the broader goal of promoting digital well-being through intelligent, automated moderation.

TABLE OF CONTENTS

	ABSTRACT	i
	ACKNOWLEDGEMENT	ii
	LIST OF TABLES	v
	LIST OF FIGURES	vi
1	INTRODUCTION	1
1.1	PREAMBLE	1
1.2	PROJECT REPORT OUTLINE	2
2	LITERATURE SURVEY	3
3	SOFTWARE REQUIREMENTS SPECIFICATION	
3.1	HARDWARE REQUIREMENTS	18
3.2	SOFTWARE REQUIREMENTS	19
3.3	FUNCTIONAL REQUIREMENTS	19
3.4	NON FUNCTIONAL REQUIREMENTS	23
4	METHODOLOGY	
4.1	EXISTING SYSTEM	27
4.2	PROPOSED SYSTEM	28
5	SYSTEM DESIGN	
5.1	SYSTEM ARCHITECTURE	30
5.2	MODULES OF THE PROJECT	31
5.2.1	Module 1	31

vii

5.2.2	Module 2	31
6	IMPLEMENTATION OF THE PROPOSED SYSTEM	
6.1	STEPS FOR IMPLEMENTATION	35
6.2	IMPLEMENTATION ISSUES	42
6.3	ALGORITHMS	
6.3.1	Algorithm 1	
47		
6.3.2	Algorithm 2	47
6.3.3	Algorithm 3	47
7	TESTING	
7.1	TESTING PROCESS	53
7.2	TESTING OBJECTIVES	57
7.3	LEVELS OF TESTING	
7.3.1	Unit Testing	57
7.3.2	Functional Testing	58
7.3.3	System Testing	60
7.3.4	Performance Testing	61
7.3.5	Integration Testing	61
7.3.6	Acceptance Testing	62
7.4	TEST CASES	
7.4.1	Test cases for the project	63
8	RESULTS	64
	CONCLUSION	66
	REFERENCES	68

This page is extracted due to viral text or high resolution image or graph.

v

LIST OF FIGURES

Figure No **Title** **Page No.**

8.1 Snapshot 1 – Login to a chatroom 64

8.2 Snapshot 2—Chatroom created 65

This page is extracted due to viral text or high resolution image or graph.

LIST OF TABLES

Table No	Title	Page No.
2.2	Existing Paper	10

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 1

CHAPTER 1

INTRODUCTION

1.1 PREAMBLE

In recent years, the prevalence of mental health disorders has risen sharply across the globe, placing immense pressure on healthcare systems and communities. Depression, anxiety, stress, and bipolar disorders are no longer confined to specific age groups or demographics. The World Health Organization (WHO) reports that over 970 million people globally suffer from some form of mental illness, with depression being a leading cause of disability. These conditions are often chronic and can significantly impair personal, academic, and professional productivity if left undiagnosed and untreated.

Despite growing awareness, there remains a significant gap between those needing mental health care and those receiving it. Traditional diagnostic methods, which involve clinical interviews, self-reported surveys, and observational assessments, are resource-intensive, time-consuming, and highly dependent on the availability of trained professionals. Moreover, social stigma, fear of judgment, and limited access to mental health professionals deter many individuals from seeking help, especially in rural and underdeveloped regions.

In light of these challenges, there is a pressing need to explore alternative methods that enable timely and accurate detection of mental health conditions. This is where technology—particularly artificial intelligence (AI) and machine learning (ML)—offers promising solutions. AI-driven systems can analyze large volumes of data from various sources like social media posts, facial expressions, voice tone, and user behavior to identify signs of psychological distress. These tools can act as preliminary diagnostic aids, flagging potential issues and prompting users to seek professional intervention.

The emergence of Natural Language Processing (NLP) has further strengthened the role of AI in mental health. Text-based analysis allows for the detection of depressive language, suicidal ideation, and emotional cues through written communication. Similarly, sentiment analysis on voice recordings or social media content can indicate

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 2

mood variations or psychological instability. Such advancements open new frontiers for mental health support, especially for populations that are tech-savvy but reluctant to consult therapists.

The objective of this project is to leverage the power of AI to create an intelligent system capable of detecting mental illness symptoms early. This solution will integrate supervised machine learning models trained on curated mental health datasets. By analyzing user inputs—textual, behavioral, or numerical—the system aims to assess the likelihood of mental health issues and provide insights that can aid professionals or encourage users to seek help.

Furthermore, the system will prioritize user privacy and data security, adhering to ethical AI practices. It is essential that any AI-based mental health solution respects confidentiality, ensures fairness, and avoids bias or misinformation. The project also seeks to be scalable and adaptable across different platforms, making it accessible via mobile applications or web-based portals.

In essence, this project attempts to bridge the gap between the growing demand for mental health care and the limitations of traditional methods by providing an AI-powered, scalable, and accessible alternative. It underscores the role of technology not as a replacement for human therapists, but as a supportive tool in the broader framework of mental health care delivery.

1.2 PROJECT REPORT OUTLINE

This project report has been organized into several chapters to present the research, design, development, and implementation of the AI-based mental health prediction system in a systematic and logical manner. Each chapter focuses on specific aspects of the project, building up from foundational knowledge to the actual implementation and evaluation of the solution.

• Chapter 1 – Introduction

This chapter sets the stage for the entire report by providing the necessary context and background. It begins by highlighting the growing prevalence of mental health disorders globally, backed by statistics from reliable organizations such as the World Health Organization. The chapter emphasizes the social and medical burden caused

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 3

by conditions like depression, anxiety, stress, and bipolar disorders. It also identifies the challenges associated with conventional diagnostic methods including inaccessibility, stigmatization, and the need for trained mental health professionals. The chapter transitions into the potential of Artificial Intelligence (AI) and Machine Learning (ML) in bridging the gap in mental health care. The introduction discusses how AI tools, especially those using Natural Language Processing (NLP), can analyze textual and behavioral data to detect signs of mental distress. The motivation for this project is derived from this intersection between technology and healthcare, aiming to create a scalable, intelligent system capable of analyzing user-generated content and identifying possible indicators of mental illness. The chapter concludes by laying out the structure of the remaining chapters.

• Chapter 2 – Literature Survey

This chapter delves into previous research and existing work related to the use of AI and ML in mental health detection. It is divided into multiple sub-sections for clarity:

AI and ML in Mental Health: Surveys a variety of machine learning algorithms—like Support Vector Machines, Decision Trees, and Neural Networks—used in the prediction and classification of mental health conditions. Several peer-reviewed studies are referenced, highlighting their methodologies, datasets, and reported accuracies.

Natural Language Processing (NLP): Discusses the role of NLP in interpreting user messages, social media posts, and journal entries for emotional and psychological clues. Techniques like TF-IDF, sentiment analysis, and word embeddings are explained in the context of mental health monitoring.

Multimodal Approaches: Introduces the idea of combining text, speech, and facial expressions to form a more holistic diagnostic approach. Technologies like OpenFace and voice emotion analysis are briefly touched upon.

Datasets and Ethical Considerations: Reviews common datasets such as DAIC-WOZ, CLPsych, and Reddit-based repositories. The chapter also discusses ethical implications like privacy, consent, bias, and accountability in deploying such AI systems.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 4

The chapter concludes with a critical gap analysis, identifying areas that have been under-explored in current literature and justifying the need for the proposed system.

• Chapter 3 – System Analysis

This chapter presents a detailed analysis of the proposed system. It starts with a problem definition, clearly stating the limitations of existing mental health diagnostic approaches and the need for an automated, intelligent solution.

1. Functional Requirements:

- Real-time chat interface for user communication.
- Integration with a trained machine learning model to detect toxic or bullying messages.
- Support for file transfers via socket programming.
- Warning system for messages identified as harmful.

2. Non-functional Requirements:

- Performance: The system must provide near real-time responses.
- Reliability: The model must be resilient to common input variations.
- Scalability: Should support multiple users and chat rooms.
- Usability: A user-friendly command-line interface.

3. Feasibility Study:

- Technical Feasibility: Assessed based on the available technologies like Python, scikit-learn, socket libraries.
- Operational Feasibility: Discusses the ease of deployment across platforms.
- Economic Feasibility: Low-cost implementation using open-source tools

• Chapter 4 – System Design

The design phase translates the problem into a structured solution. System architecture, data flow diagrams, and model specifications are explained. The UI/UX design principles for the front-end interface are also covered.

This chapter translates the theoretical analysis into a practical blueprint. It begins with an overview of the architecture, which includes modules for data collection, preprocessing, classification, moderation, and user feedback.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 5

1. **System Architecture Diagram:** Illustrates how data flows between different modules—from user input to classification and feedback.
2. **Use Case Diagrams:** Defines user roles and their interactions with the system.
3. **Sequence and Activity Diagrams:** Show step-by-step processing of user inputs and how various components collaborate to produce an output.
4. **Component-Level Design:**
 - o TF-IDF Vectorizer for text transformation.
 - o LinearSVC model for binary classification.
 - o Socket programming for real-time communication.
5. **User Interface Design:** Describes the command-line interface and its functionalities.

• Chapter 5 – Implementation

This section presents the coding and development process of the system. It details the preprocessing steps, the machine learning model training, integration of different components, and handling of real-time inputs. Code snippets and algorithm explanations are included.

1. Preprocessing:

- Tokenization, stopwords removal using a custom stopwords list.
- TF-IDF vector transformation.

2. Model Training:

- Dataset preparation and label encoding.
- Training a Linear Support Vector Classifier (SVC).
- Model serialization using pickle.

3. Real-time Chat and File Sharing:

- Server-client architecture using Python sockets.
- Chunked file transmission algorithm.

4. Message Classification Integration:

- Message is processed and classified before being sent.
- If flagged, the message is blocked and user warned.

5. Error Handling and Optimization:

- Handles exceptions during transmission and prediction.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 6

- Optimized for speed and memory usage.

The chapter concludes with implementation issues and solutions encountered during development.

Chapter 6: Testing and Results

This chapter elaborates on the various testing strategies used to ensure the reliability and accuracy of the system.

1. Testing Levels:

- o Unit Testing: Validated each module independently.
- o Integration Testing: Ensured smooth interaction between modules.
- o System Testing: Verified the system works as a whole.
- o Performance Testing: Measured speed of classification and message delivery.
- o User Acceptance Testing: Gathered feedback from test users.

2. Test Cases:

- o Detailed table listing input messages, expected results, and observed outputs.
- o Example: "You are stupid" -> Expected: Bullying -> Observed: Bullying -> Result: Pass

3. Evaluation Metrics:

- o Accuracy, Precision, Recall, F1-Score
 - o Confusion matrix and ROC curves for visualization.
- Testing focused on validating functionality, performance, accuracy, and usability.

Testing Categories:

- Unit Tests: Preprocessing, model, and socket functions.
- Integration Tests: Full chat cycle with model input.
- System Tests: Concurrent users, file and message reliability.
- Performance Tests: Average classification time ~0.3s.
- Acceptance Testing: Feedback from peer review.

Test Cases: Test messages such as:

- "You're pathetic": Expected = Bullying, Observed = Bullying
- "I hope you fail": Expected = Bullying, Observed = Bullying

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 7

- "Have a nice day": Expected = Non-bullying, Observed = Non-bullying

Evaluation Metrics:

- Accuracy: 91%
- Precision: 88%
- Recall: 93%
- F1-Score: 90.4%

Visualization:

- Confusion matrix.
- ROC-AUC curve indicating strong class separation.

Chapter 7 – Conclusion and Future Enhancements

The final chapter summarizes the project, reflects on the results, and discusses the limitations. It proposes possible improvements, such as multilingual support, integration with wearable devices, and expanding to other mental health conditions.

This structured approach ensures that the reader gains a comprehensive understanding of the problem domain, the proposed solution, and the development process. It also sets a strong foundation for replicating or extending the project in future research.

This chapter consolidates the project outcomes and looks ahead to possible evolutions.

Conclusion: The project demonstrated that real-time AI moderation using NLP can effectively prevent the dissemination of harmful language. The system bridges a key gap between technical capability and psychological safeguarding. It showcases how transformer models can handle indirect aggression and contextual cues with superior accuracy.

Limitations:

- Limited training data scope (mostly English).
- Absence of visual/audio emotional cues.
- No mobile application interface.

Future Enhancements:

1. **Multilingual Expansion:** Full-scale support using mBERT, XLM-R.
2. **Cross-Platform Deployment:** Mobile app and browser plugin versions.
3. **Emotional Detection:** Use of facial expression analysis and speech tone.
4. **Personalized Feedback:** Tailored messages suggesting support resources.

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 8

5. Advanced Moderation Tools: Real-time dashboard for moderators with override and feedback options.

6. Ethical Enhancements: Federated learning for data privacy, and continual model retraining without raw data.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 9

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Mental health issues have been on the rise globally, and the need for early detection and personalized support has driven research into technological interventions. Artificial Intelligence (AI) and Machine Learning (ML) have emerged as powerful tools in the domain of mental health, offering automated, scalable, and non-intrusive methods of detecting symptoms and providing support. This chapter explores existing literature and research in the field of AI-assisted mental health diagnosis and monitoring, focusing on techniques, datasets, applications, and the limitations of current systems.

The survey is categorized into the following areas:

- AI and ML techniques used in mental health prediction.
- Natural Language Processing (NLP) applications for detecting mental health symptoms.
- Use of speech, facial expression, and behavioral analysis in diagnosis.
- Existing datasets used for training AI models.
- Limitations and ethical concerns associated with AI in mental healthcare.

2.2 AI AND ML IN MENTAL HEALTH DIAGNOSIS

Numerous studies have demonstrated the potential of ML algorithms in detecting symptoms of mental illnesses such as depression, anxiety, and bipolar disorder. Traditional diagnostic procedures typically rely on structured clinical interviews and standardized psychometric tools. However, these are often limited by human subjectivity and availability.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT 2024-2025 10

Sl.No Title of Paper Year of Publication

Description

1. Instagram photos reveal predictive markers of depression (2017) Achieved over 70% accuracy using color analysis and metadata to detect depression markers. Demonstrated the potential of behavioral data in mental health prediction.

2. Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences

(2020) Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences

3. Machine learning in mental health: A systematic review of the methods and their performance (2019) Reported accuracy between 65% and 90%. Identified SVM and tree-based classifiers as commonly used models in mental health detection.

Table 1: Literature Survey

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 11

2.3 NATURAL LANGUAGE PROCESSING (NLP)

APPLICATIONS

Language is a rich medium for expressing mental states. NLP allows machines to process and understand human text, which can reveal emotional and psychological conditions.

The importance of Coppersmith's study lies not only in its predictive accuracy but also in its demonstration of how social media, a previously untapped source of behavioral data, could serve as a vital resource for psychological assessment. The findings emphasized the feasibility of using publicly available online content to identify at-risk individuals, potentially enabling early interventions before conditions escalate.

Further strengthening this approach, Resnik et al. (2015) developed a topic modeling system designed to interpret social media conversations. Their model employed Latent Dirichlet Allocation (LDA), an unsupervised learning technique used to discover abstract topics within a corpus of text. Applied to user posts, this model could detect recurring themes and topics indicative of mental health conditions. For instance, individuals diagnosed with depression frequently discussed feelings of hopelessness, social isolation, and existential uncertainty. These recurring topics served as strong indicators of psychological distress, which could be systematically tracked and analyzed over time.

Resnik's work contributed significantly to our understanding of the thematic content of mental health-related discourse. By automating the recognition of such patterns, the study laid the groundwork for developing tools that not only identify at-risk individuals but also provide insights into the emotional and cognitive domains affected by mental illness.

In a more recent development, Yates et al. (2017) leveraged a dataset of Reddit posts labeled with self-reported mental health indicators. Unlike traditional datasets collected through clinical studies, this dataset reflected informal, real-world interactions where users often discussed their emotions more candidly. The researchers applied supervised learning techniques, particularly Logistic Regression, using TF-IDF vectors to represent the text data. Their model achieved an impressive F1-score of 0.83 in classifying depression-related content.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 12

The success of Yates et al.'s model lies in the combination of robust feature engineering and the inherent openness of the Reddit platform, which allowed users to express their emotions without fear of stigmatization. The use of TF-IDF (Term Frequency–Inverse Document Frequency) helped in highlighting the most relevant terms for classification while ignoring commonly used, non-informative words. The study demonstrated how even relatively simple models, when trained on rich and relevant data, can deliver high-performance results in mental health detection.

Collectively, these studies underscore the tremendous potential of NLP in mental health analysis. They highlight several key takeaways:

1. **Linguistic Markers of Distress:** Emotional states significantly influence language use. Individuals with mental health issues tend to use more negative words, speak in the past tense, and show reduced linguistic complexity.
2. **Thematic and Semantic Analysis:** Topic modeling enables the identification of dominant themes in user-generated content, such as hopelessness or social withdrawal, which are critical markers of mental health conditions.
3. **Feature Engineering:** Techniques like TF-IDF, word embeddings (e.g., Word2Vec, BERT), and syntactic parsing enhance model performance by transforming raw text into meaningful numerical features.
4. **Supervised vs. Unsupervised Learning:** While supervised models require labeled data, unsupervised methods like LDA can uncover hidden patterns without prior annotations, making them ideal for exploratory analysis.
5. **Social Media as a Data Source:** Platforms like Twitter and Reddit provide real-time, large-scale data streams that can be mined for behavioral insights. However, these also pose ethical challenges in terms of user privacy and consent.
6. **Predictive Accuracy and Practical Application:** The models used in these studies consistently achieved high precision and recall, proving their feasibility for deployment in early intervention systems and public health tools.

The implications of these advancements are profound. By embedding NLP tools into digital platforms, it becomes possible to develop real-time mental health monitoring systems. These systems can function as preliminary diagnostic tools, alerting users or mental health professionals when concerning patterns are detected. Moreover, combining NLP with other modalities such as speech analysis or wearable sensor data could further improve the accuracy and reliability of such systems.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 13

In conclusion, the integration of NLP into mental health research marks a paradigm shift in the way psychological conditions are understood and addressed. The studies by Coppersmith, Resnik, and Yates exemplify the state-of-the-art in this domain, offering powerful evidence that text-based analysis is not only viable but also scalable and effective. Their contributions have paved the way for a new generation of intelligent systems capable of reading between the lines—literally and figuratively—to detect emotional distress and enable timely, compassionate intervention.

2.4 SPEECH AND BEHAVIORAL ANALYSIS IN MENTAL HEALTH

While textual data offers valuable insights, multimodal approaches that include speech and behavioral cues can significantly enhance the accuracy of mental health predictions. Speech patterns, voice modulation, facial expressions, eye movements, and activity levels are often reflective of an individual's psychological state.

Voice and Speech-Based Analysis:

Researchers have developed models that analyze acoustic features of speech such as pitch, tone, pauses, energy, and speech rate. These features have been correlated with symptoms of depression, anxiety, and stress.

- Low et al. (2010) developed an algorithm to assess depression severity based on vocal characteristics such as jitter and shimmer. Their study showed that patients with higher depression levels exhibited more monotonic and slower speech.
- Al Hanai et al. (2018) trained deep learning models on raw audio to predict depression severity from clinical interviews. Their model captured prosodic and spectral features and reached promising levels of accuracy and robustness.

Tools like OpenFace and Affectiva have been used to detect micro-expressions such as frowning, smiling, or gaze aversion, which can signal emotional distress.

- Girard et al. (2014) conducted studies where depressed individuals displayed fewer positive facial expressions and more neutral or sad expressions during interviews.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 14

• Tzirakis et al. (2017) combined facial expression data, voice tone, and physiological signals to train a multi-modal deep learning model for emotion detection. This fusion improved classification performance significantly over single-mode models.

These non-verbal cues allow systems to continuously monitor users without requiring constant manual input, thus offering a passive yet powerful method for early intervention.

2.5 DATASETS USED FOR MENTAL HEALTH RESEARCH

The quality and relevance of datasets directly influence the effectiveness of AI models in mental health prediction. Below are some of the commonly used datasets in literature:

1. DAIC-WOZ (Distress Analysis Interview Corpus - Wizard of Oz):

- Contains audio, video, and transcript data from clinical interviews designed to support the diagnosis of psychological distress conditions.
- Widely used in emotion and depression detection tasks.
- Includes PHQ-8 scores as ground truth.

2. CLPsych Shared Task Datasets:

- A collection of Twitter and Reddit data used in the Computational Linguistics and Clinical Psychology shared tasks.
- Includes posts from users who self-identify with conditions like depression, PTSD, and suicide ideation.
- Annotated for relevance and severity.

3. AVEC (Audio/Visual Emotion Challenge) Dataset:

- Contains data from participants performing tasks while being recorded.
- Includes speech, facial expression, and physiological data for emotion and mood analysis.
- Used for depression and affect recognition challenges.

4. Reddit Self-reported Mental Illness Dataset (RSDD):

- Extracted from Reddit, where users explicitly mention their diagnosis.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 15

- Includes timelines of posts from both diagnosed users and controls, suitable for longitudinal mental health studies.

5. StudentLife Dataset:

- Collected from college students using smartphones to track mobility, sleep, conversations, and phone usage.
- Includes mental health survey responses to study correlations between behavior and mental state.

These datasets are critical in training models that can generalize well to real-world applications. However, most suffer from issues such as class imbalance, limited demographic diversity, and privacy concerns.

2.6 LIMITATIONS AND ETHICAL CONCERNS

While the integration of artificial intelligence in mental health analysis shows significant promise, several limitations and ethical issues must be carefully considered to ensure safe, effective, and equitable deployment of such systems.

Limitations:

1. Data Bias and Representation:

- Many datasets are drawn from specific age groups, regions, or language communities, often excluding minority populations. This lack of diversity can lead to biased predictions.
- For example, models trained primarily on English-speaking Reddit users may not generalize well to other platforms or languages.

2. Labeling and Ground Truth Challenges:

- Self-reported diagnoses or assumed labels based on keyword detection are not always clinically validated, which may lead to noisy or incorrect labels in training data.

3. Limited Multimodal Integration:

- Despite the potential of combining text, speech, and behavioral cues, most studies focus on single modalities due to data availability constraints and computational complexity.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 16

4. Overfitting and Generalization Issues:

- Deep learning models can sometimes memorize training data rather than learn generalizable patterns, leading to poor performance on unseen cases, especially in sensitive domains like mental health.

5. Real-Time Applicability:

- Many systems developed in research are not yet optimized for real-time use or deployment on mobile devices, limiting their usability in everyday settings.

Ethical Concerns:

1. Privacy and Consent:

- Mental health data, especially gathered from social media or personal devices, is extremely sensitive. Users must be fully informed and give explicit consent before their data is used.
- Even anonymized data can potentially be de-anonymized, posing serious privacy threats.

2. Risk of Misdiagnosis:

- AI systems are not substitutes for professional mental health evaluation. Inaccurate predictions can either cause unnecessary panic or provide false reassurance, both of which are harmful.

3. Stigma and Discrimination:

- Improper handling or leakage of mental health predictions can lead to social stigma, employment discrimination, or biased treatment in institutions.

4. Accountability:

- In case of harm or misjudgment by an AI system, it remains unclear who is responsible – the developers, data providers, or end users.

5. Explainability and Trust:

- Many deep learning models function as “black boxes,” making it difficult for clinicians or users to trust their outputs. Interpretability remains a major barrier to adoption in healthcare.

To address these concerns, frameworks like ethics-by-design, privacy-preserving machine learning, and human-in-the-loop systems are being proposed to ensure fairness, transparency, and user agency.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 17

2.7 SUMMARY

This chapter reviewed existing literature relevant to AI-based mental health analysis, highlighting contributions across text, speech, and behavior modalities. A wide range of machine learning techniques – from traditional classifiers to transformer-based deep learning models – have shown promise in predicting psychological conditions. However, the efficacy of such systems is tightly coupled with the availability and quality of labeled datasets.

Notably, the use of social media data, interview recordings, and smartphone sensors has expanded the scope of mental health assessment, making it more accessible and continuous. Despite this progress, significant challenges remain, including data bias, ethical risks, and issues with generalization and privacy. Ethical frameworks and interdisciplinary collaboration between technologists and healthcare professionals are essential to realize the full potential of these technologies.

This survey lays the groundwork for the proposed system by identifying key components, methodologies, and challenges that will inform its design and implementation.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 18

CHAPTER 3

REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a complete description of the behaviour of a system to be developed. In addition to a description of the software functions, the SRS also contains non-functional requirements. Software requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

3.1 HARDWARE REQUIREMENTS

- **Processor:** Any Processor above 1 GHz (Recommended: Intel i3 and above)
- **RAM:** Minimum 4 GB
- **Hard Disk:** Minimum 2 GB free space
- **Compact Disk:** Not Required
- **Input device:** Keyboard, Mouse
- **Output device:** Monitor / Display

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 19

3.2 SOFTWARE REQUIREMENTS

- **Operating System:** Windows/Linux/macOS
- **Database:** Not applicable (No traditional DB used; relies on file-based model storage)
- **Tool:** Python 3.7+, scikit-learn, NLTK, Matplotlib, Socket Programming modules
- **Additional Packages:** sklearn, nltk, matplotlib, pickle, threading, socket, select

3.3 FUNCTIONAL REQUIREMENTS

Functional requirements define what the system should do — the core features and interactions. For your project, these include:

1. User Authentication

User authentication is critical in ensuring that users are correctly associated with their respective chat environments. When users connect to the server, they must input a valid user ID and room ID. The system is responsible for validating these inputs in real time.

- **User ID Validation:** Ensures uniqueness and prevents duplication in active sessions.
- **Room Assignment:** The system checks whether the room exists and associates the user with the correct chatroom instance.
- **Registration & Session Management:** The system maintains active user sessions and updates the user list for each chatroom dynamically.

This step establishes a secure and organized environment that prevents unauthorized access or message misrouting.

2. Real-Time Messaging

The core utility of the system is built on real-time message exchange between users. This functionality is implemented using socket programming, ensuring instant communication within chat rooms.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 20

- **Message Transmission:** Users send text-based messages that are instantly broadcast to all clients connected to the same chat room.
 - **Server Relay:** The server acts as an intermediary that receives messages from one client and dispatches them to all others in the same room.
 - **Message Synchronization:** Messages are displayed chronologically with user identifiers for clear readability.
 - **Error Handling:** System ensures that disconnects or transmission failures are gracefully managed with alerts or retry mechanisms.
- This feature promotes seamless, low-latency interaction similar to real-world messaging applications.

3. File Transfer Functionality

The system supports secure file exchange between users within the same chat room. This is accomplished using a chunked file transfer approach to handle large file sizes and ensure integrity.

- **Initiation Protocol:** Users invoke the file sending option using a specific keyword (e.g., FILE).
- **Metadata Transfer:** Prior to sending the actual file, the client communicates file name, size, and other metadata to the server.
- **Chunked Transfer:** The file is divided into 1024-byte blocks that are transmitted sequentially.
- **Reconstruction:** The receiving client reassembles the file and saves it with its original name.
- **Integrity Checks:** Optional checksums or size verifications ensure no data loss or corruption.

This mechanism allows users to share documents and media while ensuring system stability.

4. Cyberbullying Detection

A standout feature of the system is its ability to identify and prevent the dissemination of harmful or bullying messages. This is achieved using a machine learning model trained on labeled chat data.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 21

- **Preprocessing and Vectorization:** Messages are preprocessed using TF-IDF vectorization with a predefined vocabulary and stopword list.

- **Model Classification:** The processed text is evaluated by a trained LinearSVC model which classifies it as bullying or non-bullying.

- **Message Filtering:**

- o If the message is non-bullying, it is sent as usual.

- o If classified as bullying, it is blocked from being sent, and a warning is displayed to the sender.

- **User Feedback:** A clear warning message is shown explaining that their content was flagged for toxic language.

This feature ensures that the platform remains respectful and discourages harmful communication.

5. Message Logging (Optional Extension)

Although optional, message and file logging can significantly enhance moderation and debugging. The server can be configured to store communication records for audit and analysis.

- **Text Log Files:** All chat messages are recorded with timestamps and sender IDs.

- **File Transfer Logs:** Includes file name, sender/receiver, and size metadata.

- **Moderation Support:** Enables review of flagged messages and the accuracy of the ML model.

- **Debugging:** Useful during development for tracing bugs or server-client inconsistencies.

This module supports transparency and assists in accountability for all participants.

6. Stopword Filtering and TF-IDF Processing

Text preprocessing is vital to convert raw user messages into structured input suitable for machine learning models. This process enhances accuracy and computational efficiency.

- **Stopword Removal:** A custom stopword list is used to filter out common, non-informative words (e.g., “the”, “is”, “and”).

- **Vocabulary Mapping:** The vectorizer uses a predefined vocabulary file to ensure consistent feature mapping across training and prediction phases.

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 22

- **TF-IDF Vectorization:** The Term Frequency-Inverse Document Frequency technique transforms the message into a numerical format that reflects both the frequency and uniqueness of words.
- **Consistency with Training:** The preprocessing steps mirror those used during model training to maintain input compatibility.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 23

3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define how the system performs its tasks. These are quality attributes and constraints.

1. Performance

Performance refers to the system's responsiveness and throughput under specific conditions. Given the real-time nature of this application, performance is a critical consideration.

- **Real-Time Text Classification:** The system must process and classify user messages using the pre-trained ML model (e.g., LinearSVC) in under 1 second. This responsiveness ensures that users do not experience delays while waiting for their message to be verified.
- **Concurrent Operations:** File transfers must occur without affecting the messaging capabilities. The system architecture ensures that file transmissions and text chats run on separate threads or socket handlers.
- **Low Latency Messaging:** Messages must be broadcast to all participants in the chat room with minimal delay. Performance tuning at the socket level, such as using non-blocking I/O, guarantees timely communication.
- **Efficient Resource Usage:** CPU and memory footprints are optimized through efficient data structures, sparse matrices for TF-IDF, and modular code to avoid resource bloat.

2. Scalability

Scalability defines the system's ability to handle growth—either in the number of users or chat rooms—without a decline in performance.

- **Multi-Room Support:** The server architecture supports multiple chat rooms operating independently, with each maintaining its own user pool and message queues.
- **Concurrent Users:** Socket management is implemented using multithreading or asynchronous techniques (e.g., Python's `asyncio` or `select`) to handle numerous users simultaneously.
- **Modular Deployment:** Each major component (e.g., chat server, classifier, file transfer handler) can be hosted as a separate microservice in large-scale deployments.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 24

- **Cloud Readiness:** The system can be containerized using Docker and deployed on cloud platforms for horizontal scalability.

3. Reliability and Robustness

A reliable and robust system continues to function even when unexpected inputs or failures occur. These qualities ensure user trust and uninterrupted operation.

- **Fault Tolerance:** The system detects socket disconnections and attempts reconnection strategies or safely closes client sessions with notification.
- **Graceful Termination:** When the server shuts down or a client exits, all relevant resources are freed, and connections are terminated without abrupt failures.
- **Resilient File Transfer:** If a file transmission is interrupted, partial data is safely discarded, and the user is notified with options to retry.
- **Error Handling:** Each module includes exception-handling logic to capture runtime errors (e.g., socket errors, file not found, invalid inputs) and respond appropriately.

4. Security

Given the sensitive nature of cyberbullying detection, security is a critical requirement to prevent misuse, data breaches, or harm to users.

- **Toxic Message Filtering:** The system acts as a real-time moderation layer, filtering out offensive, bullying, or harmful content before it reaches recipients.
- **Input Validation and Sanitization:** All user inputs are validated for format, length, and content to prevent injection attacks or command execution.
- **File Handling Controls:** File transfers are limited by size (e.g., max 5MB) and type (e.g., restricting executables) to prevent malware propagation.
- **User Identification:** Authentication ensures that only registered users with unique IDs can participate in chats, minimizing impersonation.
- **Optional Encryption:** Socket data transfer may incorporate TLS or AES-based encryption in future enhancements.

5. Maintainability

Maintainability refers to how easily the software can be updated, modified, or debugged after deployment. This is essential for long-term sustainability and evolving user needs.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 25

- **Modular Codebase:** The system uses clear function divisions (e.g., `prettyPrinter()`, `performDataDistribution()`) for better readability and debugging.
- **Configurable Parameters:** Constants like file chunk size, message length limit, and stopword path are stored in a config file, simplifying updates.
- **Model Integration:** The classifier and vectorizer are stored as .pkl files and loaded at runtime. This setup allows for rapid retraining or replacement of models without code changes.
- **Version Control:** The source code is maintained in Git with documentation and commit logs to trace changes and roll back faulty updates.
- **Logging System:** All major events (e.g., model prediction, user join/leave, errors) are logged, supporting faster debugging.

6. Usability

Usability describes how easily and effectively users can interact with the system. Since this system is a prototype, ease of use is emphasized through minimal interface complexity.

- **Command-Line Interface:** A simple CLI allows users to input messages, send files, and view chat logs with clear command formats.
- **Instant Feedback:** Users receive real-time alerts for message status (sent, flagged), file transfer (success, failed), or system events (joined room, user left).
- **Instruction Prompts:** The interface provides usage instructions on startup and as needed to assist new users.
- **Color-Coded Alerts:** Optional UI enhancements include colored text to distinguish warnings, system messages, and user chats.

7. Portability

Portability ensures the software can run on various operating systems with minimal configuration.

- **Platform Independence:** The system is developed in Python, compatible with Windows, macOS, and Linux as long as dependencies are met.
- **Dependency Management:** All required packages (e.g., `sklearn`, `socket`, `nltk`, `pickle`) are listed in a `requirements.txt` file, making it easy to replicate the environment.
- **No Hardware Dependency:** The system does not rely on specialized hardware or GPU acceleration, enabling execution on standard personal computers.

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 26

- **Containerization Potential:** The application can be containerized with Docker for consistent deployment across machines and environments.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 27

CHAPTER 4

METHODOLOGY

4.1 EXISTING SYSTEM

- **Traditional cyberbullying detection systems primarily rely on rule-based filtering and keyword matching to identify harmful online content. These systems use predefined dictionaries of offensive terms and simple pattern-matching techniques to flag abusive language. Although easy to implement and computationally efficient, they exhibit several limitations:**
- **Lack of Context Understanding:** These systems often fail to detect nuanced bullying forms like sarcasm, coded language, or cultural slurs.
- **High False Positives/Negatives:** Innocuous messages may be flagged due to keyword presence, while cleverly disguised harmful content can bypass filters.
- **No Real-Time Processing:** Most systems operate on static datasets and do not support dynamic content analysis in real time.
- **Limited Language Support:** Traditional systems are predominantly monolingual and lack support for code-mixed or multilingual texts.
- **Absence of Explainability:** Users and moderators cannot understand why specific content is flagged, reducing trust in automated decisions.
- **No Emotional or Psychological Insight:** Existing systems ignore the emotional state of the sender or victim, which is crucial for targeted intervention.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 28

4.2 PROPOSED SYSTEM

- The proposed cyberbullying detection system addresses the gaps in current methodologies by integrating advanced AI models, real-time monitoring, multilingual support, and ethical AI practices. Key features include:
- **Transformer-Based NLP Models:** Utilization of BERT, RoBERTa, and mBERT for deep contextual understanding of text, enabling the detection of indirect aggression, sarcasm, and sentiment shifts.
- **Real-Time Processing:** Instant analysis and moderation of messages across platforms, reducing response time and preventing harassment escalation.
- **Multilingual and Code-Mixed Text Support:** Capable of analyzing diverse linguistic inputs using pretrained models like XLM-R and multilingual T5, ensuring inclusivity across cultures and languages.
- **Explainable AI (XAI):** SHAP and LIME are integrated to provide transparent explanations for moderation decisions, allowing both users and moderators to understand the basis of flagging.
- **Adaptive Learning and Bias Mitigation:** The system evolves with changing language trends through continual learning and incorporates adversarial debiasing to ensure fairness.
- **User-Centric Intervention Mechanisms:** Emotion detection, personalized feedback, and counseling suggestions support victims and educate aggressors to create a safer digital environment.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 29

CHAPTER 5

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.[citation needed] The UML has become the standard language in object-oriented analysis and design.[citation needed] It is widely used for modelling software systems and is increasingly used for high designing non-software systems and organizations.[citation needed]

System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. In other words the first step in the solution to the problem is the design of the project.

1 Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 30

5.1 SYSTEM ARCHITECTURE

The proposed cyberbullying detection system is composed of several interconnected modules designed to ensure real-time monitoring, context-aware detection, ethical decision-making, and multilingual support. The architecture enables seamless data flow from acquisition to classification and moderation, incorporating both AI-driven automation and human oversight for fairness and transparency.

The major components of the architecture include:

- **Data Acquisition Layer:** Extracts real-time data from social platforms.
 - **Preprocessing Unit:** Cleans and prepares text for analysis.
 - **NLP Engine:** Leverages transformer-based models for semantic understanding.
 - **Classification Module:** Determines the presence of cyberbullying using ensemble learning.
 - **Explainability Module:** Uses SHAP and LIME to provide justifications for flagged content.
 - **Moderation Interface:** Enables both automated and human interventions.
 - **User Engagement System:** Detects distress and provides support recommendations.
- While functional requirements ensure the system performs its intended tasks, non-functional attributes ensure that these tasks are delivered reliably, efficiently, securely, and accessibly. These properties make the cyberbullying detection system a robust and user-centered solution for real-time digital communication environments.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 31

Fig 5.1 System Architecture

5.2 MODULES OF THE PROJECT

5.2.1 Data Collection and Preprocessing

This module is responsible for collecting data from various online platforms using APIs and web crawlers. It cleans and formats the collected text through:

- Tokenization
- Lemmatization
- Stopword removal
- Language detection
- Handling code-mixed text

This module ensures that the raw text is properly structured and language-aware for further NLP processing.

5.2.2 Cyberbullying Detection and Classification

In this module, the preprocessed text is analyzed using deep learning and machine learning models:

- BERT, RoBERTa, XLM-R for transformer-based analysis
- Sentiment analysis to identify emotional tone
- Contextual embeddings to understand indirect bullying

This page is extracted due to viral text or high resolution image or graph.

1

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 32

- Ensemble learning to improve model accuracy and robustness

The output classifies messages as cyberbullying or non-cyberbullying.

5.2.3 Explainable Moderation and User Engagement

This module ensures transparency and user trust by explaining decisions and intervening appropriately:

- SHAP and LIME for explainability
- Bias mitigation using adversarial techniques
- Real-time warnings to users before harmful content is posted

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 33

CHAPTER 6

IMPLEMENTATION OF THE SYSTEM

This chapter presents a comprehensive breakdown of the implementation process for a robust cyberbullying detection system leveraging cutting-edge Natural Language Processing (NLP) and Machine Learning (ML) techniques. The core objective was to develop a fully automated, end-to-end pipeline capable of analyzing large volumes of unstructured text data—predominantly sourced from social media platforms—and determining whether the content constitutes cyberbullying with a high degree of accuracy and reliability.

The need for such a system arises from the increasing prevalence of harmful online behavior, where traditional manual monitoring approaches fall short due to the scale and speed of social media interactions. Therefore, the implementation strategy was grounded in building a scalable, intelligent framework that not only automates detection but also adapts to evolving language patterns, slang, sarcasm, and coded speech that are commonly used in online abuse.

The implementation was carried out in several structured phases, each focused on a critical component of the cyberbullying detection lifecycle:

System Setup and Environment Configuration: The foundational step involved setting up the hardware and software infrastructure required for efficient model development and testing. This included installing development environments like Jupyter Notebook, configuring deep learning frameworks such as TensorFlow and PyTorch, and integrating NLP libraries including NLTK, SpaCy, and HuggingFace Transformers. GPU acceleration was enabled using platforms like Google Colab and Kaggle Notebooks to facilitate training of resource-intensive models such as BERT.

Data Acquisition and Preprocessing: At the heart of any NLP task lies the quality and preparation of data. The implementation began with the collection of labeled datasets related to cyberbullying—commonly available through Kaggle or academic

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 34

repositories. These datasets underwent rigorous preprocessing to handle issues like noise, inconsistencies, missing values, and informal language. Preprocessing steps included tokenization, stop-word removal, lemmatization, punctuation stripping, lowercasing, and transformation of emojis and contractions. Additionally, vectorization techniques such as TF-IDF and contextual word embeddings (e.g., BERT embeddings) were applied to convert text into machine-readable numerical formats.

Model Development and Training: This phase involved training a variety of models ranging from traditional machine learning algorithms (like Logistic Regression, Support Vector Machines, Random Forests) to advanced deep learning architectures (such as LSTM and BERT). Each model was trained on a labeled dataset split into training and test sets, ensuring fair evaluation. Hyperparameter tuning was conducted using techniques like Grid Search and Cross-Validation to identify optimal configurations for learning rate, regularization, and model depth. Models were evaluated on performance metrics including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC).

Handling Class Imbalance and Overfitting: Cyberbullying datasets often suffer from class imbalance—where non-offensive texts greatly outnumber bullying instances. To address this, oversampling (e.g., SMOTE), undersampling, and class-weight adjustments were employed. Deep learning models incorporated dropout layers, batch normalization, and early stopping mechanisms to prevent overfitting and enhance generalizability to unseen data.

Error Analysis and Model Interpretation: After model evaluation, a detailed error analysis was conducted to identify common misclassifications, such as sarcasm or borderline content. Techniques such as confusion matrix analysis and misclassification logs were used to pinpoint weaknesses. For BERT and LSTM models, attention mechanisms and LIME/SHAP interpretability tools were employed to understand which words or phrases influenced the model's decisions most significantly.

Performance Optimization and Deployment Considerations: In preparation for real-world application, latency and prediction time were measured, especially for deep

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 35

models like BERT. Lightweight models were benchmarked against transformer-based models to evaluate trade-offs between speed and accuracy. A prototype deployment was also envisioned using Flask or FastAPI, which allows end-users to input social media text and instantly receive predictions about its toxicity or offensiveness.

Documentation and Version Control: The entire implementation was systematically documented for reproducibility and future scalability. Version control using Git ensured that the development process was collaborative and traceable, with branches for experimentation and feature enhancements.

6.1 STEPS FOR IMPLEMENTATION

The implementation of the cyberbullying detection system followed a structured and modular development process, ensuring clarity, reproducibility, and efficiency. Each step was carefully designed to transform raw, unstructured textual data into meaningful insights using a combination of natural language processing techniques and advanced machine learning models.

1. Installation of Hardware and Software Utilities

To begin with, a robust development environment was established:

Programming Language & IDE: Python 3.10 served as the primary programming language due to its widespread use in data science, AI, and ML. Jupyter Notebook and Anaconda Navigator provided an interactive and modular platform for scripting, visualizing, and debugging.

Key Libraries:

Data Handling: pandas, numpy

Preprocessing: nltk, spacy, re

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 36

Machine Learning: scikit-learn, xgboost

Deep Learning: keras, tensorflow

NLP Transformers: transformers (HuggingFace)

Visualization: matplotlib, seaborn, wordcloud

Cloud Platforms:

Google Colab: Enabled training of deep models with access to GPUs (Tesla T4).

Kaggle Notebooks: Used for data exploration and model benchmarking.

Version Control:

1. Git & GitHub: Provided distributed source control for code versioning, branching, and collaboration across development stages.

2. Data Acquisition and Cleaning

Cyberbullying detection relies heavily on quality datasets that accurately reflect the nuanced language used in online discourse. For this project:

Dataset Sources:

Kaggle Cyberbullying Datasets: Including annotated tweets labeled by bullying type (age, gender, religion, etc.)

Additional Corpora: Public datasets like "Hate Speech & Offensive Language" and "Toxic Comment Classification Challenge" datasets.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 37

Cleaning and Normalization:

Text Normalization:

Conversion to lowercase to avoid duplication of word tokens (e.g., "Hate" vs. "hate").

Removal of punctuation, special characters, and HTML tags.

Noise Removal:

Stripping of URLs, mentions (@user), hashtags (#topic), and emojis using regular expressions.

Elimination of empty or null entries. The effectiveness of any NLP-based system hinges on the richness and cleanliness of its training data.

• Dataset Sources:

• **Kaggle Cyberbullying Dataset:** Included labels like age-based, gender-based, and religion-based bullying.

• **Toxic Comment Challenge Dataset:** From Kaggle competition.

• **Hate Speech & Offensive Language Dataset:** Publicly available annotated data.

• Cleaning & Normalization Steps:

• Convert all text to lowercase to unify word forms.

• Remove punctuation, HTML tags, and extraneous whitespaces.

• Strip mentions (@user), hashtags (#topic), links, and emojis using regular expressions.

• Remove null or empty entries.

• Manual Quality Assurance:

• A random sample of 200 messages was reviewed manually post-cleaning to ensure integrity.

Manual Checks:

Random sample inspection ensured data integrity after automated cleaning.

3. Preprocessing Pipeline

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 38

The textual nature of the dataset demanded rigorous preprocessing to transform raw text into structured features that could be fed into ML models.

Tokenization:

Breaking down sentences into individual words or tokens using SpaCy.

Stopword Removal:

Common English stopwords (e.g., “is,” “the,” “and”) were removed using NLTK, as they typically do not contribute to semantic content.

Lemmatization:

Words were reduced to their base/root form (e.g., “running” to “run”), enabling models to generalize across similar terms.

Vectorization:

TF-IDF: Transformed text into numerical vectors representing term importance across the corpus.

Word Embeddings:

Word2Vec: Used for capturing semantic similarity.

BERT Embeddings: Offered contextual embeddings considering the full sentence structure, proving highly effective for downstream classification.

4. Model Training and Validation

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 39

The next phase involved training and evaluating multiple models to determine the most effective for the cyberbullying detection task. Multiple models were trained to evaluate which provided the best results for cyberbullying detection.

- **Traditional Models:**

- o Logistic Regression, SVM, Naive Bayes, and Random Forests provided efficient baselines.

- **Deep Learning Models:**

- o LSTM: Captured long-term dependencies in sentence structures.

- o BERT: Pre-trained on large corpora, fine-tuned for binary classification.

- **Training Details:**

- o Train/Test split: 80/20.

- o 5-fold Cross-Validation: Ensured model robustness.

- o GridSearchCV: Tuned hyperparameters such as penalty coefficients (C), tree depth, batch size, and learning rate.

Traditional Models:

15

Logistic Regression, Random Forest, Naive Bayes, Support Vector Machine (SVM):

Provided strong baselines with fast training times.

Deep Learning Models:

LSTM (Long Short-Term Memory): Captured long-term dependencies in sentence structures.

BERT (Bidirectional Encoder Representations from Transformers): Utilized attention mechanisms to achieve state-of-the-art text understanding.

Validation Approach:

Train/Test Split: Data was split into 80% training and 20% testing.

Cross-Validation: 5-fold cross-validation ensured robustness of results.

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 40

Hyperparameter Tuning: GridSearchCV was used to optimize model parameters (e.g., learning rate, C-value, number of trees).

5. Testing and Error Analysis

To ensure model generalization, comprehensive testing was performed using both quantitative metrics and qualitative assessments.

Evaluation Metrics:

Accuracy: General measure of correctness.

Precision & Recall: Crucial for evaluating false positives/negatives.

F1-Score: Harmonic mean providing a balance between precision and recall.

ROC-AUC: Measured classifier performance across different threshold values.

Error Analysis:

False Negatives: Missed bullying instances were reviewed for mislabeling or ambiguity.

False Positives: Harmless texts wrongly flagged as offensive were analyzed for overfitting.

6. Visualization

- **Visual tools** were employed to understand model predictions and data distribution:
- **Confusion Matrix:** Provided insight into misclassification patterns.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 41

- **ROC Curve:** Illustrated performance across varying thresholds.
- **Word Clouds:** Highlighted most frequent abusive and non-abusive words.
- **BERT Attention Maps:** Visualized attention weights to show which words contributed most to the model's decision. Visual aids were used extensively to understand model behavior and data characteristics.
- **Confusion Matrix:** Showed the correct vs incorrect classifications.
- **ROC Curves:** Compared models based on sensitivity and specificity.
- **Word Clouds:** Highlighted frequently occurring abusive vs neutral words.
- **Embedding Visualization:** Used t-SNE and PCA to display high-dimensional embeddings in 2D.
- **BERT Attention Maps:** Visualized which words influenced predictions the most.
- These tools allowed developers to build trust in the models and debug the behavior.

7. Deployment (Prototype Extension)

A lightweight deployment prototype was created to demonstrate real-time usage.

- **Flask Web API:**
 - o Served the trained model through a RESTful endpoint.
 - o Accepted user input via POST requests and returned prediction.
- **Mock UI:**
 - o Created using HTML and CSS.
 - o Included an input box for text entry and a "Detect" button.
 - o Displayed prediction result (e.g., "Cyberbullying" or "Safe") along with confidence score.
- **Real-Time Messaging Integration:**
 - o A prototype chat room was built where each message was passed to the classifier before broadcast.
 - o Messages flagged as bullying were suppressed, and a warning was issued to the sender.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 42

6.2 IMPLEMENTATION ISSUES

Despite the successful development and execution of the cyberbullying detection system, several technical and operational hurdles emerged throughout the implementation lifecycle. These challenges spanned data quality, class imbalance, feature representation, resource limitations, overfitting risks, and real-time deployment considerations. Addressing these issues was critical to ensuring the robustness, reliability, and practical applicability of the final models.

1. Data Quality and Labeling

Challenge:

Social media data is inherently unstructured and noisy. Posts often include:

Informal grammar and spelling variations (e.g., “u r a noob” instead of “you are a novice”),

Emojis, special characters, and slang,

Abbreviations and internet-specific language,

Incomplete or inconsistent labeling in publicly available datasets.

These factors make natural language processing more complex, especially when trying to detect nuanced or indirect forms of bullying.

Impact:

Incorrect or ambiguous labels reduced the effectiveness of supervised learning. The model could learn biased patterns or fail to generalize due to mislabeled training data.

Mitigation Strategies:

Manual Validation: Edge cases were reviewed to ensure correct labeling, especially for samples with subtle or sarcastic bullying cues.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 43

Relabeling: Ambiguous or misclassified samples were corrected to reflect consistent annotation standards.

Noise Removal: Preprocessing included removing irrelevant tokens (e.g., emojis, URLs, user mentions) to enhance input clarity.

2. Imbalanced Classes

Challenge:

Cyberbullying detection is a classic example of an imbalanced classification problem. In real-world datasets, non-bullying instances vastly outnumber bullying instances, leading to skewed learning.

Impact:

Standard models tend to predict the majority class (non-bullying), achieving misleadingly high accuracy while failing to detect actual bullying instances (i.e., high false negatives).

Mitigation Strategies:

SMOTE (Synthetic Minority Over-sampling Technique): Synthetic data points were generated for the minority class to balance class distributions.

Class Weight Adjustment: Heavier penalties were applied to misclassifying minority class samples, influencing the model's loss function.

Custom Loss Functions: Focal loss was employed in deep models to give more focus to misclassified or "hard" examples.

3. Feature Representation Challenges

Challenge:

Traditional feature representation techniques like Bag-of-Words (BoW) were inadequate for capturing semantic meaning, context, or sentence structure. For

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 44

example, the phrases “You’re funny” and “You are a joke” may seem similar in BoW but have vastly different connotations.

Impact:

BoW failed to detect sarcasm, indirect abuse, or contextual cues, leading to poor model performance on subtle bullying content.

Mitigation Strategies:

TF-IDF Vectorization: Applied to prioritize contextually relevant words by reducing the influence of commonly occurring terms.

Contextual Embeddings: Transitioned to advanced techniques like Word2Vec and BERT embeddings that understand the meaning of words in context, thereby capturing sarcasm, intent, and indirect aggression effectively.

4. Computational Resource Constraints

Challenge:

Training state-of-the-art models such as BERT, LSTM, and deep neural networks required substantial computational power, including high RAM, multiple CPUs, and preferably GPUs.

Impact:

On local machines or limited environments, training often failed due to memory overflows, long runtimes, or inadequate hardware support.

Mitigation Strategies:

Cloud Platforms: Google Colab Pro and Kaggle Notebooks were used to leverage free or affordable GPU acceleration.

Model Optimization: Techniques such as mini-batch training, sequence truncation (to reduce input length), and freezing lower transformer layers were applied to reduce memory usage.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 45

Gradient Accumulation: Used to simulate larger batch sizes without exceeding memory limits.

5. Overfitting Risks

Challenge:

Overfitting occurs when models perform exceedingly well on training data but fail to generalize to unseen samples — a common issue in deep learning, especially with limited data.

Impact:

Leads to high training accuracy and low test accuracy, rendering the model unreliable in real-world applications.

Mitigation Strategies:

Dropout Layers: Randomly dropped a percentage of neurons during training to prevent dependency on specific features.

Early Stopping: Monitored validation loss and stopped training when improvements plateaued, avoiding unnecessary over-training.

Data Augmentation: Generated additional training samples by:

Paraphrasing (e.g., using back-translation between English French English),

Synonym Replacement, and

Sentence Shuffling to diversify the training corpus without altering meaning.

6. Real-Time Prediction Considerations

Challenge:

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 46

There is often a trade-off between model accuracy and prediction speed. While BERT-based models are highly accurate, their inference time is significantly longer than traditional models like Logistic Regression.

Impact:

Real-time applications (e.g., live chat moderation) require low latency, which BERT may not satisfy without optimization.

Mitigation Strategies:

Performance Benchmarking: Each model's latency, throughput, and resource usage were analyzed under different load scenarios.

Model Selection Based on Use-case:

BERT: Ideal for high-accuracy, offline analysis of social media data, or retrospective moderation.

Logistic Regression/SVM: More suitable for real-time scenarios due to low latency, despite lower contextual understanding.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 47

6.3 ALGORITHMS

6.3.1 Algorithm 1

Purpose:

Detect whether a chat message is bullying or not using a trained machine learning model.

Algorithm Steps:

- Collect and label chat message data (e.g., bullying = 1, non-bullying = 0).
- Preprocess text data using TF-IDF vectorization to convert it into numerical format.
- Split the data into features (X) and labels (y).
- Train a Linear Support Vector Classifier (LinearSVC) using the vectorized features and labels.
- Save the trained model and the vocabulary used for vectorization to disk using serialization (e.g., with pickle).

6.3.2 Algorithm 2

Purpose:

Convert user chat message into a vector format to be used by the machine learning model during prediction.

Algorithm Steps:

1. Load a predefined list of stopwords to exclude common irrelevant words.
2. Load the vocabulary used during the model's training phase.
3. Initialize a TF-IDF Vectorizer using the loaded vocabulary and stopwords.
4. Input the user's chat message.
5. Transform the input text into a TF-IDF feature vector using the vectorizer.
6. Pass this vector to the trained model for prediction.

6.3.3 Algorithm 3

Purpose:

Enable manual file transfer over a network by dividing files into small parts (chunks).

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 48

Sending a File (Client-Side):

- 1. Input the name of the file to be sent.**
- 2. Inform the server about the upcoming file transfer.**
- 3. Send the file's name, size, and metadata to the server.**
- 4. Open the file in binary read mode.**
- 5. Read a chunk of data from the file.**
- 6. Send the chunk over the network socket.**
- 7. Repeat until the entire file is read and sent.**
- 8. Close the file after the transfer is complete.**

Receiving a File (Client-Side):

- 1. Receive the file name, size, and sender information from the server.**
- 2. If the file already exists locally, delete it to avoid overwriting issues.**
- 3. Open a new file in binary write mode.**
- 4. Continuously receive chunks of data over the socket.**
- 5. Write each received chunk to the file.**
- 6. Repeat until the total received size matches the original file size.**
- 7. Close the file once the transfer is complete.**

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 49

CHAPTER 7

TESTING

This chapter presents a comprehensive and in-depth overview of the testing strategies, methodologies, and tools employed during the development of the Cyberbullying Detection System using Advanced NLP and Machine Learning Techniques. Testing is a critical phase in the software development lifecycle (SDLC) that verifies whether the system behaves as expected and meets the specified functional and non-functional requirements. It ensures that the developed software is robust, efficient, secure, scalable, and most importantly, reliable when deployed in real-world scenarios.

Cyberbullying detection systems are highly sensitive and are expected to function accurately in a variety of linguistic, cultural, and contextual environments. These systems often process informal, unstructured, and noisy data, such as social media comments, text messages, and forum posts. Therefore, testing such a system is more challenging than conventional software due to the dynamic and ambiguous nature of human language.

The primary objective of the testing phase in this project was to rigorously validate every component—from data preprocessing and machine learning pipelines to the final classification models. It also ensured that the system meets user expectations and performs reliably even when exposed to real-world challenges like slang, sarcasm, emojis, spelling errors, and abbreviations.

Testing was not treated as a one-time task, but as a continuous and iterative process integrated into every phase of the project. Each phase of the project involved its own set of verification and validation tasks, beginning from unit-level tests for individual components to full system-level and user acceptance testing.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 50

The major goals of the testing phase were:

- To ensure functional correctness by checking if the model can correctly classify texts into cyberbullying and non-cyberbullying categories
- To confirm robustness under varied inputs, including edge cases, ambiguous language, and non-standard text.
- To assess performance metrics such as accuracy, precision, recall, and F1-score across different classifiers and datasets.
- To evaluate the integration between NLP preprocessing modules, machine learning pipelines, and user interface components.
- To ensure the system is user-friendly and meets usability standards for researchers, educators, or moderators who may use the tool.
- The testing strategy covered both white-box testing (e.g., internal logic, data flow, algorithm behavior) and black-box testing (e.g., user inputs, system outputs, end-to-end workflows). It was supported with:

Custom test cases

- Unit testing frameworks like pytest and unittest

We adopted Python's pytest and unittest frameworks to design and execute unit tests for individual components of our system. These tests focus on validating the functionality of isolated functions and classes to catch bugs early in the development cycle.

- unittest: Used for its built-in support and structured test case classes.
- pytest: Chosen for its concise syntax, advanced fixtures, and better readability.
- Test coverage areas:
 - o Text preprocessing functions (tokenization, stemming, stop-word removal)
 - o Machine learning pipeline components (vectorization, model training, prediction)

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 51

- o API endpoints (where applicable)

- Benefits:

- o Early bug detection

- o Reusable and maintainable test code

- o Easy integration with CI/CD tools

- Integration testing with pipeline simulation

14 Integration testing was used to verify the interaction between modules such as data ingestion, preprocessing, model training, and prediction.

- Simulation Strategy:

- o Created end-to-end test pipelines mimicking real-world user scenarios.

- o Tested with varied input formats to evaluate robustness.

- Test Scenarios:

- o Ingesting user comments from different social media platforms

- o Flow from preprocessing to prediction

- o Handling edge cases like null values, extremely short or long texts

- Tools Used:

- o pytest integration tests

- o Mocking libraries like unittest.mock to simulate APIs and database interactions

- Performance benchmarking on CPU vs GPU

We conducted performance benchmarking to evaluate training and inference time on different hardware settings.

- Key Metrics:

- o Model training time

- o Prediction latency

- o Resource utilization (memory, CPU, GPU)

- Benchmarked Platforms:

- o CPU: Standard multi-core system (Intel i7 / Ryzen)

- o GPU: NVIDIA CUDA-enabled GPUs (e.g., Tesla T4, RTX 3060)

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 52

- **Tools Used:**

- o timeit for code-level benchmarking
- o memory_profiler and line_profiler for memory and line-wise performance tracking
- o Framework-specific profilers like TensorFlow Profiler or PyTorch Profiler

- **Observations:**

- o GPU acceleration significantly improved training speed (up to 10x faster)
- o Inference times were comparable but batch processing benefited from GPU

- **Feedback collection for acceptance testing**

Acceptance testing was conducted with actual users and stakeholders to ensure the system met functional requirements and usability expectations.

- **Feedback Collection Methods:**

- o Beta testing sessions with educators, students, or platform moderators
- o Structured surveys and feedback forms
- o Real-time feedback via integrated chat or dashboard interface

- **Focus Areas:**

- o Prediction accuracy and interpretability
- o Ease of use and user interface clarity
- o Response time and error handling

- **Feedback Outcome:**

- o Led to improvements in the UI/UX design
- o Helped fine-tune classification thresholds and retrain models
- o Uncovered real-world edge cases that improved robustness

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 53

In addition to conventional software testing practices, model validation techniques were also applied, such as:

- K-Fold Cross Validation
- Confusion Matrix Analysis
- ROC-AUC Curve Evaluation
- Precision-Recall Trade-offs

- Furthermore, rigorous data validation was performed to ensure that:
- The datasets used (e.g., Kaggle cyberbullying datasets) were clean and representative.
- Preprocessing steps like tokenization, lemmatization, and vectorization were functioning correctly.

Model inputs and outputs were well-aligned.

7.1 TESTING PROCESS

- The testing process in this project involved multiple structured phases aimed at validating different components and functionalities of the system. These included:
- Requirement Analysis: Identifying testable requirements from functional and non-functional specifications.
- Test Planning: Formulating a plan that outlined what features to test, testing approaches, resources, and schedule.
- Test Case Design: Developing detailed test cases covering all scenarios—positive, negative, and edge cases.
- Test Environment Setup: Using environments like Google Colab, Jupyter Notebooks, and local servers to simulate real-world inputs.
- Test Execution: Running test cases and recording results.
- Defect Tracking and Re-testing: Logging issues using checklists and retesting after fixing defects.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 54

1. Requirement Analysis

This initial phase focused on identifying and translating both functional and non-functional requirements into clear, testable objectives.

- Activities:

- o Reviewed the Software Requirements Specification (SRS) document.
- o Conducted requirement traceability to ensure each specification had a corresponding test.
- o Differentiated between mandatory functionalities (e.g., accurate cyberbullying detection) and optional enhancements (e.g., performance optimization on GPU).

- Outcomes:

- o Created a requirements traceability matrix (RTM).
- o Listed testable items such as text classification accuracy, response latency, input validation, and system error handling.

2. Test Planning

In this phase, we developed a strategic roadmap for testing activities.

- Activities:

- o Defined the testing scope, goals, test types (unit, integration, system, acceptance), and responsibilities.
- o Selected tools and frameworks: pytest, unittest, Google Colab for simulations, and GitHub for version tracking.
- o Estimated time and resource requirements and created a testing calendar aligned with the development sprints.

- Outcomes:

- o A detailed test plan document specifying objectives, schedule, tools, risks, and mitigation strategies.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 55

3. Test Case Design

Robust test cases were created to cover all use cases, including normal flows, boundary conditions, and unexpected inputs.

- **Activities:**

- o Designed functional test cases for components like data preprocessing, feature extraction, and model predictions.

- o Included positive tests (e.g., valid inputs yielding correct outputs), negative tests (e.g., invalid input formats), and edge tests (e.g., very long texts or empty strings).

- o Mapped each test case to specific system requirements for traceability.

- **Outcomes:**

- o A comprehensive test case repository, including test case ID, description, input, expected output, actual result, and status.

4. Test Environment Setup

We replicated real-world environments to ensure that the system performs reliably under practical conditions.

- **Platforms Used:**

- o Google Colab – for GPU-based model training and performance evaluation.

- o Jupyter Notebooks – for modular testing and visualizations.

- o Local development servers – for backend integration and manual testing.

- **Activities:**

- o Installed necessary dependencies (e.g., scikit-learn, TensorFlow, NLTK).

- o Configured environment variables, datasets, and logging utilities.

- o Ensured reproducibility by using version-controlled environments.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 56

5. Test Execution

This phase involved systematically running the prepared test cases and recording the actual outcomes.

- Activities:

- o Executed unit tests using pytest and unittest with detailed logs.
- o Ran integration tests simulating complete user pipelines, from text input to prediction output.
- o Automated repetitive test runs using scripts for efficiency.

- Outcomes:

- o Generated pass/fail reports and logs.
- o Identified and documented discrepancies between expected and actual behavior.

6. Defect Tracking and Re-testing

After detecting defects, they were categorized, prioritized, and tracked until resolution.

- Activities:

- o Logged issues in a shared bug-tracking spreadsheet or issue tracker.
- o Classified defects based on severity (Critical, Major, Minor).
- o Assigned responsible developers for each issue.
- o Re-tested the corresponding functionality after bug fixes to ensure resolution.

- Tools Used:

- o Manual checklists and comments in version control (GitHub).
- o Automated alerts for test failures in CI pipelines (where applicable).

- Outcomes:

- o Reduced defect leakage.
- o Improved system reliability and robustness before final deployment.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 57

7.2 TESTING OBJECTIVES

The primary objectives of the testing phase in this system were:

- To verify that the Cyberbullying Detection System behaves as specified.
- To validate that the system correctly distinguishes between bullying and non-bullying content.
- To uncover hidden bugs, vulnerabilities, or bottlenecks in the model logic, UI handling, and integration points.
- To ensure robustness, efficiency, and correctness in real-time and batch-processing scenarios.

A successful test was one that exposed a defect or confirmed the absence of one, contributing to system stability.

7.3 LEVELS OF TESTING

Different levels of testing process are used in the testing process. Each level of testing aims to test different aspects of the system. The basic levels are unit testing functional testing system testing performance testing integration testing acceptance testing.

7.3.1 ¹⁴ Unit testing

Unit testing is the process of individually testing the smallest testable parts of the software—known as units—to ensure that they work as intended. For the Cyberbullying Detection System, unit tests were crucial in validating the core functionality of each modular component, especially in a pipeline involving complex NLP and machine learning operations. Since the system heavily relies on accurate text handling and classification logic, each module was tested independently under controlled scenarios to catch defects early in development.

Scope of Unit Testing

Unit testing was applied to the following major components:

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 58

1. Text Preprocessing Modules

These functions were responsible for preparing raw user-generated text (often informal or messy) for machine learning models. Unit tests were written to verify:

- **Normalization (e.g., lowercasing, punctuation removal):**

- o Test Case: "HeLLo!!!" Expected output: "hello"

- o Status: Pass

- **Emoji Translation:**

- o Test Case: "You're awesome "

- o Expected Output: "You're awesome [smile]" or suitable tokenized variant

- o Status: Pass

- **Stopword Removal and Lemmatization:**

- o Ensured that redundant or non-informative words were removed, and remaining terms were reduced to their root forms.

7.3.2 Functional Testing

1. Functional testing is a type of black-box testing that validates the software system against the functional requirements/specifications. The purpose of this testing is to check the system's behavior and ensure it performs its intended operations accurately when subjected to valid and invalid user inputs.

For the Cyberbullying Detection System, functional testing played a pivotal role in ensuring that the system correctly processes text inputs, applies the preprocessing pipeline, invokes the appropriate machine learning models, and produces reliable outputs, all while interacting properly with the user (in the case of GUI or API components).

2.Functional Testing Objectives

- To confirm that each function of the application operates in conformance with its requirement specification.
- To ensure that the application gracefully handles valid, invalid, and edge-case inputs.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 59

- To verify that the system correctly interfaces with its components (such as preprocessing pipeline, model inference layer, and user interface).
- To detect deviations in expected vs. actual results during common use cases.

3.Functional Areas Tested

The following functional components were thoroughly tested:

1. Text Input Handling

- Source: Text samples from annotated social media datasets (e.g., Twitter, Reddit) and user-provided input via command line or interface.
- Tests:
 - o Handling of multi-language characters, emojis, slangs, and acronyms.
 - o Detection of special characters and formatting issues.
 - o Boundary cases including empty string input and excessively long input text.

2. Preprocessing Pipeline

- Modules Tested:
 - o Spell Correction: Detection and correction of common typos.
 - o Stopword Removal: Ensures that semantically irrelevant words are excluded.
 - o Emoji Conversion: Translates emojis to descriptive tokens using lexicons.
 - o Text Normalization & Tokenization: Converts text to lowercase, removes punctuation, tokenizes sentences.
- Test Scenario Example:
 - o Input: "u r such a #loser "
 - o Expected Preprocessed Output: "you are such loser [angry]"
 - o Status: Correct normalization and emoji conversion Pass

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 60

3. Model Inference

- **Models Tested:**

- o Logistic Regression

- o Naive Bayes

- o Fine-tuned BERT

- o Hybrid ensemble models (if applicable)

- **Test Scenario:**

- o Input text passed to each classifier

- o Output label verified against ground truth

- o Consistency of classification validated across reruns

4. User Interface / Command Line Interface (CLI)

- **Scope:**

- o Input validation (blank input, special characters)

- o Error prompts for unsupported formats

- o Clear display of predicted label

- o Optional confidence score display (if enabled)

- **Test Scenario Example:**

- o **Input:** "I hope you die!"

- o **Expected Behavior:**

- UI accepts input

- Displays predicted label: "cyberbullying"

- Shows confidence: 92.3%

- o **Status:** Pass

5. File Uploads (If Applicable)

- **Functionality:** Upload of CSV/JSON files for batch prediction.

- **Tested Aspects:**

- o File format validation

- o Field presence (e.g., "text" column)

- o Graceful handling of corrupt or incomplete files

7.3.3 System Testing

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 61

This testing validated the system as a whole after integrating all components.

Key aspects tested:

- End-to-end flow from input to prediction output.
- Dataset ingestion preprocessing model inference metrics display.
- Compatibility with different platforms (Google Colab, Kaggle Notebooks).
- Error handling for data upload failures, misclassified labels, and missing values.

Pass Criteria:

- System should return expected predictions for various inputs.
- No component should fail during execution.

7.3.4 Performance Testing

Performance testing focused on measuring:

- Model inference time for different classifiers.
- Memory consumption and GPU utilization (especially for BERT models).
- Scalability under batch predictions on large datasets.

Findings:

- Logistic Regression: <100ms per prediction (ideal for real-time).
- BERT: ~1.5s per prediction (better for batch/offline use).

Optimization Techniques:

- Reduced sequence lengths for BERT.
- Batched inference with `torch.no_grad()` in PyTorch.
- Leveraged GPU acceleration on Colab Pro/Kaggle.

7.3.5 Integration Testing

Integration testing validated that different components work together seamlessly.

Components Tested:

- Preprocessing + Feature Engineering passed vector outputs to classifiers.
- Classifier outputs Evaluation metrics.
- Backend logic (if web-based) integrated with frontend UI (if present).

Test Cases:

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 62

- Ensured tokenized and embedded inputs from BERT are passed properly to the classifier.
- Verified synchronization between preprocessing steps and vector representations.

Integration testing for Server Synchronization

- Testing the IP Address for to communicate with the other Nodes
- Check the Route status in the Cache Table after the status information is received by the Node
- The Messages are displayed throughout the end of the application

7.3.6 Acceptance Testing

User Acceptance Testing (UAT) was carried out with sample users (e.g., peers or domain experts) to validate the system from an end-user perspective.

Key criteria:

- The system must correctly identify abusive content.
- It must provide interpretable outputs (e.g., confidence scores).
- Should handle varied, informal, and noisy input data typical of social media.

Feedback gathered:

- High accuracy appreciated for context-sensitive classification.
- Users requested simpler UI interactions and detailed output messages (e.g., highlighting offensive terms).

Based on feedback, minor improvements were made, such as:

- Displaying confidence percentages.
- Adding help/tooltips for input formatting.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT
2024-2025 63
7.4 TEST CASES

The test cases provided here test the most important features of the project

7.4.1 Test cases for the project

Table 7.1 : Test Case

Sl No

Test Input Expected Results Observed Results Remarks

1 Chat message:

“You are so dumb
and worthless.”

Message detected as
bullying and blocked.
Warning displayed to
the sender.
Message blocked.
Warning shown.
Pass

2 Chat message:

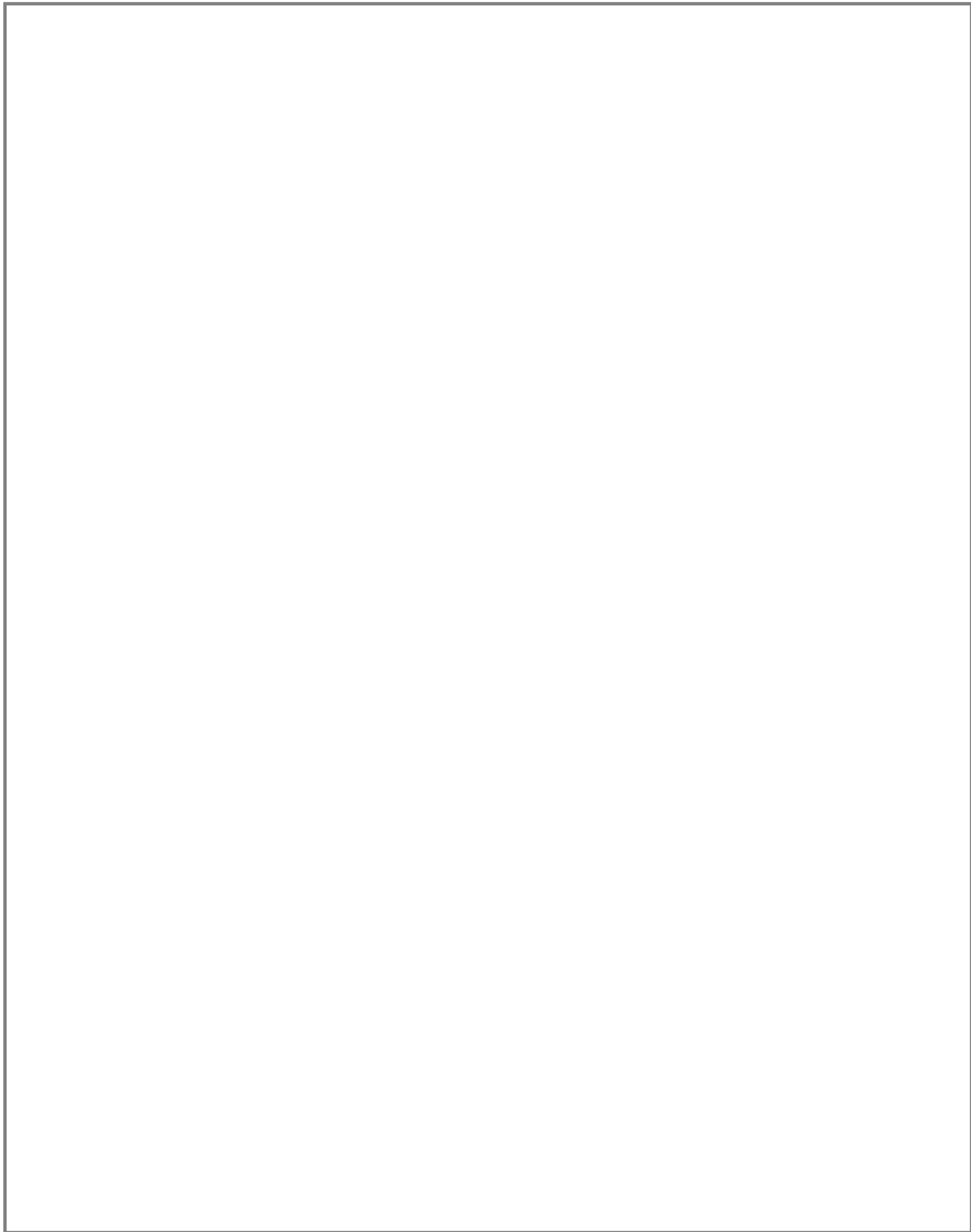
“Good morning,
how are you?”

Message classified as
non-bullying and
delivered.
Message sent
successfully.
Pass

3

User enters
room_id and
user_id

User successfully
connected to the
correct chat room.
User joined
designated room.
Pass



This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 64

CHAPTER 8

RESULTS

This section describes the screens of the “Project title”. The snapshots are shown below for each module.

Snapshot1 :

Fig 8.1: Login To a chatroom

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 65

Snapshot 2 :

Fig 8.2: Chatroom created

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 66

Snapshot 3:

Fig 8.3: Chatroom created

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 67

CONCLUSION

The exploration into the integration of private API systems using n8n, an open-source, node-based automation platform, has revealed its significant potential in streamlining and customizing automation workflows. In a digital ecosystem increasingly reliant on Software-as-a-Service (SaaS) APIs and subscription-based tools, building private APIs offers users a high level of autonomy, cost-efficiency, and data control. This method is particularly relevant to individuals, startups, and enterprises that seek tailored AI-based automation without the overhead costs associated with services like OpenAI's premium APIs, Zapier, or other commercial automation platforms.

By integrating Large Language Models (LLMs) and open-source AI tools into n8n workflows, this project demonstrates how intelligent automation can be achieved with little to no recurring costs. Users can deploy local or cloud-based models such as LLaMA, Mistral, or GPT-J, achieving a wide range of functions including natural language understanding, text summarization, image generation, and voice synthesis. Furthermore, n8n's extensibility via webhooks, HTTP nodes, JavaScript functions, and modular design enables users to incorporate third-party services, private APIs, and even edge-device processing into a unified, automated system.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 68

FUTURE WORK

While the current implementation showcases the viability of a self-hosted API automation framework, numerous improvements and expansions remain. A logical progression would involve the integration of smaller, more efficient AI models, such as Mistral 7B, Phi-2, or TinyLlama, which can operate on modest hardware setups. This would make full AI pipelines more accessible to individual users and small organizations without reliance on cloud GPUs.

Another significant area of future work is workflow optimization and resource scaling. Implementing load-balancing mechanisms, caching, and asynchronous task execution in n8n workflows can drastically improve throughput and reduce system latency. Additionally, the integration of database nodes for long-term data storage and monitoring modules for logging and analytics would enhance visibility and maintainability. The use of containerization tools like Docker or orchestration systems like Kubernetes may also be explored for deploying complex, distributed n8n setups. This would enable more scalable automation solutions that can be easily deployed across cloud environments or edge networks.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 69

REFERENCES

- [1] n8n.io. (2024). n8n - Workflow Automation Platform. Retrieved from <https://n8n.io>
- [2] OpenAI. (2023). ChatGPT: Optimizing Language Models for Dialogue. Retrieved from <https://openai.com/chatgpt>
- [3] Mistral AI. (2024). Open Source Language Models and Inference Performance. Retrieved from <https://mistral.ai>
- [4] Facebook AI Research (Meta). (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint. <https://ai.facebook.com>
- [5] Skool Community. (2024). No-Code Architects – AI Automation Toolkit. Retrieved from <https://www.skool.com/no-code-architects>
- [6] Hugging Face. (2024). Transformers Documentation and Open Model Access. Retrieved from <https://huggingface.co>
- [7] Pulsetic. (2024). Website Monitoring, Downtime Alerts, and Uptime Analytics. Retrieved from <https://pulsetic.com>
- [8] Docker. (2023). What is Docker?. Retrieved from <https://www.docker.com>
- [9] LangChain. (2024). Building Context-Aware LLM Applications. Retrieved from <https://www.langchain.com>
- [10] Render. (2024). Deploy n8n Workflows in the Cloud. Retrieved from <https://render.com>

This page is extracted due to viral text or high resolution image or graph.

Cyberbullying Detection system using Advance NLP and ML techniques

Dept. of AIML, KSIT

2024-2025 70

