**A**

**PROJECT REPORT**

**ON**

**“CYBERBULLY DETECTION”**

**SUBMITTED TO**

**SHIVAJI UNIVERSITY, KOLHAPUR**

**IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT**

**FOR THE AWARD OF DEGREE**

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

**SUBMITTED BY**

|  |  |  |
| --- | --- | --- |
| **MISS.** | **BHARATI BHASKAR PATIL** | **22UAD006** |
|  |  |  |

**UNDER THE GUIDANCE OF**

**Mr. S. P. Pise**

 



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE ENGINEERING**

**DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI**

**(AN EMPOWERED AUTONOUMOUS INSTITUTE)**

**2024-2025**

**D.K.T.E. SOCIETY’S**

**TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI**

**(AN EMPOWERED AUTONOUMOUS INSTITUTE)**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE ENGINEERING**



**CERTIFICATE**

**This is to certify that, project work entitled**

**“CYBERBULLY DETECTION”**

**is a bonafide record of project work carried out in this college by**

|  |  |  |
| --- | --- | --- |
| **MISS.** | **BHARATI BHASKAR PATIL** | **22UAD006** |
|  |  |  |
|  |  |  |
|  |  |  |

**is in the partial fulfillment of award of degree Bachelor of Technology in Artificial Intelligence and Data Science Engineering prescribed by Shivaji University, Kolhapur for the academic year 2024-2025.**

**MR. S. P. PISE**

**(PROJECT GUIDE)**

**PROF. (DR.) T. I. BAGBAN PROF.(DR.) L.S.ADMUTHE**

**(HOD AI & DS DEPT.) (DIRECTOR)**

**EXAMINER: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DECLARATION**

We hereby declare that, the project work report entitled “Cyberbully Detection” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.Tech.(AI & DS). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Name of the Student** | **PRN** | **Signature** |
| MR. | <<NAME OF STUDENT>> | <<PRN NO>> | <<SIGNATURE>> |
| MISS. | <<NAME OF STUDENT>> | <<PRN NO>> | <<SIGNATURE>> |
|  |  |  |  |
|  |  |  |  |

**ACKNOWLEDGEMENT**

With great pleasure we wish to express our deep sense of gratitude to Mr. S. P. Pise for his valuable guidance, support, and encouragement in the completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. T. I. Bagban for his cooperation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Artificial Intelligence and Data Science Department for their support, help, and assistance which they extended as and when required.

Thank you,

|  |  |  |
| --- | --- | --- |
| **Title** | **Name of the Student** | **PRN** |
| MR. | <<NAME OF STUDENT>> | <<PRN NO>> |
| MISS | <<NAME OF STUDENT>> | <<PRN NO>> |
|  |  |  |
|  |  |  |

**ABSTRACT**

Cyberbullying has become a pressing issue with the widespread use of digital communication platforms, posing serious threats to the mental health of individuals, particularly teenagers and young adults. This project presents the design and development of a Cyberbullying Detection System aimed at automatically analyzing user-input text to identify instances of offensive or bullying language. Leveraging Natural Language Processing (NLP) techniques, the system classifies textual content as either bullying or non-bullying, thereby enabling timely detection of harmful interactions. In cases where bullying is detected, the system provides users with the option to report the message for further review. The project follows the Agile Incremental Development Model, allowing for modular development, continuous testing, and iterative refinement. Developed as an individual effort, the project demonstrates disciplined task management and consistent self-evaluation. The solution is designed to be scalable and extensible, with potential future enhancements including multilingual support, real-time content monitoring, integration with social media platforms, and advanced user management functionalities. Overall, the system contributes to fostering a safer and more respectful online environment through the strategic application of technology to combat cyberbullying.

**INDEX**

1. **Introduction <<PgNo>>** 
   1. Problem definition <<PgNo>>
   2. Aim and objective of the project <<PgNo>>
   3. Scope and limitation of the project <<PgNo>>
2. **Background study and literature overview <<PgNo>>**
   1. Literature overview <<PgNo>>
   2. Investigation of current project and related work <<PgNo>>
3. **Requirement analysis <<PgNo>>**

a. Requirement Gathering <<PgNo>>

b. Requirement Specification <<PgNo>>

c. Use case Diagram <<PgNo>>

1. **System design <<PgNo>>**
   1. Architectural Design <<PgNo>>
   2. Flow Chart <<PgNo>>
   3. System Modeling <<PgNo>>
      1. Dataflow Diagram <<PgNo>>
2. **Implementation <<PgNo>>**

a. Agile Methodologies <<PgNo>>

b. Development Model <<PgNo>>

1. **Future Scope <<PgNo>>**
2. **References (Public repository GitHub source code links) <<PgNo>>**

**1. Introduction**

a. Problem definition

With the widespread use of digital platforms, users—especially students and young adults—are frequently exposed to harmful messages in the form of cyberbullying. Offensive content shared over social media and chat platforms can deeply affect mental health and emotional well-being. Traditional reporting mechanisms are often delayed or ineffective. There is a growing need for an intelligent, real-time solution that can detect harmful messages and enable users to take immediate action.  
This project addresses the problem by providing an automated cyberbullying detection system where users can log in, analyze messages, and report harmful content. It ensures a faster response by forwarding reported messages to the admin for review and necessary action. Currently, the system supports English language text only.

b. Aim and objective of the project

**Aim:**  
To develop a web-based system that allows users to analyze messages for cyberbullying, report offensive content, and enable admin intervention for timely action.

**Objectives:**

* To implement a message analysis module using a machine learning model trained on English cyberbullying data.
* To classify input text as bullying or non-bullying.
* To allow users to report messages that are flagged or perceived as harmful.
* To create an admin panel to review reported messages and take necessary actions.
* To ensure the system works efficiently for English-language input.

c. Scope and limitation of the project

Scope:

* Can be deployed in educational institutions and online platforms to reduce the impact of cyberbullying.
* Provides a streamlined process: login → message analysis → reporting → admin review.
* Enables a feedback mechanism where users actively participate in moderating harmful content.

Limitations:

* The system supports only English language inputs; other languages are not handled.
* It focuses solely on text-based content and does not process images, videos, or voice messages.
* The accuracy of the machine learning model is limited by the quality and size of the training dataset.
* The system does not automatically block users or take punitive actions—it only flags and forwards content to the admin.

**2. Background study and literature overview**

1. Literature overview

With the proliferation of social media and instant messaging platforms, cyberbullying has become a significant concern, especially among younger demographics. Early detection systems were largely rule-based, relying on predefined keyword lists to identify harmful language. However, these approaches often struggled with interpreting context, sarcasm, and evolving slang, resulting in limited accuracy and high false positives. To overcome these limitations, more recent research has leveraged machine learning and deep learning techniques to improve detection capabilities.

For instance, **Dinakar et al. (2011)** proposed a framework that utilized a combination of topic modeling and classification techniques to detect cyberbullying in online conversations. While their approach marked an important shift from rule-based to data-driven methods, it still relied on relatively shallow contextual analysis and was limited in handling nuanced or evolving language patterns. Similarly, **Badjatiya et al. (2017)** introduced deep learning models, specifically using word embeddings and recurrent neural networks, to detect hate speech in tweets. Their work demonstrated significant improvements in capturing semantic context and language complexity.

In comparison, the current project adopts Natural Language Processing (NLP) techniques to classify text input as bullying or non-bullying and provides an integrated user-reporting feature to enhance moderation. Unlike the earlier models which primarily focused on detection accuracy, this project emphasizes user engagement and ease of use, making it more suitable for real-time deployment in public platforms. However, gaps remain in scalability, language diversity, and real-time monitoring capabilities, which are areas targeted for future enhancements.

1. Investigation of current project and related work

The **Cyberbullying Detection System** developed in this project focuses on the automatic identification of offensive, harmful, or abusive text content entered by users. The project leverages **Natural Language Processing (NLP)** techniques to detect cyberbullying behavior in real-time and provides users the ability to report such instances for further review. This functionality aligns with the current research and technological trends aimed at ensuring digital safety and mental well-being.

#### **Current Project Analysis**

This project adopts a **text classification approach** wherein the input text is analyzed for specific patterns, keywords, sentiment, and contextual language. If the system determines the input to be bullying in nature, it alerts the user and provides a **"Report" feature**. The system is designed with scalability in mind and has the potential to integrate into chat applications, social media platforms, and forums. It emphasizes ease of use, accuracy in detection, and real-time response.

#### **Related Work Overview**

Several existing systems and academic studies have proposed mechanisms to detect cyberbullying:

* **Reynolds et al. (2011)** introduced a machine learning-based approach using SVM and decision trees to detect abusive content in YouTube comments.
* **Dinakar et al. (2011)** developed classifiers based on probabilistic models to detect teen cyberbullying from YouTube and Formspring posts.
* **Xu et al. (2012)** proposed using contextual and sentiment-based features to distinguish cyberbullying from casual online disagreements.
* **Badjatiya et al. (2017)** used deep learning methods like LSTM and Gradient Boosted Decision Trees for hate speech and bullying detection in tweets, achieving high accuracy levels.

While many of these systems show promise, a majority focus on large-scale platforms and require substantial computational resources. In contrast, the current project is designed to be **lightweight, adaptable, and user-friendly**, suitable for integration into smaller applications or as a standalone reporting tool.

**3. Requirement analysis**

a. Requirement Gathering

In the **Requirement Gathering** phase, the objective was to identify the key stakeholders, functional and non-functional requirements, and the technological stack necessary for developing an automated cyberbullying detection system. This phase is critical to ensuring that the final product meets the needs of both users and administrators.

#### **Stakeholders Identification:**

1. **End Users:**
   * Individuals who encounter or may be subjected to bullying behavior online.
   * Users who may inadvertently use inappropriate or harmful language.
2. **Moderators/Administrators:**
   * Responsible for reviewing flagged content and taking necessary action, such as banning or issuing warnings to offenders.
3. **Development Team:**
   * The team responsible for the technical implementation of the system, including the machine learning model and front-end/back-end development.
4. **Legal and Policy Makers:**
   * Authorities who set the standards for what constitutes bullying and ensure the system adheres to relevant laws and regulations.

#### **User Needs:**

1. **Real-Time Detection:**
   * Immediate feedback on whether the entered text is bullying or non-bullying.
2. **Easy-to-Use Interface:**
   * A simple interface for both users and moderators.
3. **Report Generation:**
   * An option to report bullying content directly, which will be reviewed by administrators.
4. **Privacy and Data Security:**
   * The system will handle sensitive data responsibly, ensuring that user privacy is maintained.
5. **Cross-Platform Functionality:**
   * The system should function effectively across different platforms, including web applications and social media environments.

#### **Functional Requirements:**

1. **Text Input:**
   * Users will enter or paste text into a simple text box.
2. **Text Classification:**
   * The system will classify text using machine learning algorithms trained on a labeled dataset of bullying and non-bullying content.
3. **Reporting Mechanism:**
   * If the text is identified as bullying, the user can report it for moderation.
4. **Actionable Feedback:**
   * The system will display a clear message stating whether the text is bullying, along with options for reporting it.

#### **Non-Functional Requirements:**

1. **Performance:**
   * The system should provide feedback within 3-5 seconds of receiving the input.
2. **Scalability:**
   * The system must scale efficiently to handle a large volume of text entries and reports.
3. **Accuracy:**
   * The classification model should achieve at least 90% accuracy in detecting bullying content.
4. **Security:**
   * All user data, including text and reports, must be stored securely with encryption to ensure privacy.
5. **Usability:**
   * The system should be intuitive and easy to use, requiring minimal user training or guidance.

b. Requirement Specification

The **Requirement Specification** defines the technical architecture and functional elements needed for the development of the cyberbullying detection system.

#### **Functional Specifications:**

1. **Text Input:**
   * Users will be able to input text manually via a web interface. The system will allow both short texts and longer passages to be analyzed.
2. **Text Classification Algorithm:**
   * The system will use machine learning models, trained on a large dataset of labeled bullying and non-bullying content, to classify the input text.
   * Natural Language Processing (NLP) techniques, such as tokenization and sentiment analysis, will be applied to understand the structure of the text.
3. **Reporting Feature:**
   * If the system detects bullying behavior, users will have the option to report the content for further review by a moderator or administrator. This report will include the text, the time it was entered, and the user ID.
4. **Feedback Mechanism:**
   * Users will receive instant feedback stating whether the text is classified as bullying or not, accompanied by a prompt to report bullying if detected.

#### **Non-Functional Specifications:**

1. **Performance:**
   * The text classification process should complete within 3-5 seconds to ensure real-time feedback.
2. **Scalability:**
   * The system should be able to handle thousands of user inputs and reports daily without performance degradation.
3. **Accuracy:**
   * The model must maintain at least 90% accuracy in distinguishing bullying text from non-bullying content.
4. **Security:**
   * The system must employ encryption for all data in transit and at rest. Access control mechanisms should be in place to prevent unauthorized access to sensitive data.
5. **Usability:**
   * The system should be designed with a focus on user experience, ensuring that both the text input and report mechanisms are straightforward to use.

#### **Hardware and Software Requirements:**

1. **Frontend:**
   * Web interface developed using **HTML**, **CSS**, and **JavaScript** for text input and report submission.
2. **Backend:**
   * The backend will be implemented using **Python** (Flask) or **Node.js** to handle text processing and interaction with the database.
3. **Machine Learning:**
   * **TensorFlow** or **PyTorch** will be used for building and deploying the machine learning model. NLP libraries such as **spaCy** or **NLTK** will assist in text processing.
4. **Database:**
   * A database (e.g., **MongoDB** or **MySQL**) will be used to store reports and logs securely.
5. **Cloud Hosting:**
   * The system will be hosted on a cloud platform like **AWS** or **Heroku** to ensure scalability and reliabil
6. Use case Diagram

A diagram of a cyberbullying system

Description automatically generated

**4. System design**

a. Architectural Design

System architecture is a conceptual model that describes the structure and behaviour of multiple components and subsystems.

A diagram of a computer program

Description automatically generated

1. Flow Chart

A diagram of a computer program

Description automatically generated

1. System Modeling
2. Dataflow Diagram

A diagram of a data analysis process

Description automatically generated

A diagram of a process

Description automatically generated

**5. Implementation**

1. Agile Methodologies

In this project, a research-driven adaptation of the **Agile Incremental Development Model** was employed to support individual work, promote continuous improvement, and maintain flexibility throughout the system's development. The methodology was tailored specifically for solo development in a research context.

* **Iterative Development:**  
  The system was incrementally built by breaking down the project into smaller functional components—such as the text input interface, NLP-based cyberbullying detection module, report handling logic, and result display. Each component was researched, developed, and tested independently before integration.
* **Research-Driven Sprints:**  
  Sprint cycles were planned around both implementation and exploration tasks. This included experimenting with different NLP models, evaluating classification techniques (like Naive Bayes), and testing various data preprocessing methods.
* **Personal Sprint Planning:**  
  Weekly goals were outlined using sprint boards and timelines. These sprints allowed for focused task execution while accommodating the trial-and-error nature of research and model fine-tuning.
* **Self-Evaluation and Reflection:**  
  At the end of each sprint, completed modules were reviewed for performance and usability. Research notes and logs were maintained to capture insights and evaluate alternate strategies for improvements.
* **User Feedback Integration:**  
  Prototype versions were shared with peers and academic mentors. The received feedback guided adjustments in both algorithmic performance and user interface design, ensuring the solution remained user-centered.
* **Documentation & Flexibility:**  
  Progress was tracked through personal journals and digital task boards. This lightweight documentation allowed flexibility to revisit earlier decisions, pivot approaches based on new research findings, and refine the system without disrupting overall flow.

This agile research approach ensured that the system evolved through continuous learning, experimentation, and structured personal development—key traits in both solo project execution and applied machine learning research.

1. Development Model

A diagram of a process

Description automatically generated

For this project, the **Agile Incremental Development Model** was adopted, as it aligns well with the solo development process while ensuring structured progress and continuous enhancement.

* **Modular Development:** The system was divided into independent modules such as user input interface, text analysis module, result display, and reporting mechanism. This approach ensured clarity and easier debugging during development.
* **Sequential and Iterative Progress:** Each module was planned, developed, tested, and refined in a sequential yet iterative manner. Adjustments were made based on observations and test outcomes at every stage.
* **Self-Driven Prototyping:** Early versions of the system were created and tested by me to validate functionality. Based on the outcomes, necessary refinements were incorporated.
* **Reduced Risk through Continuous Testing:** Each component was tested independently during its development phase, minimizing the chance of major errors later in the project.
* **Scalability and Extendibility:** The modular design allows the system to be expanded in the future. Features such as user authentication, sentiment analysis, or a dashboard for moderators can be integrated without major changes to the existing structure.

**6. Future Scope**

The Cyberbullying Detection System holds significant potential for future enhancement and real-world application. As technology advances and online interactions increase, this project can be extended and scaled in various meaningful ways:

* 1. Integration with Social Media Platforms:  
  The system can be integrated with platforms like Facebook, Instagram, or Twitter to automatically flag and filter bullying content in real-time.
* 2. Multilingual Support:  
  Currently supporting English, the model can be expanded to detect cyberbullying in regional and international languages using NLP-based translation and sentiment detection techniques.
* 3. Advanced Machine Learning Models:  
  Incorporating deep learning techniques such as LSTM or BERT can improve the accuracy and contextual understanding of bullying phrases.
* 4. User Authentication and Role Management:  
  A full-fledged user system with login, roles (user/moderator), and access control can be added to make the system more secure and personalized.
* 5. Real-time Monitoring and Alerts:  
  The system can be enhanced to monitor chat rooms, forums, and comment sections live and issue instant alerts or warnings to users or administrators.
* 6. Reporting Dashboard:  
  A dashboard for moderators or administrators to view reported messages, track patterns, and take necessary action can be developed.
* 7. Mobile App Development:  
  Creating a cross-platform mobile application will increase accessibility and allow users to report and detect cyberbullying directly from their smartphones.

**7. References (public repository GitHub source code links)**