CS 410 Lab 1 Outline

BullyBlock

Team Silver

11/6/2024

Table of Contents

List Of Figures

List Of Tables

1 Introduction

Cyberbullying has become a critical issue in K-12 educational environments, impacting not only students' mental and emotional health but also their academic performance and engagement with the school community. According to a report by the Cyberbullying Research Center, between 20-40% of students encounter cyberbullying at some point during their educational journey. These interactions, which range from harmful comments to ongoing harassment, can lead to severe emotional distress, decreased participation in school activities, and in extreme cases, heightened risk of school violence. As schools increasingly integrate digital platforms into daily learning, these platforms inadvertently create new venues for cyberbullying, where harmful interactions can be covert, difficult to monitor, and quick to escalate without timely intervention.

To address this complex problem, schools require a robust solution that not only detects cyberbullying incidents in real time but also allows for immediate intervention to maintain a safe and supportive educational environment. The need for such a solution is particularly urgent in platforms like the Canvas Learning Management System (LMS), which many schools use as a primary channel for communication and learning. Cyberbullying on these platforms often goes unnoticed by teachers and administrators, leaving students vulnerable to harmful interactions that can disrupt their academic experience and personal well-being. Traditional disciplinary approaches are insufficient to handle the covert and persistent nature of cyberbullying in digital spaces, highlighting the need for advanced technological interventions to identify and manage these incidents effectively.

To meet this need, BullyBlock is introduced as an innovative solution tailored specifically for K-12 educational environments using the Canvas LMS. BullyBlock is a web-based application designed to integrate directly with Canvas, leveraging advanced Natural Language Processing (NLP) algorithms to monitor and analyze communication data for signs of cyberbullying in real time. The system aims to detect harmful interactions as they occur, sending immediate alerts to designated school personnel to facilitate timely intervention. This solution emphasizes both student safety and educational quality by enabling proactive monitoring, minimizing the burden on educators, and creating a more inclusive learning environment for all students.

The development of BullyBlock will involve prototyping key features to demonstrate the feasibility and effectiveness of its cyberbullying detection capabilities. Through a structured prototyping approach, the product's essential functionalities—such as real-time monitoring, alert notifications, and a comprehensive dashboard for incident management—will be showcased to illustrate its impact on maintaining a safe digital learning environment. By refining these features and ensuring compliance with data privacy regulations like FERPA, BullyBlock offers a practical and scalable solution to one of today's most pressing educational challenges.

2 BullyBlock Product Description

BullyBlock is an innovative cyberbullying detection platform integrated with the Canvas LMS, specifically designed to create a safer digital environment for K-12 students. Its primary objective is to protect students from the psychological and academic impacts of cyberbullying by monitoring online communications in real time. Leveraging NLP technology, BullyBlock detects instances of harmful language, triggering immediate alerts to relevant school personnel, including educators and security staff. This real-time intervention capability empowers schools to address harmful interactions proactively, reducing the potential for escalation and supporting students' mental health and safety. In addition to alerts, BullyBlock provides a comprehensive incident management dashboard for administrators to track and review flagged incidents, making it a critical tool for K-12 schools dedicated to maintaining a secure and inclusive educational environment.

2.1 Key Product Features and Capabilities

The proposed solution is a web application seamlessly integrated with the Canvas Learning Management System (LMS) that leverages advanced Natural Language Processing (NLP) algorithms to monitor, analyze, and detect instances of cyberbullying in real-time. By continuously scanning digital interactions, such as messages, discussion posts, and comments, the system identifies harmful behavior patterns, triggering immediate alerts to school administrators and security personnel. This proactive approach ensures that cyberbullying is addressed promptly before it escalates, providing a safer digital and physical environment for students.

BullyBlock is significant because of its integration of advanced NLP algorithms directly within a platform already widely used in educational settings. It not only identifies harmful behavior instantly but also offers detailed analytics, creating a proactive and comprehensive solution. The solution's integration with Canvas ensures ease of adoption and scalability across K-12 schools.

The application enables early intervention and also offers detailed insights and analytics through a dashboard, empowering educators to make informed decisions and maintain a secure and supportive education atmosphere. This integration with Canvas ensures that the solution is both scalable and adoptable within existing digital infrastructures, making it a vital tool for safeguarding student well-being.

This solution addresses the problem of cyberbullying within K-12 schools by providing continuous monitoring and instant detection of harmful interactions on the Canvas platform.

2.2 Major Components (Hardware/Software)

The software will be developed for Canvas_LMS using the Canvas API which will be handled by using node.js express. User interactions on the Canvas platform will be monitored in real time to detect cyberbullying. The data gets retrieved from the incident management dashboard that will be stored in a database using mongoDB. The NLP_Module will trigger a notification to the people in charge of making disciplinary actions to the frontend that will be created in React.

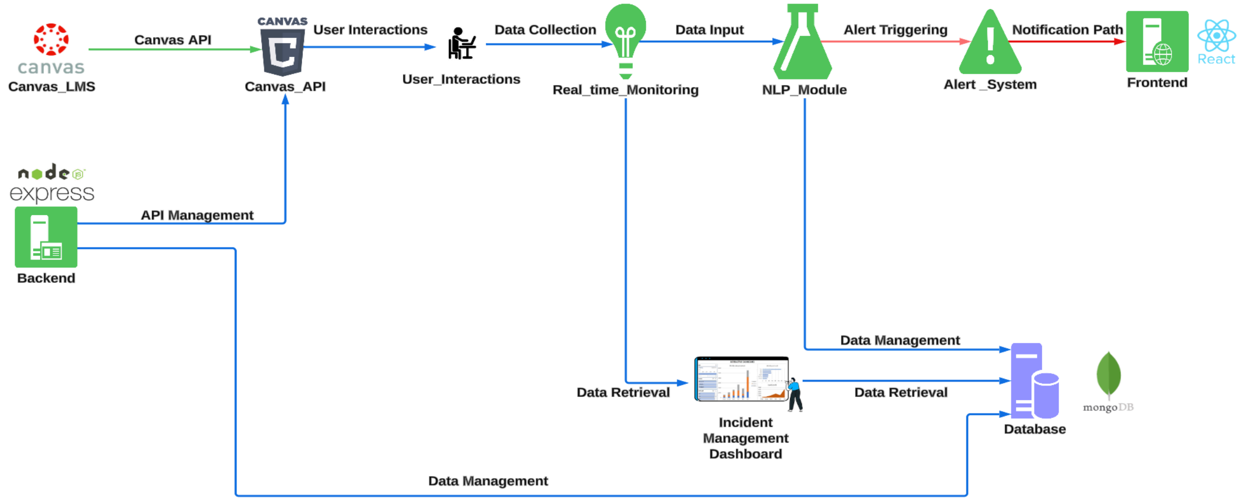

Figure 1: Major Functional Components Diagram.

3 Identification of Case Study

This product, the Cyberbullying Detection System, is being developed primarily for K-12 public schools. It is designed to integrate with the Canvas Learning Management System (LMS) to monitor and analyze communication for signs of cyberbullying in real time, enhancing the safety and educational quality within these environments. The primary reasons for its development include:

- Improving Student Safety: By detecting cyberbullying incidents early, the system aims to mitigate the adverse effects these incidents can have on students' mental and emotional well-being.
- Enhancing Educational Experience: By reducing the occurrence and impact of cyberbullying, the system helps create a more conducive learning environment where students can focus on their education without the distraction and distress caused by bullying.
- Assisting Educators and Administrators: It provides educators and administrators with tools to more effectively monitor and manage student interactions, allowing them to intervene promptly when bullying behavior is detected.

4 BullyBlock Product Prototype Description

The prototype will be somewhat reduced in capability compared to the end product, as shown in the appended table. The prototype will be more or less a scaled down version of the Real World Product such that it will be missing or contain incomplete features, considering the constraints of the project and for the purposes of demonstration. The appearance of the dashboard, for example, will be more simplified compared to the actual product. The implementation of the NLP algorithm will also be simplified to search for keywords or specific phrases, whereas the actual product would be able to analyze more complex strings, sentences, or conversations and make determinations.

Features & Functionality	RWP	Prototype
NLP-driven Cyberbullying Detection	Advanced NLP algorithms to detect subtle and overt bullying.	Basic NLP for detecting explicit keywords.
Integration with Canvas LMS	Seamless integration, with no impact on Canvas usability.	Basic integration, with some manual setup
Alert System	Automatic, real-time alerts to security personnel and educators.	Manual review required before sending alerts.
Incident Management Dashboard	Comprehensive dashboard with analytics and incident tracking.	Simple dashboard displaying recent incidents.
Data Privacy Compliance	Full compliance with FERPA	Basic data handling with some compliance gaps.

User Interface	Intuitive, accessible UI optimized for various devices.	Basic UI with limited mobile responsiveness
Scalability	Highly scalable, supporting multiple schools/districts	Limited to single schools or small districts
Customization Options	Extensive customization to fit district-specific policies	Few customization options available.

Table 1: Real World Product vs Prototype.

4.1 Prototype Architecture (Hardware/Software)

The prototype will include real-time monitoring implemented with Beautiful Soup, Scrapy, and Puppeteer to demonstrate BullyBlock’s ability to collect and process data in real time. The natural language processing utilizing NLTK, spaCy, and Stanford NLP will be capable of analyzing text between users and identifying potential bullying patterns. A basic implementation of the UI implemented with HTML and React will demonstrate how instructors will be notified and respond to instances of bullying in real time. A scaled down version of the MongoDB database will demonstrate how Bully Block will effectively handle data collection, management, and retrieval.

4.2 Prototype Features and Capabilities

The prototype will demonstrate a system that forms an alert when an instance of cyberbullying within the Canvas LMS is detected via Natural Language Processing (NLP) technology. This alert system will make it easier for cyberbullying to be detected and for appropriate disciplinary actions to be distributed in comparison to other detection methods such as teacher observation. The prototype will address all of our risk types with its risk mitigation techniques. The Customer and End User Risk will be mitigated by ensuring the application can only analyze content within the Canvas LMS. The Technical Risks will be mitigated by using the performance monitoring tool Prometheus for an expected scale of single schools or a small school district. The Security Risk will be mitigated by integrating multi-factor authentication. The Legal Risk will be mitigated by integrating basic data handling techniques, but will still experience some compliance gaps with FERPA. Overall, the prototype will analyze the Canvas LMS of single schools or a small school district using basic NLP. If the NLP detects patterns of keywords related to cyberbullying, alerts are sent to a dashboard for manual review before being passed on to school security or administrators.

4.3 Prototype Development Challenges

Some of the most challenging aspects of this project stem from missing knowledge. Many of the development tools, for example, are new to the team and will require some extent of research in order to use them effectively to produce the product. Creating algorithms to support this project will also be a challenge. For example, the team will need to research methods of implementing a Natural Language Processing algorithm to achieve some of the main functionality of the system, conversation analysis and interpretation. This might include choosing an open source generative artificial intelligence and integrating it with the rest of the project. Another challenge may be maintaining compliance with data laws. Any aspect of data handling and management will need to be checked in order to ensure that the system does not violate any laws. The significance of compliance scales as the scope of the system increases.

5 Glossary

Alert System: A feature that sends instant notifications to school staff, administrators, and security personnel when bullying behavior is detected, enabling timely intervention.

Backend: The server-side component of an application that handles business logic, database interactions, and processes requests from the frontend, typically developed using languages like Node.js, Python, or Java.

Beautiful Soup (Python): A library used for parsing HTML and XML documents, enabling easy extraction and manipulation of data from web pages.

BullyBlock: The system designed to monitor and detect cyberbullying behavior within the Canvas Learning Management System.

Canvas LMS: A widely used Learning Management System (LMS) in K-12 schools, where BullyBlock integrates to monitor and detect cyberbullying.

CI/CD: Continuous Integration and Continuous Delivery practices that automate the process of code integration and deployment, enabling developers to deliver updates and new features quickly and reliably.

Data Encryption: Security practice used to protect sensitive data within BullyBlock, ensuring unauthorized users cannot access the stored information.

Data Privacy Compliance (FERPA): Refers to adherence to the Family Educational Rights and Privacy Act, ensuring that student data is handled securely and used only within the bounds of the law.

Data Scraping: Algorithms to gather data from Canvas interactions for analysis.

Dashboard: A visual interface in BullyBlock that provides data and analytics on flagged incidents, helping educators review and manage bullying cases.

False Positives: Instances where benign interactions are incorrectly flagged as cyberbullying, potentially leading to unnecessary alerts and requiring further refinement of the detection model.

Frontend: The client-side portion of an application that users interact with directly, typically built using HTML, CSS, and JavaScript frameworks like React, designed to enhance user experience and interface functionality.

GitHub Actions: A CI/CD tool used to automate testing and deployment processes for the development of the BullyBlock application.

IDE: Integrated Development Environment, a comprehensive software suite that provides developers with tools for writing, debugging, and managing code, including features such as code completion, syntax highlighting, and version control integration.

Keras (Python): A high-level neural networks API that runs on top of TensorFlow, simplifying complex model building.

Languages: Programming languages such as Java, JavaScript, and Python that are utilized in the development of applications, each offering unique features and capabilities for implementing business logic and user interfaces.

Machine Learning (ML): A subset of artificial intelligence used to improve the cyberbullying detection algorithm by analyzing historical data to refine and enhance detection accuracy.

Machine Learning Model: A mathematical representation that learns from historical data to identify patterns and make predictions, thereby improving the accuracy of cyberbullying detection over time through continuous refinement and training.

MongoDB: A NoSQL database chosen for its flexibility in handling unstructured data, storing user messages, posts, incidents, and alerts generated by the system.

NLP (Natural Language Processing): A technology used by BullyBlock to analyze text data in real-time, enabling the detection of harmful language and patterns of bullying in online communications.

NLTK (Natural Language Toolkit): A Python library for text processing and analysis, supporting tasks like tokenization and part-of-speech tagging, essential for understanding text structure in the detection of bullying behavior.

Performance Monitoring Tool: A software application or system, such as Prometheus, that continuously tracks and assesses the performance and health of applications or services, providing insights into resource usage and operational efficiency to help identify and resolve issues.

Prometheus: A monitoring tool used in the project to track performance metrics and system efficiency, helping identify and resolve potential issues as the platform scales.

Puppeteer (Node.js): A library used for scraping dynamic content controlled by JavaScript.

Real-time Monitoring: The ability of BullyBlock to actively scan communications as they happen, allowing immediate detection and alerting for potential cyberbullying incidents.

Scalability: The system’s capability to grow and handle more users or larger datasets without performance degradation, making it adaptable to different school sizes and districts.

Scrapy (Python): A Python framework for web scraping on a large scale.

Sentiment Analysis: A component of NLP that evaluates the emotional tone behind a message to help identify negative or harmful interactions.

Stanford NLP: A comprehensive suite of NLP tools available in Python and Java, offering a range of linguistic processing capabilities, such as named entity recognition, which enhances the system's accuracy in identifying complex interactions.

Testing Frameworks: Tools, including Jest and Mocha, used to create and run automated tests for JavaScript applications, ensuring that code behaves as expected and maintaining software quality through systematic testing processes.

UI: User Interface, the space where user interaction with the application occurs.

UX: User Experience, the overall experience of a user when interacting with the application.

Version Control: A system, such as GitHub, that helps manage changes to source code over time, allowing multiple developers to collaborate effectively and track modifications to the codebase.

VADER (Python, within NLTK): A sentiment analysis tool specialized for analyzing sentiments in social media text, useful in assessing the emotional context of Canvas interactions.

6 References

Express.js. (n.d.). *Express*. https://expressjs.com/

Hinduja, S., & Patchin, J. W. (2018). *Cyberbullying identification, prevention, and response*. Cyberbullying Research Center. https://www.cyberbullying.org

IBM Corporation. (n.d.). *IBM Watson Natural Language Understanding*. https://www.ibm.com/cloud/watson-natural-language-understanding

MongoDB, Inc. (n.d.). *MongoDB*. https://www.mongodb.com/

Natural. (n.d.). *A general natural language facility for Node.js*. https://github.com/NaturalNode/natural

NLTK :: Natural Language Toolkit. (n.d.). *NLTK: Natural Language Toolkit*. https://www.nltk.org/

Node.js Foundation. (n.d.). *Node.js*. https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

Prometheus Authors. (n.d.). *Prometheus*. Prometheus. https://prometheus.io/

React documentation. (n.d.). *React: A JavaScript library for building user interfaces*. https://reactjs.org/

Securly. (n.d.). *Product Briefs.* https://ww.securly.com/product-briefs

University of Colorado Denver. (2020). *Cyberbullying Detection System*. https://engineering.ucdenver.edu/current-students/capstone-expo/archived-expos/spring-2020/computer-science/csci14-cyberbullying-detection-system.edu

U.S. Department of Education. (1974). Family Educational Rights and Privacy Act (FERPA), 20 U.S.C. § 1232g; 34 CFR Part 99. Retrieved from https://studentprivacy.ed.gov/ferpa