


# J. John Bennet

## J. John Bennet - research\_paper

 Quick Submit Quick Submit Presidency University

---

### Document Details

**Submission ID****trn:oid::1:3249657501****Submission Date****May 14, 2025, 1:02 PM GMT+5:30****Download Date****May 14, 2025, 1:07 PM GMT+5:30****File Name****research\_paper.pdf****File Size****293.5 KB****9 Pages****2,417 Words****13,966 Characters**

## \*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

**Caution: Review required.**

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

### Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

## Frequently Asked Questions

### How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (\*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

### What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



## BERT VS CYBERBULLYING: CREATING A SAFER DIGITAL SPACE

<sup>1</sup>Darshan MK, <sup>2</sup>Kusuma KN, <sup>3</sup>Rachita S, <sup>4</sup>Lipika Devaiah, <sup>5</sup>Anaiza Khan  
<sup>6</sup>Mr. J. John Bennet

Student, Department of Computer Science Engineering, Presidency University,  
Bengaluru, India

Professor, Department of Computer Science Engineering, Presidency University,  
Bengaluru, India

---

***Abstract: Cyberbullying's expanding prevalence endangers mental well-being, particularly among youth. Conventional detection schemes cannot detect subtle harassment such as sarcasm and coded messages. This paper presents an AI-based system based on BERT and deep learning to detect, understand, and respond to real-time cyberbullying attacks accurately. It exploits semantic awareness, sentiment analysis, and contextual information to detect threats on digital platforms. The system also gives automatic alerts, support utilities, and is well-integrated in multiple ecosystems, providing an active digital safety solution..***

---

## INTRODUCTION

Cyberbullying has become worse nowadays, as people can cover themselves behind screens and message people in a single stroke. Cyberbullying generally causes severe emotional trauma that remains long after abuse occurs. The majority of the current detection systems merely search for specific keywords but cannot determine the true meaning and tone of the message expressed. That's where our project "BERT vs Cyberbullying" enters the scene. We use advanced natural language processing and deep learning—specifically a powerful model called BERT to thoroughly understand the intent behind the message. This helps us to identify not just overt bullying, but indirect and subtle ones like sarcasm or coded language.

## LITERATURE REVIEW

In the “Detecting text-based cybercrimes using BERT” the authors came up with a system that keeps an eye on how users behave online. If something seems off, the system flags it as a possible threat using machine learning. It’s a smart way to spot cyber risks efficiently. This approach is quite similar to what the Cyber Threat Recommendation System does—it uses AI to give personalized security tips based on individual user behaviour, making security more relevant and effective.

In the “Detection of cyber security threats through the social media platforms” the researchers show how AI, especially machine learning, can improve how we detect the cyber threats in real time. The system can process both organized data (like logs) and unstructured data (like social media posts) to provide timely and accurate threat alerts. Similarly, the Cyber Threat Recommendation System uses GPT-4 to analyse live data, helping to quickly identify and suggest ways to counteract cyber threats, ensuring the system stays ahead of emerging risks.

Another paper, “*Detection of Cybersecurity Threats Through Social Media Platforms*,” looks at how AI, especially machine learning, helps us detect threats in real time. What’s impressive is that the system can handle both structured data, like logs, and unstructured data, like social media posts. That way, it catches threats quickly and accurately. This is also how the Cyber Threat Recommendation System works—GPT-4 is used to scan real-time data and suggest how to deal with threats right away, helping stay ahead of cybercriminals.

Finally, the paper “*Cyber Threat and Risk Detection Using ML*,” published in the *International Journal of Advance Research and Innovative Ideas in Education*, focuses on how AI is changing the way we handle cybersecurity decisions. These AI tools give customized advice on how to respond to threats, making the whole process faster and more effective. The authors also highlight the need for transparency and trust in these systems. The Cyber Threat Recommendation System builds on this idea, using GPT-4 to offer clear and practical recommendations so people and organizations can act quickly and confidently when a threat pops up.

## WORKFLOW OF DESIGN SYSTEM

The AI-powered cyber system provides a robust and integrated launching platform for secure and dignified interactions in the cyber world. It is three engines—often acting in tandem—to identify, remediate and reduce malicious activities in cyber spaces.

The elements are:

1. Detection Module:

Employing cutting-edge NLP methods, this module identifies, examines, and interprets text-based online interactions. Essentially, this module is segmented into three primary functions.

2. Detection of Offensive Content:

To locate and mark messages that have explicit, violent, or harmful content, using language as well as other information sources.

3. Sarcasm detection patterns:

It uses contextual and sentiment analysis for detecting sarcasm, a common subcategory of online hostility.

4. Harassment detection:

To identify and mark occurrences of bullying, threats, and other targeted aggression in an effort to develop a comprehensive strategy toward the identification of injurious conduct.

5. Intervention Module:

With the proactive manner of stepping in with the users, this module performs other tasks, including

Triggering alerts with an individual indulging in inappropriate behavior, cautioning them that potential action is being taken against them. Providing informative cues that promote a healthier communication trend and a better comprehension of the use of the internet and the severity of their actions.

6. Support Module:

Empowerment of the victims and delivery of desired support is achieved through this module.

7. Reporting Tools:

Simple and easy-to-use facilities for users to report cases of harassment or abuse.

8. Counselling Chatbots:

Automated conversational agents offering emotional support and advice to the victim.

## SYSTEM DESIGN AND METHODOLOGY

The cyber safety system is AI-driven and seeks to make cyberspace more respectful, supportive, and safe for everyone. It is founded on three main modules, which work together to spot malicious activity, respond appropriately, and help the victim.

### 1.Detection Module

This is the first line of defence. It uses state-of-the-art Natural Language Processing (NLP) to scan and analyze online conversations .When a user makes an obscene, violent, or offensive post, the system springs into action immediately.

- User Login: Users log into the web app through Firebase Authentication-either email/password or even Google and Facebook.When logged in,a secure session is created.
- Input comments: Logged-in members can leave comments, which are automatically screened.
- Google Perspective API: Each comment is analyzed by the Perspective API for toxicity classification i.e, insults, threats, or obscenity.
- Toxicity Score: Based on the degree of harm inflicted by the material, it's categorized (e.g.,bullying, obscene).
- Warning System: The system monitors toxic behaviour and provide users with up to three warnings:
  - 1<sup>st</sup> Offense: A gentle warning.
  - 2<sup>nd</sup> Offense:Another warning to do the right thing.
  - 3<sup>rd</sup> Offense: Final warning-this leads to a 7-day account suspension.
- Notifications: The user receives a notification on anything done on his/her account via email or app notifications.
- Admin Feedback Tools: Admins are provided with access to flagged comments,can adjust sensitivity levels, and see user appeals.
- Automation: Persistent monitoring of toxic behaviour and permanent ban of offending users.

## 2.Intervention Module

It's about rewarding improved behaviour in this part of the system.

- Warnings: The users are warned when they are acting against community guidelines.
- Learning Prompts: They also receive reflective messages intended to alert them to pause and improve their online behaviour.

## 3.Support Module

This module emphasizes the assistance of cyber harassment victims.

- Easy Reporting: Victims can easily and swiftly report offending behaviour.
- Counselling Chatbots: They can also look for emotional support and advice from AI-powered chatbots which have been created to offer comfort and care.

## API DESIGN

Method: POST

JSON response in the format of the recommended elective.

It looks for invalid or missing data in the API request. It provides sufficient error messages for the error like missing inputs and not found student IDs

### OpenAI GPT Integration:

Prompt Engineering: A structured prompt is made to ask the GPT API. This encompasses: Student ID.

Academic achievement (grades in different subjects).

Number of electives available.

API Interaction: The `openai.ChatCompletion.create()` function calls the OpenAI GPT API.

Response Handling: GPT's response is parsed and sent back as the elective required.

### Workflow of Implementation:

Create Flask app with CORS enabled to receive cross-origin requests

Import CSV dataset as Pandas Data Frame during application initialization

Handles user request via the endpoint /api/recommend

GPT Integration: Deals with input request handling including calling GPT API for fetching recommendation; then returns the recommendation.

Returns the result in structured JSON.

### Security:

Utilize os.getenv for environment variables tasked with handling OpenAI API key.

Inputs checked pretty well to prevent misuse and error

CORS: The registered domain only can talk to the backend.

### Unit Testing:

Test each individual module such as data retrieval, request validation, and calling of GPT API.

Testing Integration: Here the end-to-end process of a system needs to be tested where the dataset and the flask application should work seamlessly with the GPT API.

Performance Testing: Someone has to quantify time responses relating to API calls and process those GPT suggestions

Error Conditions: Incomplete or incorrect input, Invalid student id, GPT API fail.

Deployment:

Hosting: The application can be hosted in AWS, Heroku or even on local server.

Production Configurations: Debug mode should be off.

API key, environment settings: should be kept secret in the config file.

### Detailed workflow:

User Interface (Frontend) - The Students will use the system through a web interface (index.html). They enter their Student ID and list of available elective courses in the form. When the user submits, POST request is made to the backend API (/api/recommend) with the input data.

In the Backend API the request is processed where the Flask app (app.py) processes incoming POST requests to the /api/recommend endpoint. The API pulls student\_id and elective\_options from the JSON payload received from the frontend.

The backend is responsible for loading a CSV file with student data, such as their grades in different subjects and electives. The student\_id is utilized in retrieving the record of the corresponding student. In case the student\_id does not exist in the dataset, an error is sent to the frontend.



Elective-Specific Data Preprocessing - The student's performance in crucial elective-specific courses (e.g., Cloud Computing, Data Analysis, etc.) is pulled out of the dataset as a dictionary. The student's list of electives which is entered is cross-referenced with the dataset to confirm compatibility.

## RESULTS AND OUTCOMES

The model utilized GPT-4 to provide recommendation based on the academic performance of each student in your previous courses such as their student marks and their elective options. Therefore, each student receives personalized recommendation when they put in their Roll number and the elective options.

As the model suggests the student to choose the elective with appropriate reasoning on why the student must choose the suggested elective subject based entirely on the past academic record and gained marks. Thus an easier decision-making process.

The model captures the information from student performance dataset and utilizes that to input into GPT-4 such that the suggestions become data-driven. This perfectly aligns with working with structured data using AI even where data is sparse or missing.

The Flask application utilized in the model to design an interface through which the students can input their roll number and elective course choices to get the recommendation so that the students find it easy to interact with the application.

The design of the model utilizing the Flask application ensures that the model is capable of several users and datasets with minimal modifications making the model scalable and efficient for the users.

## CONCLUSION

The recommendation system is a good use of both machine learning and natural language processing techniques in the process of guiding students to more informed academic choices. The OpenAI GPT model is used, and a student's performance record from a given student is fed into the system to produce personalized elective recommendations. This methodology not only makes the decision-making process easier for the users but also demonstrates how the education system can be improved with the use of AI. The use of a Flask-based API allows the smooth interaction between the recommendation engine and the user interface, making it scalable and user-friendly. Further development might be done on enhancing the system's functionality to accommodate more than one language which can boost exposure to various

education institutions across different places and distribute it to a large number of people. Overall, this project demonstrates how AI can transform education and lead the way in innovation in the support and guidance system of schools.

## REFERENCES

- [1] *"A Review of Recommender Systems for Choosing Elective Courses"- International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 11, No. 9, 2020*
- [2] *"A Personalized Course Recommendation System Based on Collaborative Filtering" - Journal of Computer Science and Education, Vol. 13, No .4, 2021*
- [3] *"Hybrid Recommendation System for Course Selection Using Ontology and Machine Learning Techniques" - Journal of Computer Science and Education, VOL.72, No.37-48, 2017*
- [4] *"Elective Subject Selection Recommender System" - International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), Vol 5, 2017*
- [5] *"Personalized Course Recommendation System Based on Hybrid Approaches"- Procedia Computer Science, Gulzar A., Anny Leema, Vol.125, NO. 518-524, 2018*

