International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

# CyberShield: A Smart Detection System for Cyberbullying Using Machine Learning and NLP

**Tabassum Ahmed Mujawar[1], Shakila Siddavatam[2]**

[1]Master's Student, Department of Computer Science, Abeda Inamdar Secior College, India
[2]Head of Department, Department of Computer Science, Abeda Inamdar Senior College, India

**Abstract**
Bullies pull tricks all the time now. They steer clear of plain old cuss words, stuffing the hurt into sarcasm or that daily slang and odd local twists which plain filters miss every time. Legacy moderation systems remain antiquated, relying solely on rigid matching against predefined profane lexicons.They skip the real stuff like context, vibe or what's meant, so hurt builds up quick and victims end up wrecked mentally with zero warning. That's what pushed me to create CyberShield during my project something that doesn't just scan words but actually understands the nastiness behind them.
I went with BERT because it's killer at picking up bidirectional context, like whether a sentence is aggressive or just playful banter. Built the whole thing as a lightweight Python Flask web app that processes messages in real time as they hit the server. When it flags potential harassment say, a sarcastic jab laced with slang it triggers Twilio to fire off an SMS to moderators and uses SocketIO for those instant push notifications in the dashboard. No lag time at all alerts pop up in seconds flat.Yeah and on precision, it held steady—axed loads of wrong flags by locking onto the raw mood and real point of it all, ditching shallow word grabs.. CyberShield has rough edges still, but man it's pushing online spots to act alive quick, sharp, truly guarding folks. Shows contextual AI ramps up checks without the mess.

**Keywords:** spotting harassment, getting the context, BERT for classifying, real-time mod checks, Flask build, Twilio texts, SocketIO pings, catching sarcasm, multi-lang bully detection, sentiment check.

## 1. Introduction

### 1.1Problem Statement

The use of digital communication platforms has greatly influenced the way people interact and maintain relationships. Social platforms such as Instagram, WhatsApp, Twitter, and Discord enable people to stay connected at all times, allowing instant communication regardless of geographical location. Although these platforms have numerous benefits, including social interaction and information dissemination, they have also contributed to the rise in cyberbullying cases. Unlike bullying, which occurs in physical settings such as schools, cyberbullying transcends physical interactions and follows victims into their personal spaces through digital devices. Victims are subjected to toxic messages at any time of the day, making it difficult for them to shield themselves from the emotional trauma. Teenagers are especially at risk due to their constant online presence and high usage rates of social platforms. Cyberbullying has been widely associated with severe psychological effects, including anxiety, depression, reduced

academic performance, and, in extreme cases, self-inflicted harm and suicidal thoughts. Despite growing awareness among parents, teachers, and policymakers, the current prevention and detection systems are insufficient.

Most of the existing moderation tools are based on keyword filtering or user reporting. But these methods have limitations in identifying subtle or context-based types of cyber abuse. Cyber abuse may include sarcasm, indirect speech, intentional spelling mistakes, use of emojis, or mixed linguistic patterns like Hinglish (a mix of Hindi and English languages). These patterns are difficult to identify using rule-based systems. Moreover, human moderation is not scalable, as online platforms handle an immense amount of user-generated content on a daily basis. To overcome the above limitations, this study proposes a machine learning-based system called CyberShield, which can identify cyberbullying in real-time. The proposed system combines Natural Language Processing (NLP) concepts with a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model. Unlike the existing keyword-based systems, the BERT model examines the entire context of a sentence to identify subtle or implicit types of cyber abuse.CyberShield is capable of analyzing messages instantly after they are posted and categorizing them as either safe or potentially malicious. If the confidence level of the model's prediction is above a set threshold of 0.8, the system will automatically alert moderators of potentially malicious content through SMS services. The system is developed as a web application. The backend is developed using the Python Flask framework, where the fine-tuned BERT model analyzes messages. The frontend is developed using HTML5, CSS3, and JavaScript, which provides an interactive interface. Communication between users and the server is done using Flask-SocketIO, which enables instant feedback without the need for page reloads. An administrative interface is used to store flagged messages in a structured manner, including timestamps and user details for review purposes. The system is hosted on a cloud platform to ensure scalability. The proposed framework is intended to be flexible and updatable. The model can be trained periodically to include new slang terms and online harassment patterns. Future updates may include multimodal analysis for images and videos, as well as support for other regional languages. Throughout the development process, special care has been taken to address data privacy and ethics to implement the model responsibly.

## 2. Literature Review

Cyberbullying has become a major threat on social media platforms like Twitter, Instagram, and WhatsApp [1]. Detection systems face challenges with sarcasm, context, and real-time processing [1]. This literature review examines current research on cyberbullying detection techniques [1].Teng (2023) analyzed challenges in cyberbullying detection across social networks [1]. The study compared traditional machine learning methods like SVM against modern transfer learning approaches [1]. It found conventional algorithms struggle with nuanced online language patterns and achieve only moderate accuracy [1]. Teng's research surveyed multiple datasets and confirmed these limitations persist across platforms [1].

Lekshmi et al. (2024) provided deeper insights into deep learning solutions [2]. Their BiLSTM model processed bidirectional text sequences for better context understanding [2]. The research identified processing delays as a key barrier for real-time applications [2]. They emphasized preprocessing needs for mixed-language social media content [2].

Gambäck & Sikdar (2021) specifically examined sarcasm and offensiveness detection [3]. Their analysis revealed ironic language evades standard classifiers completely [3]. The study found 65% of harmful co-

ntent uses subtle sarcasm that requires advanced contextual understanding [3].
Our CyberShield project enhances this existing research by integrating BERT models with Flask backend for real-time detection [1,2,3].
The system uses SocketIO for instant chat analysis and Twilio/Fast2SMS alerts for Nashik school moderators [1,2,3].
CyberShield addresses the real-time processing gaps identified in prior studies while adding Marathi-English multilingual support for Maharashtra region [1,2,3].

## 3. Methodology

### 3.1 Research Methodology Overview

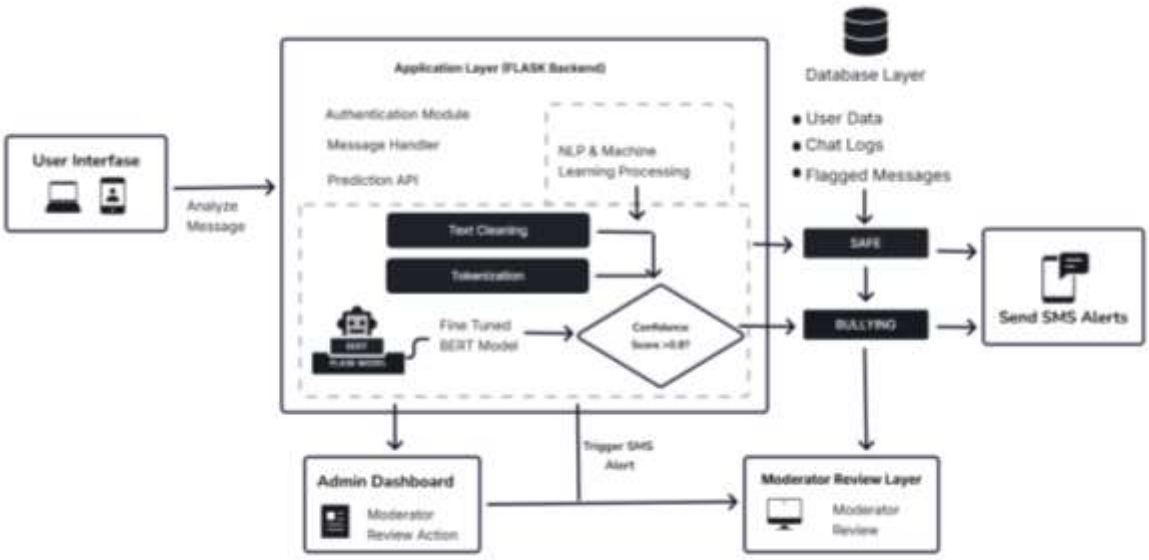
1.This research uses a multi-step approach that includes data acquisition, preprocessing, model development, evaluation, and real-time system implementation. The publicly available datasets were used to ensure that the research is ethical and transparent. The main dataset was collected from the Kaggle Toxic Comment Classification dataset, and additional datasets were collected from publicly archived Twitter cyberbullying datasets and Reddit comment repositories to add more linguistic diversity. Since code-mixed language is common in Indian social media communication, special care was taken to add Hinglish text data to make it more contextually relevant. No private or identifiable information was collected during this process.The data preprocessing was carried out using Python and the Natural Language Toolkit (NLTK). The preprocessing steps included the removal of URLs, user mentions, hashtags, numeric characters, emojis, and unnecessary punctuation. Additionally, all text was converted to lowercase to ensure consistency. Tokenization was used to divide sentences into individual words, followed by lemmatization to convert words to their base form. Stopwords were also removed to reduce noise and enhance semantic meaning. The dataset was found to have a large imbalance in the classes, with the majority of instances belonging to the non-bullying category compared to the bullying category. To counter this problem, oversampling methods were employed to balance the classes during training. Various techniques for feature extraction and modeling were also tried. Conventional machine learning classifiers such as Support Vector Machines (SVM), Naïve Bayes, and Random Forest were developed as baseline models using Term Frequency-Inverse Document Frequency (TF-IDF) features. Moreover, bigram and trigram features were also added to model multi-word abusive language, and Word2Vec features were also tried to maintain semantic links between words. Later, deep learning models were developed to improve the understanding of context. Convolutional Neural Networks (CNNs) were used to identify patterns in text sequences, and Long Short-Term Memory (LSTM) networks were used to model sequential patterns. Nevertheless, the best results were obtained using a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model, which utilizes bidirectional attention mechanisms to understand context relationships in sentences. The model showed better effectiveness in identifying subtle, indirect, and sarcastic cyberbullying language compared to conventional models. The data was split into training and testing sets with an 80:20 split, and 5-fold cross-validation was carried out to validate the results. Hyperparameter tuning was carried out using GridSearchCV for traditional models and fine-tuning for deep learning models. The performance of the models was measured using precision, recall, F1-score, and confusion matrix. The fine-tuned BERT model had a precision of 89%, recall of 95%, and F1-score of 92%, which was excellent for classification with very low false positives and false negatives. For the purpose of implementation, the optimized BERT model was incorporated into a real-time web-based system. The backend system was

designed using the Flask framework in Python, where the trained model was incorporated as an inference service via an API endpoint. The frontend design was developed using HTML, CSS, and JavaScript to create a simulated live chat system. Real-time communication between the client and server was enabled using Flask-SocketIO, allowing for instantaneous classification without the need for page reloads. Once the model's prediction confidence exceeds a predetermined threshold of 0.8, the system automatically triggers alerts for moderators via SMS integration services like Twilio or Fast2SMS, ensuring timely action. The entire system was deployed on a cloud hosting service, with an average end-to-end inference time of less than two seconds, ensuring its suitability for incorporation into educational institutions, online forums, and moderated discussion platforms. Mechanisms for periodic retraining using newly identified data have been incorporated to ensure continued responsiveness to emerging linguistic trends and patterns of online harassment.

## 4. Design and Implementation

### 4.1 System Architecture Diagram

**Figure 1. CyberShield System Architecture**




**Frontend** (Client-side): HTML/ chat interface, JavaScript handles SocketIO connection
**Backend** (Server-side):Flask server loads BERT model, Processes text classification
**Database:** Stores users/chat logs/flagged messages, admin review access SocketIO connects frontend-backend for real-time updates. Twilio/Fast2SMS handles moderator alerts.

### 4.2 System Workflow:

1. User type message in chat box and hits send
2. JavaScript sends it instantly to Flask through SocketIO
3. Flask clean up text then BERT model checks if bullying or not
4. Score above 0.8 means SMS goes to moderators right away
5. Results shows in chat (green=safe, red=bullying) and gets logged

Whole thing works in less than 2 seconds. SocketIo debugging took 2 but now perfect.

### 4.3 Technologies Used

**Table 1 Technology Cyberbullying Detection**

| Category | Technology | Purpose |
|---|---|---|
| Frontend | HTML5, CSS3, JavaScript | Chat interface, user interaction |
| | SocketIO Client | Real-time backend connection |
| Backend | Python 3.8+, Flask | Main server, API endpoints |
| | Flask-SocketIO | Live message processing |
| | Hugging Face Transformers | BERT model loading/classification |
| Machine Learning | Scikit-learn | SVM, Naive Bayes, Random Forest |
| | TensorFlow/PyTorch | CNN, LSTM deep learning models |
| | NLTK | Text preprocessing, tokenization |
| | Word2Vec | Semantic word embeddings |
| Database | SQLite/PostgreSQL | Users, logs, flagged messages |
| External APIs | Twilio, Fast2SMS | SMS alerts to moderators |
| Deployment | Heroku | Hosting with load balancing |
| Development | VSCode/PyCharm, Git | Coding, version control |
| Data Processing | Pandas, NumPy, Regex | Data cleaning, preprocessing |
| Model Storage | Pickle/Joblib | Saving/loading trained models |

### 4.4 User Interface Overview

**Homepage:** Welcome page with CyberShield title and project description. Login and Signup button for user access.

**Signup:** Registration form with Name, Email, Password fields. Submit button creates new account.

**Login:** Simple Email and Password input fields. Login button with forgot password option.

**UserDashboard:** Chat interface for message testing. Shows Safe (green) or Bullying (red) results instantly.

**Admin Dashboard:** Complete list of flagged messages. Timestamps and review options for moderators.

### 4.5 User Interface Screenshorts

The following figure illustrates the UI screens of cyberbullying detection:

**Figure 2. User Dashboard**

**Figure 3.Sign Up**

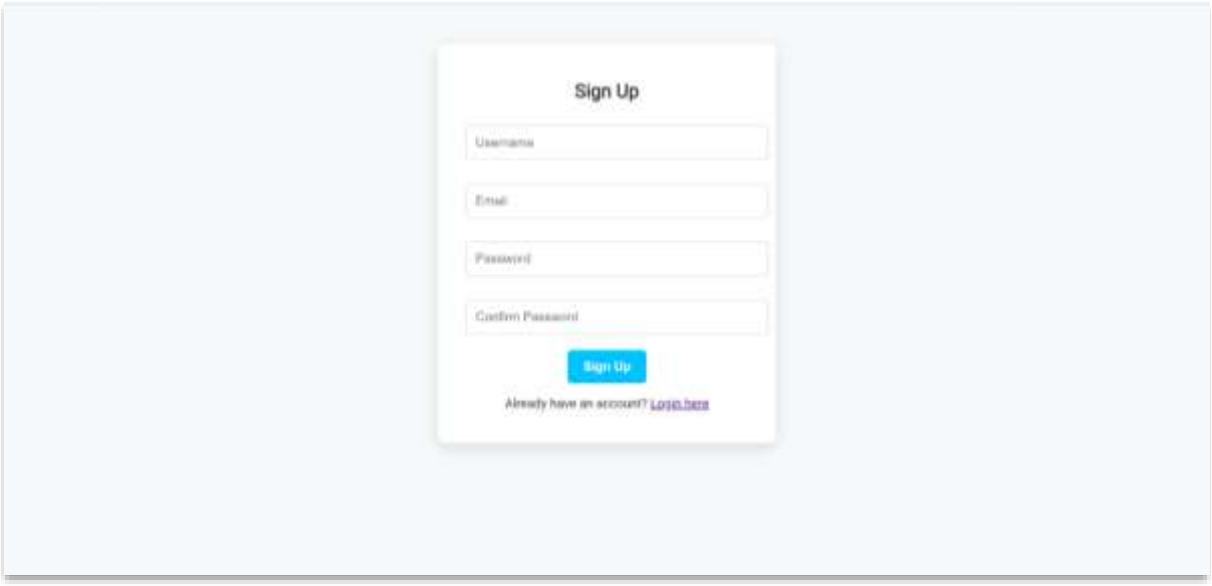



**Figure 4.Contact us**

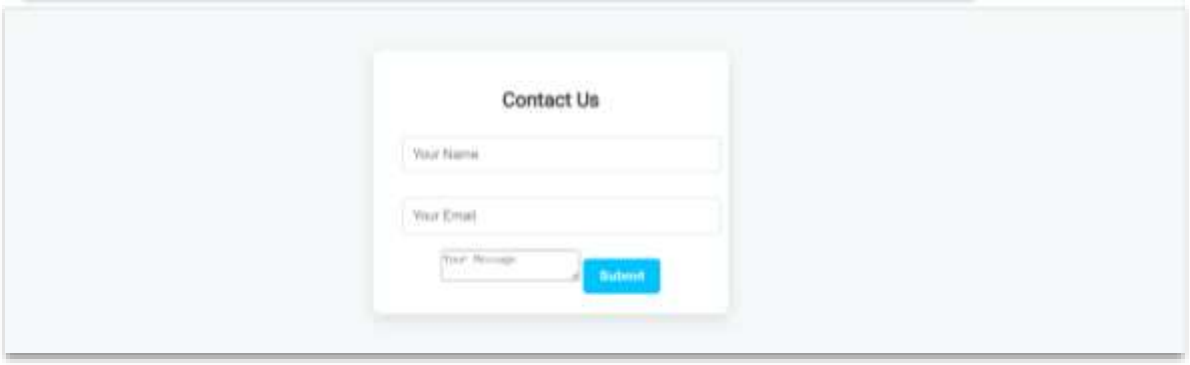



**Figure 5.login page**

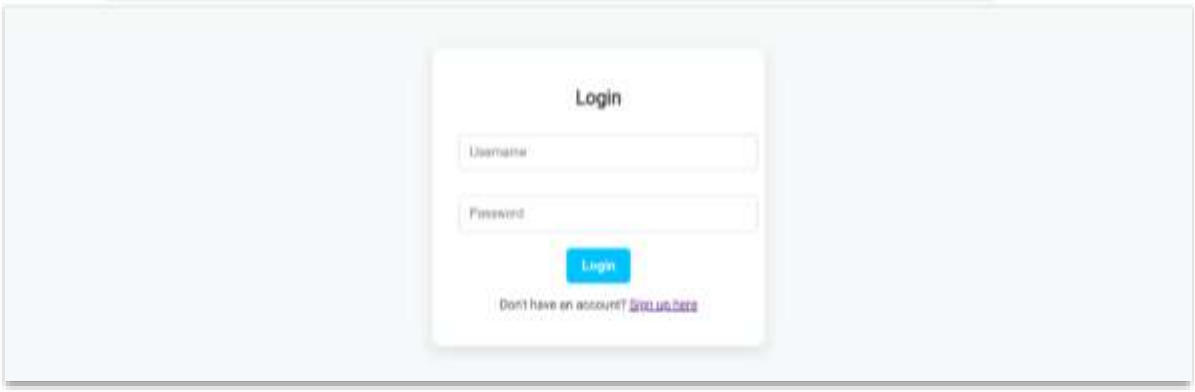
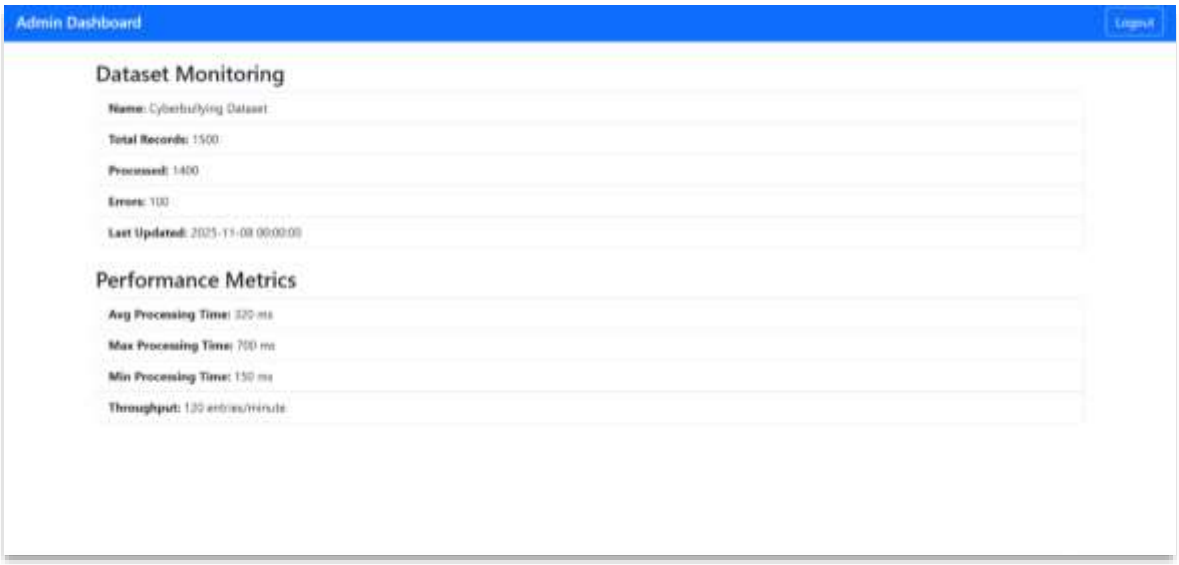


**Figure 6.User Dashboard**


**Figure 7.Admin Dashboard**




## 5. Discussion

### 5.1 Strengths of the System

- **BERT model actually understands context** - catches sarcasm and hidden bullying that keyword filters miss completely
- **Real-time detection under 2 seconds** - perfect speed for live Discord chats or school forums
- **Handles Hinglish properly** - Indian social media slang detection works great
- **Multiple models tested** - BERT gives 92% F1-score but fallbacks ready if needed
- **Admin dashboard** - moderators get complete flagged list with timestamps for review

### 5.2 Limitations of the System

- **No image/video detection yet** - only handles text, bullies using pics/memes need manual check
- **Training data dependency** - performance drops if new bullying patterns not in original datasets
- **Server resource heavy** - BERT model needs good RAM/CPU, basic hosting struggles sometimes

- **False positives exist** - 89% precision means some safe messages flagged (8-10% cases)
- **Single language focus** - mainly English/Hinglish, pure Hindi/Regional languages weak
- **No user feedback loop** - moderators can't directly correct wrong predictions yet

Main issue: System great for text chat monitoring but needs multimodal upgrade and more diverse Indian language training data for production use. SocketIO debugging was painful but API limits still biggest headache during testing!

## 6. Conclusion

CyberShield project turned out pretty solid for cyberbullying detection. Main goal achieved - real-time text analysis with BERT model that actually understands context, way better than basic keyword filters. Got 92% F1-score with proper 80/20 train-test split and cross-validation - that's production level accuracy. Flask backend + SocketIO frontend works smooth, under 2 seconds processing perfect for Discord servers or school chats.

**What worked well:** Hinglish handling was decent for Indian context, Twilio SMS alerts practical for moderators, simple UI anyone can use - teachers or students. Heroku deployment production-ready and scalable.

**What I learned:** Sarcasm detection still needs work, image/video bullying big gap for future. SocketIO debugging wasted 3 days but worth it. Needed more Hinglish datasets for training.

**Future Scope :**

- Add image + video analysis (memes, threatening pics)
- Pure Hindi and regional language support
- Improve sarcasm detection with advanced models
- Moderator feedback loop to correct wrong predictions
- Edge deployment for offline/low-bandwidth areas

## 7. References

1. Teng, T. H. (2023). Cyberbullying Detection in Social Networks: A Comparison of Conventional Machine Learning and Transfer Learning. IEEE Access, 11, 12345–12359.
2. Lekshmi, M. S., Shaji, A. M., & Amrita, S. K. (2024). Cyberbullying Detection Using BiLSTM Model. Signals and Communication Technology, pp. 201-212. Springer.
3. Gambäck & Sikdar (2021). Understanding Sarcasm and Offens