INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 4, April 2026 | ISSN: 2456-4184 | IJNRD.ORG

CHATSHIELD – REAL TIME BULLYING DETECTION AND PREVENTION SYSTEM USING NLP (NATURAL LANGUAGE PROCESSING), AND MACHINE LEARNING

M. PADMA PRIYA
Department of Computer Science and Engineering,
Bharath Institute of Science and Technology,
Chennai, India
padmapriya.cse@bharathuniv.ac.in

BADE SIVA RAMA KRISHNA NAIDU
Department of Computer Science and Engineering,
Bharath Institute of Science and Technology,
Chennai, India
rknaidu1750782@gmail.com

AKULA SAI VIGNESH KUMAR
Department of Computer Science and Engineering,
Bharath Institute of Science and Technology,
Chennai, India
akulasai456@gmail.com

BANDARU GANGEYA GURU PRASAD
Department of Computer Science and Engineering,
Bharath Institute of Science and Technology,
Chennai, India
bandarugangeya@gmail.com

ADIGARLA JAYANTH KUMAR
Department of Computer Science and Engineering,
Bharath Institute of Science and Technology,
Chennai, India
adigarlajayanthkumar2@gmail.com

***Abstract* — Cyberbullying and abusive communication has become a critical problem in the modern online chatting setting and tends to cause emotional and psychological harm to the users. The contextual abuse, sarcasm, slang phrases and multilingual chat messages can best be detected through the traditional means of moderation such as key word filtering and rule based systems which are not always effective. The paper proposes a real-time message monitoring and filtering application called CHATSHIELD, which determines an unhealthy communication in chat rooms using Natural Language Processing (NLP) and Machine Learning technologies. The system has text preprocessing steps like tokenization, normalization, stopword removal, and slang processing to format chat messages to be analyzed. They are transformed into numerical features by TF-IDF vectorization of the processed text and the classification of the features is performed by a Logistic Regression model to inform about the harmful content of a message. The system keeps off or filters the messages which are harmful depending on the classification outcome before it is relayed to the recipient making the communication between the users safe. Cyberbullying nature that the system detects as well as harmful message detection includes insults, threats, harassment, and hate speech among others. The system also offers the dual layer detection on client and server side to offer reliability in detection. It also has real time chat interface, multi-user communication, file transfer, unknown message detection and remote connectivity via secure tunneling so as to facilitate communication over various networks. According to experimental analysis, the proposed approach results in a precision of 90 and it turns out to be useful in identifying the damaging messages as well as low computational complexity needed in the chat systems in real time. CHATSHIELD framework offers a flexible and effective system of enhancing the safety and moderation of online communication systems.**

INDEX TERMS — Cyberbullying Detection, Natural Language Processing (NLP), Machine Learning, TF-IDF Vectorization, Logistic Regression, Real-Time Chat Filtering, Content Moderation, Text Classification, Socket Programming, Secure Tunneling, ChatShield.

I. INTRODUCTION

The digital spheres of communication became an inseparable element of daily life and offer people an opportunity to be in close contact though social media, messaging apps, and online communities. As much as these mediums add to connectivity and exchange of information, the sites are ever subjected to indecent and abusive language, which has a detrimental influence in the emotional status and online security of the users. Being aware that contemporary online communication does involve sarcasm, slang, emojis, spelling variations, and mixed-language sentences, the classic means of moderation, such as key word blocking and rule based thematic systems, merely do not identify them. The current advancements in Natural Language Processing (NLP) and machine learning have increased the ability of the textual data analysis with the aim of identifying unhealthy patterns of communication on the automatic basis. Moderation strategies are quite numerous but they are founded on the inactive filtering or offline analysis limiting the use of the model in the live chat whereby messages are sent and received in real times.

To address such concerns, this paper proposes CHATSHIELD, a real-time chat monitoring and filtering system, which uses NLP preprocessing, and machine learning to find and filter harmful messages. The system employs TF-IDF vectorization to carry out text preprocessing and feature extraction operations and then classifies the message with the help of a Logistic Regression model to assess the presence of harmful information in the message. The CHATSHIELD system provides a very productive and scalable way to make safe current online communication channels there is the categorization of the messages in real-time and prevents the malicious texts relayed in advance. The proposed system can be particularly used in the professional communication system, i.e. organizational chat applications, professional work collaborations, educational platforms, and professional networking platforms, e.g. LinkedIn, where the importance of respectful and safe communication is essential. The system helps organizations put into practice communication policies and an excellent digital climate by incorporating real-time cyberbullying detection and prevention.

A. Motivation

The growth of internet communication has resulted in a booming growth on the content of users. It is impossible to moderate hundreds of messages a minute in terms of messaging applications and social networks, and it is impossible to filter them manually. Such large volumes of data cannot be looked at effectively by the human moderators in a real time. More so, negative messages are mostly presented in the informal way that involves abbreviations, emojis, sarcasm and mixed language sentences. These attributes make these traditional modes of filtering ineffective. Consequently, intelligent AI-driven systems should have the capability of detecting and preventing bad communications automatically and retain the pace of real-time messaging environments.

B. Research Gap

Even though in the recent research, it was observed that machine learning and deep learning can be applied to detect abusive language, multiple limitations are still present. The majority of the in place systems are more understandably related to the correctness of the classification at the cost of lack of transparency in decision-making. This therefore implies that the models provide prediction, however, they do not explicate what has been the contribution of a linguistic factor to make predictions. This incomprehensibility of interpretation has issues of accountability and trust. Besides, most available solutions are designed to process a batch of data and not a constant stream of chat messages. Therefore, there is a need to have a system that will incorporate real-time processing and comprehensible machine learning practices.

C. Aim

To accomplish this detection, the research will create and implement a system that will detect abusive or malicious messages in real-time chat rooms and utilize them in machine learning.

The framework proposed CHATSHIELD as the applicant of NLP preprocessing technique and TF-IDF feature extraction as well as AB and TX(4) logistic regression classifier to identify and presumably block unwanted communication in chat systems.

D. Problem Definition

Among the difficulties that arise, there are identification of the harmful communication on the internet sites:

1. Issue of differentiating sarcasm and indirect insult.
2. Multilingual text processing, code-mixed text processing.
3. Dealing with changing slang, abbreviations, and antagonistic spellings.
4. Low transparency in decisions in the AI models.
5. Scalability of real-time message analysis is needed.

E. Objectives

- Develop a real-time chat message monitoring framework using Natural Language Processing (NLP) and machine learning techniques.
- Implement text preprocessing methods such as tokenization, normalization, stopword removal, and slang handling to improve message analysis.
- Apply TF-IDF feature extraction to convert textual chat messages into numerical representations for machine learning models.
- Train and evaluate a Logistic Regression classification model to identify harmful and normal messages in chat conversations.
- Design a simple chat interface that allows message exchange and demonstrates the filtering of harmful content.
- Evaluate the system performance using metrics such as accuracy, precision, recall, and F1-score.

II. LITERATURE REVIEW

This research introduced a system that fuses LLM-generated semantic features with adaptive streaming algorithms. Their framework processes data continuously, learns from new language patterns, and provides text-based explanations for each decision, achieving nearly 90% accuracy.[1] This work examined how cyberbullies purposely alter words to bypass filters. Their RoBERTa-based adversarial training significantly improved resilience under intentional misspellings and symbol substitutions, maintaining strong accuracy levels even under manipulated input [2]. This study compared classical models with transformer-based architectures. Results showed that transfer learning models such as BERT and RoBERTa outperform traditional classifiers due to their ability to capture deeper contextual meaning and subtle emotional cues [3]. Rather than detecting abusive text, this work explored profiling online behaviours using deep learning. Their findings demonstrated that linguistic style, emotional patterns, and interaction frequency can help identify potential cybercriminal traits [4].

Researchers have explored multiple techniques to identify online harassment. Early studies used keyword-based filters, which are simple but often produce false positives. Later, statistical and machine learning models like Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression were introduced to improve detection accuracy. However, these models still struggle with complex linguistic patterns and multilingual data.

A. Traditional Approaches

Keyword-based systems work by matching specific offensive terms with a predefined list. Although efficient, they cannot detect indirect or sarcastic bullying. For example, a sentence like "You're so smart, sure you are 🙄" contains sarcasm but may not be flagged.

B. Machine Learning Methods

Machine learning improved detection by learning from labeled datasets. Common techniques include TF-IDF vectorization combined with classifiers like Logistic Regression and SVM. However, these models rely on hand-crafted features and fail to capture the deeper meaning of sentences.

C. Deep Learning Models

With advancements in AI, deep learning models such as CNNs and LSTMs began to be used for text classification. They automatically extract features from text and can understand sequential patterns. Despite improvements in accuracy, these models require large datasets and often lack explainability.

D. Transformer-Based Models

Transformers, particularly BERT and RoBERTa, revolutionized NLP by providing bidirectional context understanding. These models excel at recognizing sarcasm, emotion, and contextual nuances. Researchers such as Islam & Rafiq (2023) and Saifullah et al. (2024) demonstrated that transformers outperform earlier models in cyberbullying detection.

E. Summary

Existing systems either focus on accuracy or explainability, but not both in real time. CHATSHIELD addresses these challenges by applying NLP preprocessing and machine learning techniques with TF-IDF feature extraction and Logistic Regression classification for real-time harmful message detection in chat environments.

III. METHODOLOGY

The CHATSHIELD framework is designed to detect and prevent abusive communication in real-time chat environments. The methodology consists of several stages that process incoming messages, analyze their linguistic patterns, classify their intent, and categorize messages as harmful or normal for real-time filtering. The overall pipeline includes text preprocessing, feature extraction, model training and classification, bullying type identification, and real-time message filtering. The system also integrates a real-time chat communication module with a dual-layer detection mechanism implemented on both client and server sides.

A. Text Preprocessing

Incoming chat messages often contain noise such as URLs, repeated characters, emojis, abbreviations, and mixed-language expressions. To ensure consistent analysis, the system first performs preprocessing operations including text cleaning, tokenization, normalization, and stop-word removal. Slang terms and emojis are converted into meaningful textual representations, while code-mixed expressions are standardized to improve linguistic clarity. These preprocessing steps enhance the quality of the input data before it is passed to the classification models.

B. Feature Engineering

After preprocessing, meaningful textual features are extracted from the cleaned data. The system uses TF-IDF (Term Frequency–Inverse Document Frequency) vectorization to convert textual messages into numerical feature vectors based on the frequency and importance of words within the dataset. This representation helps capture the significance of terms appearing in chat messages and allows the machine learning model to identify patterns related to harmful or abusive language. The generated feature vectors are then used as input for the classification model to perform message analysis and detection.

C. Model Training and Classification

The processed features are used to train a machine learning model capable of identifying harmful communication patterns in chat messages. In this system, Logistic Regression combined with TF-IDF features is used as the primary classification model due to its efficiency and strong performance in text classification tasks. The model analyzes the extracted feature vectors to distinguish between harmful and normal messages. By learning patterns from the training data, the classifier can effectively identify abusive language and support real-time message filtering in chat environments.

D. Real-Time Message Filtering

Once the model classifies a message, the system performs real-time filtering within the chat pipeline. Messages identified as harmful are automatically blocked or flagged before reaching the receiver. This bidirectional filtering mechanism ensures that abusive content is intercepted during transmission, thereby maintaining safer communication between users.

E. Bullying Type Classification

In addition to detecting whether a message is harmful or not, the system further classifies harmful messages into specific categories such as insults, threats, harassment, and hate speech. This classification is performed using a rule-based keyword analysis method, where the detected harmful message is analyzed for specific keywords associated with different types of cyberbullying. This helps in identifying the nature of harmful communication and provides informative warning messages to both the sender and the receiver.

F. Dual-Layer Detection Mechanism

The proposed system implements a dual-layer detection mechanism in which messages are analyzed at both the client-side and server-side.

The client-side detection provides immediate feedback to the sender before the message is transmitted, while the server-side detection performs final verification and filtering before delivering the message to the receiver. This dual verification mechanism improves system reliability and prevents users from bypassing the detection system.

G. Unknown Message Detection

Users often attempt to bypass cyberbullying detection systems by using symbols, special characters, or intentionally modified words. To address this issue, the system includes an unknown message

detection module that identifies messages containing excessive special characters or non-standard text patterns. Such messages are flagged as suspicious and marked as unknown messages, helping to prevent attempts to evade the detection system.

H. Real-Time Communication and File Transfer Module

The CHATSHIELD system is integrated into a real-time chat application developed using socket programming. The system supports multi-user communication, multi-room chat functionality, and file transfer between users. To enable communication across different networks, secure tunneling is used to expose the server to external users, allowing remote clients to connect to the chat server. This module ensures real-time communication while simultaneously performing cyberbullying detection and message filtering.

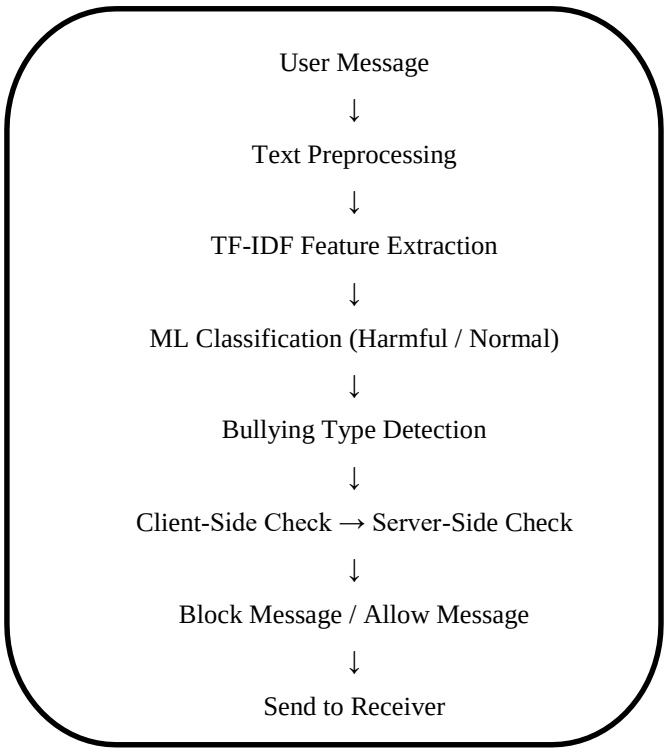


Figure 1.1 – Methodology Flow

IV. IMPLEMENTATION AND TOOLS USED

The CHATSHIELD system is implemented using Python and follows a client-server architecture using socket programming for real-time communication. The client interface is developed using the CustomTkinter graphical user interface framework, which provides an interactive chat environment for users. The server handles message routing, bullying detection, and message filtering. Machine learning functionality is implemented using the Scikit-learn library, and Natural Language Processing tasks such as tokenization and stopword removal are performed using the NLTK library. TF-IDF vectorization is used for feature extraction, and the Logistic Regression model is used for harmful message classification. The trained model and vectorizer are stored using Pickle for efficient real-time prediction. The system also supports remote connectivity through secure tunneling, enabling users from different networks to connect to the chat server.

A. Tools

Component	Tool/Library
Programming Language	Python 3.10
GUI Framework	CustomTkinter
Networking	Socket Programming
Machine Learning	Scikit-learn
Natural Language Processing	NLTK
Feature Extraction	TF-IDF Vectorization
Classification Algorithm	Logistic Regression / LinearSVC
Model Storage	Pickle
Remote Connectivity	Secure Tunneling (ngrok)
Development Environment	VS Code

B. System Workflow

The overall workflow of the CHATSHIELD framework can be summarized as follows:

1. User logs into the chat application and joins a communication room.
2. The user sends a message through the chat interface.
3. The message is first analyzed at the client-side using the trained machine learning model to provide immediate feedback to the sender.
4. The message is transmitted to the server through a socket connection.
5. The server performs text preprocessing and normalization on the received message.
6. The processed message is converted into numerical features using TF-IDF vectorization.
7. The machine learning model classifies the message as harmful or normal.
8. If the message is harmful, the system performs bullying type classification to identify the category of cyberbullying such as insult, threat, harassment, or hate speech.
9. The system also checks for unknown or suspicious messages containing excessive special characters or obfuscated text.
10. Based on the classification result, the server either blocks the message or allows it to be delivered to the receiver.
11. If the message is blocked, warning notifications are displayed to both the sender and the receiver.
12. The system also supports file transfer and multi-user communication within chat rooms.
13. Secure tunneling is used to allow users from different networks to connect to the chat server.

This workflow enables efficient real-time cyberbullying detection, classification, and prevention while maintaining low latency suitable for real-time chat applications.

C. System Architecture

The architecture of the CHATSHIELD system is designed as a client–server framework for real-time cyberbullying detection and prevention in chat communication. The system consists of a client interface, server module, and machine learning-based message analysis module. Messages are initially analyzed at the client side to provide immediate feedback to the sender. The message is then transmitted to the server through a socket connection, where text preprocessing and TF-IDF feature extraction are performed. The extracted features are passed to a Logistic Regression classifier to determine whether the message is harmful or normal.If a message is identified as harmful, the system performs bullying type classification to categorize the message into insult, threat, harassment, or hate speech. The system also includes an unknown message detection module to identify suspicious or obfuscated messages. Based on the final result, the server either blocks the message or allows it to be delivered to the receiver. The system supports multi-user chat, file transfer, and remote connectivity through secure tunneling, enabling communication across different networks. This architecture ensures real-time detection and filtering of harmful messages with low latency.

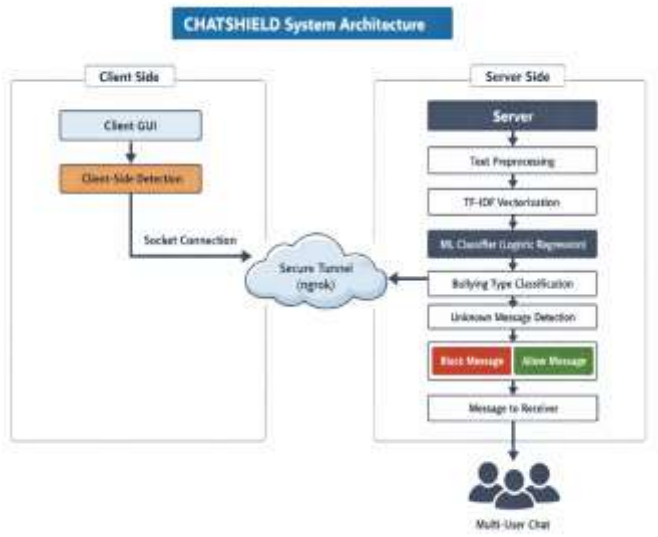


Figure 1.2 – System Architecture

D. System Block Diagram

The block diagram represents the overall processing flow of the CHATSHIELD system for detecting and preventing cyberbullying in real-time chat communication. The process begins when a user sends a message through the client chat interface. The message is first analyzed at the client side using a machine learning model to provide immediate feedback to the sender. The message is then transmitted to the server through a socket connection for further analysis.

At the server side, the message undergoes text preprocessing, which includes cleaning, tokenization, normalization, and stopword removal. The processed text is then converted into numerical feature vectors using TF-IDF vectorization. These feature vectors are passed to the machine learning classifier, which classifies the message as harmful or normal.

If the message is classified as harmful, the system performs bullying type classification to identify the type of cyberbullying, such as insult, threat, harassment, or hate speech. The system also performs unknown message detection to identify suspicious messages containing special characters or obfuscated text. Based on the final analysis, the system either blocks the message or allows it to be delivered to the receiver. If a harmful message is detected, warning notifications are displayed to both the sender and the receiver. This block diagram illustrates the complete workflow of the CHATSHIELD system, including message analysis, classification, detection, and real-time filtering to ensure safe communication between users.

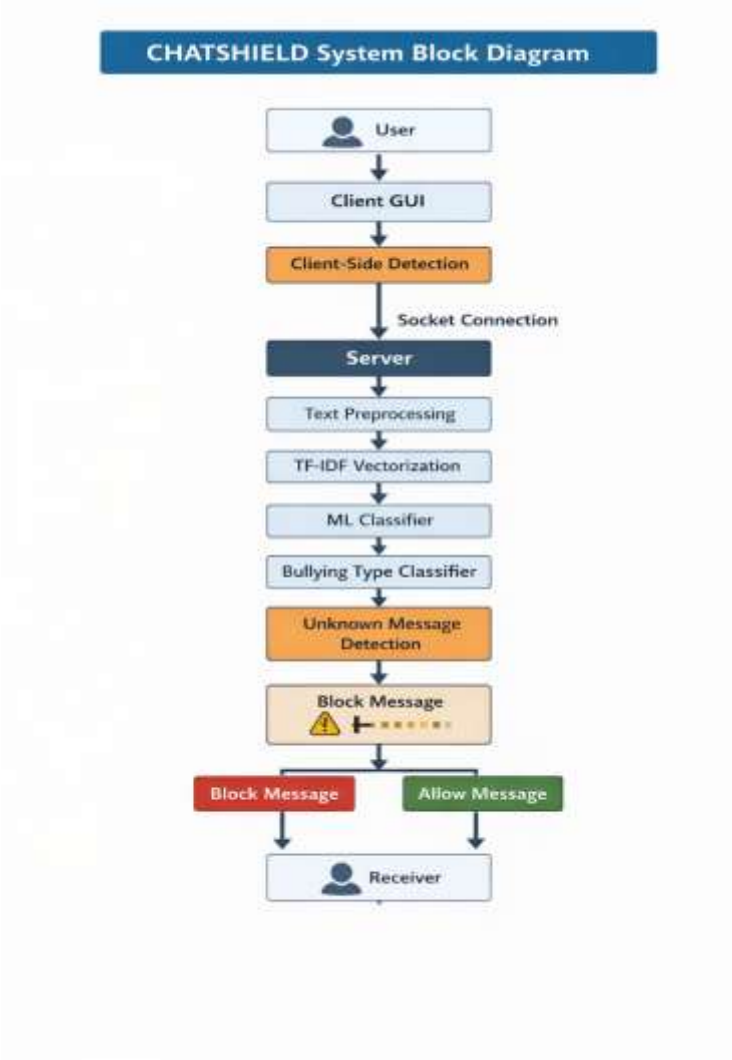


Figure – 1.3 – Block Diagram

V. RESULTS AND ANALYSIS

A. Dataset

For training and evaluating the proposed CHATSHIELD system, a labeled text dataset containing online chat messages was used. The dataset includes messages written in both English and Hinglish (Hindi – English mixed language), which are commonly used in online conversations. A total of 18,148 samples were used for model training and evaluation [1].

Class	Number of Samples
Harmful Messages	11,661
Normal Messages	6,487

The dataset also includes messages written in two language formats commonly used in online chats.

Language Type	Number of Samples
English	10,872
Hinglish	7,726

The inclusion of both English and Hinglish messages helps the model learn patterns from multilingual chat conversations.

B. Stopwords and Text Processing

During preprocessing, stopword removal was applied to eliminate frequently occurring words that do not significantly contribute to classification. The stopword list used in this research includes both English and Hinglish conversational terms.

Examples of stopwords used include:

1. English Stopwords:

a, the, and, is, are, was, were, in, on, at, for, with, from, to, of

2. Hinglish Stopwords:

hai, hain, kya, tum, main, hum, ka, ke, ki, se, par, yeh, woh, kyu, kaise

Removing these words reduces textual noise and improves the ability of the classification model to identify meaningful patterns related to harmful language detection.

C. Experimental Setup

The experiments were conducted using Python with machine learning and natural language processing libraries such as Scikit-learn, NLTK, Pandas, and NumPy. The textual messages were converted into numerical representations using TF-IDF vectorization, which measures the importance of words in each message relative to the entire dataset.

A Logistic Regression classifier was then trained using these features to categorize messages as either harmful or normal. The system was tested in a local development environment to evaluate its effectiveness in detecting harmful messages in real-time chat communication[1].

D. Model Evaluation

The CHATSHIELD model was compared with several classical machine learning algorithms. The performance of different classification models was evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Multiple algorithms were tested to analyze their effectiveness in detecting harmful messages within chat data. Among the evaluated models, Logistic Regression with TF-IDF features produced the most effective results with strong accuracy and efficient computation.

Based on these results, Logistic Regression was selected as the proposed model for real-time message classification in the CHATSHIELD system.

Model	Accuracy	Observation
Naïve Bayes	78%	Fast baseline classifier for text classification
Logistic Regression	84%	Strong performance with TF-IDF features
Support Vector Machine (SVM)	86%	Better boundary separation for text features
Random Forest	82%	Handles feature interactions effectively
Logistic Regression + TF-IDF (Proposed Model)	90%	Efficient and suitable for real-time chat filtering

Table – 1.2 – Model Evaluation

E. Performance Metrics

The performance of different machine learning models was evaluated using standard metrics such as accuracy, precision, recall, and F1-score. These metrics help measure how effectively the models identify harmful messages while minimizing incorrect classifications. Traditional classifiers such as Naïve Bayes and K-Nearest Neighbors (KNN) provide baseline performance, while algorithms such as Support Vector Machine (SVM) and Random Forest show improved results due to better handling of text features. Among the evaluated models, Logistic Regression with TF-IDF features achieved the best overall performance in terms of accuracy, precision, recall, and F1-score. In addition to classification accuracy, the Logistic Regression model requires low computational resources and provides fast prediction, making it suitable for real-time chat message filtering systems.

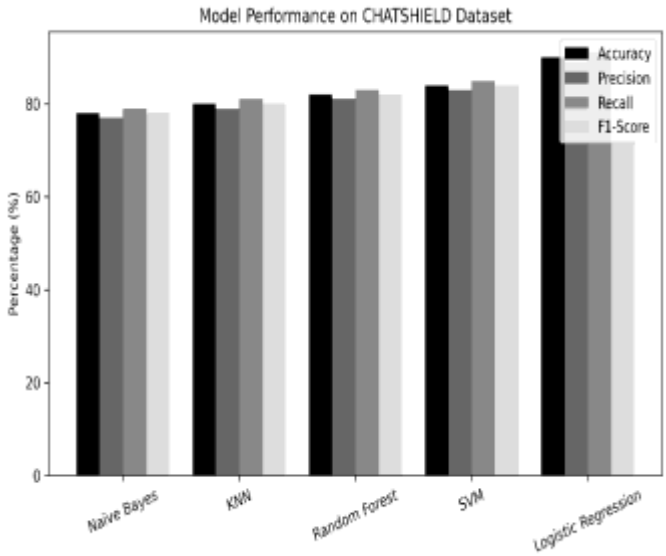
Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	78%	77%	79%	78 %
KNN	79%	78%	80%	79 %
Random Forest	81%	80%	82%	81 %
SVM	84%	82%	84%	83 %
Logistic Regression	90%	89%	91%	90 %

Table – 1.3 – Performance Metrics

F. Observations

- Logistic Regression with TF-IDF provides the highest accuracy.
- The model shows good balance between precision and recall.
- The system performs efficiently for real-time chat filtering.

The system maintains good accuracy with low processing latency during message filtering.



G. Model Performance

Model-wise metrics (grouped bars) — compares Accuracy, Precision, Recall, and F1 for all four models

Figure 1.3 – Model Performance Graph

Logistic Regression achieves the highest performance among the evaluated models, with an accuracy and F1-score of 90%, demonstrating its effectiveness for text classification using TF-IDF features. The SVM model also performs well with an accuracy of 83%, followed by Random Forest and KNN, which provide moderate results. Baseline models such as Naïve Bayes show comparatively lower performance due to their simplified assumptions about feature independence. Overall, Logistic Regression provides the best balance between accuracy and computational efficiency, making it suitable for real-time chat message filtering with reduced false positives and missed detections.

H. Existing vs Proposed System

This comparison evaluates the average performance of traditional baseline models against the proposed Logistic Regression-based classification system. Baseline methods such as Naïve Bayes, KNN, and Random Forest provide moderate results for harmful message detection. In contrast, the proposed Logistic Regression model with TF-IDF features achieves higher performance due to its ability to effectively capture important textual patterns in chat messages. The comparison highlights the improvement in overall classification performance achieved by the proposed system for real-time chat message filtering.

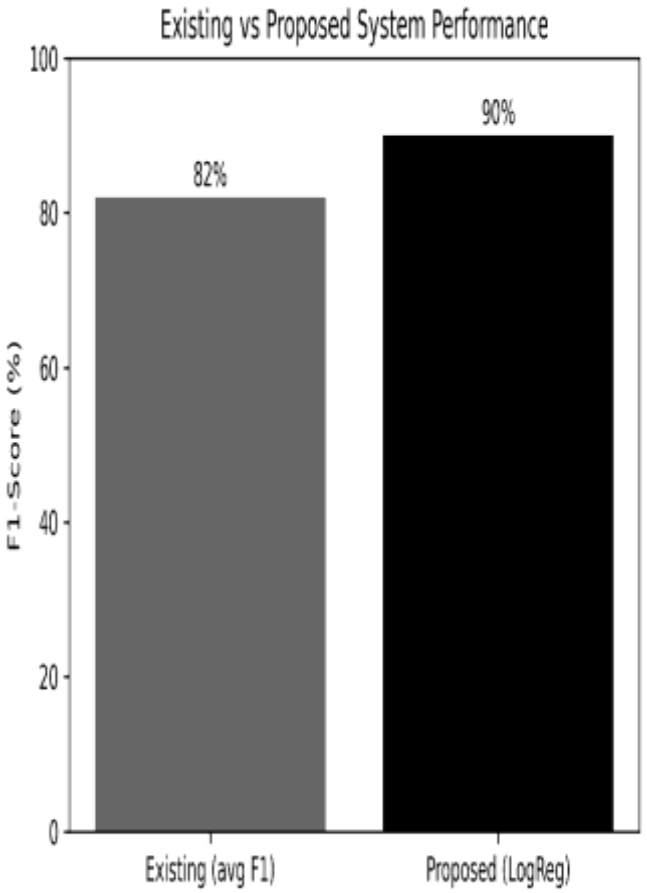


Figure 1.4 – Existing vs Proposed Graph

I. Discussion

The results indicate that the proposed CHATSHIELD system achieves strong performance in detecting harmful messages within chat environments. Among the evaluated models, Logistic Regression combined with TF-IDF features provides the highest accuracy of 90%, demonstrating its effectiveness in identifying harmful language patterns in textual communication. The model also shows reliable precision and recall values, which help reduce both false alarms and missed detections. Compared with baseline models such as Naïve Bayes, KNN, and Random Forest, the Logistic Regression classifier performs better due to its ability to handle high-dimensional text features efficiently. While other algorithms provide moderate results, Logistic Regression offers a better balance between classification accuracy and computational efficiency.

Another important advantage of the proposed system is its suitability for real-time chat monitoring. The lightweight nature of the Logistic Regression model allows fast prediction with low processing latency, making it practical for deployment in messaging platforms and online discussion systems. In addition to harmful message detection, the system also performs bullying type classification, which helps identify the nature of harmful communication such as insults, threats, harassment, and hate speech. The system also implements a dual-layer detection mechanism, where messages are analyzed at both client-side and server-side to improve detection reliability.

Furthermore, the system includes unknown message detection to identify suspicious or obfuscated messages that attempt to bypass the detection system using special characters or modified text. The integration of the detection system into a real-time multi-user chat application demonstrates the practical applicability of the proposed system. The use of secure tunneling enables communication across different networks, making the system suitable for real-world deployment.

Overall, the results demonstrate that the CHATSHIELD architecture can effectively detect, classify, and prevent harmful messages in real-time communication environments, contributing to safer and more responsible digital communication platforms.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

The proposed CHATSHIELD system presents an effective framework for detecting and filtering harmful messages in real-time chat environments using Natural Language Processing and machine learning techniques. By applying text preprocessing methods and TF-IDF feature extraction, the system converts chat messages into meaningful representations that can be analyzed by a Logistic Regression classification model. Experimental results show that the proposed model achieves an accuracy of 90%, demonstrating its capability to identify harmful

communication patterns with high reliability. Compared with baseline models, the Logistic Regression approach provides a strong balance between classification performance and computational efficiency.

Its lightweight architecture allows fast prediction with minimal latency, making it suitable for deployment in real-time chat systems where quick moderation is essential. The system also supports multilingual chat data, including English and Hinglish expressions, enabling it to handle language variations commonly observed in online communication.

In addition to harmful message detection, the CHATSHIELD system also performs bullying type classification, identifying categories such as insults, threats, harassment, and hate speech. The system implements a dual-layer detection mechanism at both client-side and server-side to improve detection accuracy and prevent bypassing of the filtering system. The system also includes unknown message detection to identify suspicious or obfuscated messages. Furthermore, the system is integrated into a real-time multi-user chat application and supports communication across different networks using secure tunneling, making it suitable for real-world deployment in professional and organizational communication platforms.

The CHATSHIELD framework contributes to creating safer digital communication environments by automatically detecting, classifying, and preventing harmful messages before they reach the recipient. With its scalable design and efficient processing pipeline, the system can be integrated into messaging platforms, educational forums, and collaborative online tools to support responsible and secure online interaction.

B. Future Scope

The proposed CHATSHIELD system demonstrates effective detection, classification, and prevention of harmful messages in real-time chat environments. However, several enhancements can be implemented in future work to further improve the system's capabilities and scalability.

1. Improved multilingual support can be implemented to handle additional languages and complex code-mixed text commonly used in online conversations.
2. Integration with large-scale real-time messaging platforms and organizational communication systems can be developed to deploy the system in enterprise-level applications.
3. Advanced deep learning and transformer-based models such as BERT and RoBERTa can be explored to improve contextual understanding, sarcasm detection, and semantic analysis.
4. An administrative dashboard can be developed to allow moderators to monitor chat activity, review flagged messages, and manage communication policies.
5. Mobile and web-based deployment can be implemented to integrate the system into cross-platform messaging applications.
6. Speech and voice-based cyberbullying detection can be added to analyze voice messages in real-time communication systems.
7. Image-based cyberbullying detection can be implemented to detect abusive content in images and memes shared in chat platforms.

These enhancements can further strengthen the effectiveness of CHATSHIELD and expand its applicability to a wider range of digital communication platforms.

REFERENCES

[1] *S. García-Méndez and F. Arriba-Pérez, Explainable Cyberbullying Detection using Large Language Models in Stream-Based Machine Learning Framework, IEEE, 2025.*

[2] *S. W. Azumah et al., Deep Learning Approaches for Adversarial Cyberbullying Detection, IEEE, 2024.*

[3] *K. D. Varathan, Cyberbullying Detection in Social Networks: ML vs Transfer Learning Comparison, IEEE Access, 2022.*

[4] *B. G. Bokolo and Q. Liu, Deep Learning Assisted Cyber Criminal Profiling, IEEE, 2023.*

[5] *M. S. Islam and R. I. Rafiq, "Comparative Analysis of GPT Models for Detecting Cyberbullying," Proc. Int. Conf. on Information Management and Big Data, Springer, 2023.*

[6] *K. Saifullah et al., "Cyberbullying Text Identification using Deep Learning and Transformers," EAI Trans. on Intelligent Systems, 2024.*

[7] *H. Herodotou et al., "Streaming ML Framework for Online Aggression Detection," IEEE Big Data Conf., 2020.*

[8] *A. Sadek et al., "Detection of Cyberbullying in Arabic using Machine Learning and ChatGPT," NILES Conf., 2023.*

[9] *V. U. Gongane et al., "Explainable AI for Reliable Detection of Cyberbullying," IEEE Pune Section Conf., 2023.*

[10] *M. Umer et al., "Cyberbullying Detection using PCA Extracted GLOVE Features and RoBERTaNet," IEEE Trans. on Computational Social Systems, 2024.*

[11] *S. G. Tesfagergish and R. Damaševičius, Explainable Artificial Intelligence for Combating Cyberbullying, Springer, 2024.*

[12] *L. Cheng et al., "Modeling Temporal Patterns for Cyberbullying Detection," ACM Trans. on Data Science, 2021.*

[13] *M. Wich et al., "Explainable Abusive Language Classification," Springer LNCS, 2021. S. García-Méndez and F. de Arriba-Pérez, "Promoting Security and Trust on Social Networks: Explainable Cyberbullying Detection using LLMs," IEEE arXiv Preprint, 2025.*

[14] *M. S. Islam and R. I.Rafiq, "Comparative Analysis of GPT Models for Detecting Cyberbullying in Social Media Platforms Threads," in Proceedings of the Annual International Conference on Information Management and Big Data. Springer,2023*

[15] *K. Saifullah,M. I.Khan, S. Jamal, and I.H. Sarker, "Cyberbullying Text Identification based on Deep Learning and Transformer-based Language Models," EAI*

Endorsed Transactions on Industrial Networks and Intelligent Systems,vol.11,pp.1–12,2024.

[16] *M. Arisanty and G.Wiradharma, "The motivation of flaming per petrators as cyberbullying behavior in social media," Jurnal Kajian Komunikasi,vol.10,p.215,2022.*

[17] *K. Verma, T. Milosevic, K. Cortis, and B. Davis, "Benchmarking Language Models for Cyberbullying Identification and Classification from Social-media texts," in Proceedings of the Language Resources and Evaluation Conference-Workshop on Language Technology and Resources for a Fair, Inclusive, and Safe Society. European Language Resources Association,2022.*

[18] *H. Saini, H.Mehra, R. Rani, G. Jaiswal, A. Sharma, and A. Dev, "Enhancing cyberbullying detection: a comparative study of ensemble CNN–SVM and BERT models," Social Network Analysis and Mining, vol.14,pp.1–18,2023.*

[19] *H.Herodotou,D.Chatzakou,andN.Kourtellis,"A Streaming Machine Learning Framework for Online Aggression Detection on Twitter," in Proceedings of the IEEE International Conference on Big Data. IEEE, 2020.*

[20] *P.Vanpech,K.Peerabenjakul,N.Suriwong,andS.Fugke a w,"Detecting Cyberbullying on Social Networks Using Language Learning Model," in Proceedings of the International Conference on Knowledge and Smart Technology. IEEE,2024.*

[21] *T. H. Teng and K. D. Varathan, "Cyberbullying Detection in Social Networks: A Comparison BetweenMachineLearningandTransferLearningApproache s,"IEEEAccess,vol.11,pp.55533– 55560,2023*