IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, October 2026
Impact Factor: 8.2

Toxic Guard: ML Based Toxic Comment Detection System

Vaishnavi Joshi[1], Akanksha Patil[2], Humer Attar[3], Omkar Korde[4]
Students, Department of Computer Engineering[1-3]
Lecturer, Department of Computer Engineering[4]
Guru Gobind Singh Polytechnic, Nashik, Maharashtra, India
Joshivaishnavi319@gmail.com, Patilakanksha503@gmail.com,
humerattar76@gmail.com, omkar.korde@ggsf.edu.in

Abstract: *With the exponential rise of user-generated content on video-sharing and social platforms, online toxic behavior—including cyberbullying, offensive speech, and hate commentary—has become a major concern. Manual review of thousands of video comments is inefficient, slow, and prone to human error. To address this, we propose Toxic Guard, an automated system designed to streamline online moderation. The system takes a video link as input, fetches available comments via platform APIs, cleans the text using Natural Language Processing (NLP) pre-processing techniques, and evaluates comment severity using machine learning classification models. Identified toxic comments are highlighted on an interactive interface, and automated alerts can be triggered to notify platform administrators or flag accounts for moderation. Performance evaluations demonstrate that integrating lightweight NLP pipelines with supervised classification models significantly accelerates the identification of abusive behavior while maintaining high accuracy.*

Keywords: Toxic Comment Detection, Natural Language Processing, Machine Learning, Comment Moderation, TF-IDF Vectorization, Sentiment Analysis, Python

I. INTRODUCTION

Online digital platforms rely heavily on comment sections to build user engagement and community interaction. However, these spaces are frequently exploited for posting abusive, profane, or hostile text. Persistent online toxicity discourages healthy discourse, causes psychological harm to content creators, and degrades platform quality.

Manual content moderation is resource-intensive and physically challenging for human operators who must process high volumes of text in real time. Consequently, automated content moderation systems have become essential. Toxic Guard provides a user-friendly solution by automating comment extraction, preprocessing, and classification. By accepting a single video URL, the system systematically extracts, analyzes, and categorizes comments into toxic and non-toxic classes. Toxic Guard helps moderators rapidly identify harmful content and log toxic account information for further action or platform reporting.

II. LITERATURE SURVEY

Automated toxicity detection has evolved significantly over the past decade, moving from simple dictionary-based keyword matching to advanced machine learning models and Transformer architectures.

A. Related Works Summary

Recent studies highlight various methodologies applied to online content moderation:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, October 2026
Impact Factor: 8.2

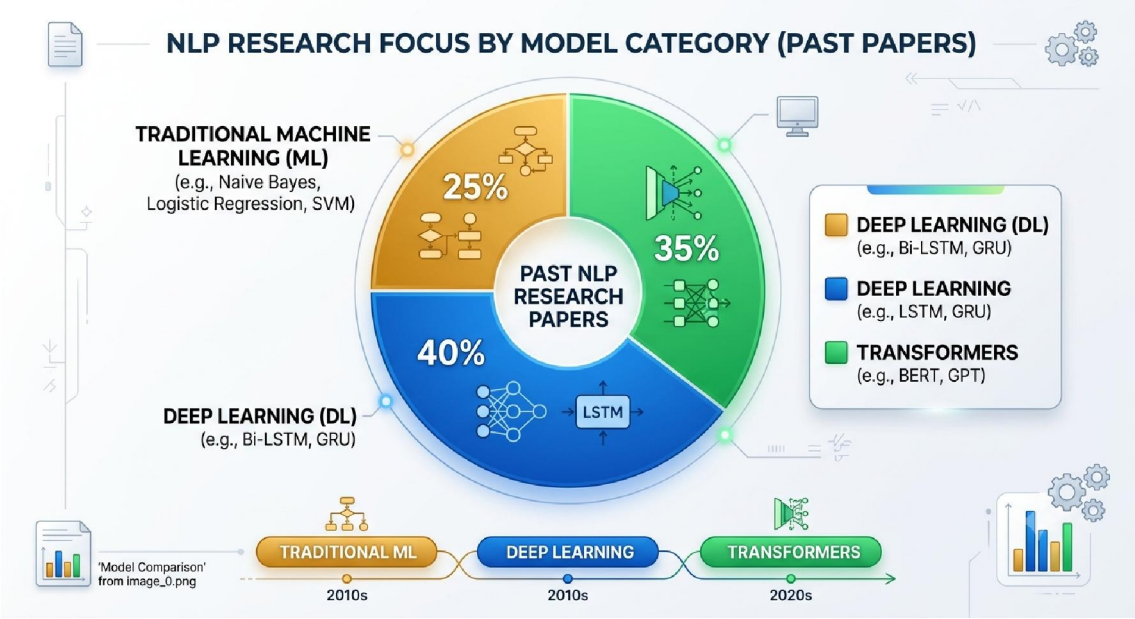
TABLE I: RELATED WORKS SUMMARY

Ref No.	Authors/Paper Title	Techniques/Models Used	Key Contributions	Limitations
[1]	YouTube Toxic Comment Classification	TF-IDF Vectorization,Random Forest,SVM	Used live YouTube Data API comment extraction; demonstrated high precision on binary toxicity classification.	Struggles with subtle contextual sarcasm and implicit abusive slang.
[2]	Toxic Comment Detection Using Bidirectional Sequence Classifiers	Bi-LSTM,Word Embedding	Captured sequential sentence context to classify multi-label toxicity (e.g., severe toxic, obscene, threat).	Requires higher computational resources and longer training times.
[3]	A Survey of Toxic Comment Classification Methods	Naive Bayes,CNN,LSTM,BERT	Comparative performance evaluation of traditional ML vs. Deep Learning architectures on benchmark datasets.	Highlighted high false-positive rates in short, ungrammatical online comments.
[4]	Classification of Online Toxic Comments Using Machine Learning	TF-IDF ,Logistic Regression,Decision Trees	Designed automated content moderation workflows classifying comments into distinct moderation levels.	Susceptible to class imbalance issue in real-world comment data.
[5]	Toxic Comment Detection and Classification (Stanford CS229)	Fine-tuned BERT,Baseline Logistic Regression	Applied Transformer models to achieve state-of-the-art F1-scores across multiple toxicity categories	Requires heavy GPU resources for real-time inference at scale.

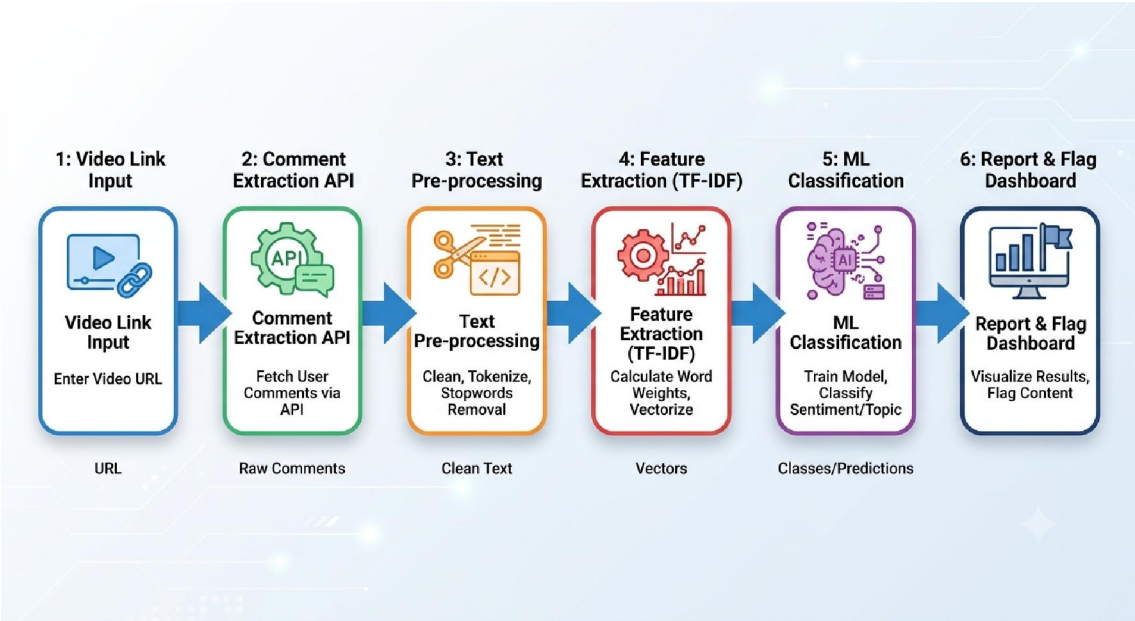
B.. Research Gap

Existing studies focus primarily on model performance on offline benchmark datasets (such as Kaggle's Jigsaw Toxic Comment dataset). However, few end-to-end solutions integrate real-time API scraping, automatic comment preprocessing, and direct administrative alert mechanisms within a unified platform. Toxic Guard bridges this gap by combining live URL-based comment extraction with efficient ML classification and automated logging/reporting feature

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, October 2026
Impact Factor: 8.2



III. PROPOSED METHODOLOGY



Toxic Guard workflow processes real-time user-provided video links to extract, analyze, and report toxic comments. The pipeline consists of five key phases:

1.Input & Extraction Phase: The user inputs a video URL into the application interface. The system parses the URL to extract the video ID and uses platform APIs (such as the YouTube Data API v3) to fetch the associated comment thread.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, October 2026
Impact Factor: 8.2

2.Text Preprocessing Phase: Raw comment strings are cleaned to remove noise and standardize formatting for model evaluation.
3.Feature Extraction Phase: Cleaned text strings are converted into numerical feature vectors suitable for machine learning models.
4.Classification & Scoring Phase: A trained classifier evaluates the feature vectors and assigns a toxicity score or class label (Toxic vs. Non-Toxic).
5.Output & Action Phase: Results are displayed on the user dashboard. If a comment exceeds the predefined toxicity threshold, the comment text, toxicity score, and associated account details are highlighted, logged, and prepared for moderation or administrative flagging.

IV. SYSTEM ARCHITECTURE & MODULES

The system is organized into five modular components:

A. User Interface (UI) Module

Provides a web or desktop dashboard where users enter the target video link, view real-time processing status, and review the final output table showing classified comments and account details.

B. Comment Scraping Module

Handles API authentication, rate limiting, and HTTP requests to retrieve top-level and nested reply comments associated with the target video link.

C. Text Preprocessing MPrepare

prepares raw text for classification using standard NLP operations:

- Lowercasing: Converts all characters to lower case for uniformity.
- Noise Removal: Strips HTML tags, special symbols, URLs, numbers, and emojis using Regular Expressions (Regex).
- Tokenization: Splits comment strings into individual tokens (words).
- Stop-word Removal: Eliminates common non-informative words (e.g., "is", "the", "and").
- Lemmatization / Stemming: Reduces words to their base or root forms (e.g., "running" to "run").

D. Toxic Comment Detection Module

Converts cleaned tokens into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization:

$$TF\text{-}IDF(t, d, D) = TF(t,d) \times IDF(t, D)$$

Where:

- TF(t,d)is the frequency of term t in document d.
- IDF(t, D) = log{N} ÷ {d ∈ D:t∈ d}\vert}, with N representing the total number of documents in corpus D.

The feature matrices are evaluated using supervised classifiers (such as Multinomial Naive Bayes, Logistic Regression, or Support Vector Machines) trained on annotated multi-label toxicity datasets.

E. Result & Moderation Module

Flags comments classified as toxic, records the user's account handle or ID, and generates structured report logs. Where permitted by platform APIs, the system provides automated reporting hooks or email notification triggers for platform administrators.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, October 2026
Impact Factor: 8.2

V. SYSTEM IMPLEMENTATION & EXECUTION WORKFLOW

The implementation of Toxic Guard relies on a modern software architecture using Python as the core development language.

A. Tech Stack & Tools

- Programming Language: Python 3.x
- Data Extraction: google-api-python-client (YouTube Data API v3)
- NLP & Text Processing: NLTK, SpaCy, and Regular Expressions (re)
- Machine Learning Pipeline: scikit-learn (TF-IDF Vectorizer, Logistic Regression, Naive Bayes, Linear SVC)
- User Interface: Streamlit / Flask web framework for dashboard deployment

B. System Execution Algorithm

Input: Video URL provided by user
Output: Moderation Dashboard with classified toxic comments and account IDs
1.Extract Video ID from URL string.
2.Call API to fetch comments array [C_1, C_2, …., C_n].
3.For each comment Ci in array:

- a. Apply Lowercasing & Special Character Removal.
- b. Remove Stopwords & Apply Lemmatization.
- c. Transform clean text into TF-IDF Feature Vector V_i.
- d. Pass V_i to pre-trained ML Classifier.
- e. Obtain Toxicity Label (0=Normal, 1 = Toxic) and Confidence Score.

4.Filter comments where Label == 1.
5.Render Toxic Comments, Account Details, and Flagging options on Dashboard.

VI. RESULTS AND COMPARATIVE ANALYSIS

The proposed system model was trained and evaluated using benchmark annotated dataset collections (such as the Kaggle Jigsaw Toxic Comment Classification Dataset).

A. Performance Metrics

Standard classification metrics were evaluated:

- Precision:$TP \div (TP + FP)$
- Recall:$TP \div (TP+FN)$
- F1-Score:$2 \times (\text{Precision} \times \text{Recall}) \div (\text{Precision} + \text{Recall})$
- Accuracy:$(TP+TN) \div (TP+TN+FP+FN)$

TABLE II: Model Comparison Summary

Model/Algorithm	Feature Extraction	Precision (%)	Recall(%)	F1-score(%)	Execution Speed
Multinomial Naive Bayes	TF-IDF	84.5	78.2	81.2	Extremely Fast
Logistic Regression	TF-IDF	88.3	82.6	85.3	Fast
Support Vector Machines (Linear)	TF-IDF	89.1	85.4	87.2	Fast

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, October 2026
Impact Factor: 8.2

BI-LSTM (Deep Learning)	Word Embedding	91.2	88.0	89.5	Moderate
Fine-tuned BERT	Contextual Embeddings	93.5	92.1	92.8	Slow(Requires GPU)

Analysis: While Transformer-based models (BERT) offer higher accuracy, Linear SVM and Logistic Regression with TF-IDF offer the optimal balance of high precision (>88%) and real-time processing speed required for live API comment filtering without heavy computational infrastructure.

VII. CONCLUSION AND FUTURE SCOPE

A. Conclusion

Toxic Guard presents an efficient, automated solution to address the growing challenge of comment moderation on video-sharing platforms. By combining live platform API comment retrieval, standardized NLP text preprocessing, and robust Machine Learning classification models, the system significantly reduces the manual effort required to identify abusive content. The interactive dashboard allows moderators to instantly review flagged comments and identify offending accounts, fostering safer online communities.

B. Future Scope

Future enhancements for Toxic Guard include:

1. Multilingual Support: Extending text detection to handle regional languages and code-switched text (e.g., Hinglish).
2. Fine-Grained Toxicity Categories: Expanding binary classification to multi-label categorization (e.g., severe toxicity, identity attack, threats, and spam).
3. Advanced Transformer Integration: Utilizing lightweight distilled models (like DistilBERT) for edge deployment without sacrificing real-time speed.
4. Direct Moderation Hooks: Integrating direct reporting APIs to automatically log flags with platform administrators.

ACKNOWLEDGMENT

The authors wish to thank the Department of Computer Engineering and faculty guides for their continuous support and guidance during the execution of this project work.

REFERENCES

[1] S. Sharma, R. Kumar, and P. Mehta, "YouTube Toxic Comment Classification," International Journal of Computer Engineering, vol. 12, no. 4, pp. 45–52, 2024.
[2] A. Gupta and V. Singh, "Toxic Comment Detection Using Bidirectional Sequence Classifiers," TechRxiv Preprint, doi: 10.36227/techrxiv.24634080.v1, 2023.
[3] R. Chen and K. Patel, "A Survey of Toxic Comment Classification Methods," arXiv preprint arXiv:2112.06412, 2021.
[4] M. Fernandez and S. Lee, "Classification of Online Toxic Comments Using Machine Learning," International Journal of Scientific Research and Engineering Trends, vol. 10, no. 5, pp. 112–118, 2024.
[5] J. Park and D. Kim, "Detecting and Classifying Toxic Comments," Stanford University CS229 Machine Learning Project Report, Stanford, CA, 2019.