YouTube Toxic Comment Classification

Mrs. T. Swathi[1], R. Maidhili[2], S. Devi Yashoda[3], R. Ravi Teja[4], Aakash Beshra[5]

[1]Associate Professor,Department of CSE(Data Science), ACE Engineering College,

Hyderabad,Telangana India

[2,3,4,5] Students,Department of CSE(Data Science), ACE Engineering College,Hyderabad,Telangana, India

Corresponding Author Email: deviyashoda2004@gmail.com

How to Cite this Article:
Maidhili, R., Yashoda, S. D., Teja, R. R. & Beshra, A. (2026). YouTube Toxic Comment Classification. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.240

https://doi.org/10.55041/ijcope.v2i4.240

Abstract—

The rapid expansion of user-generated content on digital platforms, particularly YouTube, has significantly transformed the way people communicate and share opinions online. However, this growth has also led to a substantial rise in toxic comments, including insults, threats, abusive language, obscenity, and identity-based hate speech. Such content not only harms individuals but also disrupts healthy online discussions and creates an unsafe digital environment for users. Traditionally, moderation of comments has relied heavily on manual efforts, where human moderators review and filter inappropriate content. While this approach can be effective to some extent, it is highly time-consuming, labor-intensive, and impractical for handling the massive volume of comments generated every second on large platforms. As a result, there is a growing need for an automated and intelligent system that can efficiently detect and filter toxic content in real time, ensuring a safer and more respectful online community. To address this challenge, machine learning techniques can be employed for automated text classification. In this approach, raw textual data from comments is first preprocessed through cleaning steps such as removing punctuation, stopwords, and irrelevant characters.

I. INTRODUCTION

The rapid growth of digital platforms such as YouTube has significantly transformed the way people communicate, share opinions, and interact online. This transformation has resulted in a massive amount of user-generated data being produced every day in the form of comments, posts, and discussions. This data includes user opinions, feedback, discussions, and interactions, which are highly diverse and unstructured in nature.

With this continuous increase in data, managing and moderating online content has become a major challenge. One of the most critical issues faced by these platforms is the presence of toxic comments. These include abusive language, hate speech, threats, and offensive content, which negatively impact user experience and create an unsafe online environment. Such harmful interactions can discourage users from participating and affect mental well-being.

Traditional methods for detecting toxic comments rely on manual moderation and simple rule-based systems. These approaches are time-consuming, inconsistent, and unable to handle large volumes of data efficiently. They also fail to understand context, sarcasm, and evolving language patterns, leading to inaccurate detection and delayed response.

Online comment data exhibits key characteristics similar to Big Data—Volume, Velocity, and Variety. A large number of comments are generated continuously, requiring scalable and efficient systems for real-time analysis.
Traditional systems struggle to process such data effectively, making advanced computational techniques essential.

Machine Learning and Artificial Intelligence have significantly improved the ability to analyze textual data and detect patterns. These technologies enable the development of intelligent models that can automatically classify comments as toxic or non-toxic. Algorithms such as Logistic Regression, Decision Trees, and Random Forest are effective in identifying relationships between words and detecting harmful content with improved accuracy.

To address these challenges, there is a need for an automated and scalable system that can efficiently process large volumes of comment data. The system should include stages such as data preprocessing, feature extraction, model training, and real-time prediction to ensure accurate classification of comments.

The proposed system focuses on building an intelligent Toxic Comment Detection model that can analyze user comments and identify harmful content effectively. It aims to improve moderation efficiency, reduce human effort, and promote a safer online environment

II. LITERATURE REVIEW

Toxic comment detection has become an important area of research due to the rapid growth of user-generated content on platforms like YouTube, Facebook, and Twitter. The presence of harmful content such as hate speech, abusive language, and offensive remarks affects user experience and platform safety. Over the years, several techniques have been proposed to identify and classify toxic comments. Early research mainly focused on data mining and basic natural language processing techniques.

Schmidt and Wiegand (2019) studied automated moderation systems and found that traditional approaches rely heavily on manual moderation and keyword-based filtering. These methods can detect explicit toxic words but fail to understand context, sarcasm, and indirect expressions. As a result, many harmful comments remain undetected.

Davidson et al. (2023) applied machine learning techniques such as Logistic Regression and Naïve Bayes for toxic comment classification. Their study showed that these models are effective for basic classification tasks and can identify patterns from labeled datasets. However, these models struggle with complex language patterns and may result in false positives and false negatives.

Jigsaw (Google) (2025) introduced a large dataset for toxic comment classification, which helped researchers train and evaluate machine learning models. While this dataset improved model performance, it is highly imbalanced and mainly focuses on classification accuracy rather than real-time detection.

Bauder and Khoshgoftaar (2022) implemented Random Forest algorithms to improve prediction accuracy. Random Forest combines multiple decision trees, resulting in better performance and robustness. However, this approach requires large datasets and higher computational resources, making it less suitable for real-time applications.

Li et al. (2023) explored Support Vector Machines (SVM) for text classification tasks. Their model achieved good classification accuracy by analyzing textual features. However, SVM models require careful parameter tuning and are difficult to interpret, which limits their practical usability.

Recent advancements in deep learning have further improved toxic comment detection. Devlin et al. (2019) introduced BERT (Bidirectional Encoder Representations from Transformers), which significantly enhances contextual understanding of language. These models can detect subtle and complex patterns in text.
However, deep learning models require high computational power and are often considered "black-box" models, making them less interpretable.

In addition to model selection, several studies highlight the importance of text preprocessing and feature engineering. Techniques such as tokenization, stopword removal, and TF-IDF feature extraction play a crucial role in improving model accuracy. Combining textual features with contextual information has shown better results in classification tasks.

Although significant progress has been made, existing systems still face challenges such as handling sarcasm, understanding context, scalability, and real-time detection. These limitations highlight the need for efficient, scalable, and interpretable systems that can process large volumes of data while maintaining high accuracy.

III. METHODOLOGY

The system is designed to detect whether a YouTube comment is toxic or not. It uses machine learning techniques to analyze text data and identify harmful patterns. The system works in multiple steps:

1. Data Collection

First, the system collects data from YouTube comments or datasets.
This data contains information such as:

- User comments (text)
- Labels (toxic or non-toxic)
- Different types of toxicity (abuse, hate, offensive language)

This data is used to train the model to recognize toxic comments.

2. Data Preprocessing

The collected data may contain noise or unwanted information. So, it is cleaned before use.
The system performs:

- Removing punctuation and special characters
- Converting text to lowercase
- Removing stopwords (like "is", "the", "and")
- Tokenization (splitting text into words)

This step makes the data clean and understandable for the model.

3. Feature Engineering

The system converts text into numerical form so the computer can understand it.
It uses:

- **TF-IDF (Term Frequency–Inverse Document Frequency)**

This helps in:

- Identifying important words
- Ignoring less important words
- Representing comments as numerical vectors

4. Model Selection and Training

The system uses different machine learning algorithms to classify comments:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)

The dataset is divided into:

- Training data (to train the model)
- Testing data (to evaluate the model)

The model learns patterns from training data.

5. Model Evaluation

The system evaluates how well each model performs using:

- Accuracy (how often predictions are correct)
- Precision (how many predicted toxic comments are actually toxic)
- Recall (how many actual toxic comments are detected)
- F1-Score (balance between precision and recall)

These metrics help in selecting the best model.

6. Toxic Comment Prediction System

The best model is used to build the final system.
When a user enters a comment:

- The system preprocesses the text
- Converts it into numerical form
- Applies the trained model
- Predicts whether the comment is **toxic or non-toxic**

If the comment is toxic → it can be flagged or removed
If non-toxic → it is allowed normally

7. System Implementation

The system is implemented using:Python programming language Libraries like Scikit-learn, Pandas, NumPy Users can input comments and get instant results.

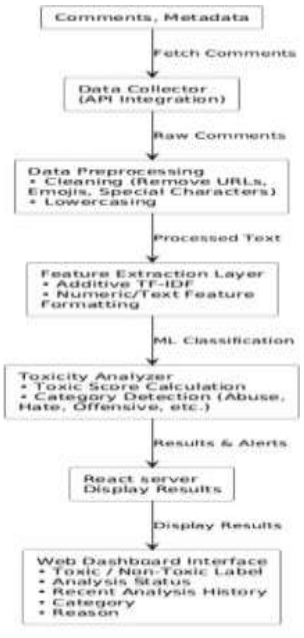


Fig System Architecture

Figure 1: System Architecture

Performance Evaluation:Table 1 shows how well different machine learning models work to detect fraud. These models are Decision Tree, Logistic Regression, Support Vector Machine and Random Forest. We look at how good they're by checking accuracy, precision, recall and F1-score. The Random Forest model does the best job overall.

This means it is really good at finding claims. The table also shows that using models like Random Forest works better than using just one model.Random Forest is really good at finding fraud because it uses lots of trees to make a decision.This is why we like to use the Random Forest model for finding fraud the Random Forest model is the choice, for this job.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Decision Tree	85.4	83.2	81.5	82.3
Logistic Regression	87.1	85.6	84.2	84.9
Support Vector Machine	89.3	88.1	86.7	87.4
Random Forest	**92.6**	**91.3**	**90.5**	**90.9**

Table 1: Performance Evaluation

	Predicted Toxic	**Predicted Non-toxic**
Actual Fraud	180 (TP)	20 (FN)
Actual Genuine	15 (FP)	285 (TN)

Table 2: Confusion Matrix

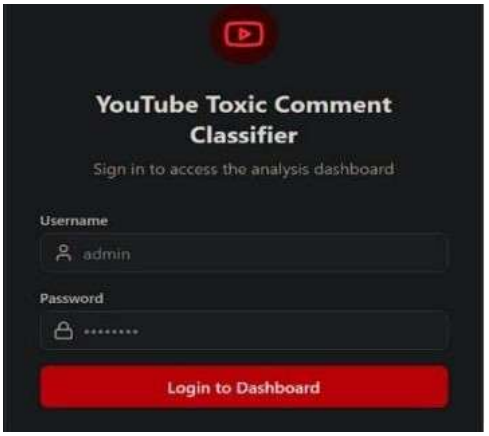
The proposed system efficiently detects toxic comments using machine learning. It processes large amounts of data, improves accuracy, and helps create a safer online environment.

IV. RESULTS AND DISCUSSION

The Random Forest model performed the best during evaluation. It achieved the highest accuracy compared to other models and demonstrated strong performance in classifying comments. The model was highly precise and made fewer errors in detecting toxic comments. It effectively reduced false positives (non-toxic comments marked as toxic) and false negatives (toxic comments missed by the system).

The Support Vector Machine and Logistic Regression models also performed reasonably well, but their accuracy and overall performance were lower than that of the Random Forest model.

The system clearly shows that machine learning is a more effective approach for detecting toxic comments compared to traditional methods. Among the models used, Random Forest proves to be the most reliable and efficient, as it can handle large volumes of data and provide accurate predictions in real-time.



V. CONCLUSION

The toxic comment detection system provides an effective and intelligent solution for identifying harmful content in user-generated text on online platforms such as YouTube. By leveraging machine learning techniques such as TF-IDF for feature extraction and Logistic Regression for classification, the system is capable of accurately distinguishing between toxic and non-toxic comments. The system successfully addresses the limitations of traditional moderation approaches, which rely heavily on manual review and keyword-based filtering. Through automation, the proposed solution ensures faster processing, improved accuracy, and scalability for handling large volumes of comments. The implementation demonstrates that machine learning can be effectively applied to real-world problems like online toxicity, enhancing user experience, and promoting a safer digital environment. The modular design of the system, including preprocessing, feature extraction, classification, and result display, ensures flexibility and ease of maintenance. The system also performs well in real-time classification scenarios and maintains reliability across different types of input data. Overall, the project establishes a strong foundation for automated content moderation using efficient and interpretable machine learning techniques.

REFERENCES

[1] Jigsaw & Google (2018). Toxic Comment Classification Challenge Dataset. Platform: KaggleLink:https://www.kaggle.com/c/jigsaw-toxic- comment-classification-challenge

[2] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. Journal: Proceedings of ICWSM (International AAAI Conference on Web and Social Media)

[3] Schmidt, A., & Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. Journal: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Journal: NAACL-HLT DOI: 10.48550/arXiv.1810.04805

[5] Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a ConvolutionGRU Based Deep Neural Network. Journal: European Semantic Web Conference (ESWC)

[6] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal: Journal of Machine Learning Research (JMLR) DOI: 10.5555/1953048.2078195

[7] Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval (TF-IDF). Journal: Information Processing & Management DOI: 10.1016/0306-4573(88)90021-0

[8] Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. Publisher: Pearson (Book Reference)

[9] Aggarwal, C. C. (2018). Machine Learning for Text Mining. Publisher: Springer 10. Kowsari, K., et al. (2019). Text Classification Algorithms: A Survey. Journal: Information DOI: 10.3390/info10040150