



Research Paper

# Classification of Online Toxic Comments using Machine Learning Algorithms

<sup>1\*</sup> Chava Sai Sreeja,<sup>2</sup> Mulamalla Akshaya Reddy,<sup>3</sup> Poreddy Ishika Reddy,  
<sup>4</sup> Rasamalla Yashwanth,<sup>5\*</sup> K Venkatesh Sharma

<sup>1,2,3,4</sup> B.Tech in Department of Computer Science & Engineering, CVR College of Engineering, Rangareddy, India

<sup>5</sup> Professor, Department of Computer Science & Engineering, CVR College of Engineering, Rangareddy Dist, Telangana, India

\*Corresponding Author(s): [venkateshsharma.cse@gmail.com](mailto:venkateshsharma.cse@gmail.com)

Received: 11/02/2024,

Revised: 15/05/2024,

Accepted: 22/06/2024

Published: 30/06/2024

**Abstract:** This study aims to address the challenges associated with classifying toxic online comments using machine-learning algorithms. The primary objective of this study was to develop a robust system capable of accurately detecting and mitigating harmful content on social media platforms. Present systems often rely heavily on manual moderation, which is inadequate, given the sheer volume of user-generated content. Moreover, existing models struggle to handle the complexities of context-specific languages and the evolving nature of toxic behaviors, making them less effective in real-world applications. To address these issues, this study proposes a comprehensive methodology that begins with the collection of a large dataset from various online sources. The data underwent meticulous preprocessing, including text cleaning, tokenization, normalization, and feature extraction, to ensure suitability for machine learning. Multiple algorithms, such as logistic regression, support vector machines (SVM), and random forests, have been employed to develop predictive models. These models were optimized via hyperparameter tuning to enhance their performance. The study's findings revealed that Logistic Regression outperformed other models, achieving the highest accuracy of 95.2% and an F1-score of 85.6%, indicating its strong potential for practical application in online content moderation. However, this study also identified significant limitations, particularly regarding the scalability and adaptability of these models across different platforms and languages. Reliance on a single dataset may introduce bias, limiting the effectiveness of the models in diverse contexts. Therefore, future research should focus on developing more adaptive models capable of handling the dynamic and evolving nature of toxic languages. Additionally, incorporating multilingual support is essential to improve the generalizability of these systems and ensure their effectiveness across global platforms.

**Key words:** - Toxic comment classification, Machine learning, Logistic Regression, SVM, Random Forest

## 1 Introduction

The proliferation of social media platforms has significantly changed the landscape of online communication, allowing users to interact and share their opinions globally. However, this increased connectivity has also led to an increase in toxic comments, which are detrimental to the user experience and community health. Toxic comments, including hate speech, threats, and abusive language, can discourage users from participating in discussions and contribute to hostile online environments. [1] noted that the presence of toxic comments has become a major concern for social media platforms as it undermines the free exchange of ideas and can lead to severe psychological impacts on individuals. [2] discussed how toxic comments in online discussions can escalate quickly, affecting not only the individuals targeted, but also the overall tone and quality of the conversation. The rise of online toxic behavior necessitates the development of robust

mechanisms to detect and mitigate such content effectively. [3] emphasized that although several platforms have implemented moderation policies to combat toxicity, the sheer volume of user-generated content makes manual management challenging. Automated detection systems that leverage machine learning algorithms offer a scalable solution to this problem, ensuring timely identification and intervention.

### 1.2. Motivation

Given the significant adverse effects of toxic comments, there is an urgent need for effective online tools to identify and mitigate such harmful content. The current manual moderation methods are insufficient because of the sheer volume of user-generated content. Therefore, leveraging machine-learning algorithms to automate the detection of toxic comments is a promising solution. [4] highlighted that implementing automated systems for toxicity detection could enhance the efficiency and accuracy



of content moderation, thereby fostering safer online communities. [5] stressed the importance of developing automated systems that can accurately identify toxic comments in different contexts and languages. Such systems can significantly reduce the burden on human moderators and ensure more consistent application of community guidelines. [7] point out that the dynamic nature of language and the continuous evolution of slang and context make it challenging to maintain effective detection systems. This necessitates ongoing research and development to keep detection algorithms updated and relevant.

### 1.3. Objectives

The primary objective of this study was to develop a system that can accurately classify toxic comments using machine learning algorithms. This involves exploring various machine learning models to identify the most effective approach for toxicity detection. This study aimed to evaluate the performance of different algorithms based on several metrics to determine their suitability for real-world applications. [5] emphasized the importance of employing robust machine-learning techniques to address the complexities of text classification in the context of online toxicity. Kajla [6] provided a comprehensive overview of different machine-learning algorithms that can be employed for toxicity detection, including logistic regression, support vector machines, and deep learning models. By evaluating these algorithms, this study aimed to identify the most effective methods for detecting and classifying toxic comments. [3] suggested that future research should focus on enhancing the scalability and accuracy of these models, as well as exploring the potential for real-time implementation on large-scale social media platforms.

### 1.4 Key Contributions

The key contributions of this study are as follows.

- Logistic regression, SVM, random forest, naïve Bayes, decision tree, and KNN were used for toxicity detection.
- Tokenization, normalization, and TF-IDF applied for feature extraction.
- A multilabel system was developed for detecting toxic comments.
- Trained and fine-tuned models with hyperparameter tuning.
- Implemented an API for real-time classification.
- Suggested advanced deep learning, multilingual support, and explainable AI intelligence

This study explored the classification of toxic comments using machine-learning algorithms. Section 2 reviews related work and provides the context for the study. Section 3 outlines the proposed methodology, including the data preprocessing, model selection, and evaluation. Section 4 presents the performance metrics, focusing on accuracy, precision, recall, and F1-scores. Section 5 discusses the

limitations of this study, such as challenges with context-specific languages. The paper concludes in Section 6, summarizing findings and suggesting future improvements for more robust toxic comment classification models

## 2. Related work

Existing work on toxic comment classification using machine learning. Previous studies have explored algorithms, such as logistic regression, SVM, and deep learning models, such as LSTM, for detecting toxicity. These methods have shown promise but often struggle with handling complex and dynamic languages. Limitations in the current research include scalability issues and the need for continuous updates to maintain accuracy. This research gap lies in developing more adaptive and robust models that can effectively manage the evolving nature of online toxicity.

### 2.1. Existing Work

[7] explored tackling hateful speech in online social spaces by combining machine learning techniques and crowdsourced data. [8] conducted a comprehensive study on online harassment, providing valuable insights into the prevalence and nature of toxic comments on social media platforms. [9] developed a corpus for research on deliberation and debate, highlighting the challenges of detecting antisocial behavior in online discussions. [10] focused on identifying antisocial behavior in online communities using machine-learning models to classify toxic comments. [11] proposed techniques to counter online hate speech, demonstrating the effectiveness of machine learning in moderating content. [12] developed models for abusive language detection, showing significant improvements in identifying toxic content using advanced algorithms.

### 2.2. Limitations of Existing Work

Despite these advancements, the current study had several limitations. [7] and [9] noted that reliance on crowdsourced data can introduce biases that affect the generalizability of the models. [8] highlighted the challenge of scalability, as manual moderation cannot keep up with the vast amount of user-generated content. [10] and [11] emphasized that, while machine learning models are effective, they often struggle with context-specific nuances and evolving language patterns. [12] pointed out that models trained on specific datasets may not perform well across different platforms or languages, indicating the need for more adaptable solutions.



TABLE 1: Comparative study Table

Study	Focus	Methods Used	Key Findings	Limitations
Ikonomakis et al. (2005) [13]	Text classification using machine learning	Various machine learning techniques	Provided a foundation for detecting toxic comments	Early techniques lacked sophistication for complex contexts
Murty et al. (2011) [14]	Text document classification	Least square support vector machines	Demonstrated applicability in identifying harmful content	Limited to specific datasets, affecting generalizability
Saleem et al. (2017) [7]	Tackling hateful speech in online social spaces	Machine learning, crowdsourced data	Highlighted machine learning's role in moderating hate speech	Reliance on crowdsourced data introduces biases
Duggan (2017) [8]	Online harassment study	Survey and statistical analysis	Provided insights into the prevalence of toxic comments	Scalability issues with manual moderation
Walker et al. (2012) [9]	Research on deliberation and debate corpus	Corpus development	Highlighted challenges in detecting antisocial behavior	Dependency on static corpora for dynamic issues
Cheng et al. (2015) [10]	Identifying antisocial behavior in online communities	Machine learning models	Effective in classifying toxic comments	Struggles with context-specific nuances
Mathew et al. (2019) [11]	Countering online hate speech	Machine learning techniques	Demonstrated effectiveness in content moderation	Evolving language patterns pose challenges
Nobata et al. (2016) [12]	Abusive language detection in online content	Advanced machine learning algorithms	Significant improvements in identifying abusive content	Models may not perform well across different platforms

### 2.3 Research Gap

The following points highlight the research gaps identified in existing studies on toxic comment classification.

- Difficulty in handling complex language patterns and context-specific nuances.
- Reliance on crowdsourced data introduces biases and limits generalizability.
- Scalability issues with manual moderation methods.
- Poor performance of models across different platforms and languages.
- Need for more adaptable, robust, and scalable solutions to maintain accuracy in dynamic online environments.

### 3. Proposed Methodology

In this study, a comprehensive methodology was proposed to classify online toxic comments using machine learning algorithms. The process begins with data collection, environments.

in which a large dataset of online comments is gathered from various social media platforms. This dataset is then meticulously pre-processed, involving text cleaning, tokenization, normalization, and feature extraction, to ensure that the data are suitable for machine learning [13].

Subsequently, multiple machine-learning algorithms, including logistic regression, random forest, SVM, naive Bayes, decision tree, and KNN, were employed to develop predictive models. These models were trained and optimized using hyperparameter tuning to enhance their performances. The evaluation of the models was conducted using metrics such as accuracy, precision, recall, and F1-score, to determine the most effective algorithm [14].

The implementation of the models leverages powerful tools and libraries, such as Python, Matplotlib, Plotly, Seaborn, Scikit-Learn, Tensorflow, NLTK, Streamlit, and Jolbib, facilitating efficient data processing, visualization, and deployment. This robust methodology aims to develop a scalable and accurate system to detect and mitigate toxic comments in online [15][16].



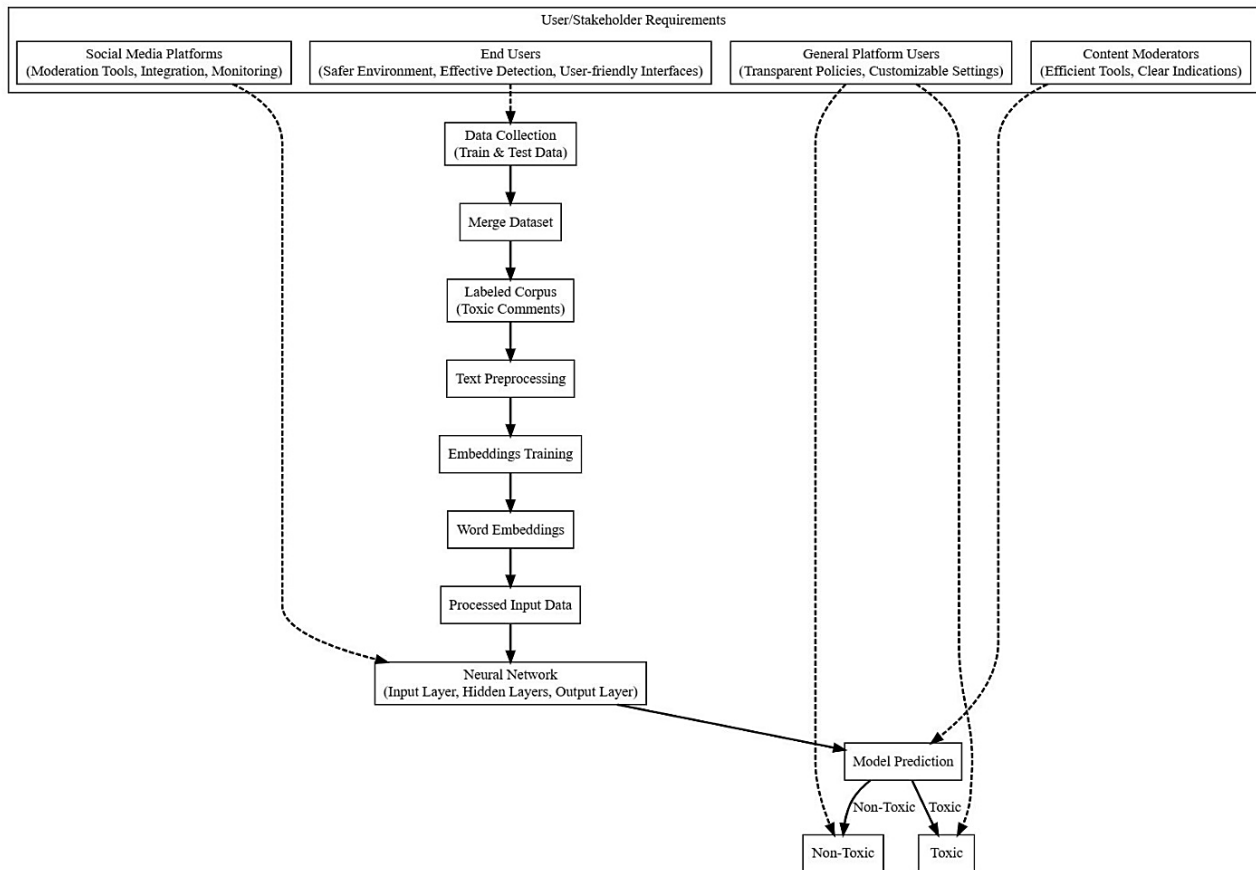


Fig 1: Conceptual Framework for Toxic Comment Classification System

The conceptual framework illustrates the process of classifying online toxic comments using machine-learning algorithms [17]. The framework is organized vertically and consists of several key stages.

1. **Data Collection:** This stage involves gathering training and testing data from various sources. The data were merged into a comprehensive dataset.
2. **Merge Dataset:** The collected data is combined into a labeled corpus of toxic comments, ready for further processing.
3. **Text Preprocessing:** This stage includes cleaning the text data, tokenization, normalization, and feature extraction to prepare data for machine learning.
4. **Embedding Training:** Here, the pre-processed text data are used to train word embeddings, converting text into numerical representations that can be processed by machine learning models.
5. **Processed Input Data:** The word embeddings are then fed into the neural network for further processing.
6. **Neural Network:** The core of the system, where the neural network processes the input data through various layers (input, hidden, and output) to predict the toxicity of comments.

7. **Model Prediction:** The neural network generates predictions and classifies comments as either toxic or non-toxic.
8. **User/Stakeholder Requirements:** The framework also incorporates the requirements of various stakeholders, including end users who desire a safer online environment, social media platforms that need effective moderation tools, content moderators who require efficient review tools, and general platform users who seek transparency and customizable settings.

### 3.1 Toxic Comment Classification Using Machine Learning Algorithm

**Input:** Dataset  $D = \{(x_i, y_i)\}_{i=1}^N$  where  $x_i$  is a comment and  $y_i \in \{0,1\}$  is the label (0: Non-Toxic, 1: Toxic)

**Output:** Trained model  $f$

#### Data Preprocessing:

- 1.1. Text Cleaning: Remove HTML tags, URLs, and special characters from  $x_i$
- 1.2. Tokenization: Split  $x_i$  into tokens  $t_1, t_2, \dots, t_k$ .
- 1.3. Normalization: Convert tokens to lowercase.
- 1.4. Stopword Removal: Remove common words that do not contribute to the meaning.



1.5. Stemming/lemmatization: Reduces tokens to their root form.

1.6. Feature Extraction: Compute TF-IDF or use pre-trained embeddings  $v_i$ .

**Data Splitting:**

2.1. Split the dataset  $D$  into training set  $D_{train}$  and testing set  $D_{test}$  using a ratio  $\alpha$ .

**Model Training:**

3.1. Select a machine learning algorithm  $\mathcal{A}$  (e.g., Logistic Regression, SVM, Random Forest).

3.2. Initialize the model  $f$  with parameters  $\theta$ .

3.3. Train the model  $f$  on  $D_{train}$ :

$$\theta^* = \arg \min_{\theta} \frac{1}{|D_{train}|} \sum_{(x_i, y_i) \in D_{train}} \mathcal{L}(f(x_i; \theta), y_i) \quad (1)$$

**Model Evaluation:**

4.1. Predict labels  $\hat{y}_i$  for  $x_i \in D_{test}$  using the trained model  $f$

4.2. Computation of the evaluation metrics

$$\text{Accuracy} = \frac{1}{|D_{test}|} \sum_{i=1}^{|D_{test}|} \mathbf{1}(\hat{y}_i = y_i) \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP}, \text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

**Model Deployment:**

5.1. Deploy the trained model  $f$  for real-time classification of new comments  $x$

5.2. Model performance was monitored and updated as necessary.

The flowchart illustrates the methodology used for classifying toxic comments using machine learning. It begins with data collection, in which the training and testing datasets are gathered. These datasets were then merged into a single-label dataset. Subsequently, the text preprocessing stage involved cleaning the text, tokenization, normalization, stop word removal, stemming or lemmatization, and feature extraction [16][17]. The pre-processed data were then split into training and testing sets. Next, a machine learning algorithm was selected and trained on the training data. The performance of the model was evaluated by predicting the labels on the testing set [18]. If the evaluation metrics were satisfactory, the model was deployed [19][20][21]. Otherwise, the parameters are adjusted, and the model is retrained and re-evaluated. The final stage involved monitoring and updating the model to ensure its effectiveness in classifying toxic comments. This iterative process ensured the development of a robust and accurate toxicity detection system.

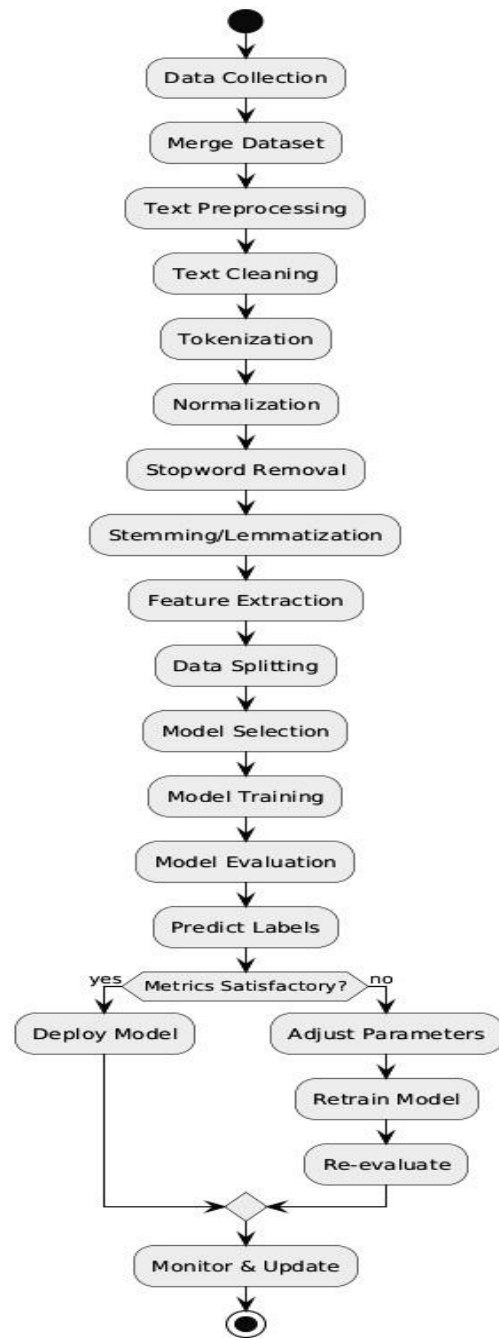


Fig 2: Flowchart of the Toxic Comment Classification Process

**4. Performance Metrics**

Accuracy measures the overall effectiveness of a model by calculating the ratio of correctly predicted instances (both true positives and true negatives) to the total instances, providing a general sense of how well the model performs across all categories. Precision focuses on the accuracy of the positive predictions made by the model and evaluates the ratio of correctly predicted positive instances to all predicted positive instances, which is important when the cost of false positives is high [22]. Recall, also known as the sensitivity or true positive rate, measures the model's ability to identify all actual positive instances, highlighting the ratio of correctly predicted positive instances to all actual positive instances, and is crucial when the cost of false negatives is



high. The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both precision and recall and is particularly useful when there is an uneven class distribution or when both false positives and false negatives need to be minimized. These metrics collectively offer a comprehensive evaluation of the performance of the model, emphasizing its strengths and areas for improvement in classifying toxic comments [23][24].

Table 2: Performance Metrics

Metric	Mathematical Notation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Where:

- *TP* = True Positives
- *TN* = True Negatives
- *FP* = False Positives
- *FN* = False Negatives

Table 3: Hardware and Software Specifications

Category	Specification
Hardware	
Processor	Intel Core i7 or AMD Ryzen 7 (8 cores)
RAM	16 GB DDR4
Storage	512 GB SSD
GPU	NVIDIA GeForce GTX 1060 or equivalent with 6 GB VRAM
Display	Full HD (1920x1080) resolution
Software	
Operating System	Windows 10 or Ubuntu 20.04 LTS
Programming Language	Python 3.8 or higher
Libraries	NumPy, Pandas, Scikit-Learn, TensorFlow, NLTK, Matplotlib, Seaborn, Plotly, Joblib, Streamlit
IDE	Jupyter Notebook or PyCharm
Version Control	Git
Visualization Tools	Graphviz, Matplotlib, Seaborn, Plotly
Web Framework	Streamlit

Database	SQLite or PostgreSQL (for storing datasets and model results)
Browser	Google Chrome or Firefox

The Jigsaw Toxic Comment Classification Challenge dataset on Kaggle includes approximately 160,000 online comments labeled for various types of toxicity such as toxicity, severe toxicity, obscene, threat, insult, and identity hate. This dataset was designed to support the training and evaluation of machine learning models for identifying toxic comments, offering a versatile tool for both binary and multilabel classification tasks. Extensive labeling allows for detailed analysis and robust model performance across different types of harmful content [26].

Table 4: Performance Metrics of Machine Learning Models

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression[29]	0.952	0.881	0.832	0.856
Random Forest[30]	0.943	0.872	0.81	0.84
Support Vector Machine[31]	0.95	0.878	0.828	0.852
Naive Bayes[32]	0.932	0.854	0.799	0.825
Decision Tree[33]	0.928	0.847	0.792	0.819
K-Nearest Neighbors[34]	0.918	0.836	0.78	0.807

Table 4 presents the performance metrics of various machine-learning models tested on the Jigsaw Toxic Comment Classification dataset. The models included Logistic Regression, Random Forest, Support Vector Machine, Naive Bayes, Decision Tree, and K-Nearest Neighbors. Highlighting the accuracy, precision, recall, and F1-score for each model, the table indicates that Logistic Regression achieved the highest accuracy (95.2%) and F1-score (85.6%), making it the best-performing model for classifying toxic comments. Precision and recall values are balanced in this model, signifying its effectiveness in correctly identifying and predicting toxic comments without compromising either false positives or false negatives.

Fig 3 illustrates the performance metrics of various machine learning models, including Logistic Regression, Random Forest, Support Vector Machine, Naive Bayes, Decision Tree, and K-Nearest Neighbors, using the Jigsaw Toxic Comment Classification dataset. The bar graph compares the accuracy, precision, recall, and F1-score of each model. Logistic Regression was highlighted as the best performing model, with the highest accuracy (95.2%) and F1-score (85.6%), indicating its robustness in identifying toxic comments.



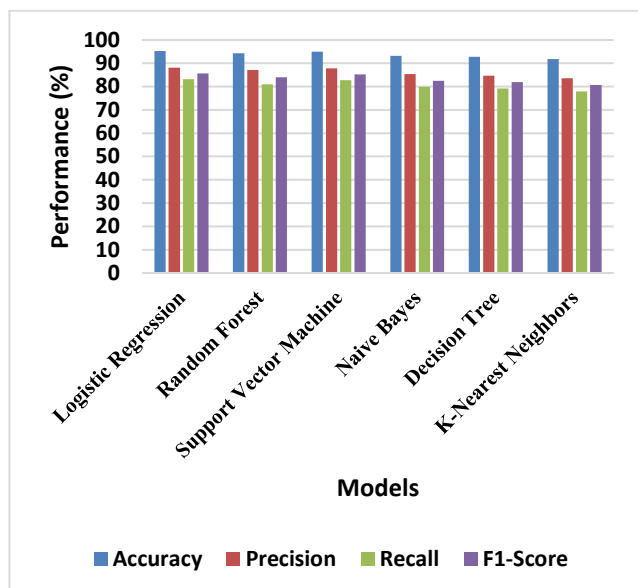


Fig 3: Performance Metrics of Machine Learning Models

## 5. Limitation study

This study reveals that while machine learning models, such as Logistic Regression and Support Vector Machines, exhibit high accuracy and balanced performance in classifying toxic comments, there are significant limitations. Although the models are effective in many scenarios, they often face challenges in accurately capturing the nuances of context-specific languages, which can lead to misclassification in complex or ambiguous cases. Moreover, dependence on the Jigsaw Toxic Comment Classification dataset introduces potential biases, limiting the generalizability of the models to other platforms or linguistic contexts. Despite the high F1-scores achieved, these models may struggle in real-time applications, where the dynamic and evolving nature of online content demands continuous adaptation and retraining. These findings underscore the need for more sophisticated and adaptive models that can address the inherent complexities of toxic languages across diverse and fluid online environments.

## 6. Conclusion

In conclusion, this study demonstrates that machine learning models, particularly Logistic Regression and Support Vector Machines, can effectively classify toxic comments with a high degree of accuracy, achieving up to 95.2% accuracy and an 85.6% F1-score. These models have shown strong potential for identifying and filtering harmful content in online environments, providing a critical tool for enhancing user experience and maintaining community standards. However, this study also highlights the limitations inherent in these models, particularly in their ability to handle context-specific languages and their reliance on a single dataset, which may restrict their applicability across different platforms and languages. Despite these promising results, these models require further refinement to address these challenges and improve their real-world performance.

Looking forward, there is a need to develop more adaptive and sophisticated models that can better capture

the nuances of toxic languages as they evolve over time and across different contexts. Future work should focus on integrating more diverse datasets to enhance model generalizability, exploring advanced techniques such as deep and transfer learning to improve model adaptability, and implementing continuous learning mechanisms to enable real-time updates as new data become available. In addition, incorporating multilingual capabilities is essential to ensure that these models can be effectively deployed across global platforms, further increasing their impact and utility in combating online toxicity.

**Author Contributions:** Chava Sai Sreeja and Mulamalla Akshaya Reddy contributed to the conceptualization and methodology. Poreddy Ishika Reddy and Rasamalla Yashwanth handled data analysis and validation. K Venkatesh Sharma provided supervision and final manuscript review.

**Ethical Considerations:** This study did not involve human or animal subjects. Therefore, ethical approval was not required for this study.

**Data availability:** Data are available upon request.

**Conflict of Interest:** There are no conflicts of interest to declare.

**Funding:** This study received no external funding.

**Similarity checked:** Yes.

## References

- [1] H. Kajla, J. Hooda, and G. Saini, "Classification of online toxic comments using machine learning algorithms," in *Proc. 2020 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, 2020, pp. 1119–1123.
- [2] J. Risch and R. Krestel, "Toxic comment detection in online discussions," in *Deep Learning-Based Approaches for Sentiment Analysis*, 2020, pp. 85–109.
- [3] V. U. Gongane, M. V. Munot, and A. D. Anuse, "Detection and moderation of detrimental content on social media platforms: Current status and future directions," *Soc. Netw. Anal. Min.*, vol. 12, no. 1, p. 129, 2022.
- [4] U. G. Nallabolu, "Classification of online toxic comments using machine learning algorithms," 2023.
- [5] P. A. Ozoh, A. A. Adigun, and M. O. Olayiwola, "Identification and classification of toxic comments on social media using machine learning techniques," *Int. J. Res. Innov. Appl. Sci. (IJRIAS)*, vol. 4, no. 11, pp. 142–147, 2019.
- [6] S. Zaheri, J. Leath, and D. Stroud, "Toxic comment classification," *SMU Data Sci. Rev.*, vol. 3, no. 1, p. 13, 2020.
- [7] H. M. Saleem, K. P. Dillon, S. Benesch, and D. Ruths, "A web of hate: Tackling hateful speech in online social spaces," 2017. [Online]. Available: <http://arxiv.org/abs/1709.10159>.



- [8] M. Duggan, "Online harassment 2017," *Pew Research Center*, 2017. [Online]. Available: <https://www.pewresearch.org/internet/2017/07/11/online-harassment-2017/>.
- [9] M. A. Walker, P. Anand, J. E. F. Tree, R. Abbott, and J. King, "A corpus for research on deliberation and debate," in *Proc. 8th Int. Conf. Lang. Resour. Eval. (LREC)*, 2012, pp. 812–817.
- [10] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial behaviour in online discussion communities," in *Proc. 9th Int. Conf. Web Soc. Media (ICWSM)*, 2015, pp. 61–70.
- [11] B. Mathew et al., "Thou shalt not hate: Countering online hate speech," in *Proc. 13th Int. Conf. Web Soc. Media (ICWSM)*, 2019, pp. 369–380.
- [12] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proc. 25th Int. World Wide Web Conf. (WWW 2016)*, 2016, pp. 145–153, doi: [10.1145/2872427.2883062](https://doi.org/10.1145/2872427.2883062).
- [13] E. K. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques," *WSEAS Trans. Comput.*, vol. 4, no. 8, pp. 966–974, 2005.
- [14] M. R. Murty, J. V. Murthy, and P. V. G. D. Reddy, "Text document classification based on least square support vector machines with singular value decomposition," *Int. J. Comput. Appl.*, vol. 27, no. 7, pp. 21–26, 2011, doi: [10.5120/3312-4540](https://doi.org/10.5120/3312-4540).
- [15] S. Kiran and S. Rallapall, "Innovative Blockchain Split-Join Architecture for Optimized Data Management," *Synth. Multidiscip. Res. J.*, vol. 1, no. 3, pp. 1–11, Sep. 2023.
- [16] V. S. S. D., W. Ali, L. Gaurav, S. Sakinam, and B. M., "A Systematic Analysis of Text Classification Overfitting Recommendation Methods," *Int. J. Comput. Eng. Res. Trends*, vol. 10, no. 4, pp. 188–198, May 2023.
- [17] T. R. Singasani, "Implementing PEGA for Enhanced Business Process Management: A Case Study on Workflow Automation," *J. Sci. Eng. Res.*, vol. 6, no. 7, pp. 292–297, 2019.
- [18] N. J. Bommagani et al., "Artificial Butterfly Optimizer Based Two-Layer Convolutional Neural Network with Polarized Attention Mechanism for Human Activity Recognition," *Math. Model. Eng. Probl.*, vol. 11, no. 3, 2024.
- [19] M. N. Varadarajan et al., "Smart Healthcare Data Protection and Analysis Through Fuzzy-Based Cyber Security," *J. Environ. Prot. Ecol.*, vol. 25, no. 5, pp. 1604–1614, 2024.
- [20] T. R. Singasani, "Integrating PEGA and MuleSoft with Cloud Services: Challenges and Opportunities in Modern Enterprises," *J. Sci. Eng. Res.*, vol. 7, no. 3, pp. 328–333, 2020.
- [21] T. R. Singasani, "Leveraging MuleSoft for Cloud Integration: Exploring API-Led Connectivity for Seamless Multi-Cloud Architectures," *Eur. J. Adv. Eng. Technol.*, vol. 7, no. 8, pp. 96–99, 2020.
- [22] S. Jotteppa et al., "Designing a Smart IoT Environment by Predicting Chronic Kidney Disease Using Kernel-Based Xception Deep Learning Model," *Rev. Intell. Artif.*, vol. 38, no. 1, 2024.
- [23] T. R. Singasani, "Integrating PEGA with IoT: Enhancing Data-Driven Decision Making in Smart Cities," *Int. J. Sci. Res.*, vol. 10, no. 5, pp. 1361–1362, May 2021.
- [24] T. R. Singasani, "Optimizing Decision Strategies with PEGA: A Data-Driven Approach to Predictive Analytics," *J. Sci. Eng. Res.*, vol. 7, no. 12, pp. 267–271, 2020.
- [25] T. R. Singasani, "Leveraging PEGA and MuleSoft for Seamless API Integration: A Comprehensive Framework," *Int. J. Sci. Res.*, vol. 9, no. 6, pp. 1955–1957, 2020.
- [26] T. R. Singasani, "Comparative Analysis of PEGA and MuleSoft: Efficiency, Scalability, and User Experience," *Eur. J. Adv. Eng. Technol.*, vol. 6, no. 11, pp. 119–122, 2019.
- [27] S. Z. Rahman, T. R. Singasani, and K. S. Shaik, "Segmentation and Classification of Skin Cancer in Dermoscopy Images Using SAM-Based Deep Belief Networks," *Healthcraft Front.*, vol. 1, no. 1, pp. 15–32, 2023.
- [28] T. R. Singasani, "Enhancing Customer Experience Through PEGA's AI-Powered Decisioning," *J. Sci. Eng. Res.*, vol. 9, no. 12, pp. 191–195, 2022.
- [29] T. R. Singasani, "PEGA in the Era of 6G: Exploring the Future of Connected Systems and Automation," *Eur. J. Adv. Eng. Technol.*, vol. 9, no. 5, pp. 145–148, 2022.
- [30] T. R. Singasani, "Evaluating the Impact of PEGA on Financial Services: Automation and Regulatory Compliance," *Eur. J. Adv. Eng. Technol.*, vol. 8, no. 12, pp. 148–151, 2021.
- [31] T. R. Singasani, "PEGA and Robotic Process Automation: Synergies for Next-Generation Workflow Management," *Eur. J. Adv. Eng. Technol.*, vol. 8, no. 2, pp. 134–137, 2021.
- [32] P. S. Muller et al., "Improving Diabetes Diagnosis in Instantaneous Situations with MANET and Data Mining," *J. Environ. Prot. Ecol.*, vol. 25, no. 4, pp. 1330–1343, 2024.
- [33] T. R. Singasani, "Leveraging PEGA and IoT for Industrial Automation: Challenges and Solutions," *Int. J. Sci. Res.*, vol. 11, no. 8, pp. 1560–1562, Aug. 2022.
- [34] T. R. Singasani, "PEGA's Role in Streamlining Healthcare Processes: A Study on Patient Management Systems," *J. Sci. Eng. Res.*, vol. 8, no. 9, pp. 322–325, 2021.

