Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Toxic Comment Classification using Deep Learning

Abinayashri S
B.Tech Information Technology
Alpha College of engineering,
Chennai

Priyadharshini V M
B.Tech Information Technology
Alpha College of engineering,
Chennai

Teja V
B.Tech Information Technology
Alpha College of engineering,
Chennai

Abstract **- The rapid growth of social media platforms has led to a significant increase in toxic and abusive content, which negatively impacts user experience and online communities. Detecting such harmful content manually is inefficient and time-consuming. This paper presents a machine learning-based system for automatic toxic comment classification using Natural Language Processing (NLP) techniques. The proposed system utilizes text preprocessing, feature extraction methods such as TF-IDF and word embeddings, and classification algorithms including Logistic Regression and Long Short-Term Memory (LSTM) networks. The model supports multi-label classification to identify different categories of toxicity such as obscene, threat, insult, and identity hate. Experimental results demonstrate improved accuracy and performance compared to traditional approaches. The system can be integrated into social media platforms for real-time moderation, ensuring a safer digital environment.**

Keywords: Toxic Comment Classification, NLP, Machine Learning, Deep Learning, Text Mining, Multi-label Classification, Social Media Analysis

I. INTRODUCTION

The exponential growth of social media platforms and online communication channels has led to a significant increase in user-generated textual content. While these platforms encourage free expression and global interaction, they have also become a medium for spreading toxic and abusive language, including hate speech, insults, threats, and offensive remarks. Such content can negatively impact individuals and communities, making it essential to develop automated systems for detecting and filtering toxic comments.

Toxic Comment Classification is a key task in the field of Natural Language Processing (NLP), aimed at identifying and categorizing harmful text. Traditional moderation techniques, such as manual review and rule-based filtering, are inefficient and not scalable due to the massive volume of online data. Although classical machine learning methods have been applied to this problem, they often rely on handcrafted features and fail to capture the contextual and sequential nature of language effectively. To overcome these limitations, deep learning techniques have been widely adopted for text classification tasks. Among these, Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), have shown significant success in processing sequential data. LSTM models are designed to retain long-term dependencies in text through specialized memory cells and gating mechanisms, allowing them to understand context, word order, and semantic relationships more effectively than traditional approaches. This makes LSTM particularly suitable for analysing user comments, where the meaning of a sentence often depends on the sequence of words. In this project, an LSTM-based deep learning model is proposed for the classification of toxic comments. The system processes textual data through several stages, including data preprocessing, tokenization, and sequence padding. Word embeddings are utilized to convert textual input into dense vector representations, enabling the model to learn semantic features. The processed sequences are then fed into the LSTM network, which captures contextual information and performs multi-label classification of comments into categories such as toxic, severe toxic, obscene, threat, insult, and identity hate.

The main objective of this work is to develop an efficient and robust model capable of accurately detecting toxic content in real-time. By leveraging the strengths of LSTM networks in handling sequential text data, the proposed system aims to achieve improved performance in terms of accuracy and generalization. This approach can be integrated into online platforms to assist in automated content moderation, thereby promoting safer and more respectful digital interactions.

II. RELATED WORKS

The problem of toxic comment classification has attracted significant attention in recent years due to the rapid growth of online platforms. Various approaches have been proposed, ranging from traditional machine learning techniques to advanced deep learning models.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Early studies on toxic comment detection primarily relied on classical machine learning algorithms such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression. These methods utilized handcrafted features like bag-of-words, TF-IDF (Term Frequency–Inverse Document Frequency), and n-grams to represent textual data. While these approaches achieved moderate performance, they were limited in capturing contextual meaning and semantic relationships within sentences.

With the advancement of deep learning, researchers began applying neural network-based models to improve classification accuracy. Convolutional Neural Networks (CNNs) were initially used for text classification tasks due to their ability to extract local features and patterns. However, CNNs are less effective in capturing long-term dependencies in sequential text data

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have been widely adopted for toxic comment classification due to their ability to process sequential data and retain contextual information over long text sequences. Several studies have demonstrated that LSTM models outperform traditional machine learning techniques by effectively understanding the order and meaning of words in a sentence. Bidirectional LSTM (Bi-LSTM) models have further improved performance by capturing context from both past and future directions.

In addition to standalone models, hybrid approaches combining CNN and LSTM architectures have also been explored to leverage both local feature extraction and sequential learning. Furthermore, the use of pre-trained word embeddings such as Word2Vec and GloVe has enhanced model performance by providing rich semantic representations of text.

More recently, transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers) have achieved state-of-the-art results in text classification tasks, including toxic comment detection. However, these models are computationally expensive and require significant resources, making them less suitable for real-time or resource-constrained applications.

In comparison to existing approaches, the proposed LSTM-based model focuses on achieving a balance between performance and computational efficiency. By utilizing sequence modelling capabilities and word embeddings, the system aims to provide accurate and scalable toxic comment classification suitable for practical deployment.

III. PROPOSED WORK

The proposed system is designed to automatically detect and classify toxic comments using an LSTM-based deep learning model. The workflow follows a structured pipeline where raw user input is progressively transformed into meaningful predictions. The detailed working process is described below:

Step 1: Data Input / Collection

The system begins with a dataset containing labeled user comments. Each comment is associated with one or more toxicity labels such as toxic, obscene, insult, etc.

Step 2: Data Preprocessing

Raw text is cleaned by removing noise such as special characters, punctuation, and unnecessary words. Text is also converted to lowercase.

Step 3: Tokenization (Text → Numbers)

Each word in the comment is converted into a numerical index using a tokenizer.

Step 4: Sequence Padding

All tokenized comments are adjusted to the same length by adding padding (zeros).

Step 5: Word Embedding Layer

The numerical tokens are converted into dense vectors using an embedding layer.

Step 6: LSTM Processing (Core of the Model)

The embedded sequences are passed into the LSTM network.

For example, it distinguishes between:

“You are bad” (toxic)

“This is a bad situation” (non-toxic context)

Step 7: Feature Learning and Pattern Detection

The LSTM extracts hidden features from the sequence.

Step 8: Output Layer (Classification)

The processed data is passed to a Dense layer with sigmoid activation.

Step 9: Multi-label Decision Making

A threshold (e.g., 0.5) is applied to assign labels.

Step 10: Model Training Process

During training:

The model compares predictions with actual labels Error is calculated using Binary Cross-Entropy Weights are updated using the Adam optimizer.

Step 11: Model Evaluation

The model is tested on unseen data.

Step 12: Prediction on New Comments

When a new comment is entered:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Flow:

Input text → preprocessing

Tokenization → padding

Embedding → LSTM

Output probabilities → labels

Step 13: Deployment (How It Works in Real Use)

The trained model is integrated into an application (e.g., web app or API).

How deployment works:

User enters a comment in UI (like a text box)

Backend sends the text to the trained model

Model processes and returns prediction

System displays result or blocks the comment

Example: Input: “You are useless”

Output: Toxic, Insult

Action: Comment flagged or removed

Step 14: System Outcome

* Sequence Modelling using LSTM

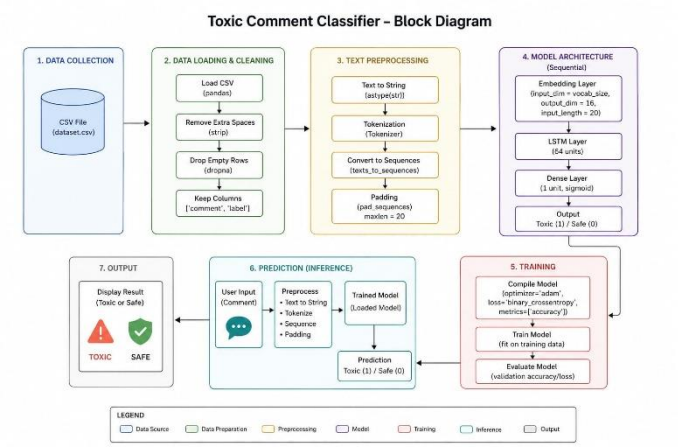
The proposed model leverages Long Short-Term Memory (LSTM) networks to effectively capture sequential dependencies in textual data. This allows the system to understand the contextual flow and order of words in a sentence, which is crucial for detecting toxic intent that depends on phrasing and context rather than individual words.

* Word Embedding Representation

To improve semantic understanding, the model utilizes word embedding techniques that convert textual data into dense vector representations. This enables the system to recognize similarities and relationships between words, helping it identifies toxic content even when different words or expressions are used.

* Multi-label Classification Capability

The system is designed to perform multi-label classification, allowing a single comment to be categorized into multiple toxicity classes simultaneously. This is important because a comment



can exhibit multiple forms of toxicity, such as being both insulting and obscene, thereby improving the accuracy and realism of classification.

VI. RESULTS AND DISCUSSIONS

A. Dataset

The model is trained using the Kaggle Toxic Comment Classification Dataset, which contains thousands of labelled comments using CSV file.

B. Performance Metrics

The system is evaluated using:

- Accuracy
- Precision
- Recall
- F1-Score
- C. Results
- Model Accuracy
- Logistic Regression - 89%
- LSTM Model - 93%

The LSTM model performs better due to its ability to understand context and sequence.

V. SYSTEM ARCHITECTURE

The system consists of the following layers:

Input Layer

User comment

Preprocessing Layer

Cleaning and tokenization

Feature Extraction Layer

TF-IDF / Word Embeddings

Classification Layer

Logistic Regression / LSTM

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Output Layer

Toxicity labels

This architecture ensures efficient and accurate classification of text data.

VI. CONCLUSION AND FUTURE WORK

The proposed toxic comment classification system successfully detects and categorizes harmful content using machine learning and NLP techniques. The system improves moderation efficiency and helps maintain a safe online environment.

Future Work

- Integration with real-time social media platforms
- Use of transformer models like BERT
- Detection of sarcasm and contextual toxicity
- Multilingual support

VII. REFERENCES

[1] Jigsaw, "Toxic Comment Classification Challenge," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proc. International Conference on Learning Representations (ICLR), 2013.
[3] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[4] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proc. EMNLP, 2014, pp. 1746–1751.
[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[6] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," Springer, 2012.
[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[9] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[10] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," in Proc. EACL, 2017.
[11] Z. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in Advances in Neural Information Processing Systems (NIPS), 2015.
[12] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, 2020.
[13] R. Socher et al., "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in Proc. EMNLP, 2013.
[14] N. Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," JMLR, vol. 15, pp. 1929–1958, 2014.
[15] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. KDD, 2016.
[16] J. Brownlee, Machine Learning Mastery with Python, Machine Learning Mastery, 2016.
[17] TensorFlow, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. [Online]. Available: https://www.tensorflow.org/
[18] Python Software Foundation, "Python Language Reference," 2023. [Online]. Available: https://www.python.org/
[19] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009.
[20] Google AI, "Text Classification Guide," 2022. [Online]. Available: https://developers.google.com/machine-learning/guides/text-classification
[21] W. Warner and J. Hirschberg, "Detecting Hate Speech on the World Wide Web," in Proc. Workshop on Language in social media, 2012.
[22] P. Badjatiya et al., "Deep Learning for Hate Speech Detection in Tweets," in Proc. WWW Companion, 2017.