Future Technology
Open Access Journal
ISSN 2832-0379

August 2025| Volume 04 | Issue 03 | Pages 76-84

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.4.3.8

Future Publishing LLC

Article

Enhanced toxic comment detection model through Deep Learning models using Word embeddings and transformer architectures

S. Sushma[1,2]*, Sasmita Kumari Nayak[1], M. Vamsi Krishna[3]

[1]Department of CSE, Centurion University of Technology and Management, Bhubaneswar, Odisha, India
[2]Aditya University, Surampalem, India
[3]Department of Computer Applications, Aditya University, Surampalem, India

ARTICLE INFO

Article history:
Received 08 April 2025
Received in revised form
21 May 2025
Accepted 31 May 2025

Keywords:
Toxic comment classification, Word embeddings, Ensemble modeling

*Corresponding author
Email address:
sushma.cse2@gmail.com

DOI: 10.55670/fpll.futech.4.3.8

ABSTRACT

The proliferation of harmful and toxic comments on social media platforms necessitates the development of robust methods for automatically detecting and classifying such content. This paper investigates the application of natural language processing (NLP) and ML techniques for toxic comment classification using the Jigsaw Toxic Comment Dataset. Several deep learning models, including recurrent neural networks (RNN, LSTM, and GRU), are evaluated in combination with feature extraction methods such as TF-IDF, Word2Vec, and BERT embeddings. The text data is pre-processed using both Word2Vec and TF-IDF techniques for feature extraction. Rather than implementing a combined ensemble output, the study conducts a comparative evaluation of model-embedding combinations to determine the most effective pairings. Results indicate that integrating BERT with traditional models (RNN+BERT, LSTM+BERT, GRU+BERT) leads to significant improvements in classification accuracy, precision, recall, and F1-score, demonstrating the effectiveness of BERT embeddings in capturing nuanced text features. Among all configurations, LSTM combined with Word2Vec and LSTM with BERT yielded the highest performance. This comparative approach highlights the potential of combining classical recurrent models with transformer-based embeddings as a promising direction for detecting toxic comments. The findings of this work provide valuable insights into leveraging deep learning techniques for toxic comment detection, suggesting future directions for refining such models in real-world applications.

1. Introduction

There is an increasing amount of harmful and toxic comments that may harm users' experience, with the exponentially growing user-generated content available on social media platforms. Hence, the need for an automated system that will detect and filter out such rotten content has become crucial to maintaining a positive and healthy environment on the internet. This motivates the adoption of more advanced ML and NLP techniques, as traditional content moderation systems have high false-positive and false-negative rates and often fail to scale well to larger volumes of data. Detecting toxic comments has been addressed with different approaches in recent years. Now, there are two things we can guess from the name of the model above: one is that it could be any classical ML algorithm, and the other is that it is used for binary classification. These techniques work reasonably well most of the time, but do not necessarily capture all the subtleties of language, particularly when processing unstructured content like social media comments. On this training, it's a great performance in sequence data modeling, especially with DL models like RNN, LSTM, and GRU. These models work proficiently with text classification. To understand contextuality and sentiment in textual data, these models have the capability to learn temporal dependencies in sequential data. Furthermore, transformer-based models such as BERT have transformed NLP tasks by capturing extensive contextual information and learning contextual word embeddings. Despite significant progress in automated toxic comment classification using ML and DL techniques, existing models often struggle with accurately capturing context, handling imbalanced datasets, and distinguishing between subtle toxic and non-toxic content. Additionally, many previous works either use only shallow feature representations or do not take full advantage of

classical and transformer-based embeddings. We aim to resolve these issues in this work by conducting experiments with hybrid DL models that rely on the TF-IDF, Word2Vec, and BERT embeddings for better toxic comment detection.

Abbreviations

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term Frequency–Inverse Document Frequency
Word2Vec	Word to Vector

This work distinguishes itself from prior studies by offering a comprehensive and unified comparison of various word embedding strategies (TF-IDF, Word2Vec, BERT) in combination with sequential models (RNN, LSTM, GRU), all evaluated under consistent preprocessing, tokenization, and training configurations. Unlike existing research that often benchmarks one or two models, this study explores a wide array of hybrid architectures (e.g., LSTM+BERT) to identify optimal pairings for toxic comment classification. The implementation is designed to reflect practical deployment scenarios using real-world metrics and balanced experimental design. The main contributions of this work are listed below:

- To design RNN and other deep learning models to compare for this toxic comment classification.
- To explore the performance of different embedding approaches, such as TF-IDF, Word2Vec, and BERT, for semantics and context features.
- Investigate using hybrid architectures that connect traditional deep learning with transformer-related embeddings.
- To determine the classification accuracy, precision, recall, and F1-score of all models on the basis of real-world toxic comment data.

2. Literature survey

A. Albladi et al. [1] investigated sentiment research specifically on Twitter, which is extensive in scale and a valuable data source for understanding public opinion. They reviewed desirable and problematic features and metrics of ML, DL, and hybrid approaches. You focus on the BERT and GPT transformer architectures, although seismic pre-processing techniques, the extraction of features, and sentiment lexicons were also summarily considered. This research aimed to give a comprehensive landscape of use cases in Twitter sentiment analysis, offering useful insights for practitioners and researchers in this area.

Z. Hao et al. [2] discussed the intricate complexities of gathering public views, particularly regarding social media and its impacts on social incidents. A two-tier hierarchical mechanism was proposed for report categorization and review-level sentiment analysis, aiming to effectively derive sentiment through a multi-step approach. This method was augmented with advances in model architecture, including embedding tables and gating mechanisms. Moreover, they proposed a new distributed DL model which is based on blockchain isomerism learning for the security risks and reducing the model silos. Through a series of extensive experiments, they showed that, for performance and aggregation efficiency, their approach vastly outperforms existing methods.

X. Wang et al. [3] addressed the challenges in sentiment analysis by introducing a novel model that integrates multimodal multiscale features based on a fuzzy-deep neural network. It utilizes intrinsic feature representations across text, audio, and image data. The model employs fuzzy logic rules to increase the adaptability to the ambiguity present in sentiment representations. Additionally, they incorporated a dual attention mechanism to flexibly attend to essential components in the multimodal data, thereby enhancing feature extraction and improving contextual awareness. We conducted extensive validation of the model's performance using several datasets and characterized its ability to model the complexities of human emotion better than existing methods. H. T. Phan et al. [4] devised a novel method of aspect-level sentiment analysis based on the application of three GCNs, which they coined as multigraphic convolutional network (MulGCN). Using the dependency parser tree, affective information from SenticNet , and inter-aspect-awareness, this approach captures both syntax and semantics as well as context. This paper presents a model that introduces a solution for aspect-level sentiment analysis and improves its performance by addressing the difficulty of effectively leveraging relevant features from different knowledge sources. Experimental evaluations over three benchmark datasets demonstrate that the MulGCN model beats the state-of-the-art and improves both accuracy and F1 score for aspect-level sentiment analysis. S. Ali et al. [5] highlighted the issue of [event classification] in low-resource languages, such as Urdu, and underline the need for well-formalised linguistic datasets. The dataset contained a total of 103,771 sentences from five different social media platforms and was obtained for the purpose of classifying text in the Urdu language in a multiclass classification approach. The 16 event categories were used in the classification task. The SMFCNN classifier showed the highest accuracy (88.29%) among the methods tested. Furthermore, on this dataset, XLM-R+ (a proposed transformer-based model) outperformed them with an accuracy of 89.8% as well.

W. Gong [6] developed a sentiment classification algorithm for textual data mining based on bidirectional long short-term memory network (BiLSTM), which demonstrated the relevance of emotion manifestation in e-commerce and social media data with DL methods. The model enhances the accuracy in distinguishing sentiment in such cases and outperforms conventional sentiment classification techniques due to the utilization of BiLSTM, which adopts bidirectional contexts of text segments [4,5]. This study demonstrates that BiLSTM can also be a viable option for sentiment analysis tasks, such as customer feedback analysis. S. A. Mostafa et al. [7] employed a sentiment analysis and classification method for Amazon Alexa products. The dataset contains 3150 reviews and was labelled for positive or negative sentiment. They therefore trained four classifiers and evaluated the performance metrics. Analysis of customer feedback was found to be highly effective using RF, which was the best-performing classifier. S. Mehta et al. [8] applied CNN as a new method for sentiment analysis with Federated Learning. It enables model-trained decentralization by processing multiple devices without transferring data to the central server, thereby improving privacy while maintaining sufficient performance for the sentiment classifier. It is based on high accuracy and a good ROC AUC score, denoting good

separation of classes as different sentiment groups. As a result, it enhanced data privacy and efficiency due to the use of federated learning. X. He et al. [9] compared SVM for sentiment analysis with other traditional classifiers such as LR, KNN, NB, and XGBoost. This study applies sentiment analysis to classify comments about "Huawei Mate60" on the Little Red Book platform. Compared to the other models, SVM proved to be more efficient for sentiment classification, making it a recommended approach for analyzing consumers' feelings, which can help brands manage their products better. M. Aamir et al. [10] conducted a similar study using different ML and DL techniques to find public sentiment on tweets about Ola and Uber. In this study, different algorithms were evaluated to understand customer feedback for these ride-hailing services. The study assisted in improving the accuracy of algorithms employed and facilitated companies to tailor their services according to user sentiments, leading to a healthier online ecosystem.

Q. Zeng et al. [11] proposed a neural network model for aspect-based sentiment classification, integrating a self-attention mechanism to capture contextual semantic information. Relative Position Representations (PRP) were produced by attending to a global context, while pairs of words were given ReLU gated convolutional networks for sentiment feature extraction. It has also been shown that their model outperforms every other model in terms of predicting valence and arousal on the SemEval dataset, Tweets, and CVAT, confirming its robustness for sentiment analysis. M. Khalid et al. [12] proposed a Sentiment Majority Voting Classifier (SMVC) to analyze the sentiment of deepfake technology-related tweets. The authors used majority voting to aggregate the predictions from several lexicon-based models and used transfer learning with LSTM and Decision Tree models. The method reached the best accuracy of 98.9%, demonstrating robustness in sentiment classification. A.L.Rao et al. [13] focused on sentiment analysis of the airline tweet or airline reviews Kaggle dataset. They suggested an ensemble architecture for CNN and LSTM for improving sentiment classification. They benchmarked this ensemble model against a standalone LSTM model, which was also trained on the same dataset, and reported the performance of each and stated that the ensemble approach provided superior performance, providing an improvement on standard methods. Y. Matrane et al. [14] performed the sentiment analysis of MD, discussing specifically the issues related to dialect-specific preprocessing techniques. They reported better results by not utilizing traditional techniques like stemming and by using the QARiB feature extractor with BiGRU. The results of DarijaBERT on the FB dataset showed their fine-tuning approach to be effective, signifying the need for dialect-specific techniques in dealing with the Arabic dialect. H. Shuqin et al. [15] implemented a BERT-BiLSTM-ATTENTION (BBA) model to recognize sentiment for course evaluation. The model was built by leveraging the BERT model for context, along with BiLSTM and attention mechanisms to better refine the focus across the relevant components of text. The BBA model beat the existing methods, showing that the deep semantic representation of education-based feedback could be achieved through this model.

A. He et al. [16] introduced a novel deep tensor evidence fusion network for multimodal sentiment classification. They introduced a joint view scoring net that integrates LSTM and tensor neural networks to extract the intermodal and intramodal rich information. They also proposed a temporal cue evaluation network based on temporal granularity and a trustworthy fusion layer to enhance the accuracy and robustness of decisions. On the CMU-MOSEI and CMU-MOSI datasets, the results were better than those of SOTA methods. Z. Wang et al. [17] proposed a multi-label classification approach for handling toxic comments in social media, specifically focusing on Indonesia's Twitter platform. Using two kinds of word vectors from BERT's hidden layers and combining both BERT + BiLSTM method, the model achieved better performance. In such a complex form of sentiment analysis task, the proposed model achieved an accuracy of 0.889, precision of 0.925, recall of 0.917, and F1 score of 0.91, thus proving the efficacy of task-specific semantic embedding and sequential learning through BiLSTM. S. K. Putri et al. [18] applied various ML models to predict toxic leadership in the Moroccan IT sector. In this regard, the study identified Undermining Behavior, Narcissistic Traits, Unjust Treatment, and Fear of Retribution as the main contributors to toxic leadership, as it sought to provide a comprehensive analysis of how toxic leadership could be predicted using different types of ML algorithms. A. Lakshmanarao et al. [19] sentiment analysis of airlines tweet gathered from Kaggle. They applied different neural network approaches to classify tweets into different categories.

S. Dutta et al. [20] addressed the toxic comment detection problem in Assamese, a morphologically rich and ambiguous language, which poses a challenge to sentiment analysis. This paper is part of a very large study of general NMF topic modeling, which indicates preceding work by manually collecting 19,550 comments from social media sites and testing a number of ML models. All these models have been trained against various text representations (count vector, count vector + TF-IDF, n-gram, etc.) The best F1-score was 94% with SVM + count vector + TF-IDF compared to the other models. Y. Mamani et al. [21] provided a summary of recent ML techniques used for sentiment analysis, developing a framework to classify sentiment models according to their structure. The paper also discussed challenges faced by the community and emerging trends, providing future directions for the research in sentiment analysis.

Rahul et al. [22] addressed the classification problem for toxic comments, which can be used to measure the severity of online harassment. They examined online comments without a focus on toxicity using six machine learning (ML) algorithms. They worked to enhance the classification of negative information within textual comments, utilizing an enhanced classification to mitigate harmful information. This enables organizations to identify toxins in discussions and take action to lead to better environments. M. Aquino et al. In Ref [23], the authors explored a new ML-based method to detect comment toxicity using text and emojis. They trained a bidirectional LSTM model using GloVe and emoji2vec combined word embeddings. It provides a new labeled dataset of text and emoji data, providing an effective means of comment toxicity detection. T. V. Sai Krishna et al. Using a range of features and both ML & DL methods, Ref [24] proposes sentiment classification on Twitter. They used several ML models. They also proposed a new end-to-end ensemble approach of using ML and DL models, which yields higher accuracy vs. these techniques when applied to i.e., real-time Twitter data for sentiment classification.

For example, Singh et al. [25] detected levels of toxicity in comments on social media using the Jigsaw dataset from Google. The dataset is a multilabel classification task with several classes. The logistic regression model performed the best in terms of accuracy and Hamming loss among the other models, indicating that it is well-suited to the task of

identifying toxic comments. Venugopal [26] explored the classification of toxic comments and identified challenges of detecting the toxicity of comments in multiple languages in a centralized system for comment detection across social media platforms. They used state-of-the-art DL architectures such as BERT or XLM-RoBERTa for multilingual toxicity detection. The paper noted that these models, when paired with appropriate preprocessing of datasets and tuning of hyperparameters, can provide significant improvement in the accuracy of detecting toxic comments compared to traditional models such as SVM. N. Boudjani et al. [27] used N-grams, linguistic features, and a lexicon of insulting words to create a supervised method for the classification of French toxic comments. Using linear SVM and decision tree classifiers, their approach yielded precision, recall, and F1-score values of 87%, 83%, and 78%, respectively. A. Jessica et al. applied BERT-CNN and BERT-LSTM hybrid models for detecting cyberbullying in online comments. This model consists of BERT and CNN, which are created through the combination of sentences by using BERT so BERT has good language understanding ability, and using CNN for feature extraction work.

3. Methodology

The proposed method for toxic comment classification is shown in Figure 1. The proposed method for toxic comment detection is designed to evaluate various DL models in combination with word embedding techniques such as Word2Vec, TF-IDF, and BERT. The method is applied to the Jigsaw Toxic Comment Dataset, which contains labeled English comments categorized as toxic or non-toxic. The dataset undergoes preprocessing to clean the text by handling inconsistencies such as missing values and ensuring that the comment texts are appropriately formatted for further analysis.

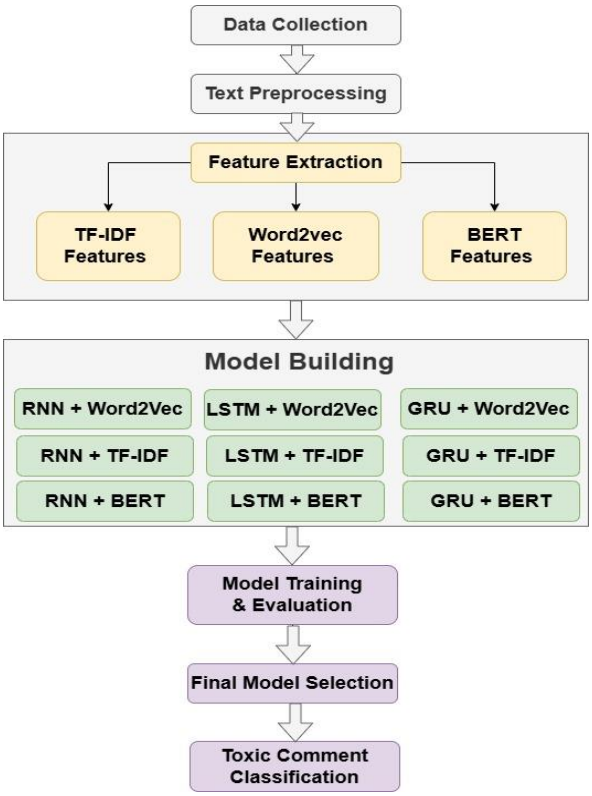

Figure 1. Proposed methodology for toxic comment classification

TF-IDF and Word2Vec are used to extract text features in two distinct ways. TF-IDF computes the relative importance of a word within a document by analyzing its frequency in relation to the entire corpus, generating sparse numerical vectors. In contrast, Word2Vec learns distributed word representations by training on the tokenized text data, capturing semantic relationships between words through co-occurrence patterns. Each comment is then represented as a fixed-length vector by averaging the embeddings of its words. These two methods provide complementary insights—TF-IDF emphasizes statistical significance, while Word2Vec captures contextual meaning. The next step involves applying different DL models, namely RNN, LSTM, and GRU, to classify the comments as either toxic or non-toxic. Each model architecture consists of an embedding or input layer, a recurrent layer (RNN, LSTM, or GRU), and a dense output layer for binary classification. The structure remains consistent across experiments, with only the embedding source (TF-IDF, Word2Vec, or BERT) varying. BERT, a transformer-based model, is also employed for richer feature extraction. This pre-trained BERT model uses a large corpus to train and fine-tunes itself on the toxic comment dataset. In this setup, token embeddings are extracted using the DistilBERT tokenizer, particularly from the [CLS] token. These embeddings are then passed through the recurrent models to improve contextual interpretation [28].

This architecture produces hybrid models such as RNN+BERT, LSTM+BERT, and GRU+BERT, where BERT serves solely as a feature extractor, providing contextual embeddings. These embeddings are fed as input to the corresponding recurrent layers for learning temporal dependencies prior to classification. BERT is not used as a classifier in this work, but it significantly enhances the input representation for downstream learning. This approach enables the models to recognize complex and contextual relationships between words in comments. Final classification is performed by the dense output layer. The entire framework is designed to output the probability of a comment being toxic based on the learned features. All models are evaluated on a range of performance metrics, including accuracy, precision, recall, and F1 score. The outcome of each model is compared to decide which is the best way to detect toxicity in online commentary. This experimental design enables a comparative evaluation of different embedding-model combinations. Using traditional feature extraction techniques such as Word2Vec and TF-IDF together with state-of-the-art transformer-based models like BERT, the proposed algorithm has high classification accuracy. It also verifies the potential for hybrid DL models as a means of fulfilling text categorisation work.

3.1 Data collection

This paper used the Jigsaw Toxic Comment Dataset, collected from Kaggle [29]. The dataset contains a collection of English language comments, each labeled as either toxic or non-toxic. The dataset consists of two important columns: comment_text, which contains the actual comment text, and toxic, a binary target variable (1 refers to toxic comments, while 0 refers to all other types of comments). This data comes from a variety of online sources, including Civil Comments and Wikipedia talk page edits. Text-based prediction of whether a comment is toxic or not.

3.2 Preprocessing

In the preprocessing phase, a variety of procedures were used to clean and prepare the data for modeling. This involved eliminating all kinds of inconsistencies, such as missing values in required columns (comment_text and toxic). Missing entries like these are corrected to preserve the completeness of the dataset and its suitability for analysis. Later, the data was tokenized and transformed into a form suitable for input into ML models, optimizing both training and validation datasets so that they would work perfectly with the model.

4. Results and discussion

4.1 Applying RNN with TF-IDF

In this phase, the TF-IDF method is applied in combination with a SimpleRNN model for toxic comment classification. The text data is first vectorized using the TF-IDF technique, which converts the text into a numerical representation based on the frequency of terms in the dataset. To perform dimensionality reduction, 5,000 input features are extracted from the text data so that the model can concentrate on the most pertinent terms. We first split our data into training and test sets as follows: 80% for training and 20% for testing. The model consists of an input embedding layer, an RNN layer with 64 units, followed by an output dense layer with a unit for binary classification. To accelerate the training process, the model is trained for five epochs with a batch size of 256. The final evaluation on the test set yields a validation accuracy of around 89%, indicating that the model successfully captures the sequential nature of the text data and performs adequately in classifying toxic comments.

4.2 Applying LSTM with TF-IDF

In this step, an LSTM model is utilized in conjunction with TF-IDF for classifying toxic comments. The text data is vectorized using the TF-IDF method, which extracts relevant features based on the term frequency-inverse document frequency. The neural network is constructed in the form of an embedding layer that takes the input dimensions, and then an LSTM layer, which has 64 units. We add a dense output layer for binary classification (toxic vs non-toxic comments). The training process gradually improves accuracy, with the training done over 3 epochs using a function to train on one set of the dataset and the given statements, with a batch size of 16. The trained model is then evaluated on the test set, giving us a validation accuracy of 90.46% , demonstrating the model's ability to capture long-term dependencies within the data whilst providing a baseline for the classification of toxic comments.

4.3 Applying GRU with TF-IDF

This section demonstrates the application of a GRU (Gated Recurrent Unit) model with TF-IDF for the task of toxic comment classification. Similar to the LSTM model, TF-IDF is used in this model. The GRU model consists of an embedding layer, single GRU layer of size 64, followed by a dense output layer for predicting binary classes. The model is evaluated on the test data after training with a batch size of 16. The final validation accuracy ~90% generally indicates that the GRU model also effectively captured the relevant patterns from the text and achieved comparable classification accuracy with the LSTM model for classifying a toxic comment.

4.4 Applying RNN with Word2Vec

In this approach, Word2Vec embeddings were combined with an RNN to classify toxic and non-toxic comments. Word2Vec embeddings were first generated by training on the tokenized comments from the Jigsaw Toxic Comment Dataset. Each comment was represented by a fixed-length vector by averaging the embeddings of its words, with zero vectors assigned to out-of-vocabulary words. The padding was applied to ensure uniform input lengths for the RNN. The model was trained for 20 epochs with a batch size of 16. Figure 2 shows epoch-wise accuracy, and Figure 3 shows loss values with the model. It achieved a validation accuracy of approximately 90.76%, demonstrating the effectiveness of using Word2Vec embeddings and RNNs for toxic comment classification.

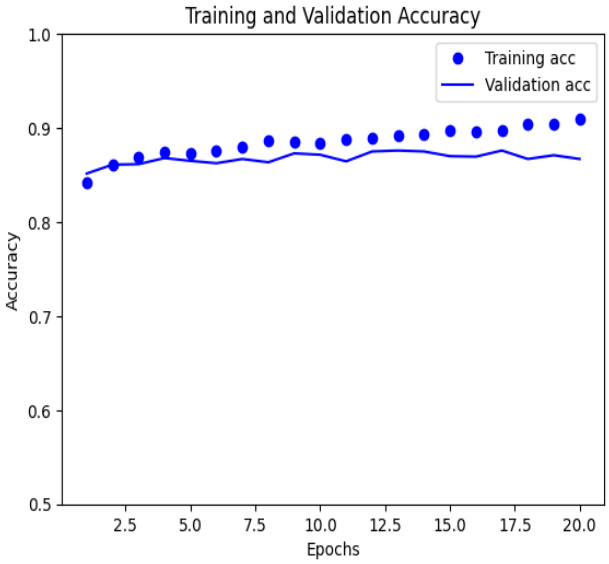

Figure 2. RNN+ Word2vec model epoch wise accuracy

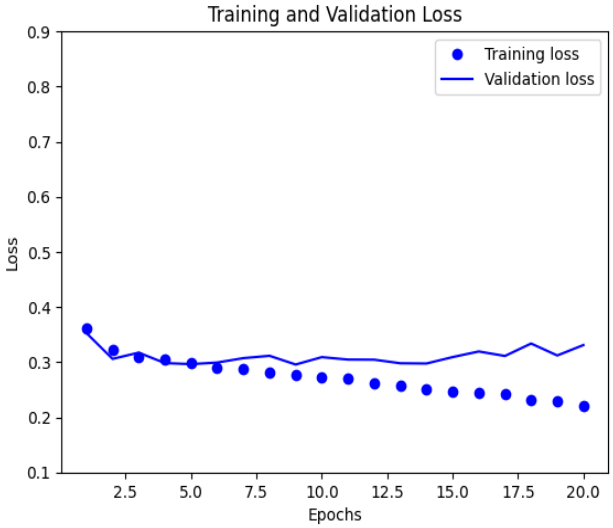

Figure 3. RNN+ Word2vec model epoch wise loss

4.5 Applying LSTM with Word2Vec

In this approach, the LSTM model was combined with Word2Vec embeddings. The Word2Vec embeddings were first generated by training a Word2Vec model on the tokenized comments from the Jigsaw Toxic Comment Dataset. Each comment was then represented as a fixed-length vector, computed by averaging the word embeddings in the comment. The data was divided into 80% training and 20% testing. Next, padding was performed to ensure uniform input lengths for the LSTM model. The model was trained for 20 epochs with a batch size of 16, achieving a 92.25% validation accuracy. Epoch-wise accuracy values with the model are shown in Figure 4. Epoch-wise loss values with the model are shown in Figure 5. The fact that LSTM with word2vec embeddings outperforms all other models and proves efficient demonstrates the potential of this combination.

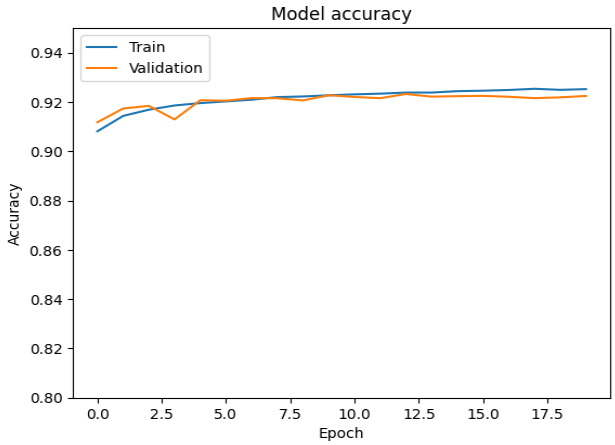

Figure 4. LSTM+ Word2vec model epoch-wise accuracy

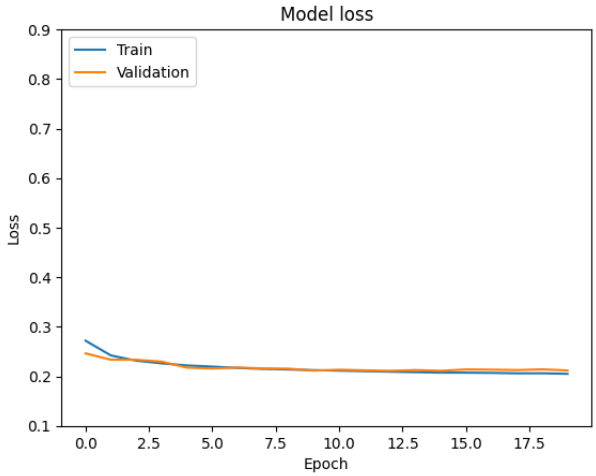

Figure 5. LSTM+ Word2vec model epoch wise loss

4.6 Applying GRU with Word2Vec

In this section, the GRU model is applied with Word2Vec embeddings for toxic comment classification. Similar to the previous methods, the Word2Vec model was trained on the tokenized comments from the Jigsaw Toxic Comment Dataset. Each comment was then transformed into a fixed-length vector using the average of word embeddings in the comment. The model was trained for 20 epochs with a batch size of 16. The GRU model achieved a validation accuracy of 92.11%, showcasing its ability to capture sequential patterns in the text and effectively classify toxic and non-toxic comments. Figure 6 and Figure 7 show epoch-wise accuracy and loss values with the model, respectively.

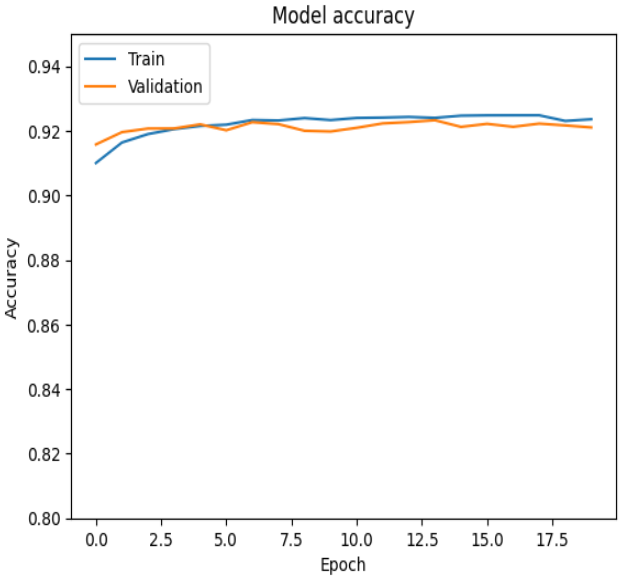

Figure 6. GRU+ Word2vec model epoch-wise accuracy

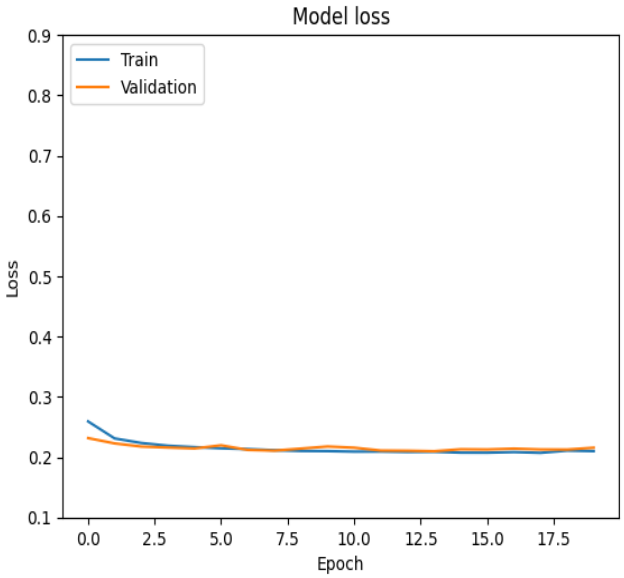

Figure 7. GRU+ Word2vec model epoch-wise loss

4.7 Applying RNN with BERT

An RNN model is combined with BERT embeddings to classify toxic comments. The dataset is preprocessed using the DistilBERT tokenizer, and embeddings are extracted from the [CLS] token. These embeddings are passed through a SimpleRNN layer and a dense layer for binary classification. After training for 10 epochs, the model achieved a validation accuracy of approximately 75.7%, demonstrating the effectiveness of RNNs with BERT embeddings for toxic comment classification. Figure 8 and Figure 9 depict epoch-wise accuracy and loss for this model.

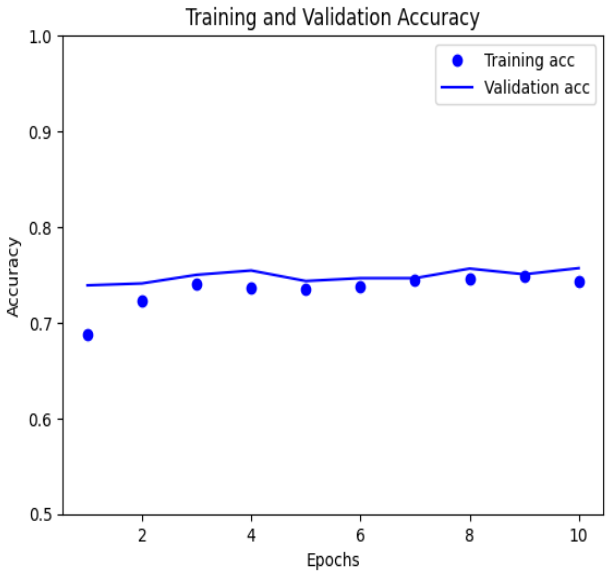

Figure 8. RNN+ BERT model epoch wise accuracy

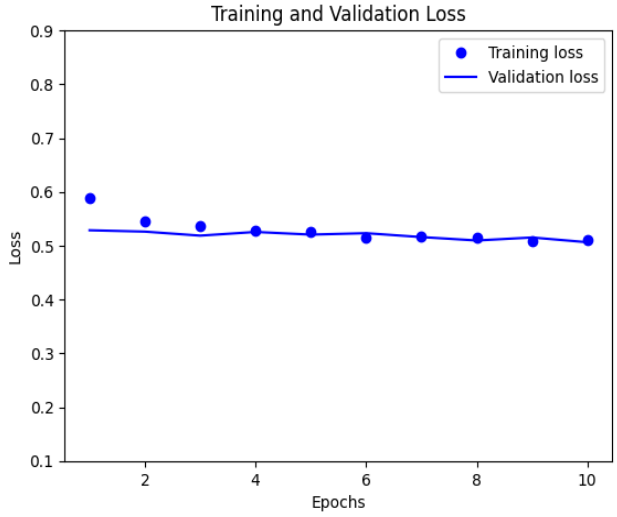

Figure 9. RNN+ BERT model epoch wise loss

4.8 Applying LSTM and GRU with BERT

In addition to the RNN-based model, the effectiveness of LSTM and GRU models combined with BERT embeddings for toxic comment classification is also evaluated. Both models leverage BERT's pre-trained embeddings, which are extracted using the DistilBERT tokenizer. The embeddings are then processed through LSTM and GRU layers, followed by dense layers for binary classification. For the LSTM model, a validation accuracy of approximately 80% was achieved, while the GRU model demonstrated a slightly lower validation accuracy of 79%.

4.9 Comparison of applied models

The performance of the proposed models was evaluated using various algorithms in combination with different feature extraction techniques, including TF-IDF, Word2Vec, and BERT. The accuracies achieved by each model are shown in Table 1 and Figure 10. As seen from the table, the combination of LSTM with Word2Vec achieved the highest accuracy of 92.25%, followed by LSTM + BERT at 92.11%. RNN-based models demonstrated competitive performance, with RNN + Word2Vec reaching an accuracy of 90.76%, and RNN + TFIDF yielding 89.46%. GRU models exhibited lower accuracies in comparison, with GRU + TFIDF achieving 75.70% and GRU + Word2Vec and GRU + BERT showing accuracies of 80.00% and 79.00%, respectively.

Table 1. Comparison of applied models

Method	Accuracy
RNN + TFIDF	89.46
LSTM + TFIDF	90.46
RNN + Word2Vec	90.76
LSTM + Word2Vec	92.25
RNN + BERT	90.76
LSTM + BERT	92.11
GRU + TFIDF	75.70
GRU + Word2Vec	80.00
GRU + BERT	79.00

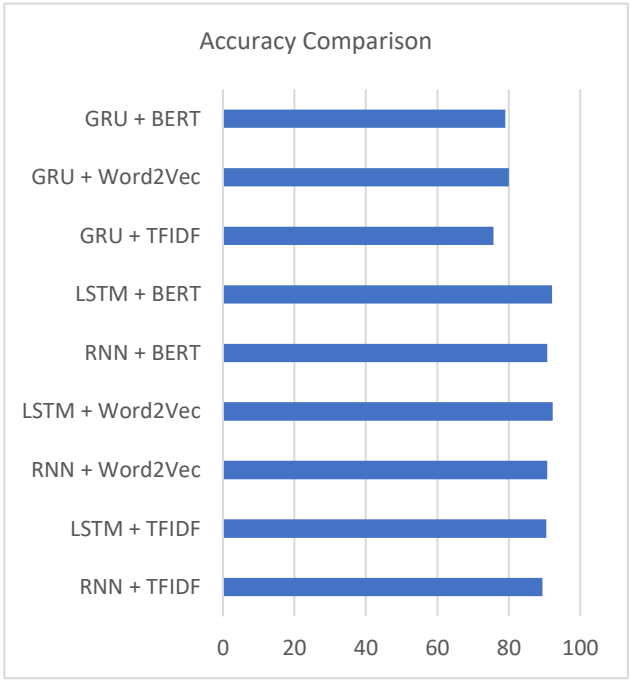

Figure 10. Accuracy comparison of applied models

5. Conclusion

In this paper, various deep learning models combined with different feature extraction techniques, such as TF-IDF, Word2Vec, and BERT, were evaluated for the task of toxic comment classification. The results demonstrated that deep learning models, particularly LSTM-based architectures, outperformed others in terms of accuracy. The highest accuracy of 92.25% was achieved with the combination of LSTM and Word2Vec, closely followed by LSTM + BERT at

92.11%. These findings indicate that LSTM networks, with their ability to capture long-term dependencies, are particularly well-suited for handling complex text data in sentiment analysis tasks. On the other hand, RNN and GRU models showed slightly lower performance but still provided competitive results, with RNN + Word2Vec achieving a 90.76% accuracy. While GRU models demonstrated lower accuracy compared to LSTM, they still hold potential for future optimization and experimentation. Future work could focus on further refining these models, experimenting with larger datasets, and exploring more advanced techniques, such as fine-tuning pre-trained models or combining multiple models, to improve classification accuracy. This paper demonstrates the potential of using advanced DL techniques and pre-trained models for effectively addressing the challenge of toxic comment classification.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically with regard to authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with policies on research ethics. The author adheres to publication requirements that the submitted work is original and has not been published elsewhere.

Data availability statement

The manuscript contains all the data. However, more data will be available upon request from the authors.

Conflict of interest

The authors declare no potential conflict of interest.

References

[1] A. Albladi, M. Islam and C. Seals, "Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review," in IEEE Access, vol. 13, pp. 30444-30468, 2025, doi: 10.1109/ACCESS.2025.3541494.

[2] Z. Hao et al., "A Novel Public Sentiment Analysis Method Based on an Isomerism Learning Model via Multiphase Processing," in IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 1, pp. 249-259, Jan. 2025, doi: 10.1109/TNNLS.2023.3274912.

[3] X. Wang, J. Lyu, B. -G. Kim, B. D. Parameshachari, K. Li and Q. Li, "Exploring Multimodal Multiscale Features for Sentiment Analysis Using Fuzzy-Deep Neural Network Learning," in IEEE Transactions on Fuzzy Systems, vol. 33, no. 1, pp. 28-42, Jan. 2025, doi: 10.1109/TFUZZ.2024.3419140.

[4] H. T. Phan, V. D. Nguyen and N. T. Nguyen, "MulGCN: MultiGraph Convolutional Network for Aspect-Level Sentiment Analysis," in IEEE Access, vol. 13, pp. 26304-26317, 2025, doi: 10.1109/ACCESS.2025.3537340.

[5] S. Ali, U. Jamil, M. Younas, B. Zafar and M. Kashif Hanif, "Optimized Identification of Sentence-Level Multiclass Events on Urdu-Language-Text Using Machine Learning Techniques," in IEEE Access, vol. 13, pp. 1-25, 2025, doi: 10.1109/ACCESS.2024.3522992.

[6] W. Gong, "Text Sentiment Classification Algorithm Based on BiLSTM Deep Learning," 2024 International Conference on Industrial IoT, Big Data and Supply Chain (IIoTBDSC), Wuhan, China, 2024, pp. 83-87, doi: 10.1109/IIoTBDSC64371.2024.00025.

[7] S. A. Mostafa, W. S. Al-Dayyeni, A. N. Kareem, M. A. Jubair, M. M. Jaber and B. A. Khalaf, "Classification and Sentiment Analysis of Amazon Alexa Reviews," 2024 1st International Conference on Logistics (ICL), Jeddah, Saudi Arabia, 2024, pp. 1-5, doi: 10.1109/ICL62932.2024.10788570.

[8] S. Mehta and A. Bhalla, "Enhanced Sentiment Classification with Federated Learning CNNs: Exploring Five Sentiment Categories," 2024 3rd International Conference for Advancement in Technology (ICONAT), GOA, India, 2024, pp. 1-5, doi: 10.1109/ICONAT61936.2024.10774751.

[9] X. He, "Sentiment Classification of Social Media User Comments Using SVM Models," 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 2024, pp. 1755-1759, doi: 10.1109/AINIT61980.2024.10581547.

[10] M. Aamir, L. J and Sweety, "A Comparative Study of ML and DL Approaches for Twitter Sentiment Classification," 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), Greater Noida, India, 2024, pp. 1-5, doi: 10.1109/ICEECT61758.2024.10738938.

[11] Q. Zeng, "Design of Intelligent Sentiment Classification Model Based on Deep Neural Network Algorithm in Social Media," in IEEE Access, vol. 12, pp. 81047-81056, 2024, doi: 10.1109/ACCESS.2024.3409818.

[12] M. Khalid et al., "Novel Sentiment Majority Voting Classifier and Transfer Learning-Based Feature Engineering for Sentiment Analysis of Deepfake Tweets," in IEEE Access, vol. 12, pp. 67117-67129, 2024, doi: 10.1109/ACCESS.2024.3398582.

[13] A. Lakshmanarao, C. Gupta and T. S. R. Kiran, "Airline Twitter Sentiment Classification using Deep Learning Fusion," 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2022, pp. 1-4, doi: 10.1109/SMARTGENCON56628.2022.10084207.

[14] Y. Matrane, F. Benabbou and Z. Ellaky, "Enhancing Moroccan Dialect Sentiment Analysis Through Optimized Preprocessing and Transfer Learning Techniques," in IEEE Access, vol. 12, pp. 187756-187777, 2024, doi: 10.1109/ACCESS.2024.3514934.

[15] H. Shuqin and R. C. Raga, "A Deep Learning Model for Student Sentiment Analysis on Course Reviews," in IEEE Access, vol. 12, pp. 136747-136758, 2024, doi: 10.1109/ACCESS.2024.3463793.

[16] A. He and M. Abisado, "Text Sentiment Analysis of Douban Film Short Comments Based on BERT-CNN-BiLSTM-Att Model," in IEEE Access, vol. 12, pp. 45229-45237, 2024, doi: 10.1109/ACCESS.2024.3381515.

[17] Z. Wang, G. Xu, X. Zhou, J. Y. Kim, H. Zhu and L. Deng, "Deep Tensor Evidence Fusion Network for Sentiment Classification," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 4605-4613, Aug. 2024, doi: 10.1109/TCSS.2022.3197994.

[18] S. K. Putri, A. Amalia and T. F. Abidin, "Sentiment Analysis Multi-Label of Toxic Comments using BERT-BiLSTM Methods," 2024 International Conference on

Electrical Engineering and Informatics (ICELTICs), Banda Aceh, Indonesia, 2024, pp. 120-124, doi: 10.1109/ICELTICs62730.2024.10776338.

[19] A. Lakshmanarao, A. Srisaila and T. S. R. Kiran, "Twitter Sentiment Classification with Deep Learning LSTM for Airline Tweets," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 520-524, doi: 10.1109/ICACCS54159.2022.9785208.

[20] S. Dutta, M. Neog and N. Baruah, "Assamese Toxic Comment Detection On Social Media Using Machine Learning Methods," 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India, 2024, pp. 1-8, doi: 10.1109/ic-ETITE58242.2024.10493331.

[21] Y. Mamani-Coaquira and E. Villanueva, "A Review on Text Sentiment Analysis With Machine Learning and Deep Learning Techniques," in IEEE Access, vol. 12, pp. 193115-193130, 2024, doi: 10.1109/ACCESS.2024.3513321.

[22] Rahul, H. Kajla, J. Hooda and G. Saini, "Classification of Online Toxic Comments Using Machine Learning Algorithms," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 1119-1123, doi: 10.1109/ICICCS48265.2020.9120939.

[23] M. Aquino et al., "Toxic Comment Detection: Analyzing the Combination of Text and Emojis," 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 2021, pp. 661-662, doi: 10.1109/MASS52906.2021.00097.

[24] T. V. Sai Krishna, T. S. Rama Krishna, S. Kalime, C. V. Murali Krishna, S. Neelima, and R. R. PBV, "A novel ensemble approach for Twitter sentiment classification with ML and LSTM algorithms for real-time tweets analysis," Indonesian Journal of Electrical Engineering and Computer Science, vol. 34, no. 3. Institute of Advanced Engineering and Science, p. 1904, Jun. 01, 2024. doi: 10.11591/ijeecs.v34.i3.pp1904-1914.

[25] N. K. Singh and S. Chand, "Machine Learning-based Multilabel Toxic Comment Classification," 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 435-439, doi: 10.1109/ICCCIS56430.2022.10037626.

[26] N. L. V. Venugopal, P. Kanchanamala, S. Muppidi, T. B. Prakash, T. Neelima and S. A. Devi, "Multilingual Toxic Comment Classification using Deep Learning," 2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2024, pp. 752-757, doi: 10.1109/ICSSAS64001.2024.10760913.

[27] N. Boudjani, Y. Haralambous and I. Lyubareva, "Toxic Comment Classification For French Online Comments," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2020, pp. 1010-1014, doi: 10.1109/ICMLA51294.2020.00164.

[28] A. Jessica, M. S. Sugiarto, Jerry, S. Achmad and R. Sutoyo, "A Hybrid Deep Learning Techniques Using BERT and CNN for Toxic Comments Classification," 2024 International Conference on Information Management and Technology (ICIMTech), Bali, Indonesia, 2024, pp. 393-398, doi: 10.1109/ICIMTech63123.2024.10780934.

[29] https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/data