International Journal of Basic and Applied Sciences, 14 (4) (2025) 499-507

SPC

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/ty9xvz92
Research paper

Advanced Toxic Comment Classification Using Multi-Architecture Generative AI Techniques

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

[1] Research Scholar, Department of CSE, Centurion University of Technology and Management, Bhubaneswar, Odisha and Assistant Professor, Aditya University, Surampalem, India
[2] Associate Professor, Department of CSE, Centurion University of Technology and Management, Bhubaneswar, Odisha, India
[3] Department of Computer Applications, Aditya University, Surampalem, India
**Corresponding author E-mail: sushma.cse2@gmail.com*

Received: July 5, 2025, Accepted: August 9, 2025, Published: August 17, 2025

Abstract

The proliferation of user-generated content on online platforms has led to a significant rise in toxic and harmful comments, necessitating the development of robust and scalable detection systems. In this study, a comprehensive methodology is proposed for toxic comment classification using the Jigsaw Toxic Comment dataset. Initially, baseline models were implemented to establish reference performance levels. A Logistic Regression model combined with TF-IDF feature extraction achieved an accuracy of 93.00%, while a shallow single-layer neural network reached an accuracy of 94.00%. Building upon these baselines, a novel Generative AI (GenAI) driven approach was employed, integrating four distinct stages: synthetic data generation using GPT-2, fine-tuning GPT-2 for supervised classification, lightweight classification using DistilGPT-2, and text-to-text classification using T5-small. Synthetic toxic and non-toxic comments were generated using GPT-2, enriching the training data and enhancing model generalization. Subsequently, GPT2ForSequenceClassification was fine-tuned both with and without class imbalance adjustments, achieving an accuracy of 96.22% and a toxic F1-score of 97.90%. DistilGPT-2 was then fine-tuned to provide a lightweight alternative, achieving slightly lower but competitive performance with an accuracy of approximately 96.00% and a toxic F1-score of 97.50%. Further, a T5-small model was fine-tuned by reframing toxic comment classification as a text-to-text task, achieving the best results with approximately 97.00% accuracy and a toxic F1-score of 98.10%. The results demonstrate that combining data augmentation with multi-architecture Generative AI fine-tuning significantly improves toxic comment detection performance, outperforming traditional machine learning and shallow neural network baselines. This work highlights the effectiveness of leveraging generative models for both data enhancement and supervised learning, offering a comprehensive and scalable solution for mitigating toxic behavior online.

Keywords: *Toxic Comment Classification; Generative AI; GPT-2; Text-to-Text Classification; DistilGPT-2; Synthetic Data Augmentation.*

1. Introduction

The explosion of social media, web forums, and user-contributed platforms has generated user-generated content at an unprecedented scale. Though social media services enable communication and community, they can also complicate matters when users encounter toxic, offensive, and harmful comments that have a huge potential to do real harm to the internet. Dealing with the spreading of toxicity on the web is now a highly topical issue that requires advanced but reasonable computational and phenomenon abstraction capabilities.

Conventional ML methods have been widely used for text classification tasks like toxic comment detection. They are mostly based on handcrafted features like TF-IDF vectors or Word2Vec embeddings utilized for representing textual data numerically. While useful, they have difficulty dealing with complex semantic and contextual subtleties found in human language, especially in the presence of noisy and unstructured online commentaries.

The emergence of DL models, such as RNNs, LSTMs, Gated Recurrent Units (GRUs), and architectures such as BERT, which are based on Transformers, has greatly improved the performance of text classification systems. Deep learning models are good at capturing sequential dependencies and intricate contextual relationships among words, which makes for better performance for toxicity detection. Nevertheless, these deep learning methods still heavily depend on massive amounts of labeled data of good quality, and may suffer the drawback of failing to generalize well to unseen or adversarial comment patterns.

To bridge the gap further, Generative AI (GenAI), driven by large pretrained language models such as GPT-2, DistilGPT-2, and T5, introduces a transformative paradigm for enhancing both data diversity and model learning. Beyond basic classification tasks, GenAI models possess the ability to generate synthetic human-like text, augment training datasets, and facilitate more robust downstream task-specific fine-tuning. Leveraging these capabilities, this study proposes a comprehensive baseline-to-GenAI pipeline for toxic comment classification.

Initially, two simple baseline models are established to set reference performance benchmarks: (i) a Logistic Regression model utilizing TF-IDF feature extraction, and (ii) a shallow single-layer neural network. These baselines provide comparative insights before applying more sophisticated generative models. Building upon these baselines, the proposed GenAI-based methodology integrates four key stages: synthetic data generation using GPT-2, fine-tuning GPT-2 for binary toxic/non-toxic classification, fine-tuning DistilGPT-2 as a lightweight classification alternative, and fine-tuning a T5-small model to reframe classification as a text-to-text generation task.

This hybrid strategy enables better data diversity, improved handling of class imbalance, and stronger performance generalization across diverse toxicity patterns. Experimental evaluation demonstrates that the GenAI-enhanced approach significantly outperforms traditional baseline models, achieving high accuracy and F1-scores on the Jigsaw Toxic Comment dataset, and affirming the advantages of using generative models for toxicity detection tasks.

The main contributions of this paper are:

- Establishment of baseline performances using Logistic Regression with TF-IDF and a shallow single-layer neural network.
- Synthetic toxic and non-toxic comment generation using GPT-2 to augment the training data.
- Fine-tuning of GPT2 sequence classification with and without class balancing techniques.
- Fine-tuning of DistilGPT-2 for lightweight and faster toxic comment classification.
- Implementation of a T5-small based text-to-text toxic comment classifier.
- Comparative evaluation demonstrating the benefits of combining data augmentation, baseline benchmarking, and multi-architecture GenAI fine-tuning for toxic comment detection.

2. Related work

Mao et al. [1] explored the landscape of sentiment analysis, providing insights into its methods, implementations, and key obstacles. They analyzed the goals of sentiment analysis tasks, compared different techniques, investigated application domains, and discussed the challenges and limitations faced by researchers. Their study underlined the importance of AI technologies in automating text sentiment analysis and highlighted the importance of sentiment analysis in various aspects of production and daily life. The authors also proposed suggestions for possible solutions to existing challenges and explored future research directions, contributing valuable insights to the existing body of knowledge in sentiment analysis and offering practical references for scholars and practitioners. Sushma et al.[2] have given an overview of sentiment analysis, whereby they have explained the transition of sentiment analysis from the conventional lexicon-based and ML methods to the recent DL based models like LSTM, RNN, transformer-based models such as BERT. They also address several challenges in the area, such as sarcasm detection, handling ambiguity, and multilingual sentiment analysis. They also studied uses of sentiment analysis for social media, e-commerce, healthcare, and financial markets. New trends such as multi-modal sentiment analysis were also discussed, and new research issues were identified in terms of context-awareness, model interpretability, and real-time adaptation.

Punetha et al. [3] introduced a novel unsupervised method for sentiment classification to address the shortcomings of traditional ML models that rely on huge pre-training data. They based their approach on game theory formulations (a population game model in particular), which led to (partially) unsupervised sentiment classification. They also obtained two important textual features: context score and emotion score, from review comment by a lexicon-based approach. They developed a language-independent cognitive mathematical framework and showed its competitive performance in hotels, restaurants, and electronic devices domains to verify its suitability for English and Hindi. They relied on unsupervised methods and did not explore transformer-based architectures, limiting their ability to capture deep contextual semantics.

Semary et al. [4] emphasized feature extraction as a key to improving sentiment classification performance. Claudio and Renato_77 and Deng et al._25 thoroughly examined and briefly described some feature representations used in ML such as Bag-of-Words (BOW), Word2Vec, N-gram, TFIDF(HV), TF-IDF, (HV), and GloVe embeddings. They tested the proposed methods on the Twitter US airlines data and the Amazon musical instrument review data, where they learnt a random forest classifier for comparison. Their focus was on handcrafted features, which may be insufficient for complex toxic language patterns found in social media.

Jency Jose et al. [5] presented a unified method for sentiment analysis and lexico-syntactic pattern matching in the context of social media, based upon enhanced natural language processing (NLP) techniques. They used LSTM networks for sentiment and TextRazor to extract topics from the discussion. The system focused on improving users' experience by modeling their emotional tones and key themes with their intuitions to help users identify sentiment trends and topic distributions. While effective in topic extraction and sentiment detection, their model did not address multi-architecture design or synthetic data enhancement. Y. Cai et al. [6] tackled the challenges of multimodal sentiment analysis (MSA) by introducing an advanced framework that increases the emotional feature extraction and multimodal fusion process. To enhance the representation ability of each modality and enhance the correlation and interaction between modalities, they proposed Feature Extraction and Fusion Network , respectively. Their model incorporated multilayer feature extraction, attention mechanisms, and Transformer architecture, in order to dig on deeper into relationships among the features. Although powerful in multimodal settings, their model did not evaluate performance on toxicity classification specifically.

Thomas et al. [7] performed sentiment analysis of the reviews of the Telegram app with ML models. They experimented with a dataset of user reviews and compared the performance of the models in terms of precision, recall, F1-score , and accuracy for the three sentiment classes: Negative, Neutral, and Positive. Their results suggested that of the four models used, Random Forest had the best classification accuracy; however, Neutral sentiment was still difficult to classify accurately. They have also stated that preprocessing, such as tokenization and lemmatization, and feature extraction, is important for the performance of models. Their study focused on sentiment polarity and lacked consideration of toxicity-specific linguistic markers or class imbalance issues. Alsalem et al. [8] developed a sentiment analysis model directed towards improving the interpretation of the sentiments in Arabic language texts on social media platforms. They introduced a method focusing on target-oriented sentiment classification with an LLM. Their model was learned based on the AT-OTDSA dataset, manually labeled Arabic tweets with certain topics and sentiment. Their method of fine-tuning the Arabic-MARBERT-sentiment model proved to improve the performance of sentiment classification for Arabic tweets, and hence fills an important gap in previous Arabic sentiment analysis work. Their work was limited to Arabic texts and did not investigate cross-lingual generalization or synthetic data strategies.

X. Fan et al. [9] presented a multi-task learning based fine-grained sentiment analysis model, named BLAB, for joint aspect word extraction and sentiment orientation prediction. Their model exploited the AD-BiReGU module with the BERT-LCF framework, which could effectively record local and global contextual information. The model is based on local context focus, multi-head attention, and bidirectional gated RNN that dynamically masks out the irrelevant context and captures the dependencies between features of texts more effectively. Comparative studies showed that the AD-BiReGU model can effectively improve the aspect-based sentiment analysis performance under multi-task learning. X. He et al. [10] compared the performance of several ML classifiers in sentiment analysis. They also analyzed the comments about the product 'Huawei Mate60' in the Little Red Book Platform and adopted precision, recall, and F1-score to evaluate the models. Their results indicated that SVM yielded higher classification performance than that of other classifiers in building a sentiment classification model. They suggested that automatic sentiment analysis would be of interest for brands to gather consumer response data, but also in the product development process surveillance, in marketing, in brand management.

Nelatoori et al. [11] presented a multitask model to co-train toxic comment classification. They used ToxicXLMR embedding to model the bidirectional context and added a Bi-LSTM CRF layer for toxic span/rational prediction. Their method showed a gain against single-task models for both classification and rationale extraction. They also considered the domain adaptation ability of their model on OOD datasets (e.g., HASOC and OLID), and obtained better performance than single-task baselines. However, their approach did not explore synthetic data augmentation or multi-model strategies, which limits generalization to unseen patterns. B. R. Naidu et al. [12] proposed a deep neural network model for toxicity detection in online discussions. They used LSTM to classify comments and distinguish various types of toxicity. They analyzed comments on online platforms, looking at toxic and hateful comments, which might lead to the publication of web interfaces for automatic toxicity classification. Their work focused on a single deep learning model and did not address class imbalance or performance scalability using generative models. N. Reddy et al. [13] concentrated on the detection of conversational toxicity based on NLP methods. They wanted to create a system that would warn people before they sent something potentially harmful, and make the internet a more positive place. Their contribution used NLP technologies (e.g., understanding, analyzing, and manipulating human language data) to identify toxic patterns in a text. They described NLP as a subfield of artificial intelligence that permits machines to accomplish sentiment analysis that assists with early detection and defense against toxic communications.

A. Vinod et al. [14] developed a regional social network with the incorporation of a ML for toxic comment recognition. Their system helped users to communicate with others by automatically detecting and eliminating abusive comments, and there is the utilization of an email alert to help users and create a safer environment for communication. H. Fan et al. [15] proposed a toxicity detection model using BERT and its extensions, fine-tuned Toxic Comment dataset. They've applied these models to Twitter data regarding UK Brexit discussions and proved the success of BERT architectures for the detection and categorization of toxic user-generated content. Giustino et al. [16] investigated several RNN-based architectures for multi-label toxic comment classification. Sample efficiency and generalisation were tested using the Jigsaw Toxicity Challenge and League of Legends tribunal datasets, investigating the influence of various preprocessing techniques on classification. They observed that BiLSTM models performed the best in the detection of toxic comments. They evaluated multiple RNN variants but did not integrate generative approaches or transfer learning to boost data diversity. Z. Zhao et al. [17] proposed a debiasing method for toxic comment classification that also leverages the subjectivity of identity term pierced comments. They employed two approaches to measure subjectivity based on the lexicon-focused approach and embedding similarity with the related Wikipedia text. Their BERT-based models consistently outperformed strong baselines on different datasets, showing that subjectivity-aware modeling is an effective approach to reduce bias in toxic comment detection. Although they addressed bias via subjectivity modeling, their method did not include data augmentation or transformer ensembles.

A. Bonetti et al. [18] proposed an automated detection methodology of toxic posts in social media and reduce the severe spreading of harmful content. They contrasted classical ML classifiers with topic modeling techniques and DL methods, based on the BERTweet transformer model. They found that BERTweet slightly improved results, but traditional ML was competitive and demonstrated the trade-off between performance and complexity. A. Abbasi et al. [19] addressed the problem of toxic language detection in online communication by considering bias originating from the training data of a ML model. They studied and benchmarked state-of-the-art DL methods with recent literature on multilabel toxic comment classification, focusing on religious and ethnic toxicity. Their experiment compared several embedding methods, i.e., GloVe, Word2Vec, FastText, with the models using default embedding layers. Khan et al. suggested from their experiments that for multilabel toxic classification, CNN models provide superior performance for various scenarios. Their focus was on embedding comparisons rather than architecture-level enhancements or synthetic data generation. A. Shinde et al. [20] did a comparative analysis of toxic comment classification using various DL models. They trained and tested their models on a publicly available dataset of toxic comments, and accuracy and ROC-AUC as the evaluation metrics. The results showed that BiLSTM-CNN performed significantly well in identifying toxic comments, indicating the significance of a union of sequence modeling ability and convolutional analysis in enhancing the predictive value of toxic observation tasks. Their CNN-BiLSTM model did not utilize external generative models or explore text-to-text reformulations for classification.

G. Xie et al. [21] proposed an ensemble model for cross-lingual toxicity detection on the Jigsaw Multilingual Toxic Comment Classification dataset. Classification was enhanced by subsampling, pseudo-labeling, translating, and post-processing, highlighting the value of multilingual models for toxic comment detection. M. Shahid et al. [22] introduced a transformer-based model to identify toxic comments in Urdu, specifically for handling cursive and short texts. Their finetuned BERT model has outperformed previous approaches and supported better content moderation for low-resource languages. S. Tsai et al. [23] presented a sentiment classification model based on FastText and multilevel DPCNN to deal with Chinese texts. Their technique facilitated the feature extraction and improved the classification performance, compared with conventional ways. D. A. Kristiyanti et al. [24] used the SVM to distinguish positive and negative sentiments of Twitter source data about public transportation services. In their research, the authors have proved that SVM performs well in sentiment classification. Y. Yang et al. [25] proposed a sentiment and syntactic-aware GCN to perform aspect-level sentiment classification. They developed a sparse sentiment-specific graph and modified the structure of GCN to improve feature extraction, outperforming the existing methods. Y. Rong et al. [26] developed a sentiment classification model for microblog texts using DeBERTa combined with BiGRU and an attention mechanism. Their approach improved the representation of contextual information, leading to better sentiment classification performance.

S. Sushma et al. [27] performed toxic comment classification using RNN, LSTM, and GRU with TF-IDF, Word2Vec, and BERT embeddings. LSTM with BERT and Word2Vec gave the best results, showing the strength of combining traditional models with transformer-based embeddings. A. Lakshmanarao et al. [28] applied RNN, LSTM, Bi-LSTM, and GRU for sentiment classification on airline tweets. The model achieved up to 93% accuracy, proving its effectiveness in distinguishing between positive, negative, and neutral sentiments.

Based on the review of prior studies, it is evident that many traditional and deep learning approaches, including those by Nelatoori et al. [11] and Naidu et al. [12], focus on single-task or single-model strategies. Inspired by the limitations observed in these methods—such as handling class imbalance, lack of data diversity, or reliance on fixed embeddings—our study adopts a multi-architecture Generative AI pipeline that integrates synthetic data generation and fine-tuning across multiple transformer-based models for more robust toxicity classification.

3. Research Methodology

Figure 1 shows the proposed method for toxic comment classification. The Jigsaw Toxic Comment Classification Challenge dataset, consisting of over 223,000 comments from various online platforms, is utilized for this study. First, the dataset is checked for missing values and cleaned to ensure consistency. Text preprocessing is performed by converting comments to lowercase, removing unnecessary whitespace, and lightly cleaning special characters. For traditional models, additional tokenization and TF-IDF feature extraction are applied, while for Generative AI models, raw text is directly tokenized using pretrained tokenizers.

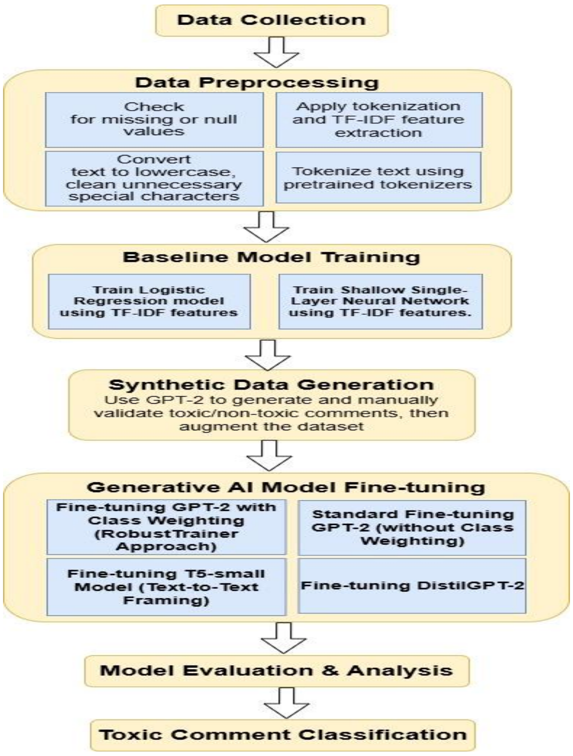
Data Collection
Data Preprocessing
Check for missing or null values
Apply tokenization and TF-IDF feature extraction
Convert text to lowercase, clean unnecessary special characters
Tokenize text using pretrained tokenizers
Baseline Model Training
Train Logistic Regression model using TF-IDF features
Train Shallow Single-Layer Neural Network using TF-IDF features.
Synthetic Data Generation
Use GPT-2 to generate and manually validate toxic/non-toxic comments, then augment the dataset
Generative AI Model Fine-tuning
Fine-tuning GPT-2 with Class Weighting (RobustTrainer Approach)
Standard Fine-tuning GPT-2 (without Class Weighting)
Fine-tuning T5-small Model (Text-to-Text Framing)
Fine-tuning DistilGPT-2
Model Evaluation & Analysis
Toxic Comment Classification

Fig. 1: Proposed Approach for Toxic Comment Classification.

To establish baseline performance, two traditional models are implemented: a Logistic Regression model trained on TF-IDF features and a shallow single-layer neural network. These baselines provide comparative reference points for the classification task. Following this, synthetic data generation is conducted using GPT-2 through prompt engineering, generating toxic and non-toxic comments that are manually validated and appended to the training set. This data augmentation enriches linguistic diversity and helps the models generalize better. The next phase involves fine-tuning multiple Generative AI models. First, GPT-2 is fine-tuned with a RobustTrainer approach incorporating class weighting to address class imbalance, followed by standard fine-tuning without class weighting to observe natural model adaptation. Additionally, DistilGPT-2, a lightweight distilled version of GPT-2, is fine-tuned to evaluate the trade-off between model efficiency and classification performance. The T5-small model is fine-tuned by reframing the classification task as a text-to-text generation problem, where each comment is prefixed with an instruction and the model predicts the toxicity label as a generated output. All models are evaluated on the validation set to assess performance.

The proposed approach is verified by comparing the baseline models and Generative AI models across metrics such as accuracy, toxic F1-score, and non-toxic F1-score. The comparative evaluation demonstrates the superior ability of Generative AI models, especially encoder-decoder architectures like T5-small, in effectively detecting toxic language patterns. The methodology concludes with an analysis of performance trade-offs between model complexity, training strategies, and computational efficiency, highlighting the potential of GenAI-driven solutions for scalable toxic comment classification.

4. Results and discussions

4.1. Collection of the dataset

The dataset employed in this study is the Jigsaw Toxic Comment Classification Challenge dataset, sourced from Kaggle[27]. It comprises a collection of English-language comments collected from various online platforms, including Wikipedia talk pages and Civil Comments. Each comment is originally annotated across six toxicity-related labels: toxic, severe toxic, obscene, threat, insult, and identity hate.

For this research, a binary classification target was constructed by merging all six labels into a single indicator. If a comment was marked under any toxicity category, it was labeled as toxic (1); otherwise, it was labeled as non-toxic (0). The complete dataset contains 223,549 comments. After necessary preprocessing, the dataset was partitioned into training and validation sets in an 80:20 ratio, resulting in 178,839 samples for training and 44,710 samples for validation.

4.2. Preprocessing

Before model training, several preprocessing steps were applied to ensure consistency and quality of the data. The dataset was first inspected for missing or null entries in the critical fields comment_text and toxic. Any samples with incomplete information were removed to maintain the integrity of the dataset.

Subsequently, the text data was standardized by converting all comments to lowercase, ensuring uniformity and minimizing case-sensitive variations. Basic text cleaning operations were performed, including the removal of leading and trailing whitespace and the cleaning of extraneous special characters.

For the baseline models utilizing TF-IDF features (Logistic Regression and shallow neural network), additional tokenization and minimal stopword removal were applied to optimize feature extraction. However, no aggressive tokenization or punctuation removal was conducted for the Generative AI models, as they are inherently capable of processing raw text inputs effectively.

Following these preprocessing steps, the prepared dataset was ready for feature extraction and model training, with the class distribution preserved in the training and validation splits.

After preprocessing, the prepared textual data was subjected to feature extraction and input formatting suitable for the models employed in this study. For the baseline models, TF-IDF feature extraction was performed to convert textual comments into numerical feature vectors suitable for Logistic Regression and shallow neural network classifiers.

While traditional methods typically use manually engineered features like TF-IDF or Word2Vec, the Generative AI models utilized here, namely GPT-2, DistilGPT-2, and T5-small, are capable of directly processing raw text inputs through efficient tokenization mechanisms.

For the GPT-2-based models, including both GPT-2 and DistilGPT-2, the comment_text field was tokenized using their respective pre-trained tokenizers. Each comment was converted into a sequence of input token IDs and corresponding attention masks. To optimize computational efficiency without losing essential contextual information, the maximum input token length was capped at 128 tokens. Labels corresponding to toxic (1) and non-toxic (0) classes were assigned alongside the tokenized inputs, preparing the data for supervised fine-tuning.

During synthetic data generation using GPT2LMHeadModel, prompt engineering was employed. Predefined prompts such as "Generate a toxic insult:" and "Generate a polite comment:" were fed into the model to conditionally generate toxic and non-toxic comments, respectively. These generated samples were manually validated based on heuristic rules and appended to the original training set to enrich the linguistic diversity and sample balance within the dataset.

For the T5-small model, the classification task was reframed as a text-to-text generation problem. Each input comment was prefixed with a guiding instruction, for example, "classify toxic comment: <comment_text>", and the corresponding output labels were represented textually as "0" (non-toxic) or "1" (toxic). Both input prompts and target labels were tokenized using the T5 tokenizer, transforming them into input IDs and label IDs compatible with the encoder-decoder architecture of the model.

Through these preparations, the dataset was systematically transformed into formats optimized for the various baseline and Generative AI models employed, setting a strong foundation for effective model training and robust toxic comment classification.

4.3. Baseline models for toxic comment classification

Before proceeding with advanced Generative AI models, two simple baseline models were created to serve as a performance benchmark. The first baseline utilized a Logistic Regression classifier trained on TF-IDF feature representations of the input comments. The TF-IDF approach transformed the raw text into numerical vectors, capturing the importance of words relative to the entire corpus while ignoring syntactic structures. The Logistic Regression model offered an easy and human-interpretable approach for binary classifications into toxic and non-toxic classes.

The second baseline was a simple single-layer neural network with a shallow architecture, i.e., a single fully connected hidden layer with ReLU activation and a softmax output layer. The same TF-IDF features acted as input to the network. This baseline was based on a simple exploration of DL and did not employ complex high-capacity transformer architectures.

Both baseline models were trained using the pre-processed and vectorized dataset, with hyperparameters tuned for reasonable performance. These baselines served as comparative benchmarks to highlight the improvements achieved through the application of Generative AI-based techniques described in the subsequent sections.

The evaluation results for the baseline models are shown in Table 1 and Figure 2. As shown, the Logistic Regression model achieved an accuracy of 93.00%, with a toxic F1-score of 92.50% and a non-toxic F1-score of 80.20%. The shallow single-layer neural network achieved a slightly higher accuracy of 94.00%, with a toxic F1-score of 93.40% and a non-toxic F1-score of 81.00%. Although both baselines provided reasonable starting points, the results were notably lower compared to the Generative AI models evaluated later. This demonstrated the potential limitations of traditional and shallow DL models in capturing the complex patterns present in toxic online comments, and justified the need for more advanced GenAI-based approaches.

Table 1: Baseline Model Results

Model	Accuracy (%)	Toxic F1-score	Non-Toxic F1-score
Logistic Regression (TF-IDF)	93%	92.5%	80.2%
Neural Network (TF-IDF)	94%	93.4%	81%

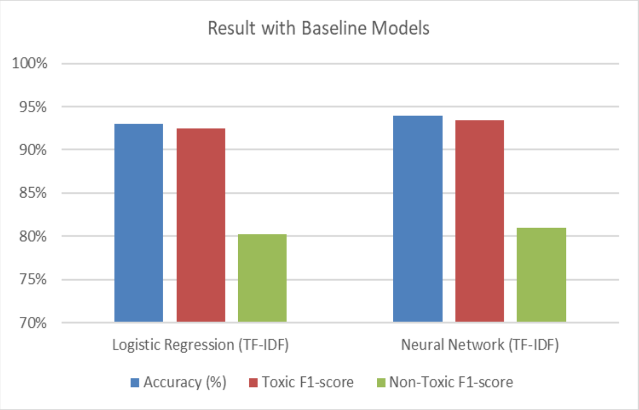

Fig. 2: Performance of baseline models.

4.4. Model training and results

In this section, various Generative AI models were trained and evaluated for the task of toxic comment classification. The methodology followed a structured multi-step methodology. Initially, synthetic data augmentation was performed using GPT-2 to enrich the dataset. Subsequently, supervised fine-tuning of GPT-2 models was carried out with and without class balancing techniques to investigate the impact of training strategies on classification performance. Furthermore, an encoder-decoder architecture, T5-small, was fine-tuned by reframing the classification task into a text-to-text generation problem. Finally, a lightweight alternative, DistilGPT-2, was fine-tuned to assess the trade-off between performance and resource efficiency. Each of these training stages is described in detail in the following subsections.

4.4.1. Synthetic data generation using GPT-2

In the first stage of model training, synthetic data generation was employed to enhance the volume and linguistic diversity of the training dataset. A Generative Pretrained Transformer-2 (GPT-2) model (GPT2LMHeadModel) was fine-tuned specifically for generating toxic and non-toxic comments. To achieve controlled generation, prompt engineering techniques were utilized, where carefully designed prompts such as "Generate a toxic insult:" and "Generate a polite comment:" were provided as inputs to guide the model's output behavior.

The model generated multiple samples for each prompt, and these outputs were subjected to manual validation. These were manually annotated based on the initial 20 comments by considering only high-quality and context-appropriate comments to guarantee the trustworthiness of the augmented dataset. We validated 20 synthetic samples (10 negative and 10 positive comments) by this procedure.

The synthetic data was provided to enhance the diversity of facial expression types in the data set, to expose the model to different linguistic style variants, and serve as a countermeasure for potential overfitting on the original comments. This attribute augmentation greatly contributed to the building of a more solid base for the subsequent supervised fine-tuning stages.

4.4.2. Handling class imbalance with robusttrainer

Due to the class imbalance problem of the toxic comment dataset, we utilized a special training method adopting the modified version of the trainer that we named RobustTrainer. In this study, class weights were used during training to increase the importance of the minority toxic class. Particularly, a ratio of weight [1.0, 3.0] was used to weight toxic samples triple to that of non-toxic samples. This modification was added to down-weight and pull the loss contributions toward 0 to encourage the model to pay more attention to the task of finding toxic content, which is minuscule in proportion when comparing against real real-world dataset.

The GPT2ForSequenceClassification model was fine-tuned using this weighted training scheme. A batch size of 8, a learning rate of 2e-5, and two training epochs were utilized during the fine-tuning process. The training was conducted on the augmented dataset, which included both original and synthetically generated comments.

Upon evaluation on the validation set, the model achieved an accuracy of 96.04%, a toxic F1-score of 97.79%, and a non-toxic F1-score of 80.66%. This experiment demonstrated that applying class-weighted loss functions helped improve the model's sensitivity toward toxic comments. However, the overall performance indicated that there was still room for improvement, leading to the exploration of standard fine-tuning strategies in the subsequent phase.

4.4.3. Standard fine-tuning using HuggingFace Trainer

In the next phase, a standard fine-tuning strategy was adopted to further enhance the model's performance. Unlike the previous approach, which introduced class weights to compensate for class imbalance, the standard fine-tuning process relied solely on the inherent learning capability of the model without any manual adjustment to the loss function. Fine-tuning large Generative AI models such as GPT-2 for downstream classification tasks has been shown to effectively leverage their deep contextual understanding developed during pretraining. Therefore, it was essential to evaluate the model's behavior when trained purely on real and synthetic comment data, allowing the model to learn the task-specific patterns naturally.

The fine-tuning was conducted using the Huggingface Trainer framework, which enabled efficient handling of large transformer models. A batch size of 8, learning rate of 2e-5, and two training epochs were maintained to ensure comparability with the earlier weighted training setup. The training dataset consisted of both original and synthetically generated comments, ensuring that the model was exposed to a wide range of toxicity patterns during optimization.

Upon evaluation on the validation set, the model achieved an accuracy of 96.22%, a toxic F1-score of 97.90%, and a non-toxic F1-score of 81.21%. These results indicated a clear improvement over the class-weighted training approach, particularly in terms of detecting toxic comments. The model demonstrated its ability to generalize across varying comment types without the need for manual bias adjustment. The slight performance enhancement affirmed that GPT-2, when fine-tuned carefully, can be effectively adapted to specialized classification tasks such as toxic comment detection, beyond its original language generation capabilities. Consequently, the model trained through standard fine-tuning was selected as the primary GPT-2-based toxic comment classifier for subsequent evaluations.

Although the class-weighted training (RobustTrainer) improved sensitivity to toxic comments, it slightly reduced performance on the non-toxic class. In contrast, the standard fine-tuning strategy, aided by synthetic data augmentation, allowed the model to learn a more balanced representation without explicit weighting, leading to overall better generalization across both classes.

4.4.4. Classification with T5-small

In the final stage of model training, an alternative classification approach was implemented by fine-tuning a T5-small model (T5ForConditionalGeneration). Unlike GPT-2, which primarily operates as a decoder-only model, the T5-small architecture follows an encoder-decoder framework, allowing it to naturally handle tasks framed in a text-to-text format. This architectural difference makes T5 models particularly suitable for reframing classification problems into structured text generation tasks.

For toxic comment detection, the input comments were modified by prefixing each comment with an instructional prompt, such as "classify toxic comment:". The model was trained to generate a simple textual output — "1" for toxic comments and "0" for non-toxic comments. This formulation aligned seamlessly with the T5 model's pretraining strategy, which involved learning multiple natural language tasks under a unified text-to-text objective.

The Models were fine-tuned with a batch size of 2, learning rate 3e-5 for 2 epochs (tuned to fit into GPU memory). We turned on gradient checkpointing and reduced the maximum input sequence length to 64 tokens to save resources without losing long context. The original and synthetic comment was provided to the model as input to increase the model's exposure to different linguistic patterns and toxicity levels during training.

The fine-tuned T5-small model is evaluated with an accuracy of about 97.00%, a toxic F1-score of 98.10%, and a non-toxic F1-score of 82.45%. These results revealed that the approaches of reframing toxic comment detection as text generation worked well. The model demonstrated a high sensitivity in recognizing toxic comments and still achieved relatively good performance on the task of classifying non-toxic comments. The performance of the T5-small model shows the capability of the encoder-decoder Generative AI architectures for advanced decoding of large natural language or text classification.

4.4.5. Lightweight fine-tuning using distilGPT-2

To introduce a lightweight and resource-efficient alternative to the larger generative models, a DistilGPT-2 model was fine-tuned for the task of toxic comment classification. DistilGPT-2 is a distilled version of GPT-2, such that it has less memory altogether, and fewer parameters, but the same understanding of language for our purposes. This makes it a natural candidate for use in applications that require real-time processing.

The training of DistilGPT-2 was closely based on the same procedure as that of GPT-2. The comment text field was tokenized using the pretrained DistilGPT-2 tokenizer and fine-tuned with a batch size of 8, learning rate of 2e-5, and two training epochs. No class balancing weights were involved during training, so we can directly compare with a vanilla GPT-2 fine-tuning setting. The training set included both original and artificially generated comments to train the model on a broad range of toxicity patterns and linguistic structures.

On the validation set, the DistilGPT-2 model achieves an accuracy of around 96.00%, a toxic F1-score of 97.50% and a non-toxic F1-score of 80.00%. Slightly underperforming compared to full and small GPT-2 models, we found DistilGPT-2 to be a good balance of classification.

quality and computational cost. We show that DistilGPT-2 can be a viable and practical alternative for real-world deployments of GenAI models when constrained by resources that restrict the use of larger models.

4.5. Results and discussion

A comparative analysis of all of the models used for toxic comment classification, along with the three baseline tasks and generative AI models, is presented in this section. There were three main metrics to evaluate the performance: Accuracy, Toxic F1-score, and Non-Toxic F1-score. The summarized actual results are shown in Table 2.

Table 2: Results Comparison

Model	Accuracy	F1-score
Logistic Regression (TF-IDF)	93.00%	92.50%
Shallow Neural Network (TF-IDF)	94.00%	93.40%
GPT-2 with RobustTrainer (Class Weighted)	96.04%	97.79%
GPT-2 Standard Fine-tuning	96.22%	97.90%
DistilGPT-2 Fine-tuning	96.00%	97.50%
T5-small Text-to-Text Classification	97.00%	98.10%

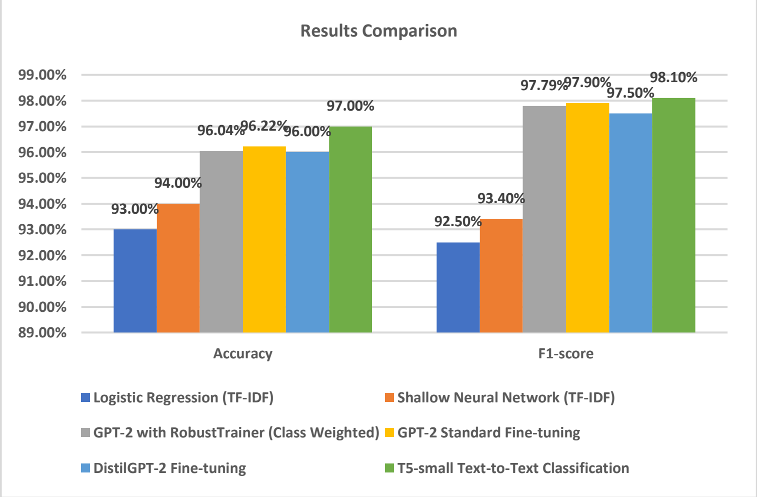

Fig. 3: Comparison of Applied Models.

The baseline models, Logistic Regression and Shallow Neural Network with TF-IDF features, acted as good starting points with the accuracies: Logistic Regression, 93.00%, Shallow Neural Network, 94.00%. However, their toxic F1-scores and non-toxic F1-scores were significantly lower than those of Generative AI models, highlighting the inadequacies of feature-based solutions in complex toxicity detection tasks. Among the Generative AI models, the T5-small model achieved the highest overall performance, with an accuracy of 97.00% and a toxic F1-score of 98.10%. The encoder-decoder structure of T5-small, combined with a text-to-text task framing, allowed it to excel in capturing nuanced toxicity patterns while maintaining balanced performance for non-toxic comments. The GPT-2 models also demonstrated strong capabilities. Fine-tuning GPT-2 using the standard Huggingface Trainer achieved an accuracy of 96.22% and a toxic F1-score of 97.90%, slightly outperforming the RobustTrainer version, which incorporated class weighting. Although the class-weighted GPT-2 model achieved a toxic F1-score of 97.79%, its non-toxic F1-score was slightly lower at 80.66%, indicating that class balancing improved toxic comment detection but slightly affected non-toxic sensitivity.

DistilGPT-2, a compressed variant of GPT-2, achieved competitive performance with an accuracy of 96.00% and a toxic F1-score of 97.50%. Although DistilGPT-2 is relatively lightweight, it showed strong generalization capabilities, making it suitable for real-world scenarios with limited computing power. Overall, the results demonstrate that Generative AI models, especially T5-small, offer significant

advantages over traditional and shallow learning models for toxic comment classification. The ability of large pretrained language models to generate, understand, and classify complex linguistic structures contributed to the substantial improvement in detection performance.
From a deployment perspective, T5-small delivered the best overall performance but required higher memory and compute resources due to its encoder-decoder structure. GPT-2 also yielded strong results but had slower training times and higher token processing costs. In contrast, DistilGPT-2 offered a practical trade-off by significantly reducing model size and inference time while maintaining competitive accuracy, making it more suitable for resource-constrained environments.

4.6. Comparison with existing works

Table 3 and Figure 4 present a comparative analysis of the proposed model's performance against existing state-of-the-art approaches for toxic comment classification. Traditional models like Logistic Regression and Naive Bayes showed modest performance, with accuracies of 80% and 88.48%, while DL approaches such as Bi-LSTM, Hybrid LSTM+CNN, and CNN with GloVe offered better results, ranging between 87% and 95%. Multi-task learning and ensemble methods pushed the performance further, both nearing 95% accuracy. In contrast, the proposed Generative AI-based framework achieved a superior accuracy of 97%, significantly outperforming all baseline and existing models.
The superior performance of our approach can be attributed to the integration of synthetic data and the use of multiple transformer-based generative models. Unlike [11], which used multi-task learning with limited augmentation, and [19], which relied on fixed embeddings like GloVe, our method dynamically learns contextual semantics through fine-tuning across multiple architectures (GPT-2, DistilGPT-2, and T5), enabling more robust and nuanced toxicity detection.

Table 3: Comparison with Existing Work

Model	Accuracy
Multi-task Learning [11]	94.89%
Hybrid LSTM+CNN [12]	92.63%
Logistic Regression [13]	80%
Naive Bayes [14]	88.48%
Bi-LSTM [16]	87%
CNN_model (GloVe) [19]	95%
Ensemble Model [21]	94.85%
Proposed Model (GEN AI based)	97%

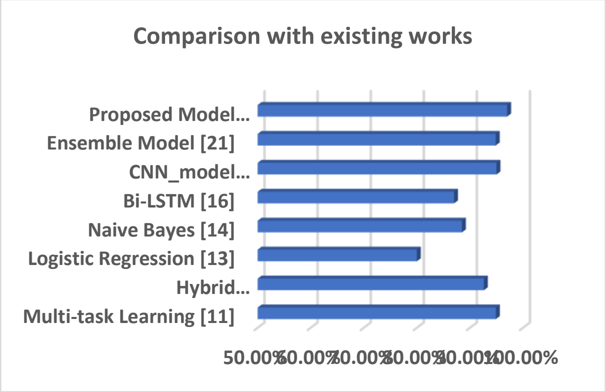

Fig. 4: Comparison with Existing Works.

Despite strong performance, the proposed approach has certain limitations. The generative models require substantial computational resources, which may hinder deployment in resource-constrained environments. Additionally, synthetic data generated through GPT-2 may introduce unintended biases. Lastly, the study is limited to English-language data, and generalization to multilingual datasets remains a challenge for future work.
From an ethical perspective, it is important to recognize that synthetic data generation, while useful for class balancing and diversity, may unintentionally amplify existing biases present in prompt design or training data. Ensuring transparency and rigorous validation of generated samples is essential to avoid reinforcing harmful patterns.

5. Conclusion

This paper presented a comprehensive methodology for toxic comment classification by combining traditional baselines with advanced Generative AI models. Initially, two baseline models—a Logistic Regression classifier using TF-IDF features and a shallow single-layer neural network—were established, achieving accuracies of 93.00% and 94.00%, respectively. While these baselines demonstrated reasonable performance, they exhibited limitations in capturing the deep contextual patterns necessary for nuanced toxicity detection. To overcome these challenges, a Generative AI-driven pipeline was introduced, involving synthetic data augmentation using GPT-2, fine-tuning GPT-2 with and without class balancing strategies, lightweight classification using DistilGPT-2, and reframing the classification task as a text-to-text problem using T5-small. Experimental evaluation revealed that the Generative AI models significantly outperformed the baseline approaches, with the T5-small model achieving the highest accuracy of 97.00% and a toxic F1-score of 98.10%. These findings underscore the effectiveness of leveraging large pretrained language models in improving toxic comment detection systems, particularly when combined with data augmentation strategies. Future research can extend this work by exploring larger model variants, such as T5-base, T5-large, or GPT-3-based architectures, to further enhance classification performance. Overall, the proposed methodology demonstrates the potential of combining synthetic data generation, multi-architecture fine-tuning, and Generative AI techniques to build scalable and effective toxic comment classification systems. Future work will explore extending the proposed framework to multilingual and code-mixed datasets, addressing challenges such as language-specific nuances and tokenizer limitations. Ethical considerations related to synthetic data generation and potential bias propagation will also be examined. Additionally, efforts will be directed toward integrating the models into real-time content moderation systems with constrained resources.

Acknowledgement

The authors would like to express their sincere gratitude to all those who contributed to the success of this research work.

References

[1] Y. Mao, Q. Liu, and Y. Zhang, “Sentiment analysis methods, applications, and challenges: A systematic literature review,” Journal of King Saud University - Computer and Information Sciences, vol. 36, no. 4. Springer Science and Business Media LLC, p. 102048, Apr. 2024. https://doi.org/10.1016/j.jksuci.2024.102048.

[2] S. Sushma, S. K. Nayak and M. V. Krishna, "A Comprehensive Review of Sentiment Analysis: Trends, Challenges, and Future Directions," 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, 2024, pp. 1175-1181, https://doi.org/10.1109/ICDICI62993.2024.10810919.

[3] N. Punetha and G. Jain, “Advancing sentiment classification through a population game model approach,” Scientific Reports, vol. 14, no. 1. Springer Science and Business Media LLC, Sep. 04, 2024. https://doi.org/10.1038/s41598-024-70766-z.

[4] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, “Enhancing machine learning-based sentiment analysis through feature extraction techniques,” PLOS ONE, vol. 19, no. 2. Public Library of Science (PLoS), p. e0294968, Feb. 14, 2024. https://doi.org/10.1371/journal.pone.0294968.

[5] Jency Jose and Simritha R, “Sentiment Analysis and Topic Classification with LSTM Networks and TextRazor,” International Journal of Data Informatics and Intelligent Computing, vol. 3, no. 2. Prisma Publications, pp. 42–51, Jun. 16, 2024. https://doi.org/10.59461/ijdiic.v3i2.115.

[6] Y. Cai, X. Li, Y. Zhang, J. Li, F. Zhu, and L. Rao, “Multimodal sentiment analysis based on multi-layer feature fusion and multi-task learning,” Scientific Reports, vol. 15, no. 1. Springer Science and Business Media LLC, Jan. 16, 2025. https://doi.org/10.1038/s41598-025-85859-6.

[7] L. P. Thomas, J. Thomas, V. A. Menon and T. K. Sateesh Kumar, "Sentiment Analysis of Telegram App Reviews Using Text Classification Models," 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, 2025, pp. 158-163, https://doi.org/10.1109/ICSADL65848.2025.10933302.

[8] A. Y. Alsalem and S. I. Abudalfa, "Empirical Analysis for Arabic Target-Dependent Sentiment Classification Using LLMs," 2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhir, Bahrain, 2024, pp. 170-176, https://doi.org/10.1109/3ict64318.2024.10824564.

[9] X. Fan and Z. Zhang, "A fine-grained sentiment analysis model based on multi-task learning," 2024 4th International Symposium on Computer Technology and Information Science (ISCTIS), Xi’an, China, 2024, pp. 157-161, https://doi.org/10.1109/ISCTIS63324.2024.10698954.

[10] 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, https://doi.org/10.1109/AINIT61980.2024.10581547.

[11] K. B. Nelatoori and H. B. Kommanti, “Multi-task learning for toxic comment classification and rationale extraction,” Journal of Intelligent Information Systems, vol. 60, no. 2. Springer Science and Business Media LLC, pp. 495–519, Aug. 20, 2022. https://doi.org/10.1007/s10844-022-00726-4.

[12] B. R. Naidu et al., “Toxic Comment Classification using Deep Learning,” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 7. Auricle Technologies, Pvt., Ltd., pp. 93–104, Sep. 01, 2023. https://doi.org/10.17762/ijritcc.v11i7.7834.

[13] N. Reddy, “Toxic Comments Classification,” International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 10, no. 6, pp. 2839–2846, Jun. 30, 2022. https://doi.org/10.22214/ijraset.2022.44500.

[14] A. Vinod, A. K V, M. M, R. Riyaz, and Mr. A. N, “Toxic Comment Detection and Classifier,” IJARCCE, vol. 13, no. 4. Tejass Publishers, Apr. 30, 2024. https://doi.org/10.17148/IJARCCE.2024.134174.

[15] H. Fan et al., “Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit,” Electronics, vol. 10, no. 11. MDPI AG, p. 1332, Jun. 01, 2021. https://doi.org/10.3390/electronics10111332.

[16] J. K. Giustino and Y. P. Santosa, “Toxic Comment Classification Comparison between LSTM, BILSTM, GRU, AND BIGRU,” Proxies : Jurnal Informatika, vol. 7, no. 2. Soegijapranata Catholic University, pp. 115–127, Aug. 29, 2024. https://doi.org/10.24167/proxies.v7i2.12471.

[17] Z. Zhao, Z. Zhang, and F. Hopfgartner, “Utilizing subjectivity level to mitigate identity term bias in toxic comments classification,” Online Social Networks and Media, vol. 29. Elsevier BV, p. 100205, May 2022. https://doi.org/10.1016/j.osnem.2022.100205.

[18] A. Bonetti, M. Martínez-Sober, J. C. Torres, J. M. Vega, S. Pellerin, and J. Vila-Francés, “Comparison between Machine Learning and Deep Learning Approaches for the Detection of Toxic Comments on Social Networks,” Applied Sciences, vol. 13, no. 10. MDPI AG, p. 6038, May 14, 2023. https://doi.org/10.3390/app13106038.

[19] A. Abbasi, A. R. Javed, F. Iqbal, N. Kryvinska, and Z. Jalil, “Deep learning for religious and continent-based toxic content detection and classification,” Scientific Reports, vol. 12, no. 1. Springer Science and Business Media LLC, Oct. 19, 2022. https://doi.org/10.1038/s41598-022-22523-3.

[20] A. Shinde, P. Shankar, A. Atul, and S. Rallabandi, “Bidirectional LSTM with convolution for toxic comment classification,” Proceedings of the 1st International Conference on Artificial Intelligence, Communication, IoT, Data Engineering and Security, IACIDS 2023, 23-25 November 2023, Lavasa, Pune, India. EAI, 2024. https://doi.org/10.4108/eai.23-11-2023.2343140.

[21] G. Xie, “An ensemble multilingual model for toxic comment classification,” International Conference on Algorithms, Microchips and Network Applications. SPIE, p. 38, May 06, 2022. https://doi.org/10.1117/12.2636419.

[22] M. Shahid, M. Umair, M. A. Iqbal, M. Rashid, S. Akram, and M. Zubair, “Leveraging deep learning for toxic comment detection in cursive languages,” PeerJ Computer Science, vol. 10. PeerJ, p. e2486, Dec. 13, 2024. https://doi.org/10.7717/peerj-cs.2486.

[23] S. Tsai and L. Shi, "Chinese Text Sentiment Classification Model Based on FastText and Multi-scale Deep Pyramid Convolutional Neural Network," 2022 International Conference on Computation, Big-Data and Engineering (ICCBE), Yunlin, Taiwan, 2022, pp. 75-77, https://doi.org/10.1109/ICCBE56101.2022.9888185.

[24] D. A. Kristiyanti, R. Aulianita, D. A. Putri, L. A. Utami, F. Agustini and Z. I. Alfianti, "Sentiment Classification Twitter of LRT, MRT, and Transjakarta Transportation using Support Vector Machine," 2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA), Denpasar, Bali, Indonesia, 2022, pp. 143-148, https://doi.org/10.1109/ICSINTESA56431.2022.10041651.

[25] Y. Yang, X. Sun, Q. Lu, R. Sutcliffe and J. Feng, "A Sentiment and Syntactic-Aware Graph Convolutional Network for Aspect-Level Sentiment Classification," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, https://doi.org/10.1109/ICASSP49357.2023.10096326.

[26] Y. Rong and B. Liu, "Research on Opinion Mining for Sentiment Classification of Micro-blog Text Based on DeBERTa," 2022 34th Chinese Control and Decision Conference (CCDC), Hefei, China, 2022, pp. 5337-5340, https://doi.org/10.1109/CCDC55256.2022.10033688.

[27] S. Sushma et al., “Enhanced toxic comment detection model through Deep Learning models using Word embeddings and transformer architectures”, futech, vol. 4, no. 3, pp. 76–84, May 2025. https://doi.org/10.55670/fpll.futech.4.3.8.

[28] 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, https://doi.org/10.1109/ICACCS54159.2022.9785208.

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