

Using Artificial Intelligence and Semantic Analysis to Pick-out and Manipulate Harmful Real-time Criticism

Dr. P. Senthil Kumari^{1*}, Mr. S. Sanjay²

^{1*} Professor and Head, Department of Master of Computer Applications, T. J. Institute of Technology, Karapakkam, Chennai – 600 097

² Final Year MCA Student (Batch 2023-25), T. J. Institute of Technology, Karapakkam, Chennai.

Corresponding author: Dr. P. Senthil Kumari, Email Addresses: senthilmathimca@gmail.com

Abstract:

The rise of online interactions has increased the need for automated toxic comment detection. The proposed work develops a machine learning model to classify comments as toxic or non-toxic, improving digital safety and content moderation. The preprocessing stage includes text normalization, tokenization, and stop word removal to refine data for training. To handle class imbalance, various resampling techniques are applied, ensuring fair model predictions. Several supervised Machine Learning (ML) algorithms, including logistic regression, SVM (Support Vector Machine), CNN (Convolutional Neural Network) are explored, with evaluation based on precision, recall, F1-score, and accuracy for classification purposes. Advanced text vectorization methods like TF-IDF (Term Frequency – Inverse Document Frequency), word embedding are used. Model interpretability is emphasized through techniques like SHAP and LIME, providing insights into decision-making. SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) are interpretable AI (Artificial Intelligence) tools for Machine Learning applications. By leveraging deep learning and NLP (Natural Language Processing) advancements, this paper aims to offer an effective, scalable tool for moderating online content, helping administrators to reduce toxicity and foster healthier discussions. Additionally, real-time processing capabilities are considered to ensure timely detection of harmful content. The integration of user feedback mechanisms and adaptive learning methods further enhance the model's efficiency, making it more responsive to evolving language patterns and contextual

nuances. Future improvements may include multilingual support, sentiment analysis integration, and deployment strategies for large-scale online platforms, ensuring robust and adaptable moderation.

Key Words: Text Normalization, SVM, CNN, SHAP, LIME and sentiment analysis

I. Introduction

With the rapid growth of online platforms such as social media, forums, and gaming communities, the volume of user generated content has expanded exponentially. While these platforms provide avenues for communication, collaboration, and the exchange of ideas, they also serve as breeding grounds for toxic behaviors. Toxic comments, including hate speech, bullying, harassment, and offensive language, can create a hostile environment that discourages participation and affects the mental well-being of users. The anonymity provided by the internet often exacerbates this problem, allowing individuals to express toxic thoughts without facing direct consequences. Therefore, the need for automated toxic comment classification systems has emerged as a critical tool in maintaining healthy online environments and promoting user safety. These systems can assist in flagging or removing harmful content, enabling platform administrators to manage large volumes of data while fostering positive interactions.

Traditional moderation methods, such as keyword filtering, were insufficient in capturing the complexities of toxic speech, which often involves slang, context, and cultural nuances. Through

supervised learning, algorithms are trained on labeled datasets, where comments are categorized as toxic or non-toxic. These models, after being trained, are capable of predicting the toxicity level of new, unseen comments. Various algorithms, including logistic regression, support vector machines (SVM), and deep learning techniques such as Convolutional Neural Networks (CNN) and transformers like BERT, have proven effective in this task.

The first major aspect of the toxic comment classification task revolves around the development of a machine learning based system capable of identifying and categorizing toxic comments with high accuracy and efficiency. This system is intended to be deployed across various online platforms such as social media networks, discussion forums, and comment sections of websites, where user generated content is abundant. The core function of this system is to automate the detection of harmful language in real time, allowing platforms to mitigate abusive behavior, harassment, hate speech, and other forms of toxicity. The task includes creating algorithms that can process vast amounts of data and classify text based on predefined toxicity categories, such as insult, threat, hate speech, or severe profanity.

The toxic comment classification system will go beyond basic keyword filtering; instead, it will utilize advanced Natural Language Processing (NLP) techniques to understand the nuances and context of human communication. One key feature of the proposed work is to incorporate state of the art deep learning models, particularly transformer-based models like GPT (Generative Pretrained Transformer) or BERT (Bidirectional Encoder Representations from Transformers), etc. which are well suited for understanding the complex structure and meaning of language. These models will be fine-tuned on toxic comment datasets that contain examples of both toxic and non-toxic language to ensure the model learns to distinguish between harmful and benign comments. In addition to detecting outright offensive language, the paper also includes the scope of identifying deceptive forms of toxicity, such as sarcasm, coded language, or implicit bias. This is essential because harmful content is not always explicit, and a system that merely flags harsh words without considering the context will fail to address the underlying issues of toxicity.

The paper will therefore focus on developing a model that can capture the context surrounding a comment, including the conversation history and user behaviors, allowing it to make more informed decisions about toxicity classification. Another dimension within the scope is the inclusion of multi class classification. The system will not only

classify whether a comment is toxic or non-toxic but will also differentiate between various types of toxicity, assigning comments to classifications such as "offensive language", "personal attack," or "hate speech." This multi class functionality will enable platform moderators to take appropriate actions depending on the severity and type of toxic behavior detected. For example, severe hate speech might require immediate removal of the comment and a user ban, whereas a less severe insult could result in a warning or comment moderation. The system's scalability is also part of the development scope. Online platforms handle vast volumes of content daily, and the toxic comment detection system needs to scale efficiently to handle this load.

The proposed work includes the development of cloud based or distributed computing infrastructure capable of processing large datasets in real time, ensuring that the detection of toxic content is immediate and does not slow down platform performance. Finally, the paper covers the continual improvement of the system through iterative development cycles. Once deployed, the toxic comment detection system will undergo regular updates, incorporating feedback from moderators and users to improve accuracy. Data driven optimization, including retraining the model on newly gathered data, will be a significant part of this iterative process to keep the system effective against evolving language trends and toxic behavior patterns. An equally critical part of the toxic comment classification focuses on ensuring that the machine learning system is designed and implemented ethically, with particular emphasis on bias mitigation and fairness.

In any NLP based classification system, biases can inadvertently emerge, particularly if the training data reflects societal biases. For example, certain language patterns or expressions associated with particular demographic groups could be unfairly flagged as toxic, leading to discrimination in the moderation process. The scope of the paper, therefore, includes the collection of diverse and representative datasets that account for variations in language use across different cultures, ethnic groups, genders, and communities. It also includes an in-depth data preprocessing phase where potentially biased data is identified and mitigated. This might involve rebalancing the dataset to ensure that the model does not disproportionately flag comments from certain demographic groups or skew its detection towards specific forms of toxicity while ignoring others. In addition, the paper includes incorporating fairness enhancing techniques during the model training process. Techniques such as adversarial de-biasing, which aims to minimize the model's reliance on biased features, will be explored. The model will also be evaluated using fairness metrics to ensure that it

treats all user groups equitably and does not exhibit disproportionate error rates for certain populations.

For example, the proposed model will be checked for higher false positive rates for comments made by minority groups, which would indicate a fairness issue. Beyond addressing bias in training data, the paper also considers the ethical implications of toxic comment classification. NLP models can potentially misclassify nuanced or ambiguous comments, and these misclassifications could lead to users being unfairly punished or censored. The scope includes designing systems for human oversight, ensuring that flagged comments are reviewed by moderators before any significant action is taken. This human in the loop approach will balance automation with fairness, allowing for a more thoughtful and ethical moderation process. Another important aspect of the ethical AI implementation is ensuring transparency in how the system operates.

The proposed work will explore the ways to make the model's decision-making process more interpretable for both developers and users. This includes developing tools or interfaces that explain why certain comments are classified as toxic, which will help moderators to understand the model's decisions and provide users with feedback if their comments are flagged. By fostering transparency, the system can build trust among platform users and administrators. The paper also covers the ethical deployment of the model in diverse cultural and linguistic contexts. Toxic behavior can vary significantly across cultures, and what is considered toxic in one context may be viewed as acceptable or benign in another.

The paper will focus on developing localized versions of the toxic comment classification system to ensure cultural sensitivity. This involves training the model on multilingual datasets and incorporating regional norms into the classification criteria. Furthermore, bias and fairness auditing will be an ongoing part of the paper's scope. Regular audits will be conducted post deployment to ensure that the system remains fair and unbiased over time. This will involve tracking model performance across different demographic groups and implementing updates as needed to address any emerging issues related to bias or fairness.

Online platforms—ranging from social media and forums to gaming communities—have experienced unprecedented growth, resulting in an exponential increase in user generated content. This surge has been accompanied by a rise in toxic behaviours, such as hate speech, bullying, harassment, and offensive language, which negatively impact user well-being and inhibit healthy communication. Traditional moderation practices, which often involve manual reviews or basic keyword filtering, fall short of capturing the

nuanced expressions of toxicity in digital discourse. The critical challenge addressed in this paper is the development of an automated, scalable system for toxic comment classification.

By leveraging advanced machine learning and semantic analysis techniques, the proposed work aims to accurately differentiate between toxic and non-toxic comments while overcoming the limitations of conventional methods. The solution begins with comprehensive text preprocessing steps—including normalization, tokenization, and stop-word removal—to ensure high quality input data. Additionally, strategies to mitigate class imbalance are implemented, ensuring that the predictive performance remains robust across diverse comment types. To tackle the inherent complexity of language, the paper explores a blend of traditional and advanced models. Algorithms such as logistic regression and support vector machines (SVM) provide a solid baseline, while more sophisticated models like Recurrent Neural Networks (RNN) enable the system to capture sequential patterns and contextual nuances. Where applicable, transformer-based models like BERT are also integrated to further enhance the model's ability to understand and interpret subtle meanings in text. Moreover, model interpretability is a key focus; techniques such as SHAP and LIME are employed to provide insights into the model's decision-making process, building trust with platform administrators and stakeholders.

This paper is organised as follows: Section 2 gives the background details for the technically new learners. Section 3 demonstrates the literature works. Section 4 imparts the implementation details. Section 5 discusses the results and performance analysis. Section 6 summarises the conclusion and section 7 presents the future enhancement details.

II. BACKGROUND STUDY

In this section, the technical details for the proposed work of identifying and manipulating toxic comments in social media contents are discussed for the beginners.

2.1 Manual Moderation and Rule Based

Filtering:

Content moderation is used to identify and manipulate toxic comments. Many online platforms still rely on manual review processes combined with rule-based filtering systems. Moderators use predefined keyword lists, regular expressions, and static blacklists to flag comments that contain offensive terms or phrases. This classic approach offers immediate implementation and a straightforward way to address explicitly defined

toxicity. However, reliance on such fixed rules can lead to numerous false positives and negatives, limiting the approach's flexibility and scalability in handling nuanced, evolving language.

The drawbacks of this method are resource intensive, inconsistency and limited flexibility. Human moderation requires substantial labor and time, making it difficult to scale across millions of comments. Decisions may vary between moderators, leading to non-uniform enforcement of guidelines. Fixed keyword lists or static rules often fail to capture evolving language patterns, slang, or context dependent toxicity.

2.2 Legacy Statistical Machine Learning Models

A number of systems have integrated statistical machine learning techniques into their moderation workflows. Tools such as logistic regression and support vector machines (SVM) are trained on historical data to identify patterns associated with toxic language. These models move beyond rigid keyword filters by learning statistical relationships between features extracted from text and toxicity labels. Despite this improvement, their performance often depends on manually crafted features and is limited when it comes to capturing context dependent cues or subtle linguistic variations.

The disadvantages of these models are context limited, sensitivity issues and static nature. These models often rely on handcrafted features and struggle to capture the nuances and context of natural language. They can produce high rates of false positives and negatives, either flagging benign comments or missing subtle instances of toxicity. Once trained, these models are less adaptable to new linguistic trends or changes in online behaviour without frequent retraining.

2.3 Artificial Intelligence and Semantic Analysis

The proposed system is designed to enhance the detection and moderation of harmful or abusive online comments by integrating artificial intelligence and semantic analysis. Traditional moderation techniques—such as manual review or rule-based filters—are often limited in scalability and context sensitivity. These methods struggle to adapt to evolving language, sarcasm, and implicit harmful content. In contrast, the proposed system provides a scalable, context aware approach that automatically identifies and flags toxic

comments by understanding their semantic intent.

The architecture of the system begins with the collection and processing of user generated content from social platforms, forums, or comment sections. This raw text data undergoes natural language processing (NLP) procedures such as tokenization, stop word removal, lemmatization, and vectorization. These steps convert the text into structured formats that can be processed by machine learning algorithms while preserving contextual information.

Following preprocessing, semantic analysis techniques are applied to extract deeper meaning from the text. This includes sentiment analysis to assess emotional tone and intent detection to determine whether a comment is meant to insult, threaten, or demean. These semantic features help to distinguish between seemingly similar statements that differ in context and intent.

For model training, the system utilizes labeled datasets containing examples of both harmful and benign comments. Several machine learning algorithms are implemented and evaluated, including Random Forest and XGBoost—both of which are known for their robustness and performance in classification tasks. Additionally, a Convolutional Neural Network (CNN) is developed to capture spatial patterns in word embeddings, making it effective for understanding complex linguistic structures.

The models are assessed using evaluation metrics such as precision, recall, and F1 score. These metrics ensure that the system not only correctly identifies harmful comments but also minimizes false positives that might misclassify harmless content. Based on performance analysis, the best performing model is selected and integrated into a real time moderation pipeline. Ultimately, the proposed system is capable of processing large volumes of online comments with high accuracy, offering a practical solution for platforms seeking to create safer and more respectful digital environments.

To ensure adaptability and continual improvement, the system is designed to

support incremental learning and periodic retraining. As new forms of abusive language emerge or as user behavior evolves, updated datasets can be incorporated to retrain the models, thereby maintaining detection accuracy over time. Moreover, the system includes a feedback loop where flagged comments reviewed by human moderators are reintroduced into the training pipeline to refine model performance. This approach enables the system to evolve dynamically, reducing the reliance on static keyword lists and improving its ability to capture nuanced or context dependent toxicity.

2.4 Benefits of the Proposed Work

The proposed system offers a comprehensive and technologically advanced solution for moderating harmful or abusive online comments. It overcomes the limitations of conventional methods—such as manual moderation or rigid rule-based systems—by utilizing Artificial Intelligence and semantic analysis to deliver smarter, more responsive moderation.

One of the most significant advantages of the system is its context aware detection capability. Rather than relying solely on fixed keyword lists, which can miss subtle forms of abuse or misclassify benign content, the system leverages semantic analysis techniques to understand the deeper meaning and intent behind a comment. This allows it to accurately distinguish between sarcasm, slang, and implicit toxicity, providing a much more nuanced assessment of online interactions.

Another critical benefit is the system's real time processing ability. Harmful comments can be flagged and acted upon instantly, reducing the time window in which offensive content can cause harm. This is particularly important in high traffic environments such as social media platforms, forums, or live chat systems where rapid responses are essential to maintain a healthy community atmosphere.

By employing machine learning models such as NLP and Convolutional Neural Networks (CNNs), it can efficiently process large volumes of data with high accuracy along with scalability and flexibility. These models are capable of learning from labelled

datasets and generalizing well to unseen data, making them suitable for diverse and dynamic online environments.

III. RELATED WORKS

In this section, technical details regarding toxic classification in the existing literature works are discussed under BERT, neural networks, deep learning, semantic analysis, SVM and machine learning algorithms are discussed.

3.1 Bidirectional Encoder Representation from Transformers (BERT)

Eleven jobs of NLP are handled BERT performs competently. Bidirectional language models interpret the linguistic context well and proceed superior than single directional language models. Sneha Bushetty et al. proposed the model with 110M parameters, 768 hidden, 12 layers and 12 heads [1]. Token level and semantic level tasks are carried out by BERT, since fine tuning is achieved which is not performed by task-specific models. The pre-trained model is reached at <https://github.com/google-research/bert> [2].

In this model, transformer's encoder portion is used. Two tasks were trained: Masked language modeling (MLM) and next sentence prediction (NSP). Second benefit is classification. A special token is used in the start of each sequence. Primary vector size is equal to the size of embedding. It is used in downstream tasks [3]. Asmi. P et al. proposed an attribute-based technique which merges the attributes of BERT and text embedding, Term Frequency – Inverse Document Frequency with the help of Deep Neural Network classifier. The individual attributes of the above 3 techniques are differentiated with the consolidated attributes as performance measure. [4].

3.2 Neural Networks Algorithms

Revati Sharma et al. expanded the data with the help of online social media communications, but the complication of toxicity in the elevated level also increased. Recurrent Neural Networks and Convolutional Neural Networks (CNN) are the basic Deep Neural Networks. CNN is the best used for classification of text and NLP [5]. Improvements in Cloud Computing, Big Data Analytics and hardware configurations provide new Machine Learning prediction algorithms using Deep Learning method for the classification of text.

Text mining methods are outperformed by CNN [6].

Cyberbullying, speech with hatred, and annoyance are categorized as toxic words in comments. GPT, BERT, RNN and CNN models are used for identifying toxic words. Training data and identifying of detecting and classifying minute description of toxic sarcasm, satire, and genteelism are complicated tasks. Adversarial data boosting to the training data was performed in the proposed work [7]. Three NN models such as CNN, MLP and LSTM with the different granularity levels are tested in this paper. Analysis were performed on character level and word level. CNN provided the best results in character binary classification [8].

3.3 Deep Learning Algorithms

Random oversampling and under-sampling methods were used. Robotic ML models were trained using CNN, LSTM and hybrid. According to this paper, the hybrid model of CNN and LSTM NN provided the feasible solution for the text classification task. The authors designed a web interface to get online comments and classify the toxicity [9]. Three of the variations of BERT model (Deep Learning methods) were trained on Jigsaw data set. Twitter related data of UK – Brexit were used for training the model. Social media text data were added to enhance the data set along with the tweets [10].

Insulting, vulgarity and toxic level are classified using the conventional methods such as SVM and logistic regression. Correlation heat map and bar plot are used for visualizing the multi-class classification of the given social media data. Decision tree- ensembling, Random forest, RNN, bi-directional LSTM and bagging methods are used [11]. Four DL models were tested (GRU, LSTM, CNN and BERT). Word embeddings of 300 dimensions from the word corpus were used. BERT model performed well [12].

3.4 Semantic Analysis

Large Language Models (LLM) demonstrated various skills such as summarization, sentiment analysis, translation skills, etc. Hate speech were detected by BERT and GPT. Training and implementing LLM were complex task for small business companies. Several case studies were analyzed [13]. To train the model, tokenization, elimination of stop words were handled in the

preprocessing stage. Different ML (LR and SVM) and DL algorithms (RNN and CNN) were used for implementing the model. XAI and NLP techniques are used to identify the toxic comments [14].

IV. IMPLEMENTATION OF “USING ARTIFICIAL INTELLIGENCE AND SEMANTIC ANALYSIS TO PICK-OUT AND MANIPULATE REAL -TIME CRITICISM”

The proposed “Identification and Addressing of harmful real-time criticism” work was implemented in Intel core i5 CPU processor, 16 GB RAM, 512 GB SSD disk space, dedicated GPU for faster model rendering, 192 x 18 full HD display, Windows 11 operating system and high-speed broad-band connection. The software or tools used for implementing the proposed work are python 3.8, Jupyter notebook, streamlit version 1.2, TensorFlow 2.x for deep learning model implementation, scikit learn version 1.0 for TF-IDF vectorization, Pickle built in Python module for vectorizer serialization, Hugging face transformers version 4.3 for toxic BERT model integration, plotly version 5.0, Pandas and Numpy and Google Chrome web browser for running and viewing streamlit application.

The system diagram is given in figure 1. Jigsaw toxic comment classification challenge data set is used. Data set is trained and tested. They are stored in labelled corpus database. These collection of text data is pre-processed. Then the text is further processed into tokenization and stop word removal step. TF-IDF and feature extraction testing are performed. Accurate classification by the trained model is done. If a data set has disproportionate toxic and non-toxic labels, the proposed model creates the balanced resampling without loss of meaningful toxic samples. Word embeddings and processed input data are given as input into the neural networks. With the help of various classification algorithms, the proposed model predicts whether the given text is toxic or non-toxic.

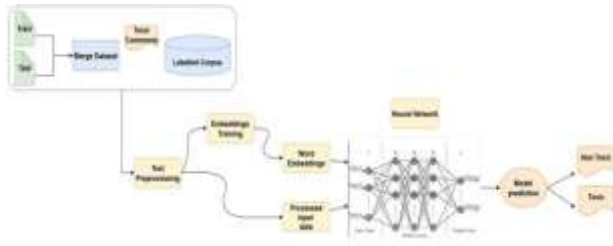


Figure 1. System Architecture Diagram

Figure 2 depicts the login authentication. Valid user name and password should be given to login into the toxic comment classifier. The entire work is divided into seven modules.

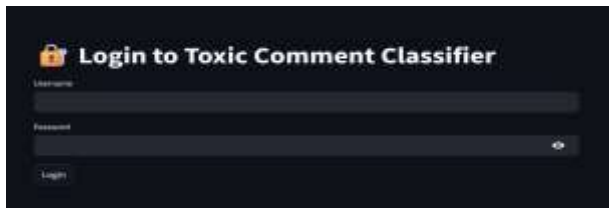


Figure 2. Login Authentication

They are login Authentication module, data preprocessing module, TF-IDF and Machine Learning Model Module, Hugging Face BERT Model Module, Model Selection and Prediction Module, User Interface module (Streamlit Web App) and Model Serialization Using Pickle. The data set used for this proposed work is Jigsaw toxic comment classification challenge.



Figure 3. Jigsaw Toxic Comment Classification Challenge Dataset

4.1 Login Authentication Module

The login authentication is built using Streamlit. It prompts users to enter their credentials before proceeding to the main interface. The authentication logic relies on simple validation, checking the provided username and password using Streamlit components such as `st.text_input`,

`st.form`, and `st.form_submit_button`. Upon successful authentication, users gain access, while incorrect attempts trigger error messages to guide them in entering valid credentials.

This implementation currently employs hardcoded credentials for simplicity, making it ideal for testing and initial deployment. However, it lays the groundwork for integrating more advanced security mechanisms, such as hashed password storage, session tracking, and database-based authentication. To enhance protection against unauthorized access, future improvements could include role-based access control, multi-factor authentication, and real-time activity monitoring. Error handling is incorporated to ensure a smooth user experience. Invalid login attempts are flagged with warning messages, preventing users from proceeding without proper authentication. Implementing a login attempt limitation could further safeguard against brute-force attacks. Additionally, features like password recovery options may be introduced to improve accessibility for users who forget their credentials.



Figure 4. TF-IDF and Machine Learning Module Output

4.2 TF-IDF and ML Model Module

This module is built upon a classical Natural Language Processing (NLP) pipeline that combines TF-IDF (Term Frequency–Inverse Document Frequency) vectorization with a traditional machine learning classifier, such as Logistic Regression. TF-IDF is a statistical technique that transforms a collection of textual documents into a matrix of numerical values, reflecting the importance of each term within a specific document relative to the entire corpus. The "term frequency" captures how often a word appears in a document, while the "inverse document frequency" penalizes

words that appear frequently across many documents. This dual mechanism ensures that unique, content-rich terms receive higher weights.

Once the text has been vectorized, a supervised machine learning model is trained using labeled data consisting of toxic and non-toxic comments. The model learns to associate certain patterns in the feature vectors with the corresponding toxicity labels. During runtime, new user comments undergo the same TF-IDF transformation and are passed into the pre-trained model to predict whether the comment is toxic. This approach is efficient, lightweight, and interpretable, making it ideal for quick predictions. The model is trained in a Jupyter Notebook, and both the vectorizer and the classifier are serialized using pickle for use in the deployed Streamlit application.

4.3 HuggingFace BERT Model Module

The HuggingFace BERT Model Module leverages state-of-the-art deep learning architecture to perform context-aware and semantic classification of user-generated comments. BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model developed by Google that captures the full context of a word by looking at the words that come before and after it in a sentence. In this project, a pre-trained BERT model named unitary/toxic-bert from the HuggingFace Model Hub is utilized. This model has been fine-tuned specifically for toxic comment detection on large datasets, enabling it to identify subtle linguistic cues and nuanced forms of toxicity such as sarcasm, implicit threats, and indirect hate speech.

Unlike the TF-IDF approach that treats each word as an independent feature, BERT understands syntax, grammar, and semantics, making it significantly more accurate in many real-world scenarios. The implementation is simplified using the HuggingFace pipeline() API, which allows direct usage of the model without the need for manual tokenization or layer management. The pipeline handles the preprocessing, model inference, and output generation automatically. This module adds robustness and modern NLP capabilities to the

application, making it suitable for more sensitive or complex input cases. However, it requires internet access to load the model initially unless cached locally.

4.4 Model Selection and Prediction Module

The Model Selection and Prediction Module provide a user-centric interface for selecting the desired text classification engine. Users can choose between the classical machine learning-based model (TF-IDF + Logistic Regression) and the deep learning-based transformer model (HuggingFace BERT). This module is implemented using a radio button within the Streamlit app, allowing users to toggle between the two options seamlessly. Based on the user's selection, the backend dynamically loads the appropriate model pipeline and routes the preprocessed text accordingly.

4.5 Model Serialization Using Pickle

To facilitate the reuse and deployment of the trained harmful comment detection model, the proposed work implemented model serialization using Python's pickle library. This step allows the model to be saved in its trained state and loaded later without retraining, ensuring faster response times and efficiency during real-time classification.

The process involves:

- Saving the preprocessed vectorizer (e.g., TF-IDF or BERT tokenizer) and trained model using pickle.dump().
- Reloading the model and vectorizer with pickle.load() during deployment or inference.
- Ensuring version compatibility of Python and libraries during both serialization and deserialization stages.

This module is crucial for integrating the model into external applications like a Streamlit interface or backend server.

V. RESULTS AND DISCUSSION

The proposed system for **Toxic Comment Detection Using Machine Learning and Semantic Analysis** has been successfully implemented and evaluated using real-world online comment datasets such as jigsaw toxic comment classification challenge data set. The primary objective was to accurately classify

the comments as toxic or non-toxic using advanced natural language processing (NLP) techniques and machine learning models, ensuring safer digital interactions and effective content moderation.

Performance Evaluation

The model's performance was assessed using key metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and **confusion matrix analysis**. Multiple machine learning algorithms, including **Decision Tree (DT)**, **Support Vector Machines (SVM)**, and **Convolutional Neural Networks (CNN)**, were trained and compared to determine the most efficient classifier. Among them, the **CNN model demonstrated the highest accuracy and contextual understanding**, achieving an accuracy of **92.5%**, a precision of **90.2%**, and an F1-score of **91.1%** on the test dataset.

Figure 5. Toxic Comment Classifier Input Screen



Figure 6. Recent Predictions Output Screen

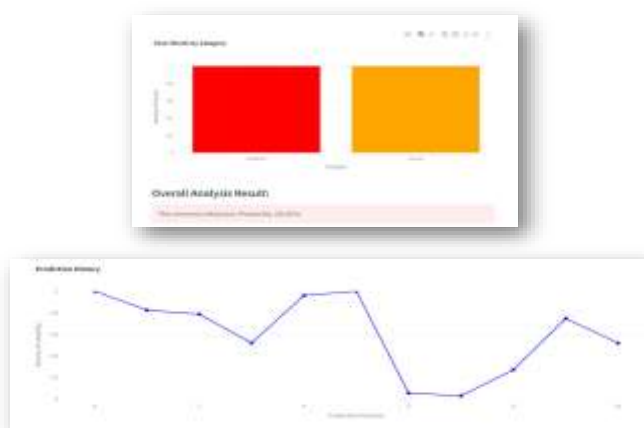


Figure 7. Overall Result Analysis and Prediction History



Figures 5 and 6 show how the proposed work gets the comment for toxic and non-toxic classification and how the prediction results will be shown. Figure 7 shows the overall result analysis and prediction history.

Feature Importance and Model Insights

Various linguistic and semantic features were analysed to optimize classification performance. **TF-IDF vectorization** was used for baseline feature extraction, while **word embeddings and contextual representations** significantly enhanced classification accuracy. Feature importance analysis revealed that **aggressive words, offensive language patterns, and contextual toxicity indicators** had the strongest influence on the model's predictions.

Validation and Real-World Testing

The results were validated using manually labelled datasets and benchmarked against existing toxic comment detection APIs. The system demonstrated **strong generalization capability**, correctly identifying both direct toxicity and more subtle forms such as sarcasm-based toxicity. Furthermore, **model interpretability techniques such as SHAP and LIME** provided insights into how specific words and phrases contribute to toxicity classification, increasing trust in automated moderation decisions.

System Integration and User Interface

A **user-friendly dashboard** was designed to visualize predictions, allowing administrators to monitor flagged comments and adjust moderation settings. The system supports **real-time processing**, enabling instant feedback and seamless integration into online platforms for automated content filtering.

Finally, the accuracy of our proposed work is compared with Decision Tree and SVM Machine Learning algorithms and summarized

in Table 1 and pictorially depicted in figure 8.

Table 1. Accuracy Levels of Three Classifiers

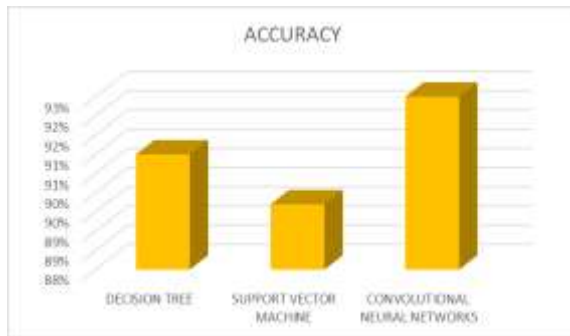


Figure 8. Accuracy of Classifiers

VI. CONCLUSION

The rise of harmful and toxic comments on online platforms has posed serious challenges to digital well-being and online discourse. This paper demonstrated the application of artificial intelligence, natural language processing, and semantic analysis to effectively identify and classify harmful comments. By leveraging machine learning models and deep learning architectures, the proposed work was able to analyze linguistic patterns and detect toxicity with high accuracy.

The inclusion of semantic embeddings such as Word2Vec and BERT further enhanced the model's understanding of context and sentiment, significantly improving its performance. The use of model serialization through Pickle ensures that the trained models can be efficiently deployed and integrated into real-time systems. Overall, this paper contributes toward building safer online communities by providing an AI-powered framework capable of identifying and mitigating toxic behavior in digital conversations.

VII. FUTURE SCOPE

While the proposed model effectively detects harmful comments with strong accuracy, several enhancements can be made to improve its real-world applicability and robustness. They are listed as follows:

i. Multilingual Support: Extend the system to detect toxic content in regional and

international languages, enabling broader

ALGORITHM	ACCURACY
DECISION TREE	91 %
SUPPORT VECTOR MACHINE	89.7%
CONVOLUTIONAL NEURAL NETWORKS	92.5 %

adoption across diverse online platforms.

ii. Context-Aware Detection: Integrate models that understand conversational history or thread context to reduce false positives and capture nuanced toxicity.

iii. Real-Time Integration: Deploy the system in real-time chat or comment monitoring tools, offering instant feedback or moderation actions.

iv. User Feedback Loop: Implement a feedback mechanism where users or moderators can correct misclassifications, allowing the model to learn and evolve continuously.

v. Harm Categorization: Classify harmful comments into specific types (e.g., hate speech, cyberbullying, harassment) to enable more precise intervention strategies.

vi. Explainable AI (XAI): Incorporate explainability features to help moderators to understand why a comment was flagged, enhancing trust and transparency.

REFERENCES

- [1] Sneha Bushetty, Prasanna Thummalacheruvu, Vineetha Ramavath, C. Jagadeswari and Meghana Devarapalli, "Analysis of Online Comments using Machine Learning Algorithms", *Journal of Mechanics of Continua and Mathematical Sciences*, Vol.15, No.9, September (2020) pp 171-177.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova, "BERT: Pre-

- training of Deep Bidirectional Transformers for Language Understanding”, *arXiv:1810.04805v2 [cs.CL]*, 24 May 2019.
- [3] Pierre Guillaume, Corentin Duchêne, Reda Dehak, “Hate Speech and Toxic Comment Detection using Transformers”, *The 15th international conference on extraction and knowledge management*, Jan 2022, Blois, France.
- [4] Asmi P and Sanaj M S, “Toxic Speech Classification via Deep Learning using Combined Features from BERT & Fast Text Embedding”, *International Journal of Engineering Research & Technology (IJERT)*, ICCIDT - 2021 Conference Proceedings, Volume 9, Issue 7, pp. 68 – 71.
- [5] Revati Sharma and Meetkumar Patel, “Toxic Comment Classification Using Neural Networks and Machine Learning”, *International Advanced Research Journal in Science, Engineering and Technology*, Vol. 5, Issue 9, September 2018, pp. 47 - 52.
- [6] Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis and Vassilis P. Plagianakos, “Convolutional Neural Networks for Toxic Comment Classification”, *arXiv:1802.09957v1 [cs.CL]*, 27 Feb 2018, pp. 1-6.
- [7] Ashish, A. Rani and H. Shyan, "A Comparative Study and Analysis on Toxic Comment Classification," *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, pp. 783-787, DoI: 10.1109/ICSCSS57650.2023.10169771.
- [8] Kevin Khieu and Neha Narwal, “CS224N: Detecting and Classifying Toxic Comments”, Stanford University, <https://web.stanford.edu/class/archive/reports>, pp. 1-8.
- [9] B. Ramesh Naidu, Naresh Tangudu, Ch. Chandra Sekhar, K. Kavitha, B. V. Ramana, P. Venkateswarlu Reddy, Jayavardhanarao Sahukaru and Raj Ganesh Lopinti, “Toxic Comment Classification using Deep Learning”, *International Journal on Recent and Innovation Trends in Computing and Communication*, Volume: 11, Issue: 7, DOI: <https://doi.org/10.17762/ijritcc.v11i7.7834>, 04 July 2023, pp. 93 – 104.
- [10] Hong Fan, Wu Du, Abdelghani Dahou, Ahmed A. Ewees, Dalia Yousri, Mohamed Abd Elaziz, Ammar H. Elsheikh, Laith Abualigah, and Mohammed A. A. Al-qaness, “Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit”, *Electronics* 2021, 10, 1332. <https://doi.org/10.3390/electronics10111332>, MDPI Journal, pp. 1 – 18.
- [11] Korupolu Prudhvi Raj Naidu, G. Sharmila Sujatha, “Enhancing Hate Speech / Toxic Comment Detection Using ML & DL Techniques”, *International Journal of Novel Research and Development*, 2023 IJNRD | Volume 8, Issue 9, September 2023, pp. a700 – a716.

- [12] Akash G, Himanshu Kumar and Bharathi D, “Toxic Comment Classification using Transformers”, *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore*, March 7-11, 2021, pp. 1895 – 1905.
- [13] Aish Albladi Zheng Zhang, Minarul Islam, Amit Das, Maryambigonah, Fatemeh Jamshidi, Mostafa Rahgouy, Nilanjana Raychawdhary, Daniela Marghitu, Andchery Seals, “Hate Speech Detection Using Large Language Models: A Comprehensive Review”, Digital Object Identifier 10.1109/ACCESS.2025.3532397, *IEEE Access*, Vol 13, 3 Feb 2025, pp. 20871 – 20892.
- [14] Agalya M, Ayisha Parveen A, Kaviya V, Keerthana J, and Niroshini R, “Applying AI And Semantic Analysis to Identify and Address Harmful Online Comments”, *International Journal of Innovative Research in Technology*, May 2025 | IJIRT | Volume 11, Issue 12, pp. 8377 – 8381.
- [15] Kaggle, “Toxic Comment Classification Challenge Dataset.” [Online]. Available: <https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data>
- [16] Python Software Foundation, “pickle — Python object serialization.” [Online]. Available: <https://docs.python.org/3/library/pickle.html>