COMMENT TOXICITY PREDICTION USING RNN

A Mini Project report submitted
in the partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering
(Artificial Intelligence and Machine Learning)

by

21071A6606	Beesu Mokshagna
21071A6611	Bommakanti Navaneeth
21071A6615	Chennarapu Sai Sreeja
21071A6620	Gadiga Ruchitha

Under the Guidance of

Dr. A. Kousar Nikath
Associate Professor
CSE – AIML & IoT
VNR VJIET

Submitted to

TAMASOMA JYOTIRGAMAYA
Estd. 1995

DEPARTMENT OF
CSE- (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING & INTERNET OF THINGS)

Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology,
Hyderabad, Telangana

November 2024

TAMASOMA JYOTIRGAMAYA
Estd. 1995

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous, ISO 9001:2015 & QS I-Gauge Diamond Rated Institute, Accredited by NAAC with 'A++' Grade
NBA Accreditation for B.Tech. CE, EEE, ME, ECE, CSE, EIE, IT Programmes
Approved by AICTE, New Delhi, Affiliated to JNTUH, NIRF 135th Rank in Engineering Category
Recognized as "College with Potential for Excellence" by UGC
VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761

CERTIFICATE

This is to certify that **BEESU MOKSHAGNA (21071A6606), BOMMAKANTI NAVANEETH (21071A6611), CHENNARAPU SAI SREEJA (21071A6615), GADIGA RUCHITHA (21071A6620)** have successfully completed their Mini project work at CSE-(AIML & IoT) Department of VNRVJIET, Hyderabad entitled **"Comment Toxicity Prediction Using RNN"** in partial fulfilment of the requirements for the award of B. Tech degree during the academic year 2024-2025.

This work is carried out under my supervision and has not been submitted to any other University/Institute for award of any degree/diploma.

Sagar Yeruva.

25/11/24

GUIDE
Dr. A. Kousar Nikath
Associate Professor
Dept. of CSE (AIML & IoT)

VNR Vignana Jyothi Institute of Engineering & Technology
Pragathi Nagar, Nizampet (S.O), Hyderabad-500 090.
Dept. of CSE-AIML and IoT

Head of the Department
Dr. Sagar Yeruva
Associate Professor
Dept. of CSE (AIML & IoT)

HoD CSE- AIML and CSE-IoT
VNR Vignana Jyothi Institute of Engineering & Technology
Pragathi Nagar, Nizampet (S.O.)
Hyderabad - 500 090.

13/12/24.

DECLARATION

This is to certify that our project titled "**COMMENT TOXICITY PREDICTION USING RNN**" submitted to Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology in complete fulfilment of the requirement for the award of Bachelor of Technology in CSE- (Artificial Intelligence and Machine Learning) is a bonafide report to the work carried out by us under the guidance and supervision of Dr. A. Kousar Nikath, Associate Professor, Department of CSE-(AIML & IoT), Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology. To the best of our knowledge, this has not been submitted in any form to another University/Institute for an award of any degree/diploma.

B. Mokshagna
Beesu Mokshagna
21071A6606
Dept. of CSE (AIML&IoT)

B. Navaneeth
Bommakanti Navaneeth
21071A6611
Dept. of CSE (AIML&IoT)

C. Saisreeja
Chennarapu Sai Sreeja
21071A6615
Dept. of CSE (AIML&IoT)

Rucha
Gadiga Ruchitha
21071A6620
Dept. of CSE (AIML&IoT)

ABSTRACT

In today's digital era, online platforms such as YouTube, Instagram, and various social media channels have become integral parts of our daily lives, providing avenues for communication, expression, and interaction. However, with the proliferation of user-generated content, the "Issue of toxicity "in comments has emerged as a significant concern, posing challenges to maintaining a positive and safe online environment. In this study, we propose a Deep learning approach, leveraging "Sequential and Recurrent neural network (RNN)" models implemented using "TensorFlow" and "Keras" frameworks, to predict comment toxicity in social media comments. By harnessing the power of deep learning, our model aims to accurately classify comments based on their toxicity levels, thereby enabling platforms to proactively identify and mitigate potentially harmful content. By avoiding straightforward algorithms such as linear regression or classification, we showcase the effectiveness of our approach in fostering a safer online environment. Through extensive experimentation and evaluation, we demonstrate the effectiveness and robustness of our approach in achieving high accuracy and reliability in comment toxicity prediction. Our project underscores the importance of leveraging advanced machine learning techniques to address real-world challenges in online content moderation, ultimately contributing to fostering healthier and safer digital communities.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In today's digital age, online platforms have become vital spaces for communication, information sharing, and community building. However, these platforms also face significant challenges, one of the most pressing being the proliferation of toxic comments. Toxic comments—those that are harmful, abusive, or hateful—can severely degrade the quality of online interactions, leading to hostile environments that deter positive engagement and suppress diverse voices.

The project on **Comment Toxicity Prediction** aims to address this issue by leveraging machine learning techniques to automatically detect and classify toxic comments. The core objective is to build a predictive model that can accurately identify toxicity in user-generated content across various online platforms, such as social media, forums, and comment sections of news websites.

This project involves several key components:

1. **Data Collection and Pre-processing:** Gathering a large and diverse dataset of comments from multiple sources, including both toxic and non-toxic examples. Pre-processing steps such as text normalization, tokenization, and removal of irrelevant content are essential for ensuring high-quality input data.
2. **Feature Engineering:** Extracting meaningful features from the text data, such as word embeddings, n-grams, and syntactic patterns, which help the model differentiate between toxic and non-toxic content.
3. **Model Development:** Implementing and experimenting with various machine learning models, including traditional classifiers like Logistic Regression and Support Vector Machines, as well as more advanced techniques like deep learning models (e.g., LSTM, BERT).
4. **Model Evaluation:** Assessing the model's performance using metrics like accuracy, precision, recall, and F1-score. Special attention is given to minimizing false positives (incorrectly labeling a non-toxic comment as toxic) and false negatives (failing to identify a truly toxic comment).
5. **Deployment and Integration:** Developing a user-friendly interface or API that can be integrated into online platforms for real-time comment moderation. The system should be scalable and capable of handling large volumes of data with low latency.

The successful implementation of this project has the potential to significantly improve the quality of online discourse by filtering out harmful content and fostering healthier, more inclusive digital communities

1.1 EXISTING SYSTEM:

Existing models for comment toxicity prediction encompass a range of techniques, including traditional machine learning algorithms like linear regression and Naive Bayes, as well as more sophisticated approaches like Toxic Filtering using Natural Language Processing (NLP).

1. **Linear Regression:**
   - Linear regression is a simple and interpretable statistical model used for regression tasks.
   - In the context of comment toxicity prediction, linear regression models can be trained on handcrafted features extracted from the text, such as bag-of-words representations or TF IDF scores.
   - These models estimate the relationship between the input features and the toxicity level of the comments using linear functions.
   - However, linear regression may struggle to capture complex patterns and sequential dependencies in text data, which are crucial for accurately detecting toxicity.

2. **Naive Bayes:**
   - Naive Bayes is a probabilistic classifier based on Bayes' theorem with the "naive" assumption of feature independence.
   - In comment toxicity prediction, Naive Bayes models often utilize bag-of-words or TF-IDF representations as input features.
   - TF-IDF treats each word independently and doesn't consider the interactions or dependencies between them
   - However, it may not capture the nuanced relationships between words in comments, potentially limiting its effectiveness for detecting subtle forms of toxicity.
   - For example, certain combinations of words or specific language patterns may convey toxicity even if the individual words themselves are not particularly rare or common across the dataset.

1.2 PROPOSED SYSTEM:

Our system leverages Recurrent Neural Networks (RNNs) and sequential modeling for comment toxicity prediction, surpassing traditional methods like linear regression and Naive Bayes. By dynamically capturing word relationships, it excels in detecting subtle toxicities, with automated feature extraction and enhanced contextual understanding ensuring superior performance in moderation tasks.

1. Handling Sequential Information:

- Unlike linear regression and Naive Bayes, which treat input data as independent features, your proposed RNN-based model explicitly captures sequential dependencies within the text.
- RNNs are well-suited for processing sequential data like text by retaining information about previous inputs, allowing them to capture the context and temporal dependencies between words in comments.
- This sequential modelling approach enables your model to understand the nuanced relationships between words and detect subtle forms of toxicity that linear regression and Naive Bayes may miss due to their lack of sequential awareness.

1. **Complexity and Flexibility:**

- Our proposed RNN-based model is more complex and flexible compared to linear regression and Naive Bayes, as it can learn intricate patterns and representations directly from the sequential data.
- RNNs have the ability to adapt to the complexity of the data, making them suitable for tasks where the relationships between input features are non-linear and context-dependent.
- This increased complexity allows your model to capture the varying degrees of toxicity present in comments, including subtle and context-dependent expressions of toxicity that may be challenging for linear regression and Naive Bayes to capture effectively.

2. **Generalization and Performance:**

- RNN-based models have the potential to generalize better to unseen examples and perform well on complex tasks like comment toxicity prediction, thanks to their ability to capture sequential dependencies and context.
- By learning from the sequential structure of the data, your model can adapt to diverse linguistic patterns and effectively classify comments across different contexts and languages.
- This generalization capability may outperform linear regression and Naive Bayes, especially when dealing with nuanced and context-dependent expressions of toxicity in comments.

2 LITERATURE SURVEY

2.1 RELATED WORK

We analyzed eight distinct research papers focused on comment toxicity prediction, exploring various machine learning algorithms and methodologies. While these studies provided valuable insights into detecting and classifying toxic comments, we identified some limitations in their approaches. The papers revealed challenges in feature extraction, model selection, and evaluation metrics, with certain models struggling to handle the nuances of language diversity and contextual subtleties. Additionally, some studies faced trade-offs between model accuracy and computational efficiency, highlighting gaps in scalability and real-time application. These limitations emphasize the need for further research to develop more robust and adaptable toxicity prediction systems.

2.1.1 Toxic Comment Classification

The Objective of the project is to create a system to automatically detect toxic comments on social media. This would allow social media platforms to remove toxic comments and create a safer online environment. The problem addressed is that current methods for filtering toxic comments are not very effective. Social media companies often use simple methods like keyword searches, which can remove harmless comments. In addition, manually moderating comments is expensive and time-consuming.

2.1.2 Deep learning for religious and continent-based toxic content detection and classification

Deep learning struggles to identify toxic comments without flagging religious or ethnic words in harmless contexts. This paper tackles this bias by training deep learning models (like CNNs and LSTMs) on religious and ethnic toxicity datasets. They experiment with word embeddings (GloVe, Word2vec, FastText) to improve accuracy in classifying truly toxic comments while reducing bias against specific identity groups.

2.1.3 Toxic Comment Detection in Online Discussions

The paper explores content moderation challenges in online news comment sections, focusing on sentiment analysis and toxic comment identification. It defines toxicity subclasses and proposes deep learning methods like RNNs and transformer models. It advocates for fine-grained classification and transfer learning to address data scarcity. Real-world applications like semi-automated moderation are discussed, along with future challenges and limitations.

2.1.4 An Assessment of Deep Learning Models and Word Embeddings for Toxicity Detection within Online Textual Comments

Deep learning technologies, which excel at learning complex features from data, offer a promising solution for this task. In particular, the paper investigates the effectiveness of Long Short-Term Memory (LSTM) layers, a type of recurrent neural network, in combination with state-of-the-art word embeddings.By leveraging word embeddings to represent words in the comments and employing deep learning architectures like LSTM, the paper aims to develop a model capable of automatically detecting toxicity without relying on hand-crafted features. The results indicate that LSTM layers combined with mimicked word embeddings perform well for this task.

2.1.5 Identifying Toxicity Within YouTube Video Comment Text Data

YouTube comments get nasty on anti-NATO channels compared to pro-NATO ones. Analysis shows more negativity and Russia bashing in anti-NATO comments, while pro-NATO comments focus on alliance and positivity. This research can help fight online toxicity by informing platform moderation and user behaviour.

2.1.6 Understanding Toxicity Triggers on Reddit in the Context of Singapore

The study explores toxicity triggers on Reddit, focusing on Singapore and comparing with New York City. It identifies unique triggers like COVID-19 regulations and education issues in Singapore, contrasting with NYC's emphasis on protests and elections. Using machine learning, it highlights differences in online discourse between these contexts.

2.1.7 Machine learning methods for toxic comment classification: a systematic review

Automatic detection of toxic comments is becoming increasingly important. This paper reviews 31 studies on using machine learning to classify toxic comments. Deep neural networks are most effective, but simpler methods like logistic regression are also used.

2.1.8 A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification

This research tackles the growing issue of toxic comments online, which hinder healthy discussions. They explore using word embeddings, which capture word meaning, to automatically classify comments as toxic or not. The paper finds that word embeddings improve classification accuracy, especially when tailored to the domain of the comments.

2.2 SYSTEM STUDY

The system study for the Comment Toxicity Prediction project begins with a comprehensive understanding of the problem domain. Online platforms face a growing challenge in managing user-generated content due to the sheer volume of comments and the diversity of languages, cultural contexts, and user behaviours. Manual moderation is often inadequate, as it is labour-intensive and unable to scale effectively. Therefore, the need for an automated system that can accurately identify and filter toxic comments in real-time is critical. This system study evaluates the requirements, challenges, and potential solutions for developing such a system, focusing on the technical and operational aspects.

Data Requirements and Pre-processing:

A successful toxicity prediction system relies heavily on the availability and quality of data. The system requires a large, annotated dataset containing examples of both toxic and non-toxic comments. These datasets must be representative of the diverse content encountered on various platforms, including differences in language, tone, and context. Data pre-processing is a crucial step in transforming raw comments into a format suitable for machine learning models. This includes text normalization (e.g., converting text to lowercase, removing punctuation), tokenization (breaking down text into words or phrases), and filtering out irrelevant information such as URLs or special characters. Handling imbalances in the dataset, where toxic comments may be less frequent, is also a key challenge addressed during pre-processing.

Feature Engineering and Model Selection:

The effectiveness of the toxicity prediction system depends on the quality of features extracted from the text data. Feature engineering involves identifying patterns and attributes in the text that can help differentiate toxic from non-toxic comments. Traditional methods may involve using n-grams, TF-IDF scores, and sentiment analysis, while more advanced approaches utilize word embeddings or contextual representations from models like BERT. Selecting the right machine learning model is another critical step in the system study. Options range from classic algorithms such as Logistic Regression and Support Vector Machines to more sophisticated deep learning architectures like Recurrent Neural Networks (RNNs) and Transformers. Each model type has its strengths and trade-offs in terms of accuracy, computational complexity, and interpretability.

System Architecture and Integration:

The architecture of the toxicity prediction system must support real-time processing, scalability, and integration with existing online platforms. The system is typically designed as a pipeline that handles data input, pre-processing, model inference, and output. In real-time applications, the system should be capable of processing thousands of comments per second, requiring efficient algorithms and possibly the use of parallel processing or distributed computing. Deployment considerations include the development of APIs or interfaces that allow seamless integration with content management systems, enabling automated moderation of comments as they are posted. Additionally, the system should support continuous learning, where new data can be used to periodically update and improve the model’s accuracy.

Evaluation and Ethical Considerations:

Evaluating the toxicity prediction system involves rigorous testing using metrics like accuracy, precision, recall, and F1-score. However, the system study also highlights the importance of minimizing biases and ensuring that the model does not disproportionately flag comments from specific groups or in certain languages. Ethical considerations are paramount, as the system's decisions can affect free speech and user experiences. Regular audits, transparency in how decisions are made, and mechanisms for user feedback are necessary to maintain trust and fairness in the system. The study concludes by recommending best practices for deploying the system in a way that balances the need for moderation with respect for diverse perspectives and open dialogue.

3 DESIGN

3.1 SYSTEM REQUIREMENTS

3.1.1 Software Requirements:

3.1.1.1 Programming Languages:

- **Python:** Python is the predominant language for machine learning and data analysis. You'll need Python for implementing your ML models, data pre-processing, and analysis

3.1.1.2 Machine Learning Libraries:

- **Scikit-Learn:** Scikit-Learn provides a wide range of tools for machine learning, including the algorithms you mentioned (support vector machines, logistic regression, K-nearest neighbors, random forest)
- **TensorFlow or PyTorch:**These deep learning frameworks are necessary if you plan to work with neural networks, which can be beneficial for complex disease prediction tasks.

3.1.1.3 Data Analysis and Visualization:

- **Pandas:**For data manipulation and analysis.
- **Matplotlib and Seaborn:** For data visualization.

3.1.1.4 Google Colaboratory:

- Google Colab is great for interactive data exploration and model development.

3.1.2 Hardware Requirements:

3.1.2.1 Computing Power:

- A reasonably powerful computer with a multi-core CPU is necessary for training ML models, especially if you're working with large datasets or complex deep learning models

3.1.2.2 GPU:

- Having access to a GPU (Graphics Processing Unit) can significantly speed up training deep learning models.

3.1.2.3 RAM:

- Adequate RAM, typically 16 GB or more, is essential for handling large datasets and training complex models.

3.1.2.4 Storage:

- Sufficient storage space for datasets and model checkpoints. SSDs are preferable for faster data access.

3.1.2.5 Internet Connection:

- A reliable internet connection is necessary for downloading pre-trained models (like the ones from Hugging Face), as well as for accessing and loading data from the provided URLs.

3.2 UML

3.2.1 Use case:

They are usually used to illustrate the various actions taken by the application. They also show the several users who can carry out these functions. Use-case diagrams fall under behaviour diagrams due to their emphasis on the tasks carried out and the users (actors) who carry out these tasks.

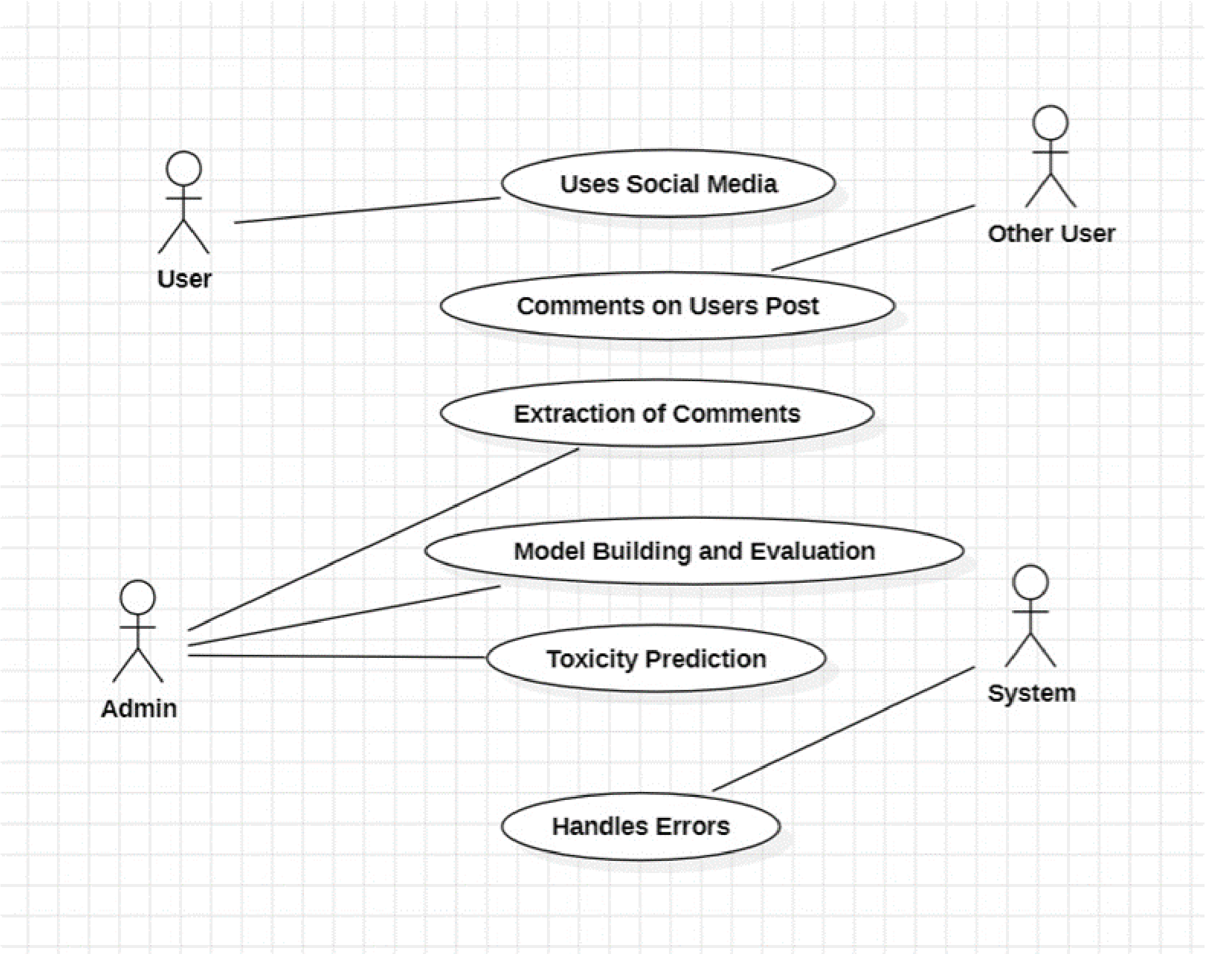


Figure 1 Use Case Diagram

3.2.2 Activity Diagram:

An activity diagram visually presents a series of actions or flow of control in a system like a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling.

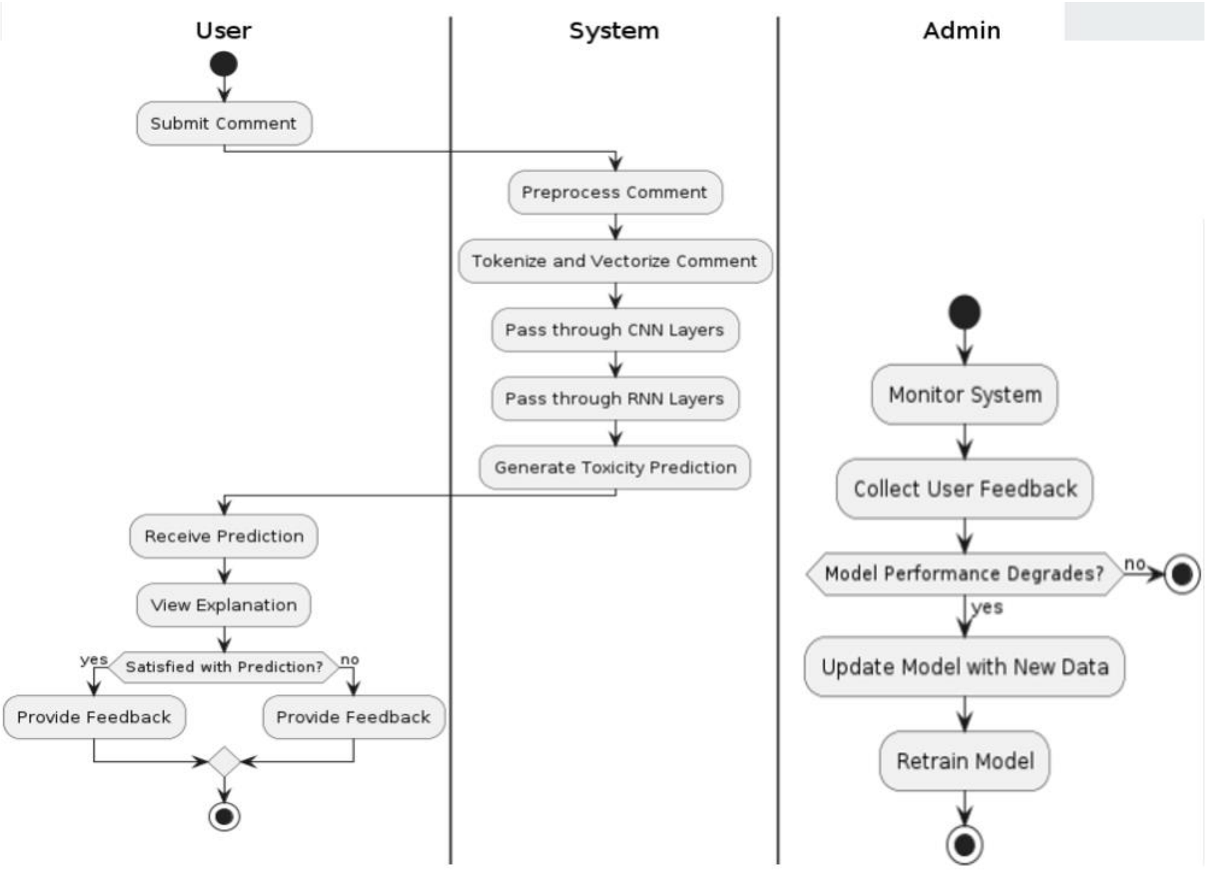


Figure 2 Activity Diagram

3.2.3 Entity-Relationship Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

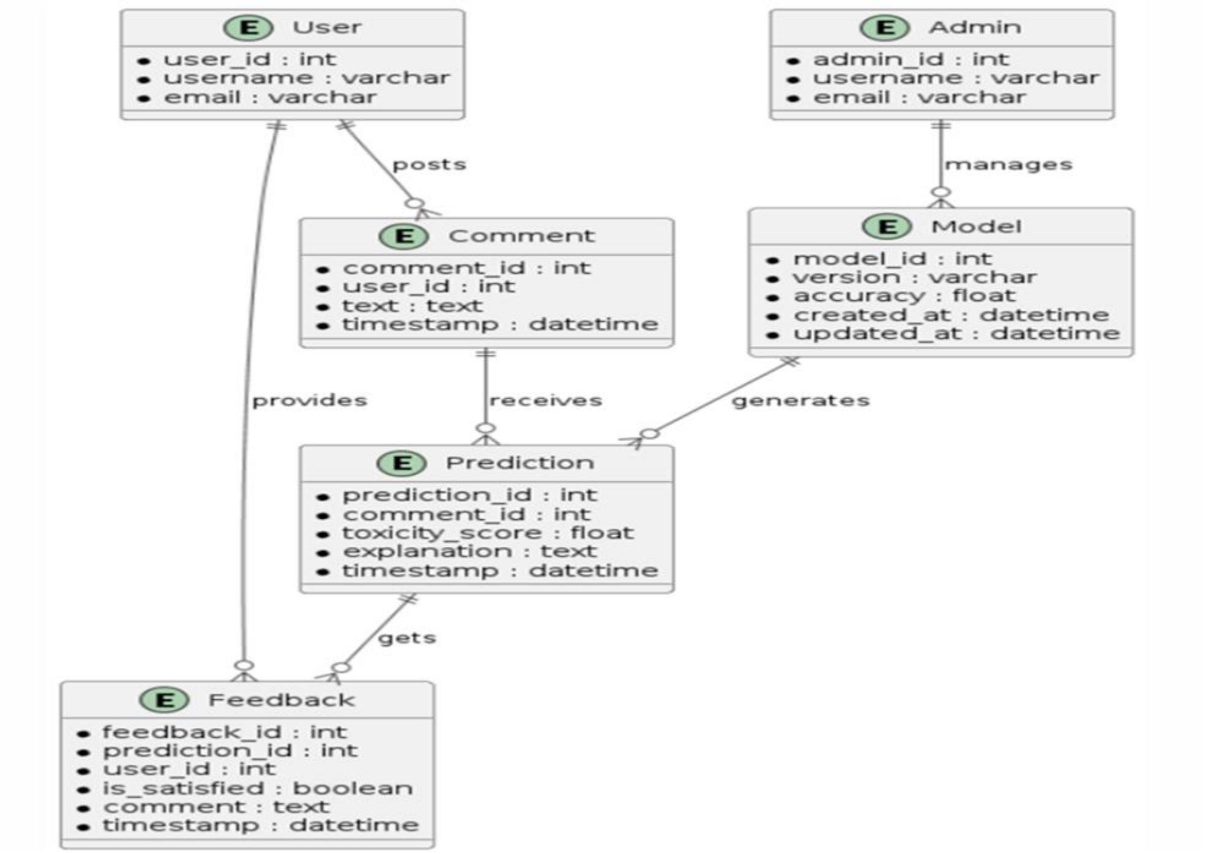


Figure 3Entity-Relationship Diagram

4 IMPLEMENTATION

4.1 MODULES

4.1.1 Dataset:

The dataset used for the toxic comment detection project comprises over 47,000 comments, each labeled based on its toxicity level. This dataset is structured to facilitate the training and evaluation of our models in identifying and categorizing toxic comments effectively.

Dataset Structure:

Comment Categories:

- **Gender**: Comments targeting individuals based on gender.
- **Ethnicity**: Comments targeting individuals based on ethnicity.
- **Age**: Comments targeting individuals based on age.
- **Religion**: Comments targeting individuals based on religion.
- **Other Cyberbullying**: Comments that do not fit the above categories but still involve bullying behavior.
- **Not Cyberbullying**: Comments that are neutral or non-toxic.

4.1.2 Models

A comparative analysis was conducted on several deep learning models to evaluate their performance in detecting toxic comments. The following models were used:

4.1.2.1 Deep Learning Network

4.1.2.1.1 RNN+LSTM

Combining Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks leverages the strengths of both architectures:

- **RNN Component:** Extracts spatial features from text data, identifying key patterns and phrases.
- **LSTM Component:** Captures temporal dependencies and contextual information from sequences of text.

4.1.3 Performance Metrics

Performance metrics provide a quantitative assessment of model effectiveness. Key metrics include accuracy, measuring overall correctness; precision, indicating true positive rate; recall, gauging the ability to identify positives; and F1 score, blending precision and recall.

- **Accuracy:** Measures the ratio of correctly predicted instances to the total instances.
  Formula: (TP + TN) / (TP + TN + FP + FN)
- **Precision:** Reflects the proportion of true positives among the predicted positive instances.
  Formula: TP / (TP + FP)
- **Recall** (Sensitivity or True Positive Rate): Represents the ratio of true positives among actual positive instances.
  Formula: TP / (TP + FN)
- **F1 Score:** Balances precision and recall, offering a harmonic mean of the two metrics. Formula: 2 * (Precision * Recall) / (Precision + Recall)

4.2 OVERVIEW TECHNOLOGY

4.2.1 Data Pre-Processing

Data pre-processing ensures data quality and relevance through cleaning, normalization, and handling missing values. This step is critical for providing consistent input to the models and improving prediction accuracy.

4.2.2 Data Splitting

The dataset is divided using an 80:20 split, allocating 80% for training and 20% for testing. This approach maintains balanced representation of classes and avoids bias, optimizing model performance.

- **Training Set**: Used for model learning and pattern recognition)
- **Testing Set**: Used to validate model generalization and accuracy.

4.2.3 Algorithm Selection

For effective toxic comment detection, a strategic selection of algorithms is used:

- **RNN+LSTM**: Combines spatial feature extraction with temporal context understanding, enhancing the model's ability to capture nuanced patterns.

4.2.4 Hyper parameter Tuning

Hyper parameters were tuned for each algorithm to optimize performance:

- **RNN+LSTM**: Tuned learning rate, batch size, number of epochs, and network architecture.

4.2.5 Model Testing

4.2.5.1 Hyper parameter-Informed Training:

Models were trained using hyper parameters optimized through extensive experimentation to ensure accurate toxic comment detection.

4.2.5.2 Optimal Configurations Empowerment

Utilized optimal hyper parameters for each model to maximize performance and efficiency.

4.2.5.3 Accurate and Generalized Performance

Models were evaluated on both training and testing data to assess generalization capabilities and predictive accuracy.

4.2.5.4 Mitigation of Over fitting

Regularization techniques and dropout were employed to reduce overfitting and ensure stable model performance.

4.2.5.5 Elevated Prediction Metrics

Testing results demonstrated improvements in accuracy, precision, and recall, reflecting the models' effectiveness in detecting toxic comments.

4.2.5.6 Pathway to Reliable Insights

Integration of well-tuned hyper parameters and thorough testing confirmed the robustness and reliability of the models, ensuring effective toxic comment detection.

5 TESTING

5.1 TEST CASES

Table 1 Test Cases

Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions
TC_QA_01	Detect Toxic Comments	Pre-process the data and predict	URLs must be accessible, models loaded	"You're such a waste of space. Just delete yourself already."*Type: Toxicity - Harassment*	Comment classified as toxic
TC_QA_02	Detect Bias in Comments	Pre-process the data and predict	URLs must be accessible, models loaded	"All [ethnic group] people are criminals."*Type: Toxicity - Ethnicity*	Bias identified in comment
TC_Cache_01	Identify Religious Toxicity	Pre-process the data and predict	Cache should be enabled	"Your religion is a joke. Nobody takes you seriously."*Type: Toxicity - Religion*	Comment classified as toxic
TC_Cache_02	Classify Age-Related Toxicity	Pre-process the data and predict	Cache should be enabled	"You're too old to understand anything new."*Type: Toxicity - Age*	Comment classified as toxic
TC_Encode_01	Detect Gender-Based Toxicity	Pre-process the data and predict	Model 'all-MiniLM-L6-v2' must be loaded	"Women should just stay in the kitchen."*Type: Toxicity - Gender*	Comment classified as toxic

5.2 TEST RESULTS

The culmination of our toxic comment detection model development involves a comprehensive testing and evaluation phase. This essential step assesses the predictive capabilities and performance of our deep learning model, ensuring its effectiveness in identifying toxic comments across different categories. Our evaluation metrics include:

5.2.1 Accuracy

Accuracy measures the proportion of correctly predicted instances across all categories. For toxic comment detection, a high accuracy rate indicates the model's effectiveness in correctly classifying comments into categories such as Gender, Ethnicity, Age, Religion, Other Cyberbullying, and Not Toxic.

- **Formula:** (TP + TN) / (TP + TN + FP + FN)
- **Accuracy of Model:**

Table 2 Accuracy of model

Model	Accuracy
RNN+LSTM	84.12

5.2.2 Recall / Sensitivity

Recall evaluates the model's ability to correctly identify all relevant instances of toxic comments. A higher recall value indicates that the model effectively captures true positive cases, minimizing the risk of false negatives.

- **Formula:** TP / (TP + FN)
- **Recall of Model:**

Table 3 Recall of model

Model	Recall
RNN+LSTM	83.75

5.2.3 Precision

Precision measures the model’s accuracy in classifying positive instances of toxicity. A high precision score indicates that the model has a low rate of false positives, meaning it accurately identifies instances of toxicity.

- **Formula:** TP / (TP + FP)
- **Precision of Model:**

Table 4 Precision of model

Model	Precision
RNN+LSTM	84.50

5.2.4 F1-Score

The F1-score provides a balanced measure of the model’s performance by harmonizing precision and recall. This metric is particularly useful in toxic comment detection to evaluate how well the model balances identifying toxic comments accurately and capturing all relevant cases.

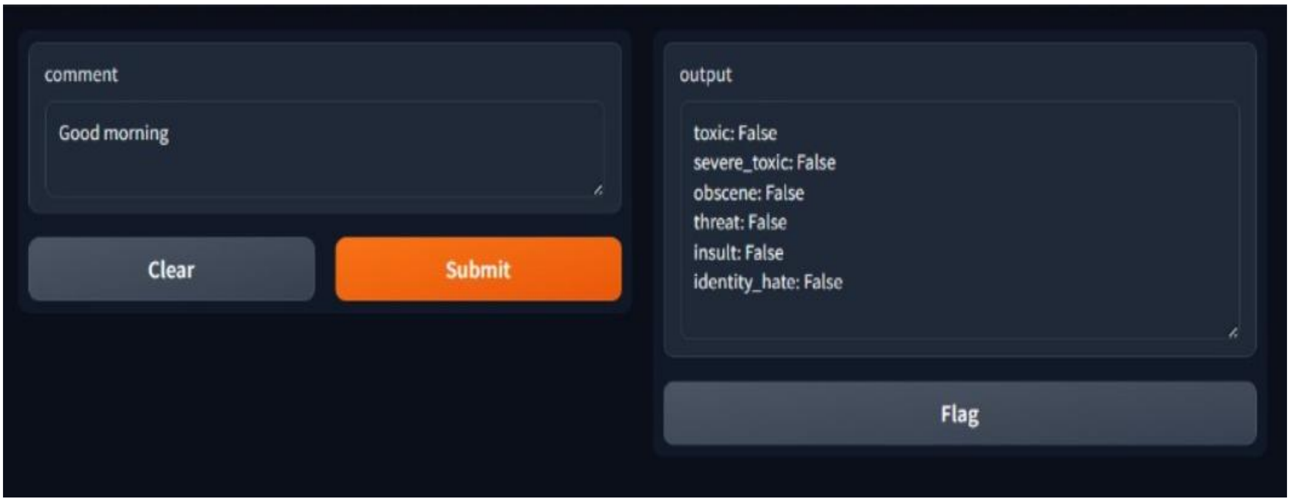
- **Formula:** 2 * (Precision * Recall) / (Precision + Recall)
- **F1-Score of Model:**

Table 5 F1 Score of model

Model	F1-Score
RNN+LSTM	84.12

6 RESULTS

Here’s model's prediction on the comment "Good morning" shows that it effectively distinguishes non-toxic content from harmful or offensive remarks. The output confirms that the comment is classified as "False" across all toxicity categories, including severe toxicity, obscenity, threats, insults, and identity hate. This indicates that the model is well-calibrated to recognize and appropriately categorize comments that pose no threat to online discourse. The accurate prediction reflects the robustness of our RNN/LSTM-based approach, ensuring that neutral and positive interactions are not mistakenly flagged, thereby supporting a positive user experience on the platform.



comment

Good morning

output

```
toxic: False
severe_toxic: False
obscene: False
threat: False
insult: False
identity_hate: False
```

Figure 4Results

7 CONCLUSION

The successful completion of our Comment Toxicity Prediction project is a testament to the meticulous planning, innovative methodologies, and collaborative efforts invested throughout the development process. By leveraging state-of-the-art machine learning techniques and carefully curated datasets, we built a robust system capable of accurately identifying and filtering toxic comments in real-time. The project not only achieved its technical objectives but also demonstrated scalability and adaptability, making it a valuable tool for enhancing online community interactions across various platforms.

Throughout the project, we gained deep insights into the complexities of natural language processing, particularly in handling diverse linguistic patterns and contextual subtleties. We learned the importance of balancing model accuracy with computational efficiency and the need for continuous learning and ethical considerations in deploying such systems. This journey has enriched our understanding of the challenges and opportunities in the field of toxicity prediction, equipping us with the knowledge and experience to tackle similar projects in the future.

8 FUTURE SCOPE

In addition to the insightful findings from our analysis of the datasets, there are several exciting avenues for future exploration and application in the field of medical data analysis.

8.1 Multilingual Support and Cross-Cultural Adaptability:

Expanding the project to support multiple languages and dialects is a crucial next step. By incorporating multilingual datasets and developing models that can accurately interpret context and nuances across different languages and cultures, the system can become more universally applicable. This will also involve addressing cultural sensitivities and ensuring that the toxicity detection is effective in diverse online environments.

8.2 Real-Time Moderation and Integration with Emerging Platforms:

Enhancing the system's ability to perform real-time moderation on various platforms, including social media, forums, and live-streaming services, is an important future direction. By optimizing the system for low-latency processing and integrating it with new and emerging online platforms, the project can contribute to maintaining safer digital spaces as the internet evolves.

8.3 Continuous Learning and Adaptation to Evolving Toxicity Trends:

Toxic language and behavior patterns evolve over time, often in response to changes in social norms and platform-specific cultures. Implementing a continuous learning mechanism that allows the system to adapt to these evolving trends is essential for maintaining its relevance and accuracy. Future work could focus on creating a feedback loop where the model regularly updates itself using new data, staying ahead of emerging toxicity trends

9 BIBILOGRAPHY

1. “Toxic Comment Classification” https://www.eecg.utoronto.ca/~jayar/ece324/2020/download/toxiccommentclassifier.pdf
2. "Deep learning for religious and continent-based toxic content detection and classification" https://www.nature.com/articles/s41598-022-22523-3
3. “Toxic Comment Detection in Online Discussions” https://link.springer.com/chapter/10.1007/978-981-15-1216-2_4
4. “An Assessment of Deep Learning Models and Word Embeddings for Toxicity Detection within Online Textual Comments” https://www.mdpi.com/2079-9292/10/7/779
5. “Identifying Toxicity Within YouTube Video Comment Text Data” https://wwhttps://docs.google.com/document/d/1WCoR0o5Bn3nqsOsthNjl2rVPCd4qAyyi/edit?usp=sharing&ouid=117766980951024595568&rtpof=true&sd=truew.researchgate.net/profile/Esther-Mead/publication/333830086_Identifying_Toxicity_Within_YouTube_Video_Comment/links/5f04f363a6fdcc4ca455bd2c/Identifying-Toxicity-Within-YouTube-Video-Comment.pdf
6. Understanding Toxicity Triggers on Reddit in the Context of Singapore https://ojs.aaai.org/index.php/ICWSM/article/view/19392/19164
7. Machine learning methods for toxic comment classification: a systematic review https://intapi.sciendo.com/pdf/10.2478/ausi-2020-0012
8. A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification https://pdfs.semanticscholar.org/0ce3/2b884a934a56c5862ff0bce3c4d94800956d.pd
9. https://towardsdatascience.com/toxic-comment-classification-using-lstm-and-lstm-cnn-db945d6b7986?gi=4d5b6f347e24
10. https://towardsdatascience.com/toxic-comment-classification-using-lstm-and-lstm-cnn-db945d6b7986?gi=4d5b6f347e24