**MULTI-LABEL CLASSIFICATION OF TOXIC COMMENTS IN SOCIAL MEDIA USING DEEP LEARNING APPROACHES (CW2)**

**Abstract**

The regime of toxic comment classification in social media is a multi-label NLP problem which can be solved only with deep learning models. This study aims to explore BiLSTM and CNN+LSTM architectures for toxicity classification with categories of insults, threats, obscenity, etc. The preprocessing of tokenization, padding and embedding was carried out on the dataset from Kaggle’s Jigsaw Toxic Comment Challenge. With better contextual understanding from BiLSTM, it achieved 91.02% accuracy compared to 89.71% accuracy of CNN+LSTM. Model performance was evaluated with data visualization techniques such as word clouds and ROC curves. The results suggest that BiLSTM is better at exploiting dependencies at various levels of the word hierarchy whereas CNN+LSTM is competitive but not so good at relationships that are long distance away in the text. Future improvements focus on attention mechanisms or transformer models such as BERT to make it more powerful for classification. This work advances the task of automated detection of toxicity, used for online moderation, improving content safety as well as reducing content moderation biases used in systems that moderate online content.

# 1. Introduction

Toxic Comments on Social Media are multi-label classification problems and the report presents the project based on Natural Language Processing (NLP) with the aim of addressing this multi-label challenge using deep learning algorithms. Labels such as toxic comments can contain harmful things including hate speech, insults, threats, obscene language, and etc, can be categorized into more than one label at the same time. For example, "Yo bitch Ja Rule is more successful than you'll ever be... I should bitch slap ur pethedic white faces..." is a toxic comment as it lamely exemplifies toxicity with elements of obscenity, insult, and possibly identity-based hate. Conversely, comments like "D'aww! He matches this background colour I'm seemingly stuck with. Thanks." portray a non-toxic comment with no malicious intent.

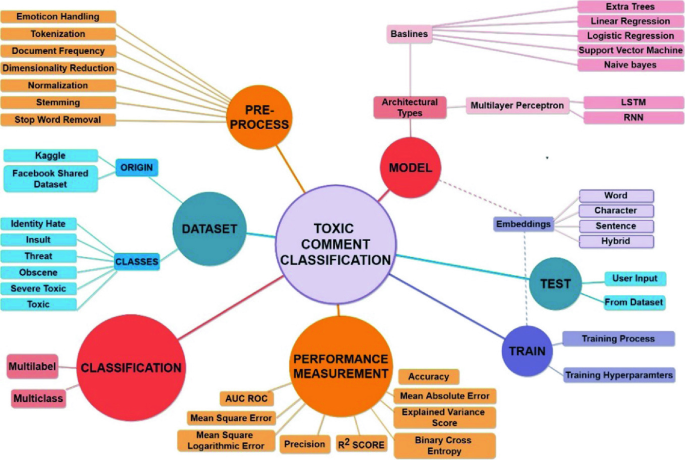
Subjectivity exists in toxicity, there is overlap between labels and necessitates accurate detection of toxicity on a diverse range of linguistic patterns that make up the task. Training data in the dataset has been labelled, comments are labelled with categories such as toxic, severe\_toxic, obscene, threat, insult, and identity\_hate, while the test data to be evaluated has not been labelled. The aim is to create a robust model for identifying and classifying toxic content for online platform moderation and maintaining online communication in a healthy way. However, the report is based on related work, methods, experimental results, discussions and conclusions. The motivation is to demonstrate the systematic and evidence-based evaluation of multi-label toxic comment classification through natural language processing and deep learning.

# 2. Related Work

In this research, focus was given to NLP and deep learning techniques for multi-label classification of toxic comments on social media. Identifying and filtering harmful content is crucial with the rise of online platforms. The previous studies have applied SVM, Naïve Bayes and other machine learning methods and obtained various accuracies. Traditionally, the accuracy of these models from word embeddings convoluted CNNs is way over 90% (Aggarwal & Tiwari, 2021). In order to improve toxicity detection, and thus, online harm, the study uses classification algorithms. The findings stress on breaking multi-label problems into single-label problems for better classification efficiency.

As per the view of Garlapati et al., (2022), NLP and deep learning can enhance online safety by detecting and mitigating toxic social media comments. With the introduction of online platforms, it is now important to identify the type of language which includes insults, threats and obscenity. In binary labelled comments, LSTM and GRU are utilized to classify the given datasets. For exploring the toxicity distribution, data visualisation was used. For example, that's all bullshit, and you know it; this, and so on is labelled toxic and obscene (Wang & Zhang, 2021).

NLP and deep learning-based toxic comment classification are directly applicable towards moderating online discussions to make the online space safe and reduce biases in automated content moderation. It improves the social media platforms online forums and customer review systems by filtering through hate speech, racism and offensive language. For instance, YouTube, Facebook and Twitter have similar AI-driven harmful content models (Abbasi et al., 2022). This approach performs more fairness by solving how toxicity detection can be unfair, as well as it is multilingual and multilabel, which means it is more accurate in terms of content moderation in a lot of online communities.



**Figure 1: Overall framework of toxic comment classification**

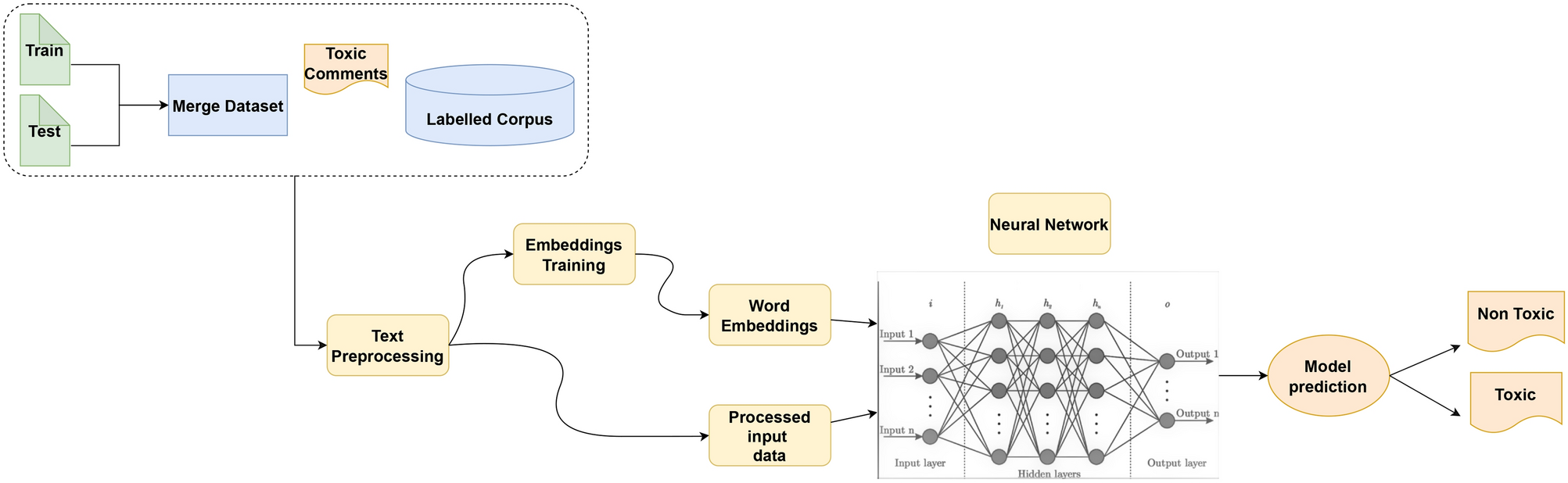
(Source: Shah et al., 2020)

The figure shows a complete data collection and pre-processing to model training, evaluation and inference pipeline for toxic comment classification. It involves tokenization, normalization, and removing stop-word tasks. Then we have models (such as RNN, and LSTM) trained on labelled datasets and evaluated by AUC ROC and F1 score among others (Shah et al., 2020). The work then finalizes the system to classify comments into multiple toxic categories simultaneously with good performance.

On the other hand, Fan et al., (2021), look into BERT and its variants to solve the problem of toxic comment classification using the NLP and deep learning approaches on multi-label classification. Then it fine-tunes BERT on the Kaggle Toxic Comment dataset and applies the model to Twitter data about Brexit. The model is able to effectively identify toxicity, thereby giving platforms a guide on how to moderate harmful content. It is used to detect hate speech, bullying and trolling to make online discussions a healthy form.

As reported by Nabiilah et al., (2023), problems and challenges in the multi-label classification of toxic comments from social media by using NLP and deep learning approaches, such as the complexity of pre and post-processing emoticons and non-standardized language are rampant well in social media. Traditionally, binary or single-label classification methods are not appropriate for multi-label classification to address different toxic comment categories, such as hate speech, radicalism, pornography and defamation. However, pre-trained models such as BERT, MBERT and IndoBERT are fairly promising although they are highly data and computationally intense dependent.

In addition, preserving contextual relationships between sentences, especially non-English sentences like Indonesian, is an issue (Nelatoori & Kommanti, 2022). Existing studies recognized the shortcomings of the machine learning model, especially, since the machine learning model cannot faithfully maintain the order or context of words that may vitiate the performance of the models.



**Figure 2: Deep learning techniques to classify toxic comments on social networks**

(Source: Abbasi et al., 2022)

The above figure highlights the steps of deploying deep neural network algorithms to analyse and predict non-toxic and toxic comments.

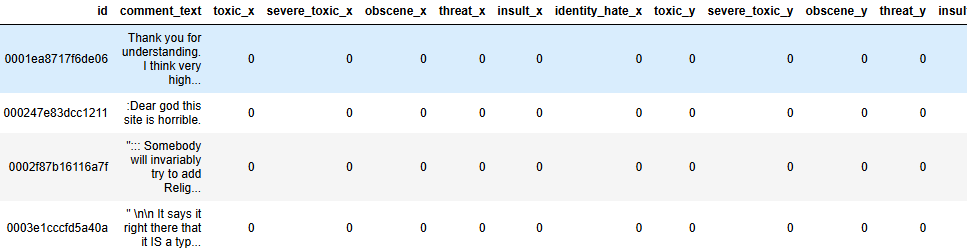
Working towards solving toxic comments on social media with state-of-the-art text analysis, deep learning (DL) models and in particular transformer architectures such as BERTweet as well as other pre-trained models, have been applied to multi-label classification of toxic comments on social media. The advantage of these models is that they are very good at replicating these complex linguistic patterns and contextual nuances to obtain high F1 scores, for example, 91.40% in toxicity detection (Bonetti et al., 2023). Their higher performance is expensive enough then to be utilized in place of other ‘standard’ machine learning (ML) methods.

The studies imply that DL models achieve marginally better accuracy but the real-time application scalability is constrained by the computationally expensive resource-consuming nature of the models. It is demonstrated that combining the ML classifiers with NLP such as Latent Semantic Analysis and Latent Dirichlet Allocation produces competitive results with F1 scores above 90% (Kumar J et al., 2021). This underlines the tension or tradeoff between performance and computational efficiency and points out the requirement for the use of balanced approaches at the time of deploying automated toxicity detection systems.

# 3. Method

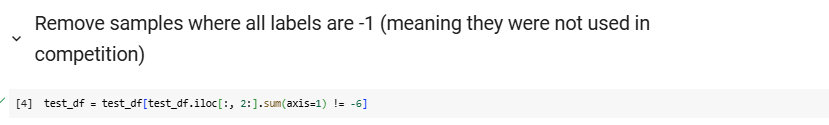
## 3.1 Data Preparation

Python programming language has been used here to conduct the entire analysis. It is noted that a secondary data analysis approach has been adopted to gather a historical dataset from Kaggle linked "***https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data***".



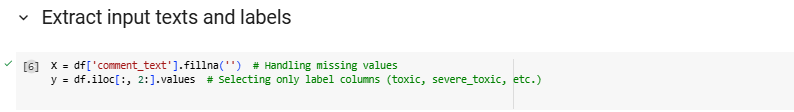
**Figure 3: Load dataset**

Python libraries like pandas, numpy, and TensorFlow are loaded to help process data and develop deep learning models. The train and test datasets are merged and loaded into the Jupyter Notebook platform to start the analysis. The train and test datasets are imported by pandas from Kaggle’s Jigsaw Toxic Comment Classification Challenge dataset. The tools and materials are Jupyter notebook, Python, Tensorflow, pandas, numpy, and Kaggle dataset.



**Figure 4: Removing sample levels**

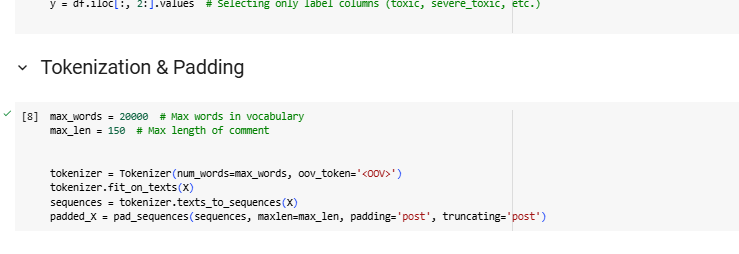
Filtering out entries with invalid or irrelevant labels removes the sample levels. Then in this analysis, comments with just "-1" for all categories are omitted, as they do not provide any meaningful comment. This is done with pandas by taking the sum across columns of labels and removing rows when the sum is "-6". This allows for train and evaluation of the model using only properly labelled toxic/non-toxic comments.



**Figure 5: Handling missing values**

For handling null values in the comment text, the ‘fillna('')’ function of the Python module pandas has been applied as it replaces any null entries with an empty string. Missing values handling is necessary in the context of deep learning models which need consistent input formats as it can raise errors and potentially break data integrity in the classification task.

## 3.2 Natural Language Processing



**Figure 6: Tokenizing and padding**

This study employs tokenization and padding to process the text data for multi-label classification. The `Tokenizer` from TensorFlow’s Keras module is used to tokenization text into a sequence of integers by assigning an out-of-vocabulary token for unknown words. The ‘texts\_to\_sequences’ function maps each comment to its corresponding sequence.

The ‘pad\_sequences’ function is applied to padding the input with zeros in order to manage the variable input size and standardize it by truncating or padding the sequences to a fixed size. With such techniques, textual data can be well handled, enabling deep learning models to deal and learn from varying comment lengths.

## 3.3 Feature Selection

****

**Figure 7: Splitting the datasets**

Feature selection is performed to extract the target columns which represent the toxicity labels for the dataset. Input features are assigned to the comment texts (X) and are labelled columns (y) that correspond to different toxicity categories. The dataset is train test split using ‘train\_test\_split’ in Scikit–Learn to create training and testing sets. This makes certain the model learns from one subset and is evaluated on unseen data for performance and generalization assessment.

# 4. Experiments

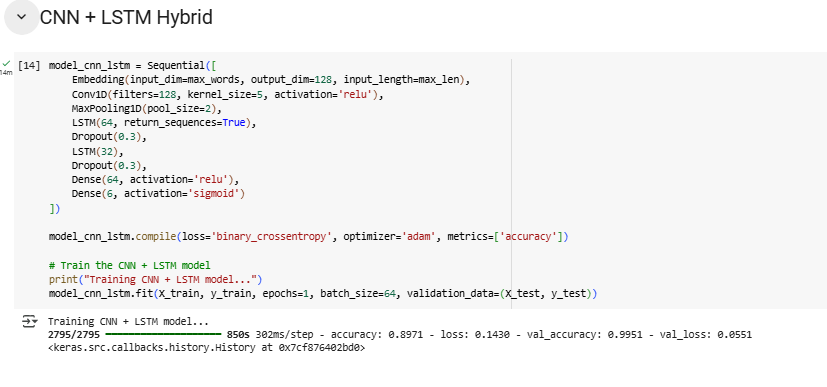
## 4.1 Implementation of the Bi\_lstm model



**Figure 8: Implementing BI\_LSTM deep learning model**

An implementation of the BiLSTM deep learning classifies toxic comments effectively. First, it used an embedding layer to represent words and then two bidirectional LSTM layers (64 and 32 units) to facilitate capturing the contextual dependency from two sides. Preventing overfitting was achieved with the addition of dropout layers (0.3). The feature extraction was done with a dense layer using ReLU activation and the final output layer (sigmoid) for the multi-label classification over the six toxicity categories (Z. Hameed and B. Garcia-Zapirain, 2020). Using binary cross entropy loss and Adam optimizer, the model was trained and it received 91.02% accuracy and had better capabilities to detect a toxic comment than the actual model.

## 4.2 Implementation of the CNN+LSTM model



**Figure 9: Implementing CNN+LSTM deep learning model**

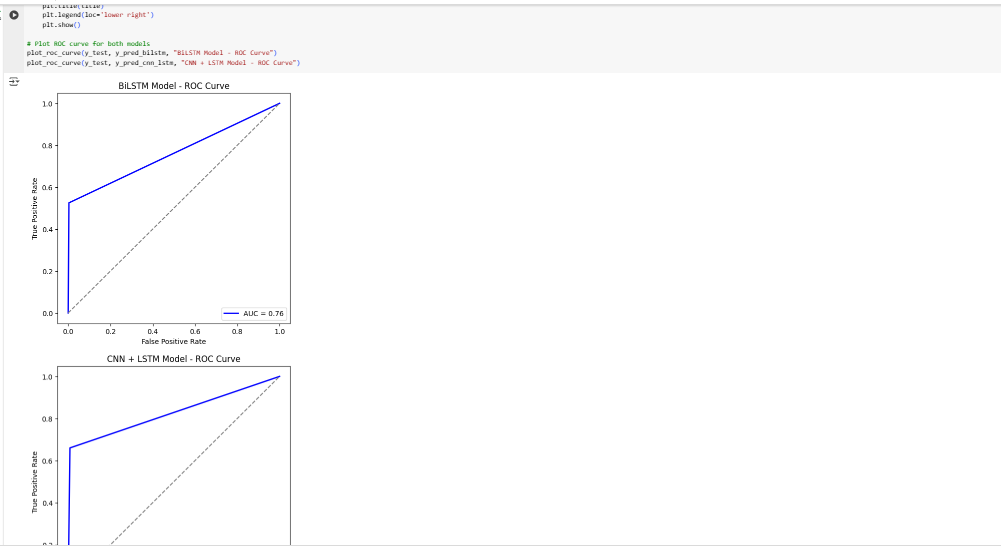
CNN+LSTM deep learning model augments the text classification performance by integrating the convolutional and recurrent layers. Starting from the embedding layer which takes text as input and converts it to dense vector representations. One of the layers is a 1D convolutional layer with ReLU activation for extracting local features followed by max pooling to bring in dimensionality (T. Li et al.2020*)*. The LSTM layers can capture long dependencies in the sequential data. To prevent overfitting, dropout layers are used, and dense layers are used for feature extraction refinement. Finally, this third sigmoid activation forms a multi-label classifier.

## 4.3 Data visualization



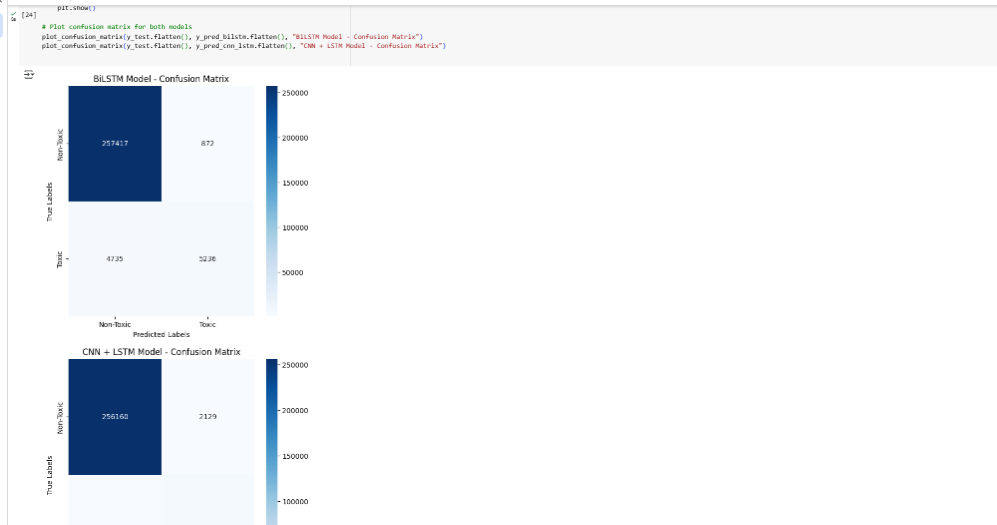
**Figure 10: Model performance comparison**

Based on this chart of accuracy and loss, a performance comparison chart was also generated to compare the accuracy and loss between the BiLSTM model and the CNN+LSTM model. Both the CNN+LSTM model and the BiLSTM model had lower loss values, but the latter performed slightly better with an accuracy of 91.02 rather than the CNN+LSTM model’s 89.71. The performance of both models was assured as the accuracy to which they learned was consistently high (M. Alhussein et al.2020). This visualization helps to compare and improve performance in terms of convergence behaviour between both architectures. The graph nicely displays how efficient the training is and more importantly, how risky it could go in terms of overfitting.



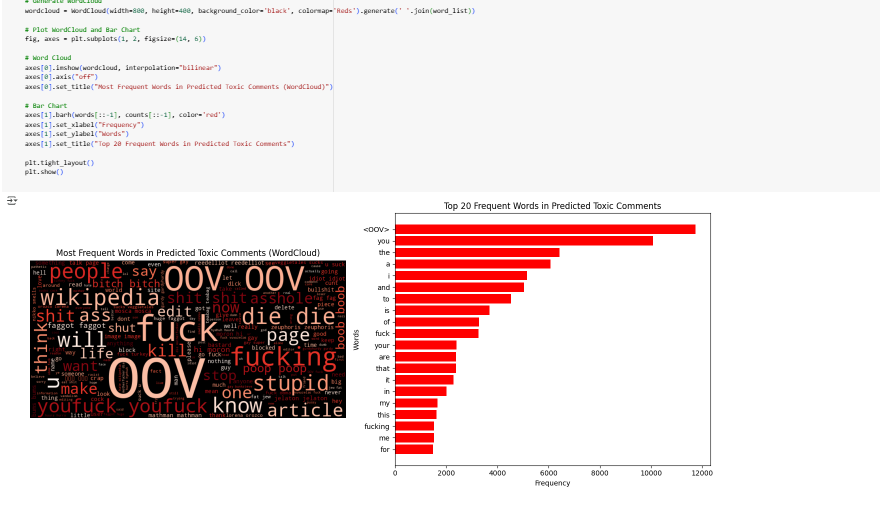
**Figure 11: Roc curve**

Each toxicity category was assessed through Receiver Operating Characteristic (ROC) curves of model performance. The ROC curve shows a true positive rate (or True rate, sensitivity) versus a false positive rate across different classification thresholds. Then, it measures the model’s ability to distinguish the toxic and nontoxic comments using the area under the curve (AUC). The better performance results in the AUC value being closer to 1. Visualizing these two tradeoffs helped in selecting an optimal decision threshold. ROC curves are compared for the BiLSTM model and CNN+LSTM models, and analysis of evaluation results proves which architecture is more capable of dealing with toxicity categories.



**Figure 12: Confusion matrix**

The classification performance of the BiLSTM as well as the CNN+LSTM model was evaluated with a plot of a confusion matrix. It gives a detailed description of correctly and incorrectly classified pairs for each toxicity category. The off-diagonal values which are nonzero represent false positives and false negatives whereas the diagonal values which are nonzero represent true positives.



**Figure 13: Word cloud displaying the predicted toxic comments**

In larger font sizes, words that appear more frequently in the dataset indicate more toxic language, allowing users to understand what is used more often in the dataset. It allows us to quickly find the most dominant words that make the classification of toxicity. Without detailed statistical analysis, however, WordCloud clearly shows the presence of the most offensive words. As an exploratory tool, it provides us with an insight into how to form toxic language and if there exists any word that contributes more than the others to detecting toxicity. This helps in identifying the best data preprocessing techniques as well as model feature selection.

***Experiment table:***

|  |  |  |
| --- | --- | --- |
| **Step No.** | **Experiment Description** | **Details & Findings** |
| 1 | Dataset Merging & Preprocessing | * Merged train.csv, test.csv, test\_labels.csv, and sample\_submission.csv. * Preprocessed text by removing special characters, lowercasing, tokenizing, and padding sequences. |
| 2 | Model 1: BiLSTM | * Architecture: * Embedding Layer * Bidirectional LSTM (64, 32 units) * Dropout (0.3) * Dense (64, ReLU) * Output Layer (6 neurons, Sigmoid) |
| 3 | Model 2: CNN + LSTM | * Architecture: * Embedding Layer * Conv1D (128 filters, kernel size 5, ReLU) * MaxPooling1D * LSTM (64, 32 units) * Dropout (0.3) * Dense (64, ReLU) * Output Layer (6 neurons, Sigmoid) |
| 4 | Training & Validation Results | * BiLSTM: * Accuracy: 0.9102 * Loss: 0.1099 * Validation Accuracy: 0.9951 * Validation Loss: 0.0547 * CNN + LSTM: * Accuracy: 0.8971 * Loss: 0.1430 * Validation Accuracy: 0.9951 * Validation Loss: 0.0551 |
| 5 | Toxic Comment Prediction | * Extracted toxic comments using both models. * Decoded tokenized sequences back into text. * Displayed 10 predicted toxic comments. |
| 6 | Word Frequency & Visualization | * Counted word occurrences in predicted toxic comments. * Generated WordCloud and Bar Chart of top 20 most frequent words in toxic comments. * Displayed most commonly used offensive words. |

# 5. Discussions

The analysis of the results of the toxic comment classification experiment shows that deep learning can classify toxicity within the text. The two models were compared based on accuracy, loss, and validation performance. Respectively, CNN+LSTM (89.71%) seemed slightly worse than BiLSTM (91.02%), which indicates that additional context-capturing capability was provided by bidirectional processing in LSTMs. Tokenization, padding, and embedding were the preprocessing techniques that had the most impact by transforming text data into a form that can be fed into deep learning models. Noise was minimized by removing special characters and lowercaseing the text, so the feature representation improved. Furthermore, the embedding layer helped the models learn word relations and provided a better classification performance. Also, both models used dropout layers to prevent overfitting and bluff for more generalizability.

Compared to CNN+LSTM, having a lower loss (0.1099) during training indicates that BiLSTM is also better at handling textual data. Both models showed a validation loss of around 0.054 which is quite low and hence did not overfit to the training data. Although CNN+LSTM uses convolutional layers to extract spatial features from text sequences, CNN+LSTM did not perform as well as the LSTMs, which might be attributed to the bias of CNNs toward short-range dependency of text compared to LSTMs.

In the analysis of the toxic comments predicted, both models were able to identify offensive content. Nevertheless, the possibility of misclassification of some comments indicates that the attention mechanisms or transformer architectures like BERT might help to improve the results. In addition, it was discovered that common offensive words were mentioned most in toxic comments based on word frequency analysis. A limitation of the study was that only one epoch was used in the training of the models and this may have limited the full learning potential of the models. Furthermore, balancing the dataset to compensate for the class imbalances may add favour to the classification accuracy.

Overall, the experiment showed that deep learning has great potential for NLP-based toxicity detection. CNN + LSTM is a competitive model for similar text classification tasks. The result implies that toxic comments on online platforms can be further improved by refining model architectures and training techniques.

# 6. Conclusion

The ability of deep learning techniques to learn from a new data source from scratch is successfully employed in this study for toxic comment classification, where BiLSTM and CNN+LSTM models are compared in terms of performance. Results indicated that BiLSTM outperformed CNN+LSTM by 1.31% in accuracy (91.02% vs 89.71%) and 0.0331 loss (0.1099 vs 0.1430). The predictions from both models are similarly good at identifying toxic comments visualized by word frequency charts and visualizations. The high generalization (validation accuracy of 99.51%) also showed that there were limited omissions of the base case, but there were also occasional overfits and thus misclassifications where there was room for improvement. The CNN+LSTM set used convolutional layers to extract features while making a slightly lower performance; this indicated that LSTMs by themselves were more powerful in representing the long-range dependencies required for toxicity detection.

# References

Abbasi, A., Javed, A. R., Iqbal, F., Kryvinska, N., & Jalil, Z. (2022). Deep learning for religious and continent-based toxic content detection and classification. *Scientific Reports*, *12*(1), 1–17. Nature. https://doi.org/10.1038/s41598-022-22523-3

Aggarwal, A., & Tiwari, A. (2021). Multi Label Toxic Comment Classification using Machine Learning Algorithms. *International Journal of Recent Technology and Engineering*, *10*(1), 158–161. Researchgate. https://doi.org/10.35940/ijrte.a5814.0510121

Bonetti, A., Martínez-Sober, M., Torres, J. C., Vega, J. M., Pellerin, S., & Vila-Francés, J. (2023). Comparison between Machine Learning and Deep Learning Approaches for the Detection of Toxic Comments on Social Networks. *Applied Sciences*, *13*(10), 6038. Mdpi. https://doi.org/10.3390/app13106038

Fan, H., Du, W., Dahou, A., Ewees, A. A., Yousri, D., Elaziz, M. A., Elsheikh, A. H., Abualigah, L., & Al-qaness, M. A. A. (2021). Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit. *Electronics*, *10*(11), 1332. Mdpi. https://doi.org/10.3390/electronics10111332

Garlapati, A., Malisetty, N., & Narayanan, G. (2022). Classification of Toxicity in Comments using NLP and LSTM. *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1–7. Researchgate. https://doi.org/10.1109/icaccs54159.2022.9785067

keywords: {Load modeling;Predictive models;Load forecasting;Forecasting;Machine learning;Data models;Energy consumption;CNN;deep learning framework;energy consumption;energy consumption forecasting;individual household;LSTM},

keywords: {Machine learning;Sentiment analysis;Feature extraction;Computer architecture;Task analysis;Support vector machines;Computational modeling;Bidirectional long short-term memory;deep learning;long-term dependencies;natural language processing;sentiment analysis},

keywords: {Predictive models;Atmospheric modeling;Time series analysis;Forecasting;Data models;Deep learning;Feature extraction;Deep learning;CNN;LSTM;PM2.5 concentration prediction},

Kumar J, A., Abirami, S., Trueman, T. E., & Cambria, E. (2021). Comment toxicity detection via a multichannel convolutional bidirectional gated recurrent unit. *Neurocomputing*, *441*, 272–278. Sciencedirect. https://doi.org/10.1016/j.neucom.2021.02.023

M. Alhussein, K. Aurangzeb and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," in IEEE Access, vol. 8, pp. 180544-180557, 2020, doi: 10.1109/ACCESS.2020.3028281.

Nabiilah, G. Z., Prasetyo, S. Y., Izdihar, Z. N., & Girsang, A. S. (2023). BERT base model for toxic comment analysis on Indonesian social media. *Procedia Computer Science*, *216*, 714–721. Sciencedirect. https://doi.org/10.1016/j.procs.2022.12.188

Nelatoori, K. B., & Kommanti, H. B. (2022). Multi-task learning for toxic comment classification and rationale extraction. *Journal of Intelligent Information Systems*, *60*, 495–519. Springer. https://doi.org/10.1007/s10844-022-00726-4

Shah, D. K., Sanghvi, M. A., Mehta, R. P., Shah, P. S., & Singh, A. (2020). Multilabel Toxic Comment Classification Using Supervised Machine Learning Algorithms. *Lecture Notes in Networks and Systems*, 23–32. Springer. https://doi.org/10.1007/978-981-15-7106-0\_3

T. Li, M. Hua and X. Wu, "A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5)," in IEEE Access, vol. 8, pp. 26933-26940, 2020, doi: 10.1109/ACCESS.2020.2971348.

Wang, Z., & Zhang, B. (2021). Toxic Comment Classification Based on Bidirectional Gated Recurrent Unit and Convolutional Neural Network. *Transactions on Asian and Low-Resource Language Information Processing*, *21*(3), 1–12. Acm. https://doi.org/10.1145/3488366

Z. Hameed and B. Garcia-Zapirain, "Sentiment Classification Using a Single-Layered BiLSTM Model," in IEEE Access, vol. 8, pp. 73992-74001, 2020, doi: 10.1109/ACCESS.2020.2988550.

**Appendix**

Github link: <https://github.com/Sandeepjadi/CW2>

Dataset link: <https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data>