CAB420 Assignment 2

Toxic Comment Classification for social media

Group Number: 60

#1	Fu-Cheng Tsai	10297171
#2	Oscar Chandra	7227116
#3	Ebenezer Antwi-Brefo	11504862

1.0 Introduction

Online platforms have become integral to our daily lives, providing us with opportunities to connect, share ideas, and engage in discussions with people from all over the world. As referenced in this article, research conducted by Global WebIndex indicates that social media is utilized by 60% of the global population. The average daily usage currently stands at precisely 2 hours and 24 minutes (April 2023). Nevertheless, within this vast realm of online interactions, a significant challenge emerges, posing a threat to the well-being and inclusivity of these virtual communities: toxic comments.

In the book "The Dark Side of Social Media: Psychological, Managerial, and Societal Perspectives" by Sheldon, Rauschnabel, and Honeycutt, the authors delve into the multifaceted consequences of negative behaviors and toxic interactions within online environments.This book highlights the urgent need to address the dark side of social media, offering insights into the psychological processes underlying toxic behavior, the managerial challenges faced by organizations in managing online communities, and the broader societal consequences. The groundbreaking study conducted by Mark Becker of Michigan State University in 2018 reveals compelling statistics, indicating that individuals who engage with social media reported an alarming 70% increase in depressive symptoms, accompanied by a notable 42% rise in social anxiety. The evidence strongly suggests that excessive involvement with social media exposes individuals to a heightened risk of experiencing depression, anxiety, and ultimately, stress.

As a result, a question arose “How can we effectively classify toxic comments in online platforms using machine learning techniques? to help the social media company can automatically delete the toxic comment when identified ". In order to create a safer online environment is not only essential for the well-being of individuals but also crucial for the educational and academic realms. The paper by Hinduja and Patchin (2012) highlights how online harassment can negatively impact academic performance, hindering learning opportunities and impeding the educational journey of students. This underscores the need for educational institutions and online platforms to proactively address and mitigate the effects of toxic comments to foster a conducive learning environment.

This research project will build upon existing research and analyze the Kaggle toxic comment classification challenge. The Kaggle challenge (2018) centers around the identification of toxic comments in online platforms. The Conversation AI team, a research initiative founded by Jigsaw and Google, aims to improve online conversations by studying negative online behaviors, including toxic comments. While the AI team have developed publicly available models, such as the Perspective API, further advancements are needed. Current models still make errors and lack the ability to filter different types of toxicity based on platform-specific requirements .

In this competition, the objective is to develop a multi-headed model capable of detecting various types of toxicity, such as threats, obscenity, insults, and identity-based hate, outperforming the Perspective API's existing models. Consists of comments from Wikipedia's talk page edits. Improvements to the current model will contribute to more productive and respectful online discussions.

In summary, the goal of this research project is to develop a classification model that accurately identifies toxic comments and enables social media platforms to remove such content. This report aims to present the motivation behind the project, outline the research question, and establish its connection to previous works conducted in this field. By addressing the issue of toxic comments, and aspire to create a safer or more inclusive online environment for all users.

2.0 Related Work

Text classification is a common machine-learning task. These three models are also the models we have chosen to perform this text classification task. The existing approach in Kaggle is an LSTM method, we did the improvement in the model architecture by using different variations of the model architecture such as dropout rate, and units in the LSTM layer. We also did a GRU and random forest version to compare and contrast our results. More detail can be found in the methodology segment.

Random forest algorithms are a common choice for text classification tasks. In the research paper ‘A comparative study of automated legal text classification using random forests and deep learning’, random forest algorithms have the best performance out of 179 classifiers with 121 data sets (Fernández-Delgado et al., 2014). Random forests are also great at text classification tasks because it is good at managing issues involved in text data such as high dimensionality, sparsity, and noisy feature space (Islam et al., 2019). From these findings, we have chosen random-forest to be one of the methods used for this text classification task.

An LSTM and GRU approach is also suitable for text classification tasks. The main advantage of LSTM and GRU models is the ability to handle sequential information. They are also able to perform well in various text classification tasks like sentiment analysis and document classification (Kim, Y et al., 2014).

LSTM was also used in a similar project. In Magzoub’s research paper titled ‘Toxic Comment Classification In Discord’, Magzoub found that the LSTM method for a binary classification task achieved 91% accuracy from a data set containing 159,571 training samples and 63978 testing samples for classifying toxic comments (Magzoub, 2023).

The objective of our work is to leverage these proven-effective methods, such as LSTM, GRU, and random forest algorithms, for text classification tasks. By exploring different variations of the model architectures and comparing their results, we aim to improve upon existing approaches and provide insights into their strengths and weaknesses. Additionally, we seek to contribute to the field by evaluating these methods in the context of the specific text classification task at hand.

3.0 Data Pre- Processing

The dataset that is used is from Kaggle. It contains many comments from Wikipedia which have been labelled with different types of toxic behaviour. The data provided has over 150,000 comments, with 7 columns, ‘comments’, ‘toxic’, ‘severe toxic’, ‘obscene’, ‘threat’, ‘insult’ and ‘identity hate’. With the data being so large, a random sample of 80,000 comments is taken having the comments column as the data to be learned and the other six columns as the prediction value with the task being to predict the probability of each toxic behaviour.

The data is then split accordingly, 60% for train data, 20% for validation data and 20% for test data. As the comments had characters in it that are not needed for the task, processing was done to have these removed and convert all the comments into lower case. Initially, the input data is preprocessed to prepare it for the model. This preprocessing includes tokenization of the text comments, where they are split into individual words or tokens. Subsequently, the tokens are converted into integer indices and padded to ensure uniform sequence length across all inputs.The comments were then made numeric as it is wanted in sequences for when the group train each of the models. Looking at how the classification is distributed and need column is created to separate the clean and toxic comments. Figure 1 displays a highly imbalanced dataset, with a significant disparity in the distribution of clean and non-clean comments.

This presents a challenge that we must consider when training our learning algorithms. It is important to note that even if 'clean' is predicted for every comment, the accuracy results may not be as poor as expected due to this imbalance.

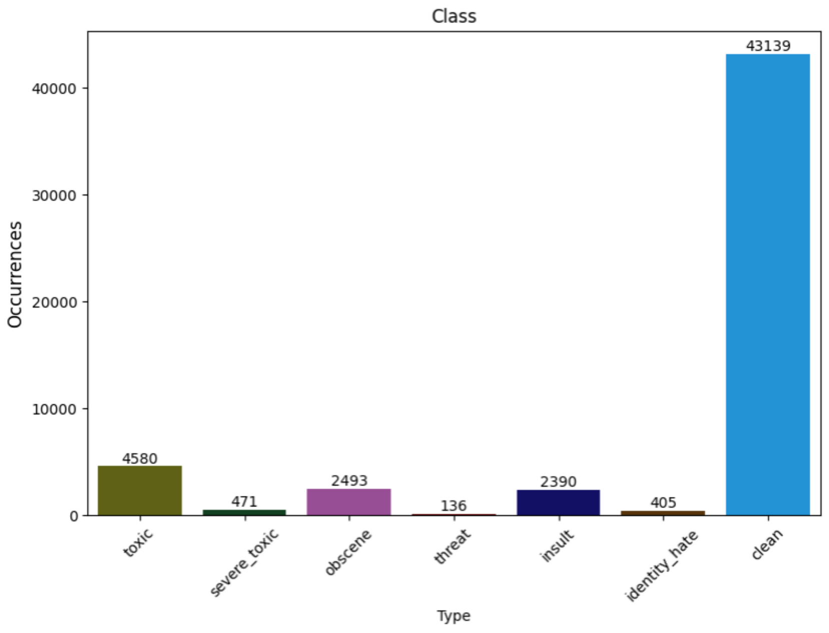


Figure 1: Graph showing the distribution between toxic and clean comments.

4. 0 Methodology

4.1 LSTM

The one of the models employed in this study involves several key steps to train a model for multi-label text classification. The model architecture utilized in this study is based on a recurrent neural network (RNN) with LSTM (Long Short-Term Memory) cells. The LSTM cells are well-suited for capturing long-range dependencies within sequential data, making them particularly effective for processing text sequences. This architecture enables the model to effectively learn and comprehend the underlying patterns and relationships within the input text.

Within the model architecture, an embedding layer is incorporated to learn and assign dense vector representations, known as embeddings, to each word in the input vocabulary. These embeddings encode semantic relationships between words, aiding the model in understanding the meaning and context of the text data.

Furthermore, the LSTM layer with 64 units is employed to process the embedded word sequences. The LSTM layer is designed to capture important temporal patterns and dependencies within the text data. It effectively learns to extract relevant features from the sequential representation of the input.

To prevent overfitting, dropout layers with a dropout rate of 0.5 are incorporated after the GlobalMaxPool1D layer and after the dense layers. Dropout randomly sets a fraction of input units to zero during training, reducing over-reliance on specific features and promoting the learning of more robust representations.

Following the dropout layers, two fully connected dense layers with 32 and 16 units, respectively, are introduced. These layers apply non-linear transformations to the learned features, enabling the model to extract higher-level representations that capture the complex relationships within the text data.

The final dense layer with 6 units and a sigmoid activation function generates the output probabilities for each class. By employing the sigmoid activation function, the model provides a probability value between 0 and 1 for each class, indicating the likelihood of the input comment belonging to that particular class.

The model is compiled using the Adam optimizer, which is a commonly used optimizer for training neural networks. The binary cross-entropy loss function is utilized, as it is well-suited for multi-label classification tasks. The model's performance is evaluated using accuracy as the metric, allowing for the monitoring of the model's predictive performance throughout the training process. It is trained on 5 epochs with a batch size of 128 the model structure shown in Appendix A.

4.2 GRU

GRU (Gated Recurrent Unit) is a type of recurrent neural network (RNN) architecture that is commonly used for sequence data processing tasks, such as natural language processing (NLP). In the context of analyzing toxic comments or text classification, GRU can be a suitable choice for several reasons:

1. Handling Sequential Data: Toxic comments in social media often contain dependencies and relationships between words and phrases that appear in a specific order. GRU, being a recurrent architecture, is designed to capture and remember information from previous steps in the sequence. This allows the model to consider the contextual information of the comment and better understand the meaning of the text.

2. Dealing with Variable-Length Inputs: The comments in the dataset are vary in length. GRU is capable of handling variable-length inputs by processing them sequentially, one step at a time. This flexibility makes it suitable for modeling toxic comments of different lengths without the need for padding or truncation.

3. Efficient Computation: Compared to other RNN architectures like LSTM (Long Short-Term Memory), GRU has a simpler structure with fewer parameters, making it computationally efficient. This can be beneficial when training on large datasets or when computational resources are limited.

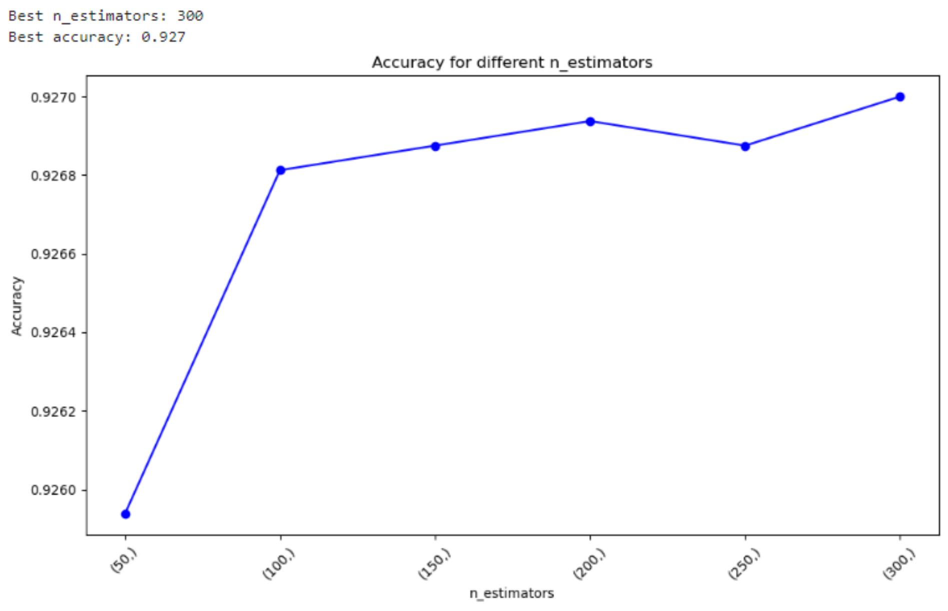
The GRU (Gated Recurrent Unit) model was specifically designed to be similar to LSTM (Long Short-Term Memory) in terms of its model architecture (as shown in Appendix B). Both GRU and LSTM are recurrent neural network (RNN) variants that excel at capturing sequential dependencies in data. By adopting a similar model architecture to LSTM, the intention was to compare and analyze the performance of these two models.

The purpose of designing the GRU model with a similar architecture to LSTM was to investigate which model performs better on a given task, such as analyzing toxic comments. By conducting a comparative evaluation, researchers can gain insights into the strengths and weaknesses of each model and determine which one is more suitable for the specific task at hand.

4.3 Random Forest

The third model used for this binary classification task is a random-forest classifier. The random forest is a prominent technique for handling imbalanced data and performs better than other machine learning models due to its parallel architecture (Jalal et al., 2022). The metrics used to determine the quality of this model is the accuracy and F1-score. Accuracy is how well the model correctly classifies the given data-set, F1-score is a combination of the precision and recall scores.

To find the best accuracy, a for-loop was used to find the n_estimators that return the best accuracy. The n-estimators is the number of trees that will be generated in the model. In this model, the n_estimator value is a list of numbers [50, 100, 150, 200, 250, 300]. By executing the code, we find that the higher the value, the better the accuracy. We are assuming that a better accuracy score can be achieved by continuing to increase the n_estimator value, but this results in much longer processing times. The execution of the code looping over the list of n_estimator values is around 4 minutes. This is a massive drawback of the random forest model. To find the best accuracy using this approach, more time is required to compute the large n_estimator values that are greater than 300. For the purpose of this task, we will use 300 as the value for n_estimator.



```
Best n_estimators: 300
Best accuracy: 0.927
```

Figure 2: Iterations over n_estimators

A similar approach has been used to tune the max_depth parameter. By using the ‘best’ n_estimator (in this case 300), a for-loop is executed (include code reference) over a list of max_depth values (Figure 2). The values used in this case are 5, 10, 15, and 20. However, by looking at the plot, we find no changes in accuracy with different max_depth values (Figure 3). We decided to use the default setting which evaluates to ‘None’ max depths. By using ‘None’ max_depths, there is no limit to the maximum number of trees generated.

```
Best n_estimators: 300
Best max_depth: None
Best accuracy: 0.927
```

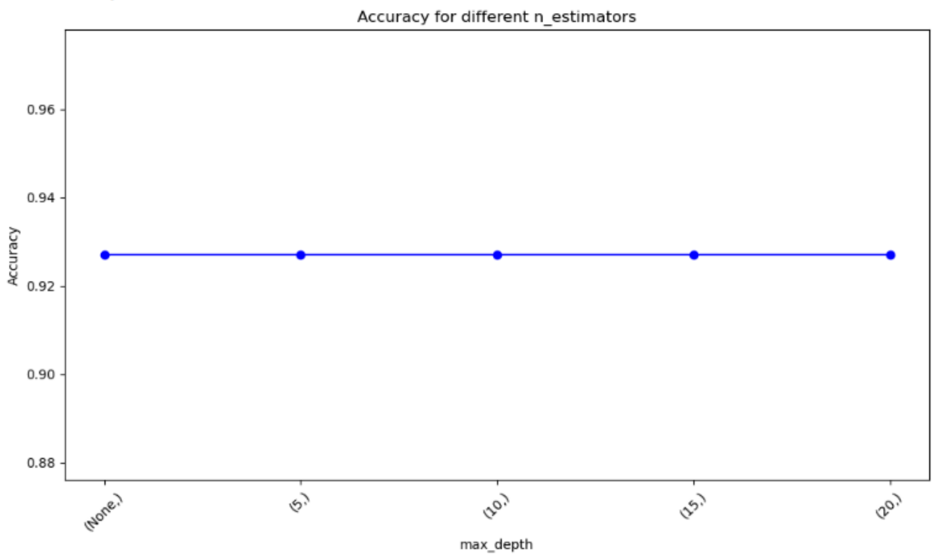


Figure 3: Iterations over max_depth

The parameters that will be used for this model will be an n_estimator of 300, and a max_depth of ‘None’.

5.0 Evaluation and Discussion

5.1 LSTM

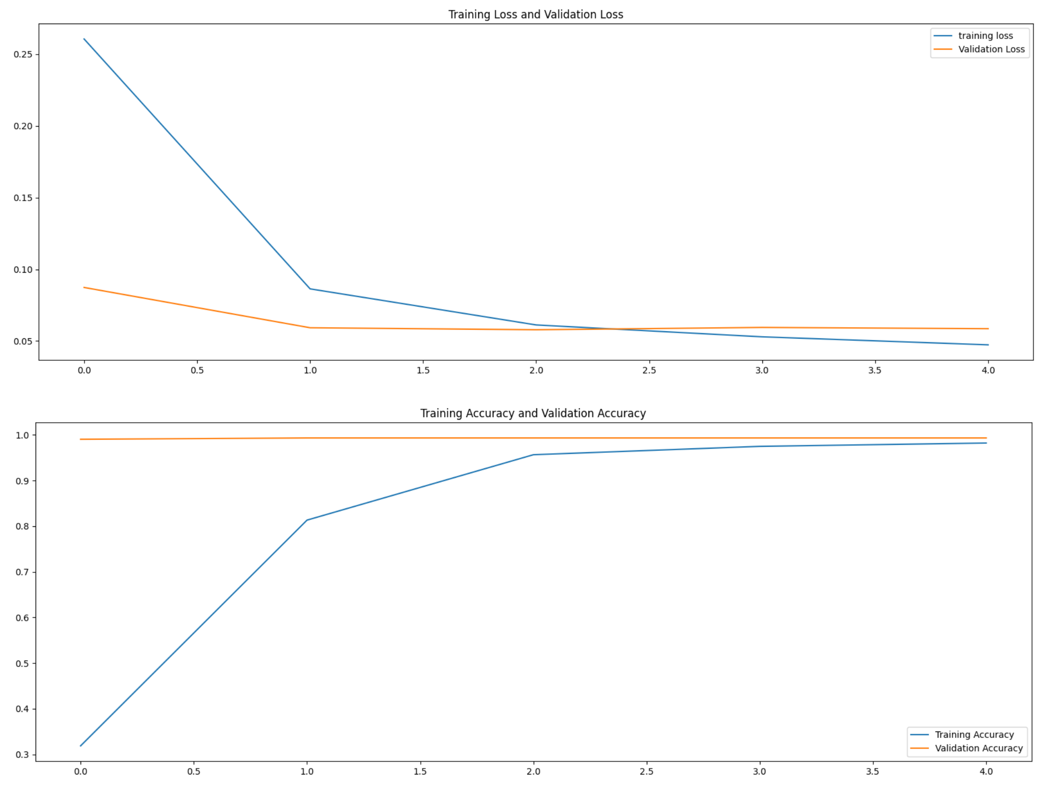


Figure 4: Model training performance, with loss(loss) and accuracy (bottom). Training converges after 2 epochs.

After training the model it was evaluated and as shown in figure 4 the LSTM network does fairly well will such a low loss rate however overfitting does occur due to the validation data's accuracy being almost perfect however this is what is expected due to the vast distribution of the data.

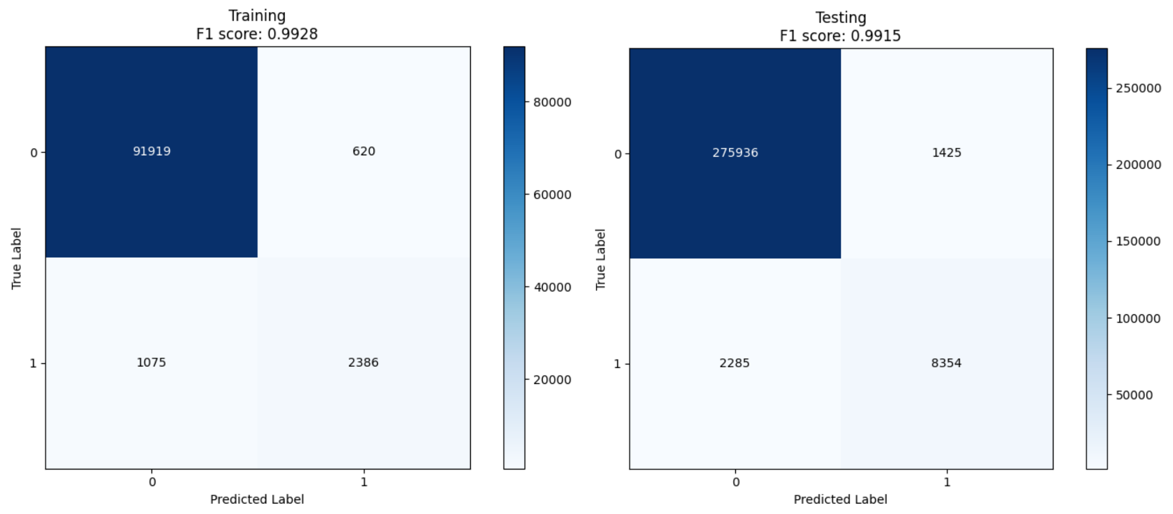
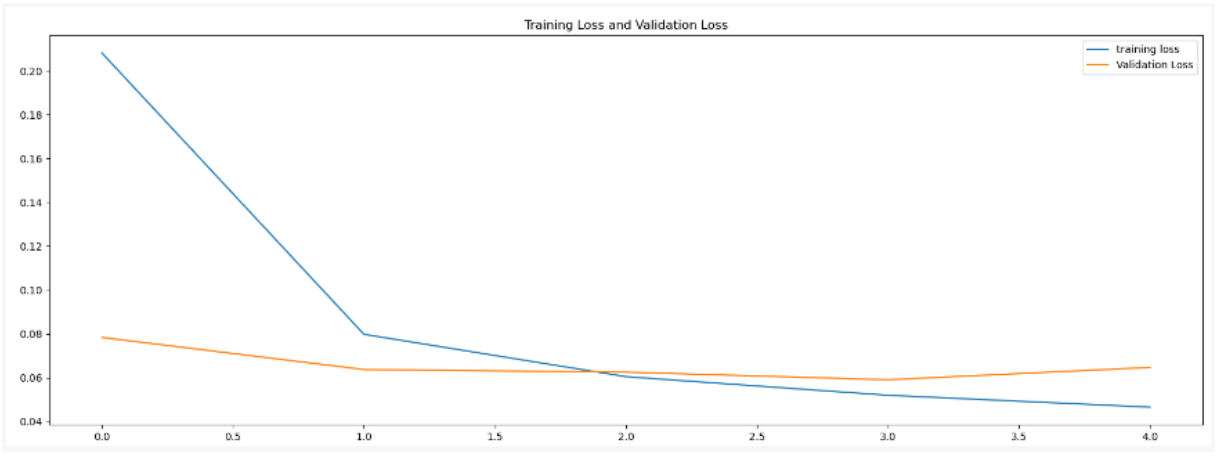


Figure 5: Confusion charts for training (left) and testing (right) sets for LSTM model.

The F1 scores of 0.9928 and 0.9915 are gathered for the training and testing sets which indicates that even though there is some overfitting of data happening the model does very well as shown in figure 5 where most of the text is classified as not toxic and a low amount of comments found to be toxic when they are not.

5.2 GRU



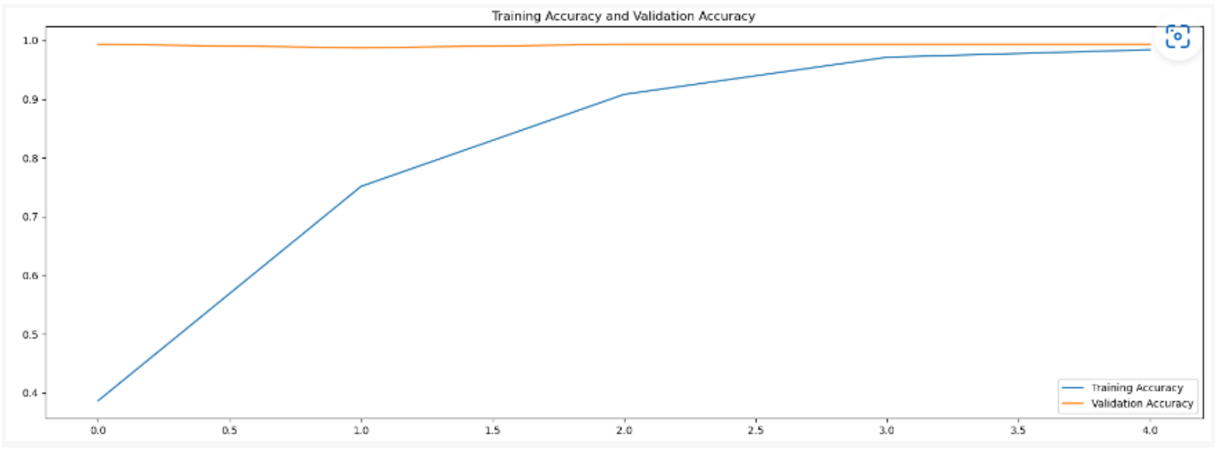
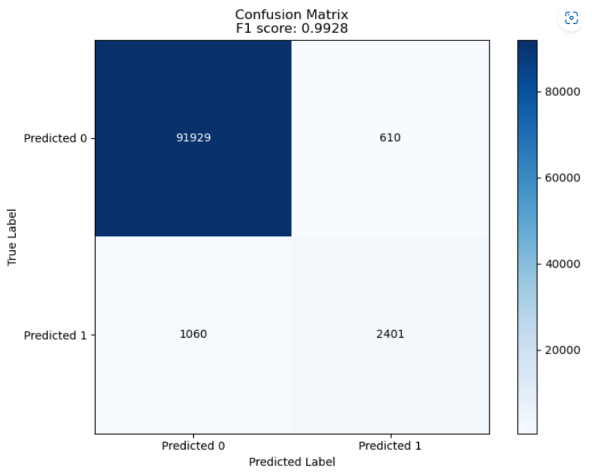


Figure 6: Model training performance, with loss(top) and accuracy (bottom). Training converges after 2 epochs.

After training the model it was evaluated and as shown in figure 6 the GRU network does fairly well will such a low loss rate however overfitting does occur due to the validation data's accuracy being almost perfect however this is what is expected due to the vast distribution of the data.



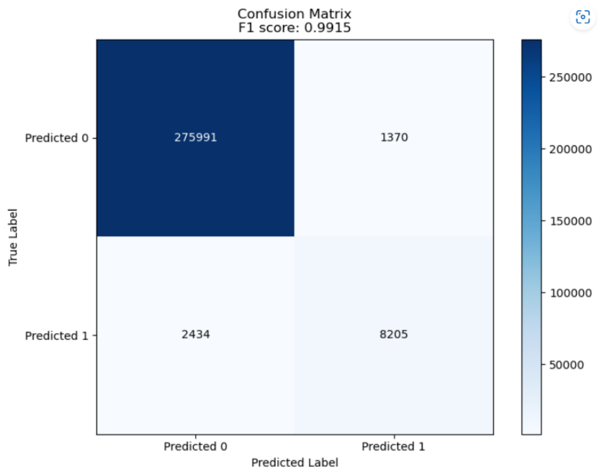


Figure 7: Confusion charts for training (left) and testing (right) sets for GRU model.

The F1 scores of 0.9928 and 0.9915 are gathered for the training and testing sets which indicates that even though there is some over fitting of data happening the model does very well as show in figure 7 where most of the text is classified as not toxic and a low amount of comments found to be toxic when they are not.

5.3 Random Forest

```
Best n_estimators for training set: 250
Best n_estimators for testing set: 300
Best accuracy on training set: 0.9997916666666666
Best accuracy on testing set: 0.927
```

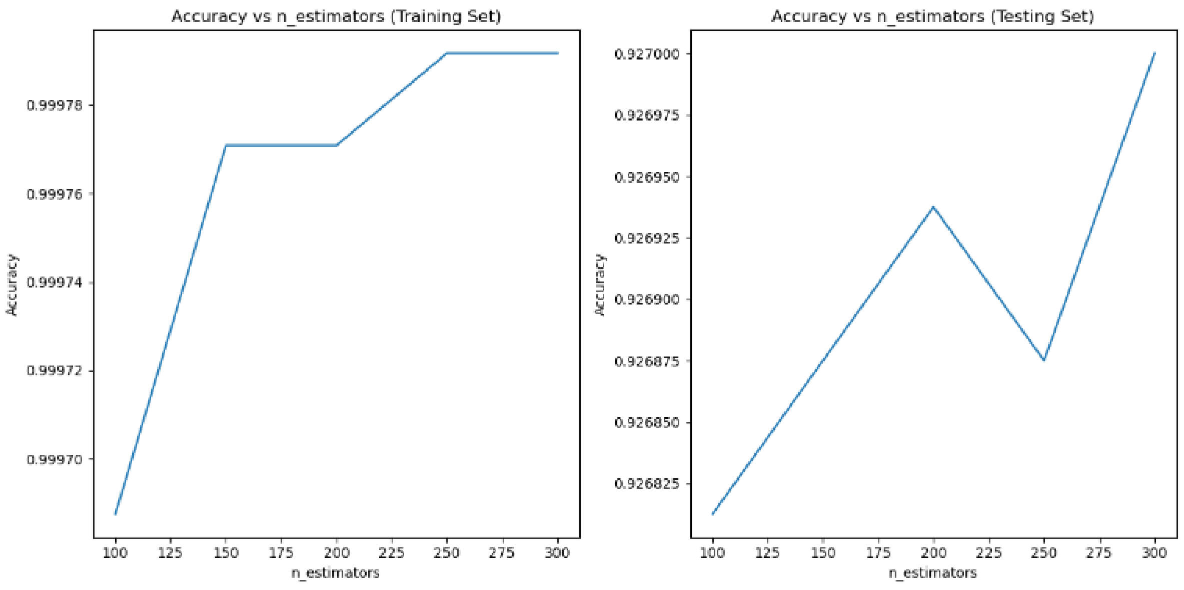


Figure 8: Random Forest training vs testing sets

We now observe the confusion matrices which is an extension of the above results. Looking at figure 8 we can see that the model performs reasonably well on the training and testing sets.

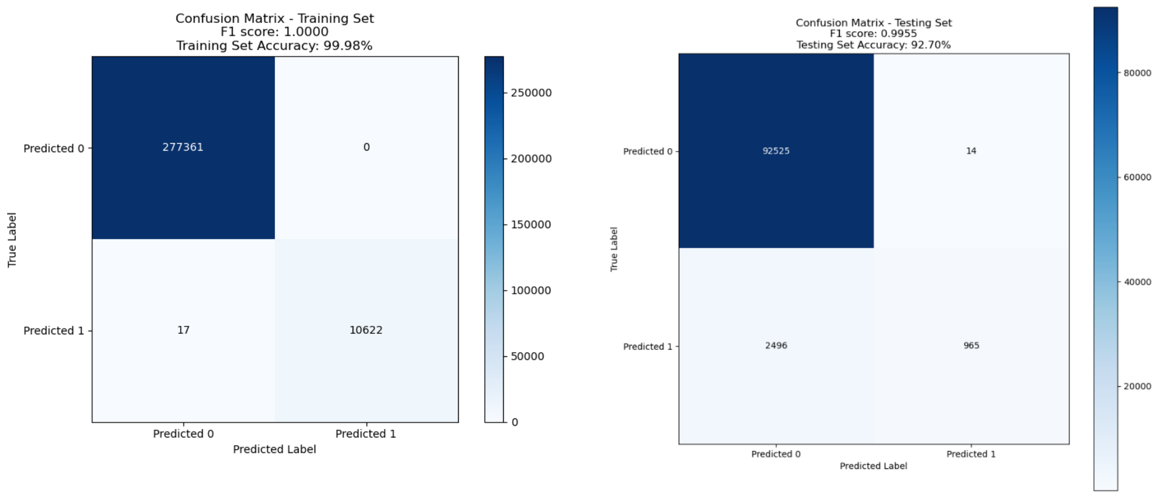


Figure 9: Confusion matrix - Training vs Testing

To summarise the random forest classifier, it was important to determine the n_estimator value. We can assume with the trend observed in figure 9 the higher the n_estimator, the higher the accuracy. For future random forest models, we can experiment with a higher n_estimator value at the cost of significantly higher processing times. We have found that changing the max_depth value brings little change to the model, so we kept it at its default state ‘None’.

model	Training f1-score	Testing f1-score
LSTM	0.9928	0.9915
GRU	0.9928	0.9915
Random Forest	1.0000	0.9955

Figure 10: Table show f1 scores for each model

After evaluating each model the random forest model achieved the best results as it had the best f1-scores for both training and testing data compared to the LSTM model and the GRU model. The latter two models both performed the same which is due to it being similar and not handling the overfitting of data as well as the random forest model did.

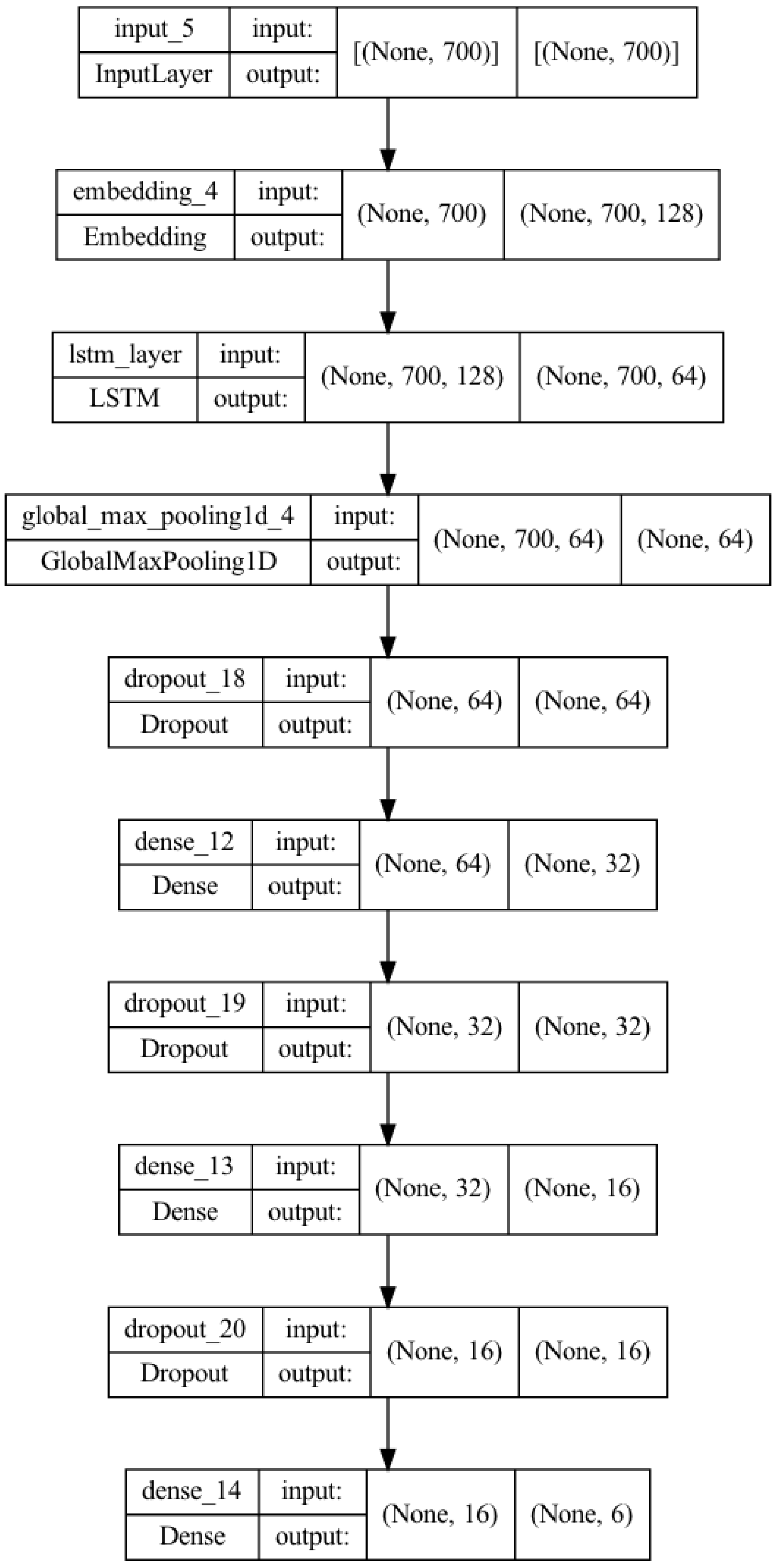
6.0 Conclusion

In conclusion each model matched the experiment goal of predicting whether a comment is one of the toxic classifiers, each model provided good accuracy results but showed the models do over fit due to the great distribution of data, some comments were still classified as clean even though it is classified as toxic displaying the shortcomings in the data. Due to lack computational power the having to reduce the data did not help with the evaluation as the number of comments that are clean are vastly superior to the number of toxic comments. To gain better results for future investigations it will be better to train these models with a more balanced dataset to reduce it being more biased towards comments that are not classified as toxic. It also displays that to eradicated toxic comments more data is needed on these types of comments however there will always be a majority of clean comments as it is a small minority of people that use social platforms to produce such toxic comments.

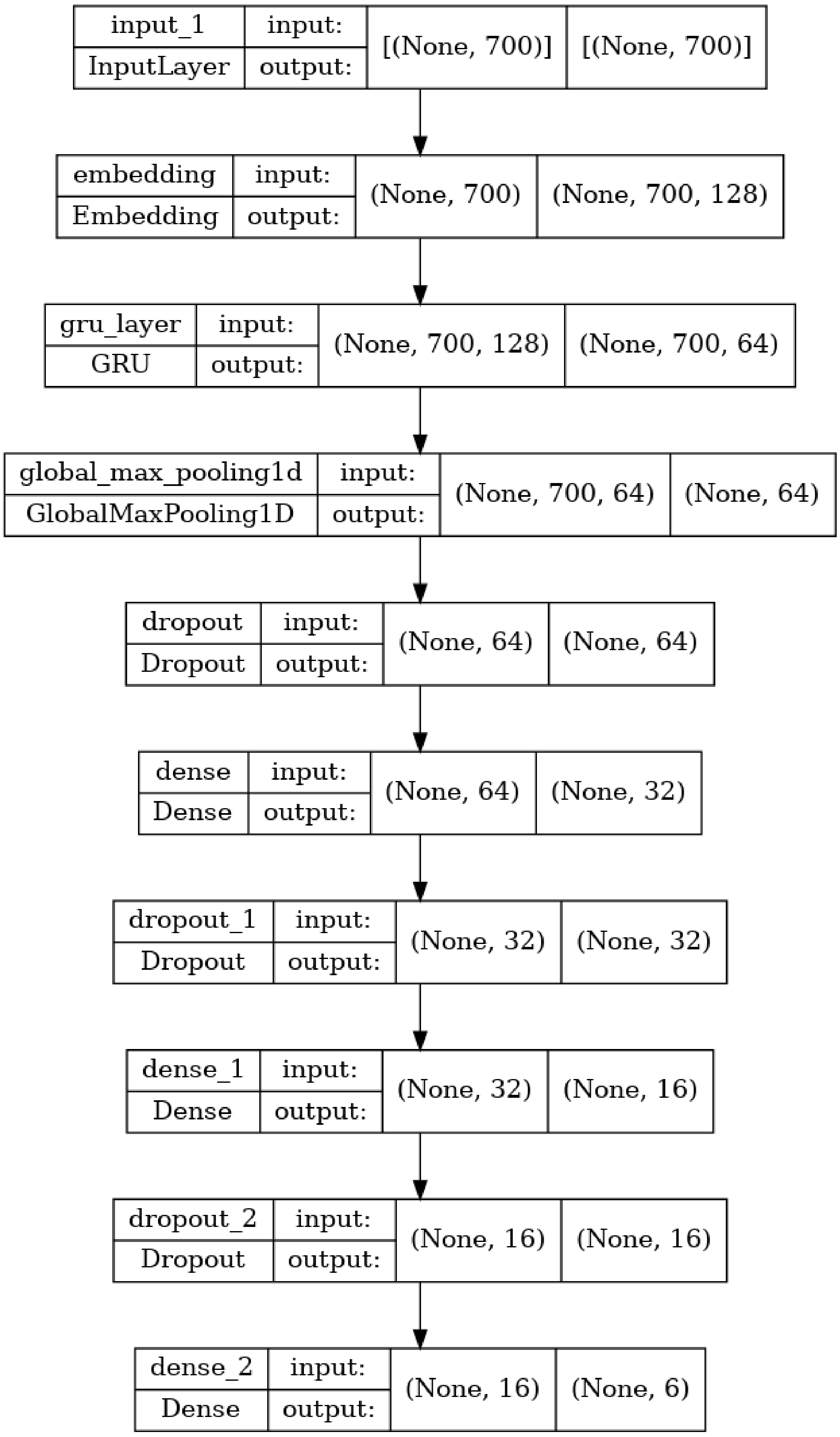
Appendix: Statement of Contribution

	Name	Student Number	Contribution	Segments
#1	Fu-Cheng Tsai	10297171	33.3%	Introduction, GRU
#2	Oscar Chandra	7227116	33.3%	Related work. Random Forest.
#3	Ebenezer Antwi-Brefo	11504862	33.3%	Data Pre-Processing, LSTM

Appendix A: LSTM Network



Appendix B: GRU Network



input_1 | input: | [(None, 700)] | [(None, 700)]
InputLayer | output:
embedding | input: | (None, 700) | (None, 700, 128)
Embedding | output:
gru_layer | input: | (None, 700, 128) | (None, 700, 64)
GRU | output:
global_max_pooling1d | input: | (None, 700, 64) | (None, 64)
GlobalMaxPooling1D | output:
dropout | input: | (None, 64) | (None, 64)
Dropout | output:
dense | input: | (None, 64) | (None, 32)
Dense | output:
dropout_1 | input: | (None, 32) | (None, 32)
Dropout | output:
dense_1 | input: | (None, 32) | (None, 16)
Dense | output:
dropout_2 | input: | (None, 16) | (None, 16)
Dropout | output:
dense_2 | input: | (None, 16) | (None, 6)
Dense | output:

References

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research, 15, 3133-3181. Retrieved from www.scopus.com

Global Webindex. (2023). Report: The biggest social media trends. Retrieved from https://www.gwi.com/reports/social

Islam, M. Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019, November). A semantics aware random forest for text classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 1061-1070).

Jalal, N., Mehmood, A., Choi, G. S., & Ashraf, I. (2022). A novel improved random forest for text classification using feature ranking and optimal number of trees. Journal of King Saud University - Computer and Information Sciences, 34(6, Part A), 2733-2742. https://doi.org/10.1016/j.jksuci.2022.03.012

Kaggle. (2018). Toxic Comment Classification Challenge: Identify and classify toxic online comments. Retrieved from https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Magzoub, Z. (2023). Toxic comment classification in discord. Retrieved from http://essay.utwente.nl/94373/

Patchin, J., & Hinduja, S. (2012). School efforts to prevent cyberbullying (Prevention Researcher). Retrieved from https://www.academia.edu/41895507/Patchin_and_Hinduja_2012_School_efforts_to_prevent_cyberbullying_Prevention_Researcher_

Shali, Sonia Kaul, The Impact of Social Networking on Society with Special Emphasis on Adolescents in India (June 6, 2018). International Journal of Social Science and Humanities Research, 6(2), 662-669. Available at SSRN: https://ssrn.com/abstract=3191756

Sheldon, P., Rauschnabel, P. A., & Honeycutt, J. M. (2019). The Dark Side of Social Media: Psychological, Managerial, and Societal Perspectives. Retrieved from https://www.sciencedirect.com/book/9780128159170/the-dark-side-of-social-media#browse-content