Analysis & Reporting of Results:
Our ML pipeline begins with industry standard data pre-processing to remove meaningless features such as stop words, ensuring more reliable training inputs. Allowing our models to focus on meaningful patterns rather than common words that contribute little to distinguishing between human and computer-generated reviews. After comparative testing, we selected ReLU activation over Leaky ReLU and Tanh due to its industry-standard status and gradient stability. We evaluated model optimisation using 5-fold cross-validation, selected for its robustness and widespread acceptance. All models shown use Word2Vec embeddings which outperformed Glove in our base models. Due to resource constraints BERT embeddings were only tested on our final optimised model.

MLP Model Performances

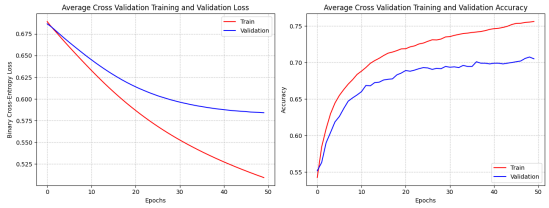


Figure 1: Base MLP1 Performance

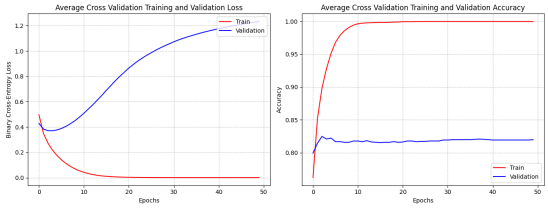


Figure 2: Base MLP2 Performance

Figure 1 and Figure 2 reveal different issues with the base Word2Vec MLP models. MLP1 shows moderate overfitting with steadily diverging training/validation loss curves. MLP 2 displays severe overfitting with validation loss increasing after the initial epochs. Despite worse overfitting, MLP2 had better cross-validation metrics (0.82 accuracy, 0.37 loss) compared to MLP1 (0.70 accuracy, 0.58 loss).

Optimisation 1 (early stopping): We will use early stopping to monitor validation loss during training. It will halt the training process when performance begins to decrease, preventing the model from memorising training data instead of learning generalisable patterns. We have set a high maximum epoch count to allow sufficient training time if needed.

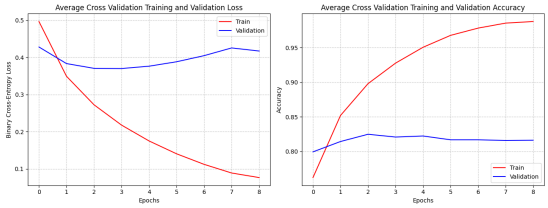


Figure 3: MLP2 Early Stopping Performance

Figure 3 displays the improved MLP2 model using early stopping, which stops training at 8 epochs. This prevents the severe overfitting seen in the base model by stopping before validation loss begins to increase. However, the model still shows overfitting, evidenced by the 30% gap between training loss (0.07) and validation loss (0.42). Early stopping had no effect on MLP1, which still trained for the same number of epochs as its base implementation.

Optimisation 2 (Regularisation): Following early stopping implementation overfitting remained a significant issue in both models. L2 regularisation was implemented to constrain model capacity by penalising large weight values. L2 regularisation proved more effective than L1 for text data with correlated features across embedding dimensions, L2 handles these correlations more effectively than L1's sparse solutions.

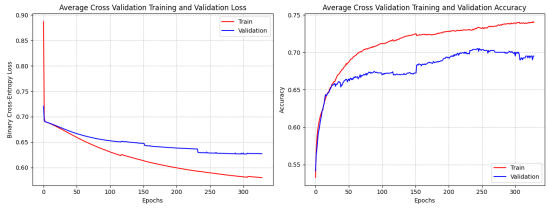


Figure 4: MLP1 L2 Performance

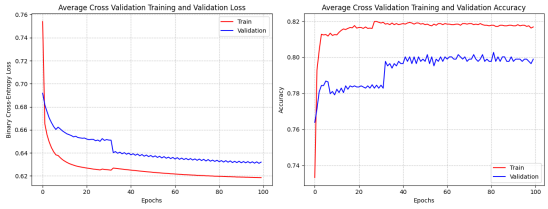


Figure 5: MLP2 L2 Performance

Figure 4 and Figure 5 demonstrate quantifiable improvements: training/validation loss divergence decreased by 2% in MLP1 and 30% in MLP2, while enabling longer training (up to 300 for MLP1 and 100 for MLP2 additional epochs) without performance decreasing. Next steps we need to focus on smoothing performance graphs and further improving accuracy and loss metrics.

Optimisation 3 & 4 (Weights Initialisation & Optimiser): To improve performance metrics and smooth training curves, we implemented HE weight initialisation and Adam optimiser. HE initialisation ensures gradients flow smoothly through ReLU networks by preventing vanishing/exploding gradients, while Adam adaptively adjusts learning rates during training. Initial implementation of HE alone showed negligible improvements, prompting us to combine both optimisations with increased model complexity (expanding the first dense layer from 5 to 16 neurons) so the model could learn more complex patterns.

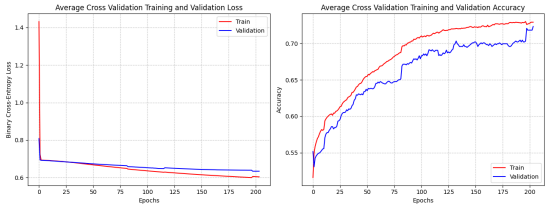


Figure 6: MLP1 HE & Adam Performance

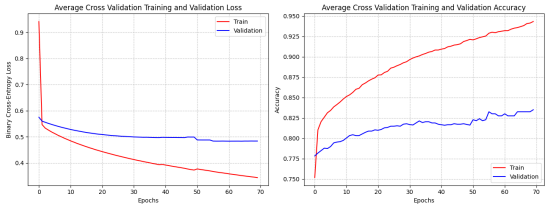


Figure 7: MLP2 He & Adam Performance

The results here are significant: MLP1 Figure 6 demonstrated faster convergence, reduced overfitting, and improved accuracy curves. MLP2 Figure 7 achieved mean accuracy improvements from 0.80 to 0.83 and reduced loss from 0.63 to 0.50, with smoother training curves overall. However, these optimisations introduced a new challenge, the increased overfitting evidenced by a 6% wider gap between training and validation loss, suggesting further adjustments of regularisation rates may be needed.

Optimisation 5 (Tuning Model Architecture) After investigating various MLP1 model architectures we found that adding more layers and neurons did not meaningfully improve performance. Simpler architectures often outperformed more complex ones. The final MLP1 model showed poor performance on the unseen test set (accuracy:0.64, precision:0.589, recall:0.86) with a clear bias towards positive predictions. However MLP2 performance once again improved after increasing model capacity to represent complex patterns and relationships in the data by adding a second hidden layer and increasing the number of neurons. Out of the two MLP architectures it is clear that MLP2 is superior as sequential embeddings preserve word order and positional relationships. The model can learn how words interact with each other in context. This is impossible with a single vector representation.

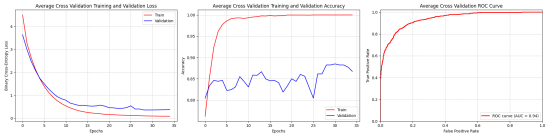


Figure 8: MLP2 Final Model Curves

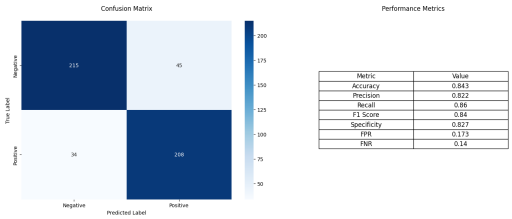


Figure 9: MLP2 Final Model results

In Figure 8 we can see that the architecture changes effectively mitigated overfitting, as evidenced by the validation loss curve, while achieving exceptional classification performance (AUC:0.94). Cross-validation across 5 folds confirmed strong performance (mean accuracy: 0.86, mean loss: 0.46). On the unseen test set in Figure 9 , MLP2 maintained this performance (accuracy: 0.84, recall: 0.86) while minimising false positives and negatives.

CNN Model:

We follow a similar methodology for optimising our CNN model as we used with the MLPs. Carrying over successful techniques like early stopping to our CNN base implementation. For brevity, only new optimisations will be explained in detail

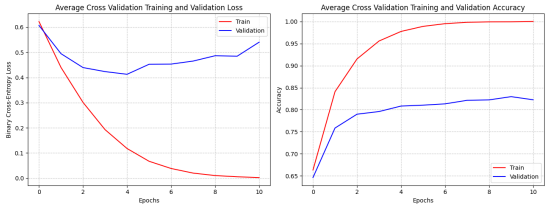


Figure 10: CNN Base Model

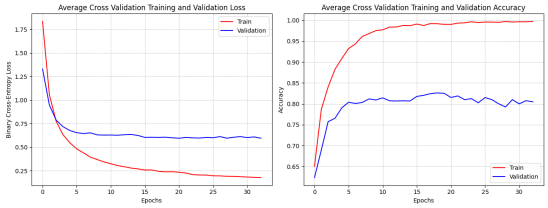


Figure 11: CNN Regularisation Model

In Figure 10, our base CNN model has a clear overfitting issue, similar to the unregularised MLPs, with validation loss increasing while training loss continues decreasing To address this we implemented two regularisation techniques in Figure 11. L2 regularisation, which adds a penalty term to the loss function discouraging the model from learning overly complex patterns and a Dropout layer which randomly deactivates neurons during training forcing the network to learn more features that don't depend on particular neuron combinations. These techniques combined show improvement, the validation loss no longer rises but instead converges closer to the training loss, a clear indication of reduced overfitting. While training time increased, the model maintains a validation accuracy of 0.80 with more generalisable performance.

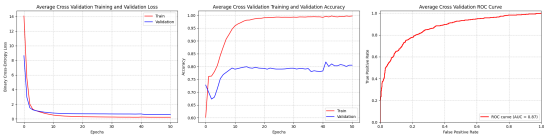


Figure 12: CNN Final Model Curves

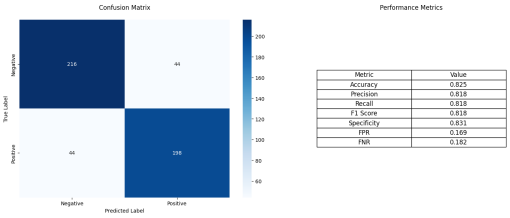


Figure 13: CNN Final Model results

Following further optimisations that I have provided rationale for earlier, including HE weight initialisation, Adam optimiser, increasing neurons in the dense layer from (10 - 16) & further fine tuning of learning rate and regularisation rates, we have arrived at these results. Displayed in Figure 12 are the improved curves with the loss curve showing less signs over overfitting and faster convergence. We can see early stopping has increased from epoch 30 to 50 due to a lower Adam optimiser learning rate & increased regularisation rate proving useful, allowing the model to extend training time resulting in a much smoother loss curve. The ROC curve shows a high AUC score of 0.87, but performs poorly compared to our other models. In Figure 13 the confusion matrix and metrics table highlight the models balanced performance across both classes, with the (specificity: 0.831) indicating a slight strength towards classifying negatives correctly compared to classifying positives correctly (Recall: 0.818). Considering our main priority is minimising false negatives this is a clear limitation of this CNN model.

RNN Model:

From our previous models it is apparent that without intially applying early stopping and regularisation the models often struggle with overfitting so we will carry over Early stopping into the RNN base model. Then we will apply L2 regularisation and dropout as our next optimisation.

Figure 14: RNN Base Performance

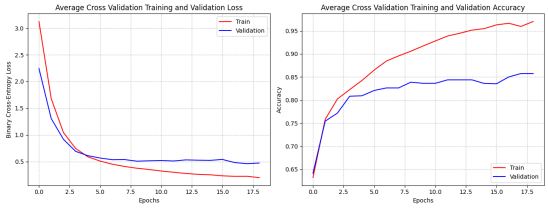


Figure 15: RNN Regularisation Performance

Figure 14 reveals that despite early stopping at epoch 10, the base model exhibits classic overfitting symptoms—decreasing training loss but increasing validation loss, with a widening gap between training and validation accuracy curves. In Figure 15, after applying L2 regularisation and dropout, we observe significant improvement: training time extended by 70%, with the validation and training loss curves now converge appropriately, indicating better generalisation. The validation accuracy stabilizes around 0.85, demonstrating that the regularisation effectively prevents the model from memorising training data while still capturing meaningful patterns.

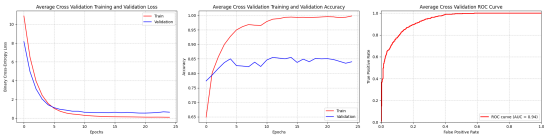


Figure 16: RNN Final Performance Curves

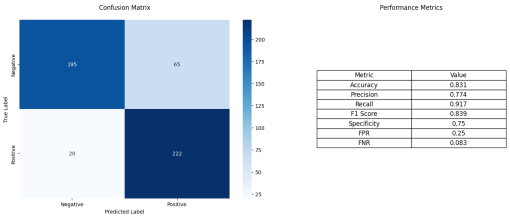


Figure 17: RNN Final Performance Metrics

In Figure 16, we can see significant performance improvements after applying several targeted optimisations. We selected Adam optimizer with a low learning rate (0.001) to provide adaptive momentum while ensuring stable convergence on our sequential text data. He weight initialisation was implemented as it performs well when paired with our Relu architecture and Adam optimiser. The architecture was fine-tuned by doubling LSTM layer capacity (64→128) to better capture complex sequential patterns, while simultaneously halving the first dense layer (256→128) to reduce the models capacity to memorise training data, this created a more balanced architecture for our task. These complementary adjustments resulted in appealing loss curves shown in Figure 16, with training and validation losses converging appropriately. The ROC curve demonstrates exceptional class separation with an AUC score of 0.94, while Figure 17 reveals strong overall performance on the test set accuracy (0.83). consistent results were produced across all validation folds mean accuracy (0.85). Notably, the metrics table shows a low false negative rate (0.083), indicating the model successfully identifies computer-generated reviews in most cases, this will prove very useful for our final model utilising BERT embeddings.

Bert Embeddings:

We selected our RNN model for BERT embedding tests based on its consistent cross-validation performance (0.85 mean

accuracy) and superior false negative rate (0.08) compared to other evaluated models. It is likely that using BERT will improve performance as it generates dynamic embeddings that change based on the surrounding context, BERT also processes text in both directions simultaneously which helps it understand the full context around each word.

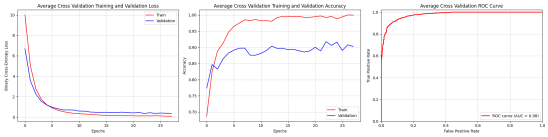
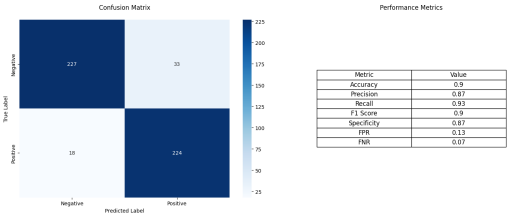


Figure 18: RNN Final Performance Curves



Metric	Value
Accuracy	0.9
Precision	0.87
Recall	0.93
F1 Score	0.9
Specificity	0.87
FPR	0.13
FNR	0.07

Figure 19: RNN Final Performance Metrics

Figure 18 displays the final results of our RNN with BERT embeddings, The model appears to be training correctly with minimal overfitting and it boasts exceptional performance with a mean validation accuracy of (0.915) and mean validation loss of (0.36). The ROC curve has an AUC score of 0.98 indicating strong class separation abilities. In Figure 19 we can visualise the models performance against the test set with notable performance in relation to accuracy (0.9), recall (0.93) and false negative rate (0.07). Across the board all metrics are high, showing minimal signs of class bias.

an important insight was discovering a 2-3% performance improvement when using unprocessed data, this suggests that for our small dataset, stop words may actually be helpful for distinguishing between human reviews and computer generated ones. However, it is worth noting that training on unprocessed data may skew evaluation results and potentially lead to worse real-world performance. In different datasets, stop word removal would likely improve model performance.

Final Outcome & Moving Forward:

The LSTM RNN model using BERT embeddings emerged as the best performing model for fake review detection. This selection is based on its remarkable accuracy of 0.9 on the test set and average of 0.915 during cross validation, ROC curve score of 0.98, and most importantly its false negative rate of 0.07 (the lowest of all models evaluated), which fulfills our primary objective of minimising the amount of computer-generated reviews that are misclassified as genuine users. LSTMs thrive when processing sequential data, and their specialised memory cells allow them to capture distant contextual relationships within reviews. This capability proved essential for detecting computer-generated reviews, as they often exhibit distinctive linguistic patterns and repetitive structures which differ from authentic human-written reviews.

The project successfully achieved the main goal of creating a model that improves upon manual detection of fake reviews which is labor intensive and increasingly unsustainable. The LSTM RNN model provides a reliable solution to minimise computer generated reviews with deceptive intent from reaching the customers platform. While our model demonstrates high accuracy it may be necessary for human judgment in some complex cases such as, a review written by a human but heavily edited using AI tools. For example, a user might write a genuine review about their experience, then use an AI tool to enhance the language, add more details, or expand the length. The "black box" nature of deep learning models makes it difficult to fully understand the specific features guiding classification decisions. Additionally, the model's effectiveness is likely tied to its training on English language data and reviews of particular lengths, which could potentially limit performance when processing reviews in other languages or those with unusual lengths compared to our training data. A significant challenge moving forward will be obtaining comprehensive, varied, and regularly updated datasets with reliable labels. Furthermore, as synthetic review generation techniques evolve, adapting the model to recognise new deceptive patterns will present ongoing challenges.

A range of strategies can be explored to improve the model models performance. We could implement controlled adversarial examples to generate borderline samples that exist at the boundary between fake and genuine reviews, challenging the models classification abilities. Back-translation is another interesting approach, where text is translated text to another language then back to the original, preserving the meaning while altering the structure. For real-world applications, training on larger and more diverse datasets would be beneficial, allowing the model to learn many more patterns and deception techniques. Furthermore, Attention mechanisms would allow the model to focus on the most relevant parts of the reviews by dynamically weighting words and phrases that indicate authenticity or deception. Finally our model could be integrated into an ensemble approach, combining its strengths with other strong classifiers to achieve better performance than any single model alone.

The model offers great value as a support tool for manual review moderation. With the models ability to automatically approve high-confidence genuine predictions and flag suspicious reviews, manual checkers can focus their time on high priority cases. The model could work alongside human moderators through a feedback system where moderator decisions refine the model overtime, gradually increasing automation as model performance improves. Additionally, The model could highlight evidence in reviews using colour coding (e.g. yellow for generic content, red for deceptive content), providing visual guidance that allows moderators to focus on specific suspicious sections of the review rather than the entire text.