
NORTHWESTERN
UNIVERSITY

# OPTIMIZING TEXT CLASSIFICATION THROUGH DEEP LEARNING: PREPROCESSING, ARCHITECTURE, AND EMBEDDING STRATEGIES

Prepared by: Eswarankarthik Paranthaman

May 12, 2025

MSDSP 458 Artificial Intelligence & Deep Learning

Professor Edward Arroyo & Professor Narayana Darapaneni

Northwestern University, USA

## 1.0 ABSTRACT

This study explores deep learning architectures for text classification using the AG News dataset, emphasizing the impact of preprocessing and model structure. A dense neural network (DNN) served as a baseline to assess vocabulary size, stopword removal, and sequence length (natural versus fixed) across twelve configurations. The optimal setup, consisting of a vocabulary size of 20,000, stopword removal, and a fixed sequence length of 128, was applied to recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and one dimensional convolutional neural network (1D CNNs) using GloVe embeddings. Key hyperparameters, including a dropout rate of 0.4, batch size of 100, the Adam optimizer, and sparse categorical crossentropy as the loss function, were held constant for consistency. Evaluation metrics included accuracy, log loss, training time, and confusion matrices. The top performing model was further tested with learnable embeddings to compare generalization. A conceptual framework for a generative model based on Chollet (2018) is also discussed.

## 2.0 INTRODUCTION

With the rapid growth of unstructured text data, automatic text classification has become essential across sectors. Businesses use these systems to extract insights, improve content recommendations, and streamline customer support. This study evaluates the performance of dense, recurrent, and convolutional neural networks for classifying news articles using the AG News dataset, a widely used benchmark in natural language processing. Beyond identifying the best performing model, the research examines trade-offs between accuracy, complexity, and training efficiency. The findings provide practical guidance for implementing NLP solutions in domains such as news categorization and conversational AI.

## 3.0 LITERATURE REVIEW

Text classification is a fundamental task in natural language processing, evolving from bag-of-words models to deep learning approaches. Early methods relied on handcrafted features and linear classifiers, while deep learning enabled models to learn hierarchical representations directly from data (LeCun, Bengio, and Hinton 2015). DNNs are effective for simple tasks but struggle with sequential patterns. RNNs, especially LSTM networks, address this by capturing temporal dependencies (Ghojogh and Ghodsi 2023). Bidirectional variants further improve context modeling (Graves and Schmidhuber 2005). CNNs, though rooted in image processing, also perform well in text classification by capturing local features through filters and pooling layers (Kim 2014).

Word embeddings such as GloVe (Pennington, Socher, and Manning 2014) have largely replaced sparse representations, enabling models to capture semantic similarity and improving downstream performance. Generative models based on RNNs and LSTMs offer potential in text generation, with techniques like temperature-controlled sampling expanding output diversity (Chollet 2018). Although this study focuses on classification, it also considers the theoretical basis for generative extensions.

# 4.0 METHODS

## 4.1 DATASET AND PREPROCESSING

The AG News dataset was retrieved from the TensorFlow datasets catalog, comprising four balanced classes: World, Sports, Business, and Sci/Tech. Each class contains approximately 30,000 training samples and 1,900 test samples. The training data was further split using an 80:20 ratio to form validation data.

Initial exploratory data analysis (EDA) revealed a highly skewed document length distribution, with the majority of articles consisting of 20 to 40 words, and an average length of 31. A histogram was plotted to visualize this distribution (Figure A2 in the Appendix). Three vocabulary sizes were tested (5,000, 10,000, and 20,000) and bar charts (Figure A.3) and word clouds (Figure A.4) were used to analyze the frequency of terms. Across all vocabulary sizes, common stop words like "the", "a", and "in" dominated the top ranks. The frequent appearance of the numeric token "39" further emphasized the need for stopword filtering. Experiments were conducted with and without stopword removal, and with both natural and fixed sequence lengths (128 tokens).

## 4.2 BASELINE MODEL: DENSE NEURAL NETWORK (DNN)

A total of 12 baseline experiments were conducted using a simple DNN to evaluate three preprocessing variables - vocabulary size (5,000, 10,000, and 20,000), stopword removal (yes vs. no), and sequence length (natural vs. fixed at 128 tokens). This factorial design ($3 \times 2 \times 2$) resulted in 12 configurations.

All DNN experiments were trained using:

- Epochs: 20
- Batch size: 100
- Dropout rate: 0.4
- Optimizer: Adam
- Loss function: Sparse categorical crossentropy
- Metrics: Accuracy

This setup ensured consistent training conditions across experiments. The best-performing configuration was the Experiment 12 used a vocabulary size of 20,000, with stopwords removed and a fixed sequence length. This configuration served as the baseline for subsequent modeling architectures (Analysis: Figure A.7 – Figure A.49 and Table A.1).

## 4.3 ADVANCED MODELS: RNN, LSTM, AND 1D CNN

Subsequent experiments maintained the baseline's optimal preprocessing settings. All models used GloVe 200-dimensional word embeddings, which were kept non-trainable to preserve semantic relationships.

- RNN and LSTM Models: Both unidirectional and bidirectional versions were tested. Each included a 64-unit recurrent layer, dropout layers (rates 0.5 and 0.4), a 128-unit dense hidden layer with ReLU activation, and a final softmax layer for classification.
- 1D CNN: Included a Conv1D layer with 128 filters, a kernel size of 5, and ReLU activation to capture local textual patterns. This was followed by dropout and dense layers similar to the RNN models.

Each model was trained for 10 epochs. Performance metrics of accuracy, loss, and training time were recorded and compared (see Table D.7).

### 4.4 EMBEDDING COMPARISON:

To assess the generalization capabilities of pretrained embeddings, the best-performing model architecture was re-trained using learnable embeddings initialized from scratch. This experiment aimed to compare the predictive performance and training efficiency of models using fixed GloVe embeddings versus models that learn embeddings directly from the dataset. The results of this comparison are discussed in the Results section (refer to Figure D.7 in the Appendix).

## 5.0 RESULTS & DISCUSSION:

### 5.1 EXPERIMENT B: SIMPLE RNN MODELS

#### EXPERIMENT B1: SIMPLE RNN UNIDIRECTIONAL

The unidirectional Simple RNN model performed poorly on the AG News classification task, as evidenced by the classification report (Table B.1 – B.3 in Appendix), which reveals that the model predicted almost exclusively Class 0 (World), leading to perfect recall for that class but zero precision, recall, and F1 scores for Class 1 (Sports), Class 2 (Business), and Class 3 (Sci/Tech). Consequently, the macro and weighted average F1 scores were near 0.10, suggesting performance akin to random guessing. Confusion matrices (Figure B.1) further corroborated this by showing that nearly all predictions fell under Class 0 (World) across training, validation, and test sets, reflecting a degenerate solution where the model memorized the most frequent label instead of learning class distinctions. During training, both accuracy and loss (Figure B.2) remained flat, with accuracy hovering around 0.25 across all epochs and loss staying high, between 1.38 and 1.39. These patterns pointed to severe underfitting and the model's inability to adjust its weights meaningfully. The training process lasted approximately 245 seconds but resulted in no practical performance improvement, significantly underperforming relative to the DNN baseline established in Experiment 12.

#### EXPERIMENT B2: SIMPLE RNN BIDIRECTIONAL:

In contrast, the bidirectional RNN model showed notable improvement, achieving an accuracy of approximately 0.89 on both training and validation sets and 0.88 on the test set (Table B.4 – B.6). Precision, recall, and F1 scores were strong across all classes, with particularly high performance for Class 1 (Sports), which recorded recall values near 0.99 and F1 scores around 0.94. Class 0 (World) also performed well, while Class 2 (Business) and Class 3 (Sci/Tech) exhibited solid, balanced performance,

indicating good generalization across the dataset. Confusion matrices (Figure B.3) revealed a clear diagonal pattern, showing accurate classification with only minor and scattered misclassifications across all data splits. The accuracy curve steadily rose and stabilized above 0.88, while loss consistently decreased to below 0.4 (Figure B.4), with training and validation curves closely aligned, indicating convergence and effective generalization. The training process lasted around 313 seconds. When compared to both the unidirectional RNN and the DNN baseline, the bidirectional RNN demonstrated superior performance, capturing richer contextual information, achieving better accuracy, reducing loss more effectively, and maintaining a more balanced class distribution. It proved to be a significantly more effective architecture for this text classification task.

## 5.2 EXPERIMENT C: LSTM MODELS

### EXPERIMENT C1: LSTM UNIDIRECTIONAL

The unidirectional LSTM model shows significant improvement over both unidirectional and bidirectional Simple RNNs, achieving 0.90 accuracy on training and validation sets and 0.89 on the test set, demonstrating strong generalization without overfitting (Table C.1-C.3). Class-specific performance is consistently strong, with Class 1 (Sports) achieving a recall of up to 0.98 and an F1 score near 0.97. Class 0 (World) and Class 3 (Sci/Tech) also perform well, while Class 2 (Business) shows slightly lower recall but still maintains a solid F1 score between 0.85 and 0.87. This balanced performance suggests that the model effectively distinguishes among all four categories. Confusion matrices (Figure C.1) reveal a dominant diagonal, indicating high classification accuracy, with most errors arising from overlap between Class 0 (World) and Class 2 (Business), and to a lesser extent between Class 2 (Business) and Class 3 (Sci/Tech), although misclassifications are limited and well-distributed. Training dynamics demonstrate steadily increasing accuracy and decreasing loss over the epochs (Figures C.2), with training and validation curves closely aligned, reflecting strong convergence and generalization. Although the training process took longer (625 seconds), the LSTM's ability to model long-range dependencies justifies the additional computational cost.

### EXPERIMENT C2: LSTM BIDIRECTIONAL:

The bidirectional LSTM model outperforms its unidirectional counterpart and previous architectures across all metrics, achieving 0.95 accuracy on the training set, 0.92 on the validation set, and 0.91 on the test set, reflecting excellent generalization with minimal overfitting (Table C.2 – C.5). Class-specific performance is particularly strong, with Class 1 (Sports) achieving near-perfect recall and precision, resulting in an F1 score close to 0.99. Class 0 (World) and Class 2 (Business) also perform well, though minor confusion occurs between them, while Class 3 (Sci/Tech) maintains strong recall and precision, achieving F1 scores around 0.93. This consistent performance across all categories suggests the model’s ability to capture nuanced patterns in text data. Confusion matrices (Figure C.3) show clear diagonal dominance, confirming high classification accuracy, with most errors occurring between semantically similar classes, especially between Class 0 (World) and Class 2 (Business), though misclassifications are still limited and evenly distributed. Training dynamics illustrate steady improvement, with accuracy

stabilizing above 0.90 and loss consistently decreasing over time (Figures C.4). The alignment of training and validation curves further suggests stable convergence and strong generalization. Despite the training time increasing to 1,110 seconds, the performance gains justify the added computational cost, and the bidirectional LSTM's ability to leverage both past and future context makes it particularly effective in handling complex classification tasks.

5.3 EXPERIMENT D: 1D CONVOLUTIONAL NEURAL NETWORK :

The 1D CNN model, which incorporates pretrained GloVe embeddings and a flatten layer, delivers strong and efficient performance on the AG News classification task by effectively capturing local n-gram patterns, making it a fast and reliable alternative to sequential models. It achieved 0.96 accuracy on the training set and 0.91 on both the validation and test sets, demonstrating solid generalization without signs of overfitting (Table D.1 – D.3 ). Class-specific performance is well-balanced, with Class 1 (Sports) showing the highest precision and recall, achieving an F1 score approaching 0.99. Class 0 (World) and Class 3 (Sci/Tech) also perform strongly, with F1 scores ranging from 0.92 to 0.95. Although Class 2 (Business) shows slightly lower recall around 0.84, it still maintains competitive F1 scores between 0.87 and 0.94. The confusion matrices (Figure D.1) exhibit a strong diagonal trend, confirming high classification accuracy, with most errors occurring between Class 0 (World) and Class 2 (Business), and between Class 2 (Business) and Class 3 (Sci/Tech), reflecting patterns observed in previous models. Training dynamics are smooth and stable, with accuracy steadily improving and plateauing above 0.90, while loss decreases consistently across epochs (Figures D.2), and the close alignment of training and validation curves suggests minimal overfitting and robust learning. The model's training time was efficient at 408 seconds, significantly faster than both LSTM variants, and despite its reduced computational demands, the 1D CNN delivered competitive performance, reinforcing its suitability for resource-constrained environments.

5.4 MODEL COMPARISON:

A comparison of the model architectures evaluated in this study demonstrates the trade-off between predictive accuracy and computational efficiency (refer to Table D.7 in Appendix). The bi-directional LSTM (Experiment C2) achieved the highest validation accuracy at 0.92 and a test accuracy of 0.91, outperforming other models in raw performance. However, its training time of 1110.05 seconds made it the most computationally expensive model in the study. In contrast, the 1D CNN (Experiment D) achieved a nearly equivalent test accuracy of 0.9111 and comparable validation accuracy, while requiring significantly less training time at 408.59 seconds. These findings indicate that the 1D CNN offers a favorable balance between performance and training efficiency, and was therefore selected as the best model for this classification task.

5.5 EMBEDDING COMPARISON:

To assess the impact of embedding strategies, the selected CNN model was retrained using one-hot encoding instead of pretrained GloVe embeddings (see Table D.7 in Appendix). The version using one-hot encoding achieved an extremely high training

accuracy of 0.9915; however, this did not translate to better generalization, as both validation (0.8998) and test accuracy (0.893) declined. Moreover, the validation loss increased substantially to 0.8051, suggesting signs of overfitting. In contrast, the original CNN model using GloVe embeddings showed more consistent performance across training, validation, and test sets. These results underscore the advantage of utilizing pretrained word embeddings, which provide rich semantic context and help improve generalization in natural language classification tasks.

### 5.6 GENERATIVE LANGUAGE MODEL DESIGN:

Inspired by Chollet (2018), this project also considered the framework for building a generative model for news text using the same AG News data. The architecture would involve an LSTM-based sequence model trained to predict the next word in a sequence. The input would consist of tokenized and padded sequences of news articles, likely using a subset of articles from a single class to improve stylistic consistency. The training objective would be to minimize categorical crossentropy while predicting the next word in the sequence. To control the creativity of the generated text, temperature-based sampling would be applied. As an example, the model could be seeded with a short phrase and generate subsequent words. While full implementation is beyond the current report's scope, this approach provides a foundational step toward building content-generating systems in future work.

## 6.0 CONCLUSION:

This study evaluated multiple deep learning architectures for text classification using the AG News dataset, comparing performance across dense, recurrent, long short term memory, and convolutional neural network models. Initial experiments with a simple dense network identified an optimal preprocessing setup, including vocabulary size, stopword removal, and sequence length, which was used consistently across models. The bidirectional long short term memory model achieved the highest accuracy but came with high computational cost. In contrast, the one dimensional convolutional neural network offered a strong balance between accuracy and efficiency, making it the most practical for scalable natural language processing applications. Models using pretrained GloVe embeddings outperformed those with one hot encodings, highlighting the value of semantic knowledge in tasks with limited context, such as news headlines.

For organizations deploying machine learning in text classification or customer support, the findings offer practical guidance. The one-dimensional convolutional model, in particular, is well suited for production environments. Looking ahead, generative models such as those proposed by Chollet in 2018 could be explored using similar datasets, although they would require more complex architectures and infrastructure for real time deployment.

In conclusion, this study underscores the need to balance accuracy with efficiency in natural language model selection. Pretrained embeddings and convolutional architectures are recommended for fast and reliable text classification, with opportunities for future work in generative language modeling.

References:

Chollet, François. 2018. *Deep Learning with Python*. 1st ed. Shelter Island, NY: Manning Publications.

Ghojogh, Benyamin, and Ali Ghodsi. 2023. “A Comprehensive Review of Recurrent Neural Networks and Their Variants.” arXiv preprint. https://arxiv.org/abs/2009.09586.

Graves, Alex, and Jürgen Schmidhuber. 2005. “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures.” *Neural Networks* 18 (5–6): 602–10. https://doi.org/10.1016/j.neunet.2005.06.042.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” Neural Computation 9 (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Kim, Yoon. 2014. “Convolutional Neural Networks for Sentence Classification.” arXiv preprint arXiv:1408.5882. https://arxiv.org/abs/1408.5882.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” Nature 521 (7553): 436–44. https://doi.org/10.1038/nature14539.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. “GloVe: Global Vectors for Word Representation.” In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–43. https://aclanthology.org/D14-1162/.

# APPENDIX A – DATA PREPARATION AND EDA (EXPERIMENT- A)

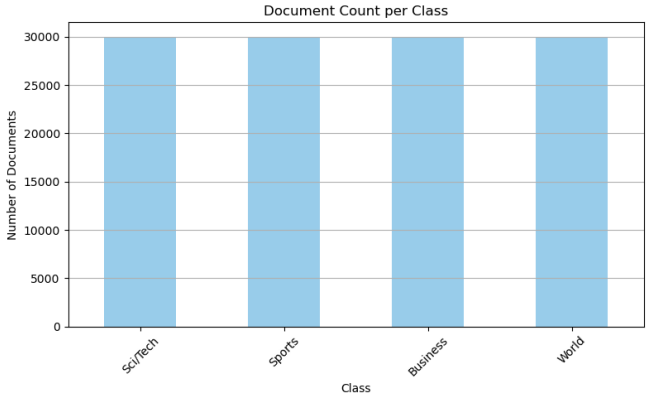


Figure A.1: Target Label Distribution

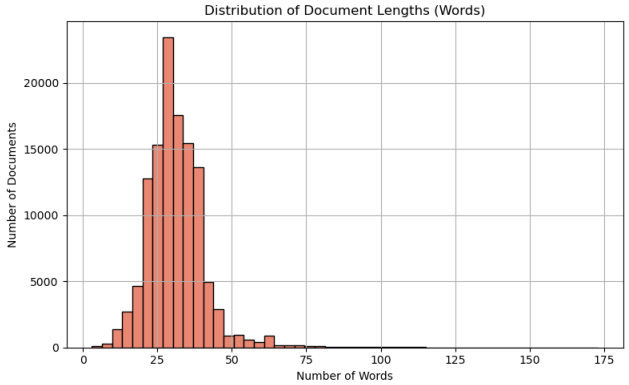


Figure A.2: Histogram Document Length Distribution

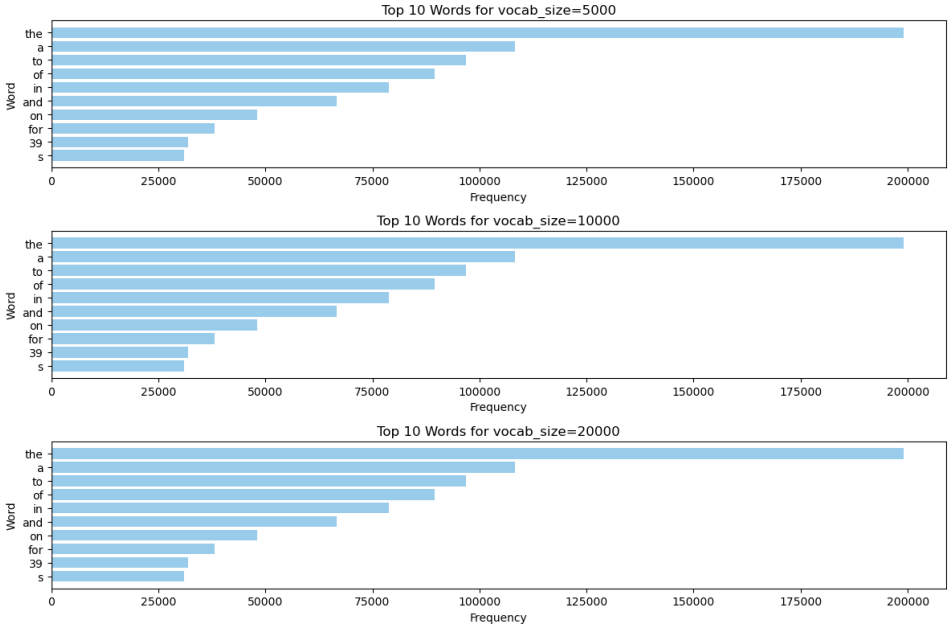


Figure A.3: Top 10 frequent words at different vocabulary size (5,000,10,000, and 20,000)


Figure A.4: Words appearance word cloud at different vocabulary size

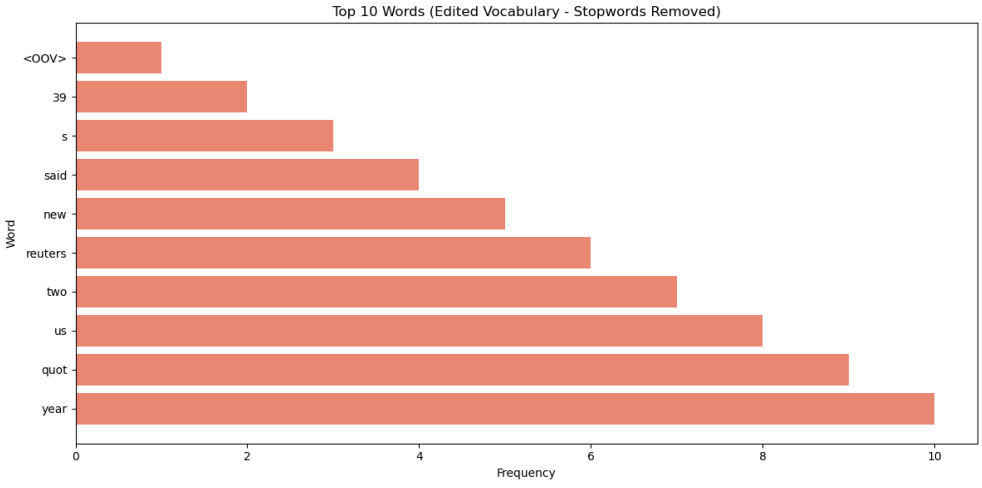


Figure A.5: Top 10 frequent words after removing stopwords.

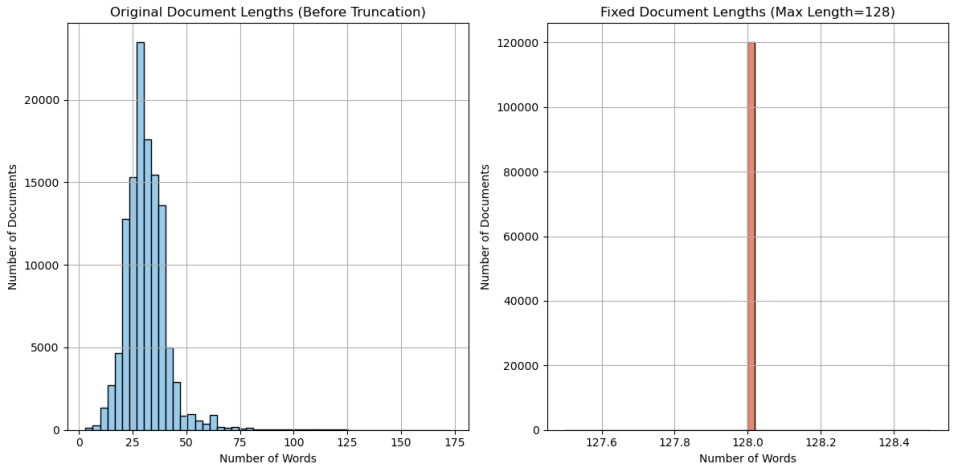


Figure A.6: Sequence length distribution Natural (left) vs Fixed length 128 (Right)

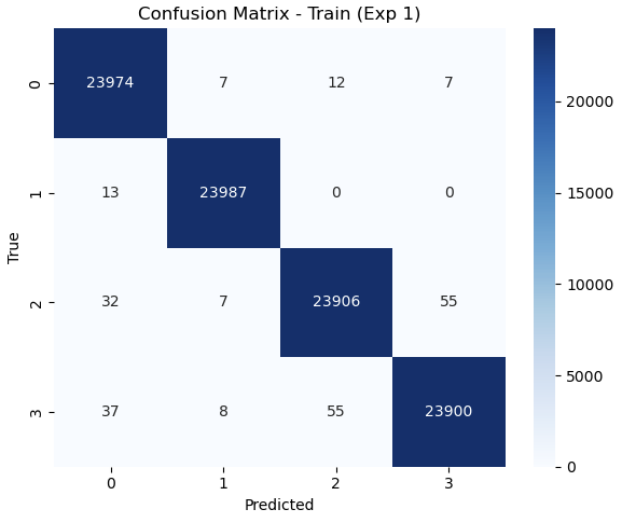


Figure A.7: Confusion Matrix - Experiment-1 training dataset

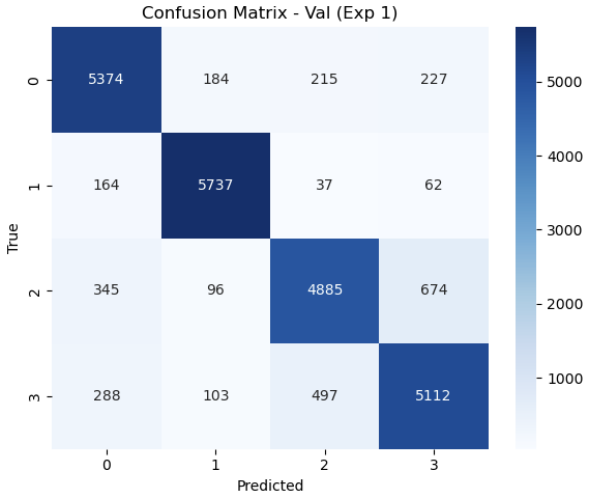


Figure A.8: Confusion Matrix - Experiment-1 validation dataset

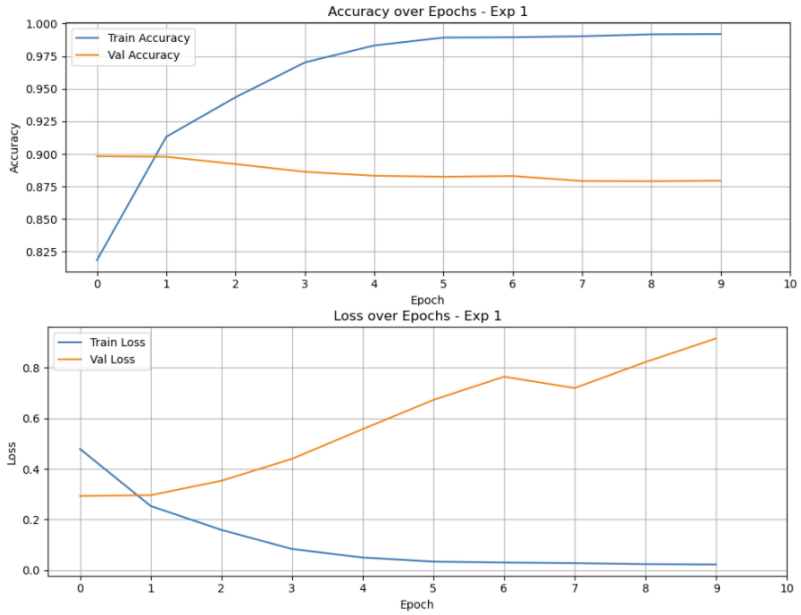


Figure A.9: Model training Exp-1 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

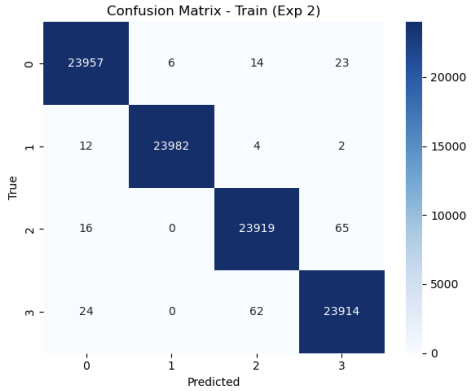


Figure A.10: Confusion Matrix - Experiment-2 training dataset

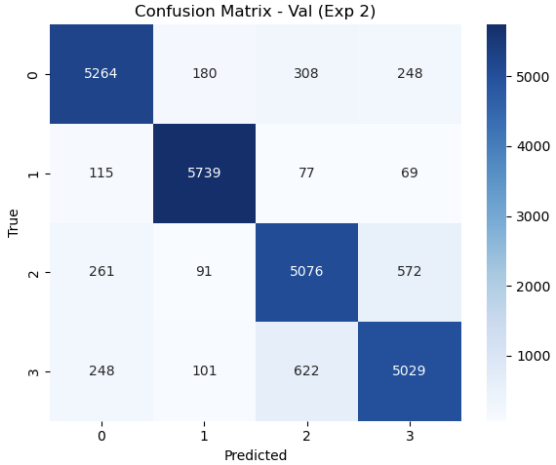


Figure A.11: Confusion Matrix - Experiment-2 validation dataset

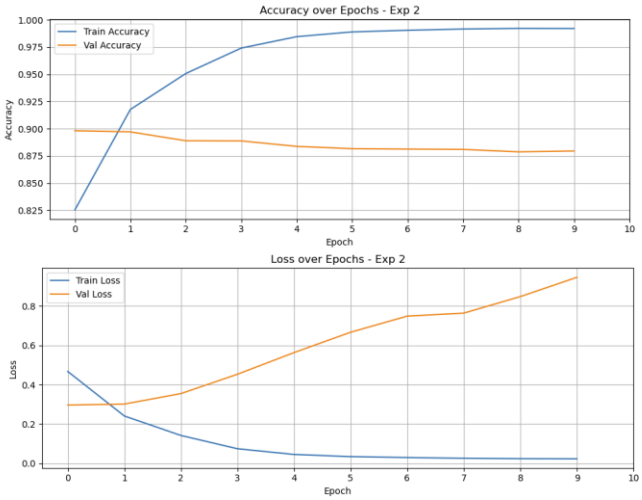


Figure A.12: Model training Exp-2 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

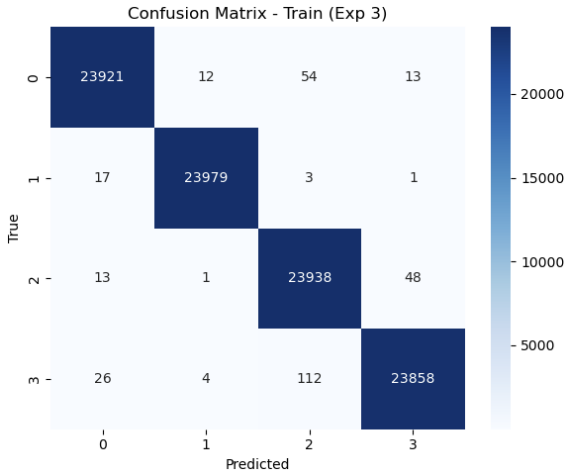


Figure A.13: Confusion Matrix - Experiment-3 training dataset

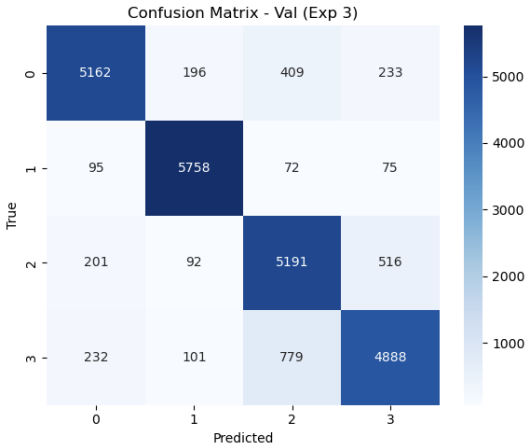


Figure A.14: Confusion Matrix - Experiment-3 validation dataset

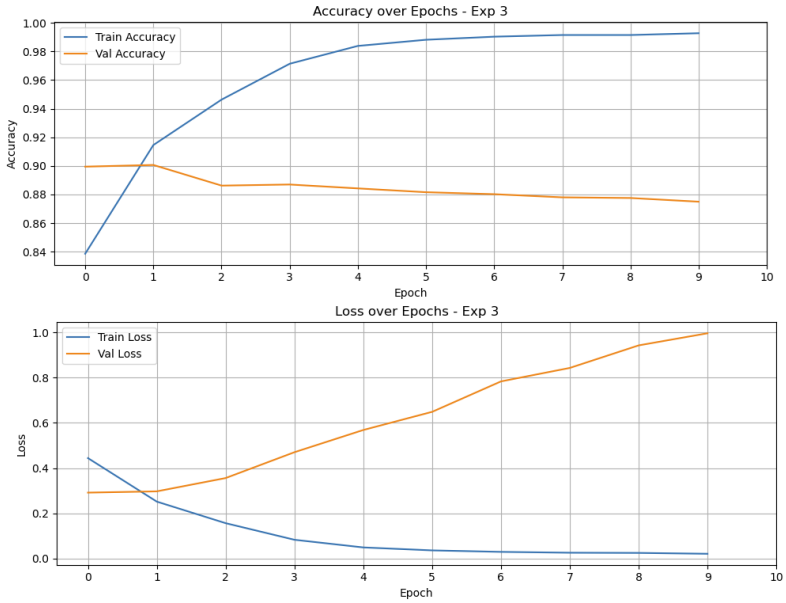


Figure A.15: Model training Exp-3 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

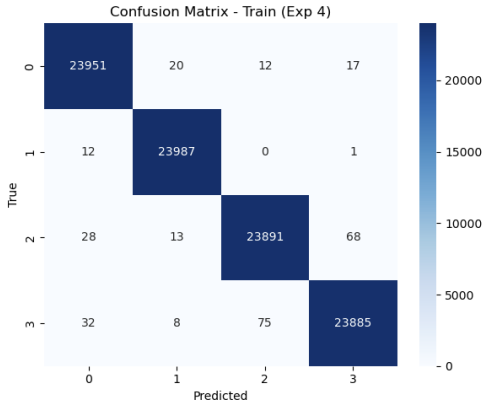


Figure A.16: Confusion Matrix - Experiment-4 training dataset

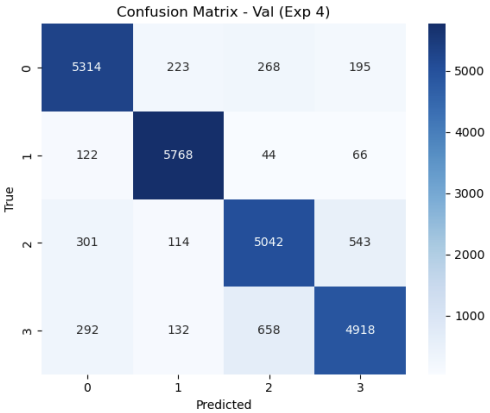


Figure A.17: Confusion Matrix - Experiment-4 validation dataset

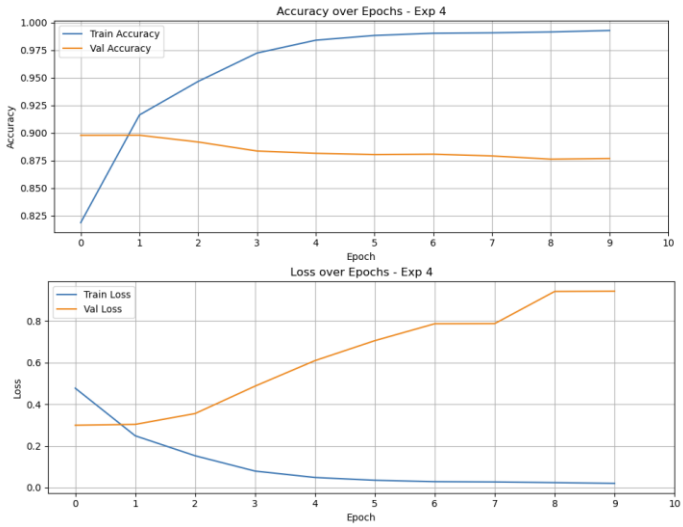


Figure A.18: Model training Exp-4 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

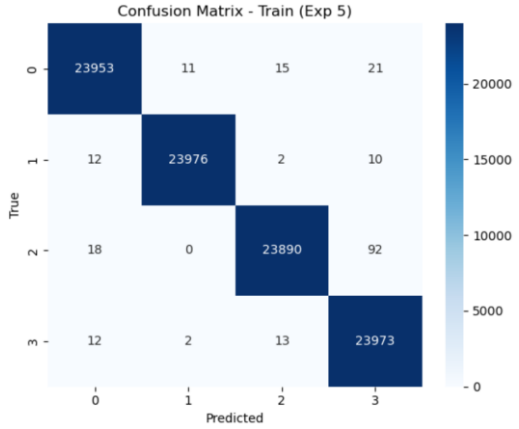


Figure A.19: Confusion Matrix - Experiment-5 training dataset

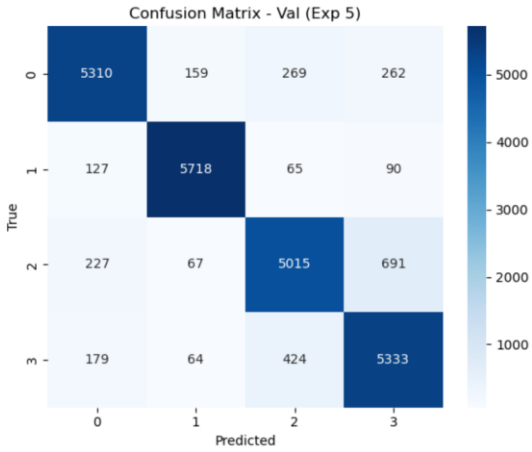


Figure A.20: Confusion Matrix - Experiment-5 validation dataset

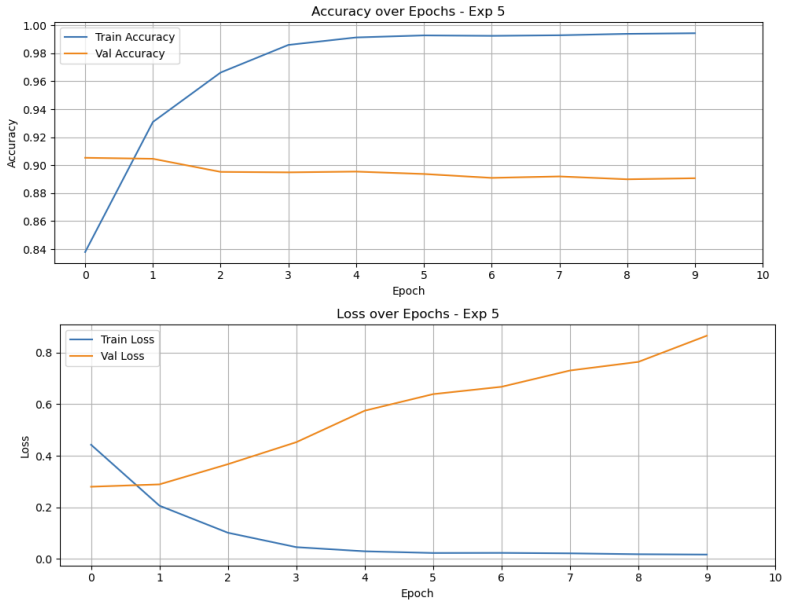


Figure A.21: Model training Exp-5 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

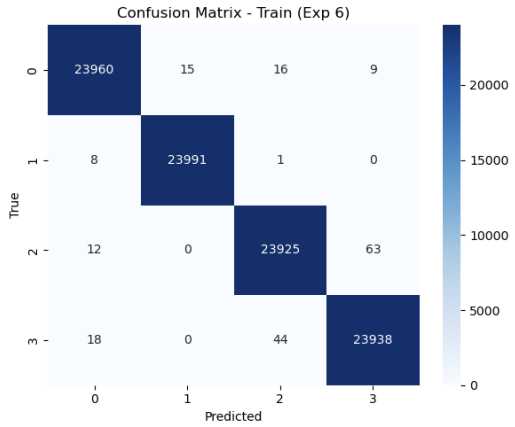


Figure A.22: Confusion Matrix - Experiment-6 training dataset

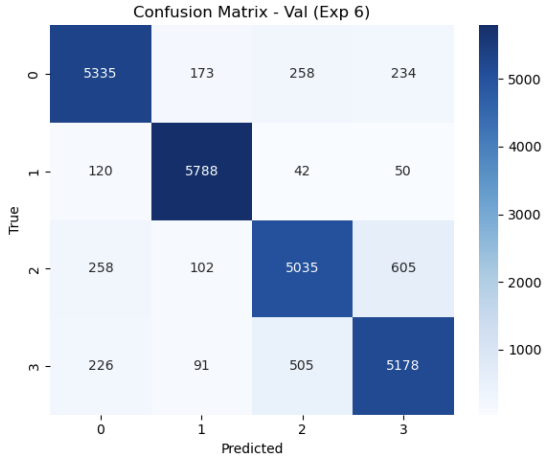


Figure A.23: Confusion Matrix - Experiment-6 validation dataset

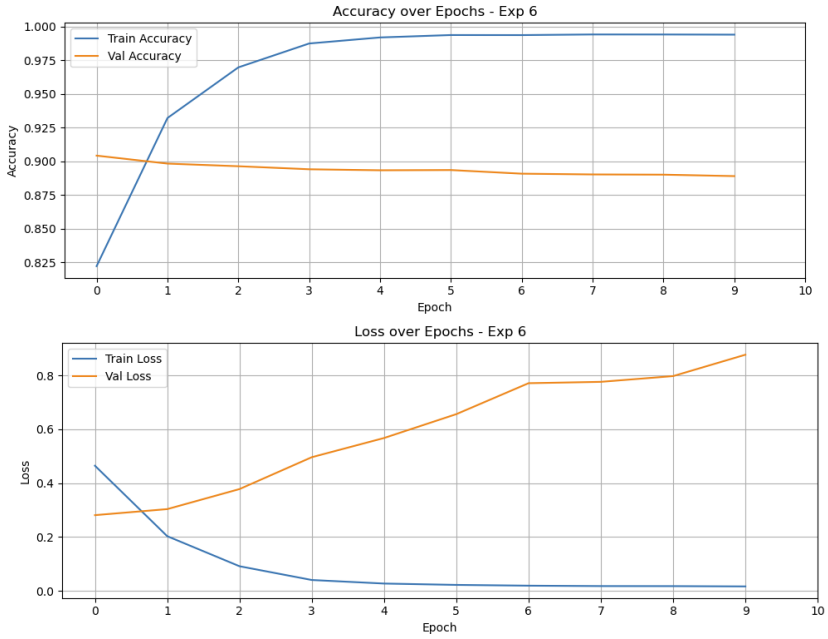


Figure A.24: Model training Exp-6 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

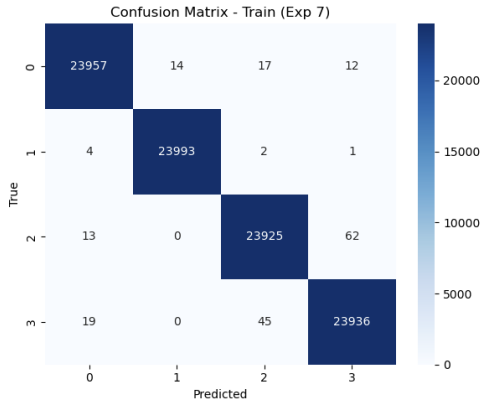


Figure A.25: Confusion Matrix - Experiment-7 training dataset

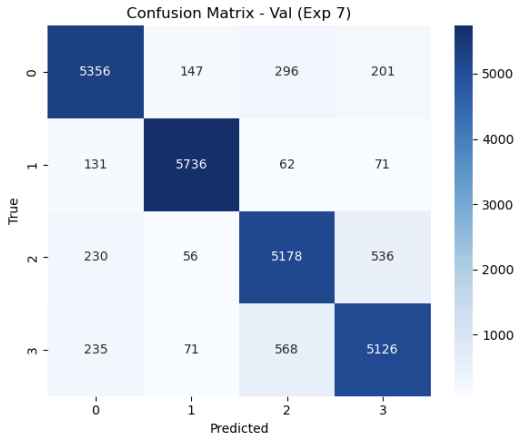


Figure A.26: Confusion Matrix - Experiment-7 validation dataset

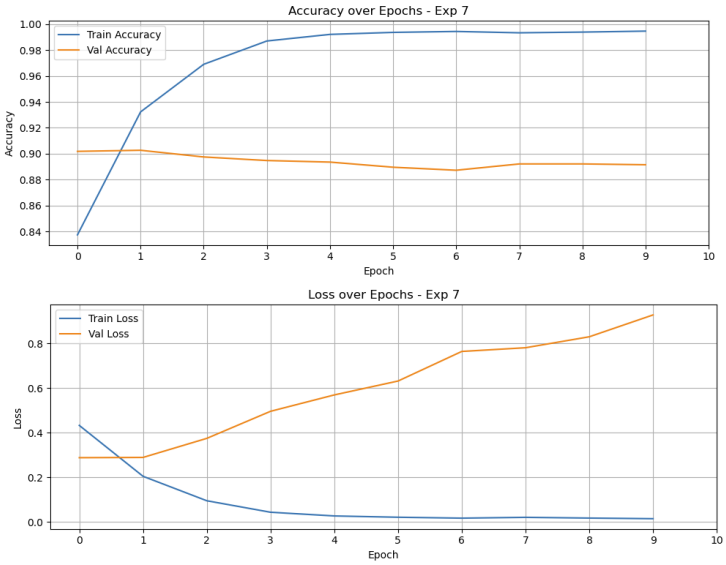


Figure A.27: Model training Exp-7 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

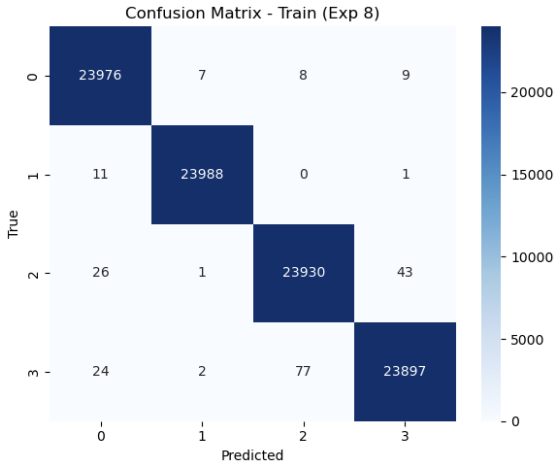


Figure A.28: Confusion Matrix - Experiment-8 training dataset

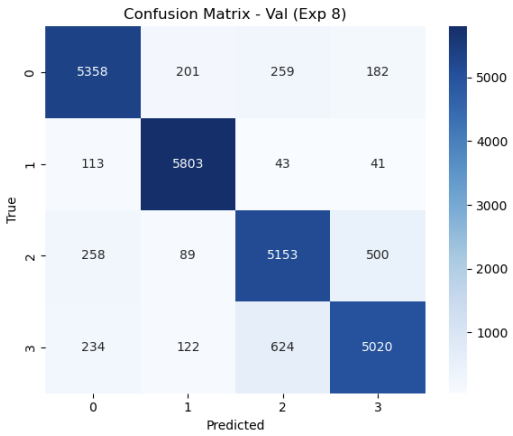


Figure A.29: Confusion Matrix - Experiment-8 validation dataset

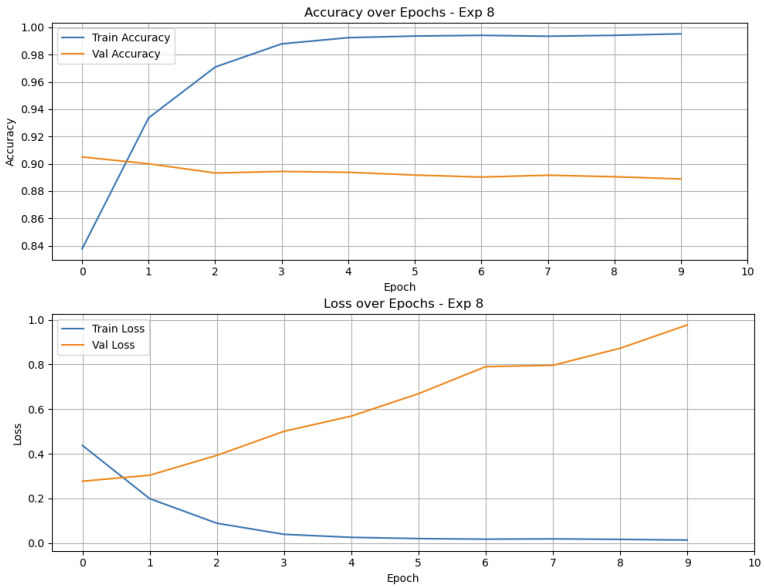


Figure A.30: Model training Exp-8 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

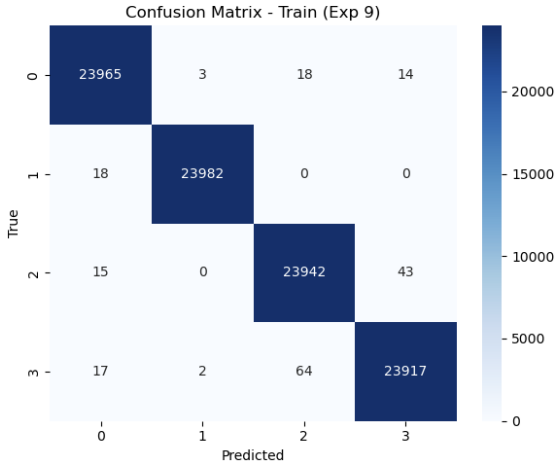


Figure A.31: Confusion Matrix - Experiment-9 training dataset

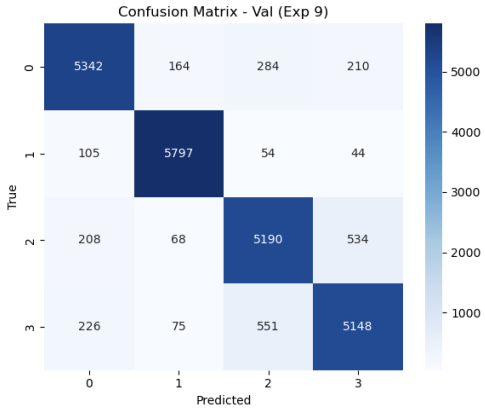


Figure A.32: Confusion Matrix - Experiment-9 validation dataset

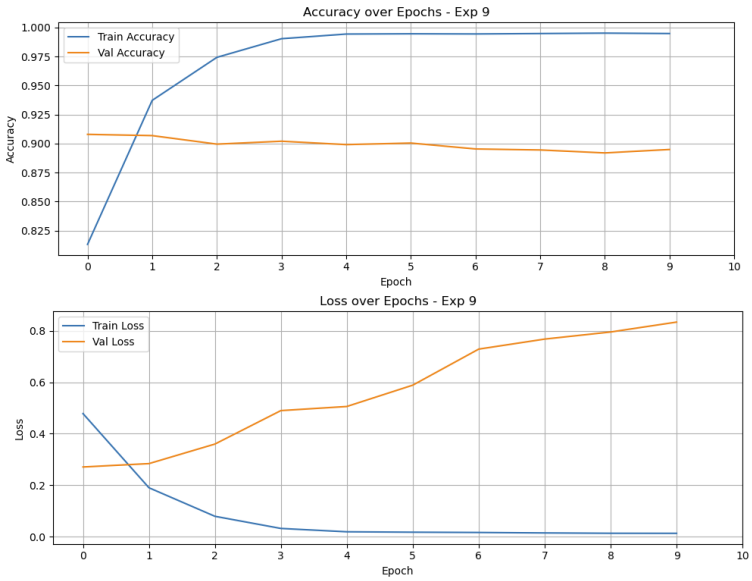


Figure A.33: Model training Exp-9 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

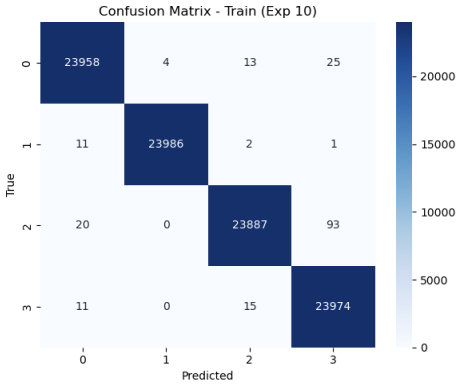


Figure A.34: Confusion Matrix - Experiment-10 training dataset

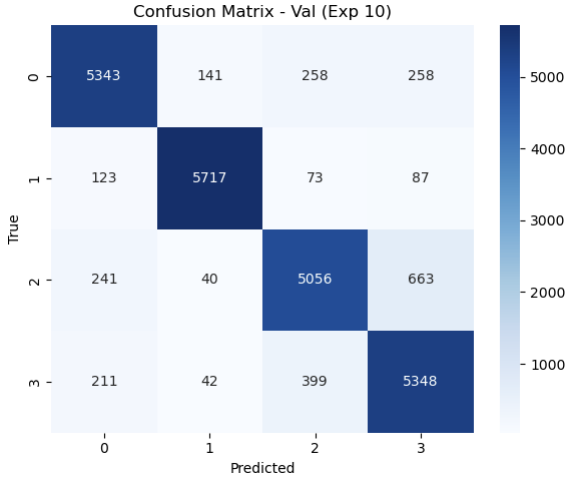


Figure A.35: Confusion Matrix - Experiment-10 validation dataset

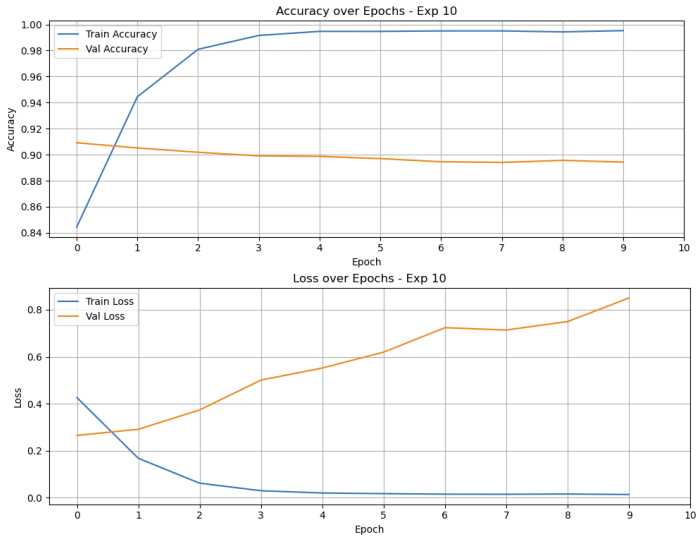


Figure A.36: Model training Exp-10 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

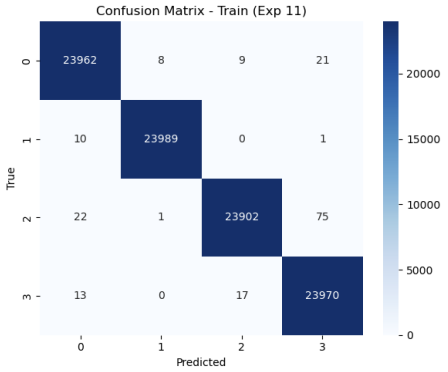


Figure A.37: Confusion Matrix - Experiment-11 training dataset

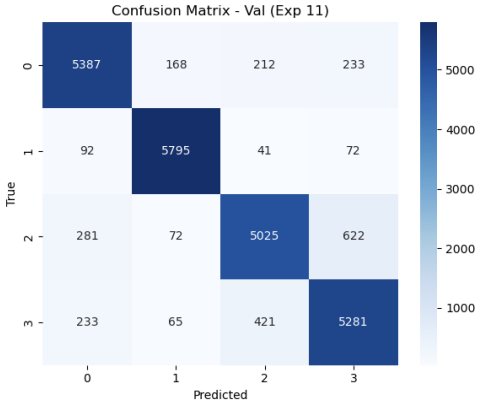


Figure A.38: Confusion Matrix - Experiment-11 validation dataset

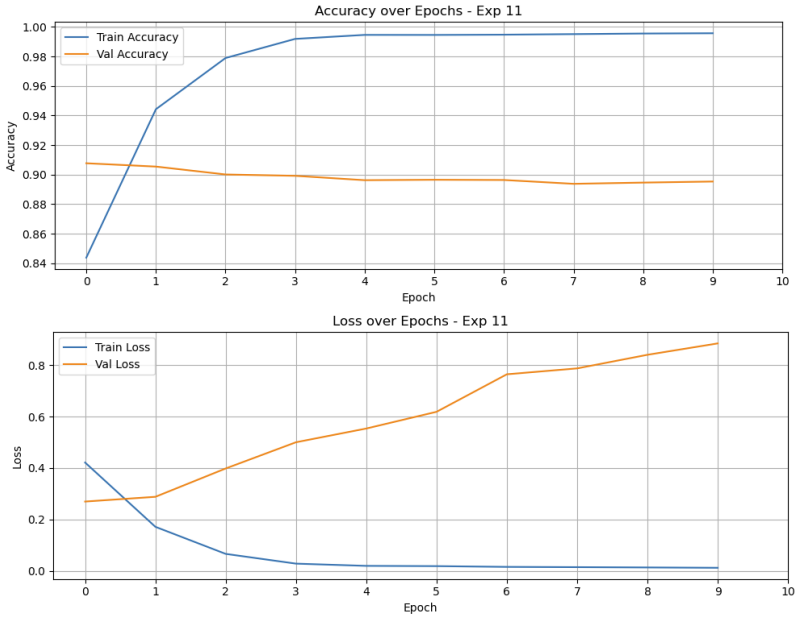


Figure A.39: Model training Exp-11 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

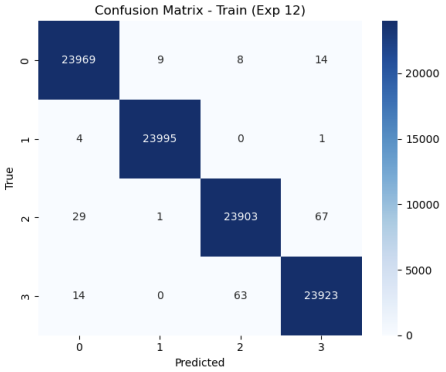


Figure A.40: Confusion Matrix - Experiment-12 training dataset

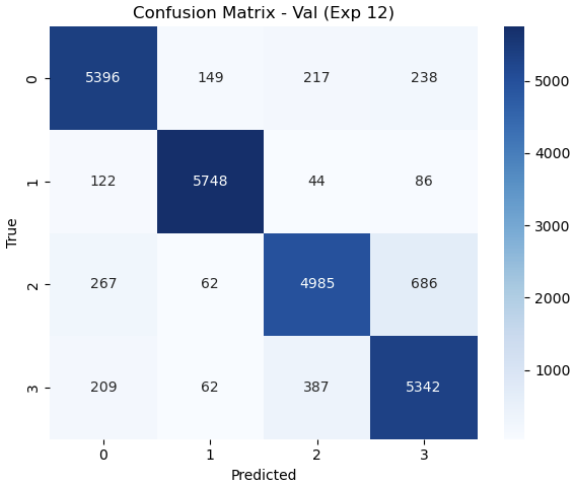


Figure A.41: Confusion Matrix - Experiment-12 validation dataset

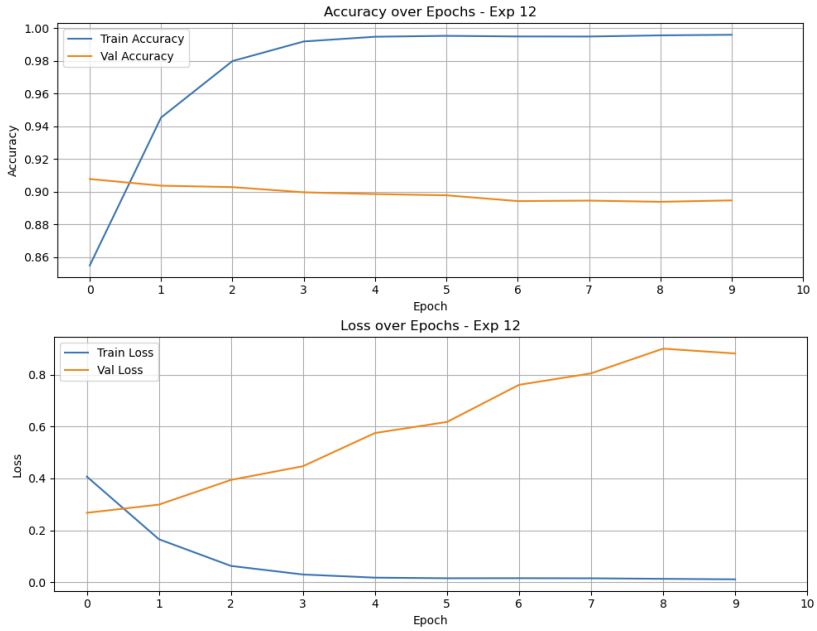


Figure A.42: Model training Exp-12 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

| Experiment | Vocab Size | Stopwords Removed | Sequence Length | Train Accuracy | Val Accuracy | Train Loss | Val Loss | Training Time (s) |
|---|---|---|---|---|---|---|---|---|
| 10 | 20000 | TRUE | Natural | 0.995719 | 0.895333 | 0.011814 | 0.88518 | 309.97 |
| 8 | 20000 | FALSE | Natural | 0.994875 | 0.894875 | 0.013742 | 0.833159 | 321.33 |
| 11 | 20000 | TRUE | Fixed(128) | 0.995844 | 0.894625 | 0.010993 | 0.881515 | 228.99 |
| 9 | 20000 | FALSE | Fixed(128) | 0.995292 | 0.894333 | 0.013681 | 0.850208 | 234.31 |
| 6 | 10000 | TRUE | Natural | 0.994677 | 0.8915 | 0.014337 | 0.928586 | 199.73 |
| 4 | 10000 | FALSE | Natural | 0.994385 | 0.890667 | 0.016715 | 0.865492 | 182.17 |
| 5 | 10000 | FALSE | Fixed(128) | 0.994073 | 0.889 | 0.016116 | 0.877041 | 169.29 |
| 7 | 10000 | TRUE | Fixed(128) | 0.99526 | 0.888917 | 0.013677 | 0.977204 | 190.75 |
| 0 | 5000 | FALSE | Natural | 0.99199 | 0.8795 | 0.023002 | 0.914442 | 211.93 |
| 1 | 5000 | FALSE | Fixed(128) | 0.992063 | 0.8795 | 0.022832 | 0.946696 | 166.5 |
| 3 | 5000 | TRUE | Fixed(128) | 0.992906 | 0.87675 | 0.0205 | 0.942232 | 167.23 |
| 2 | 5000 | TRUE | Natural | 0.992698 | 0.874958 | 0.020969 | 0.996274 | 189.31 |

Table A.1: Model Performance Under Varying Vocabulary Sizes, Stopword Removal, and Sequence Length Configurations

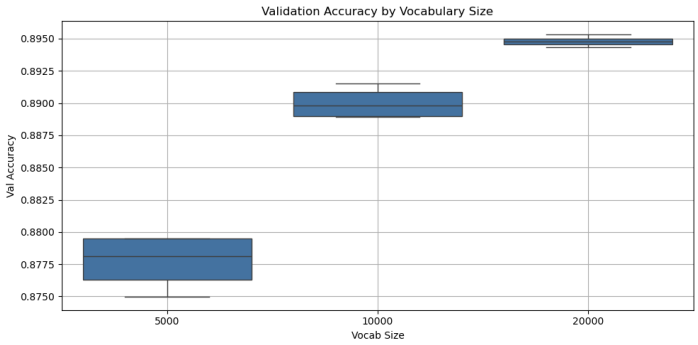


Figure A.43: Validation Accuracy Vs Vocabulary Size

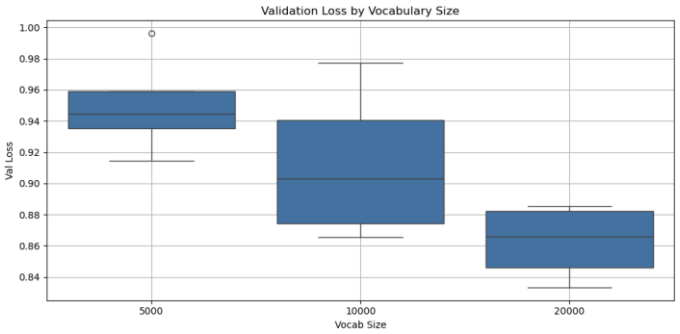


Figure A.44: Validation Loss Vs Vocabulary Size

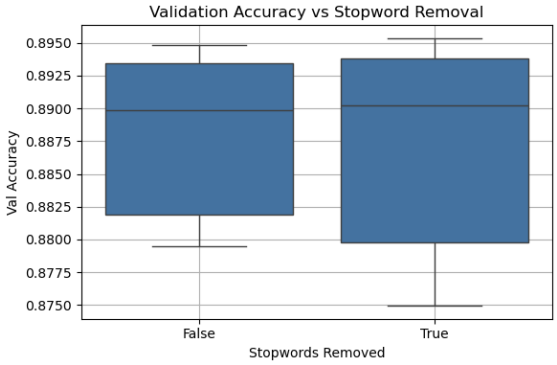


Figure A.45: Validation Accuracy Vs Stopword Removal

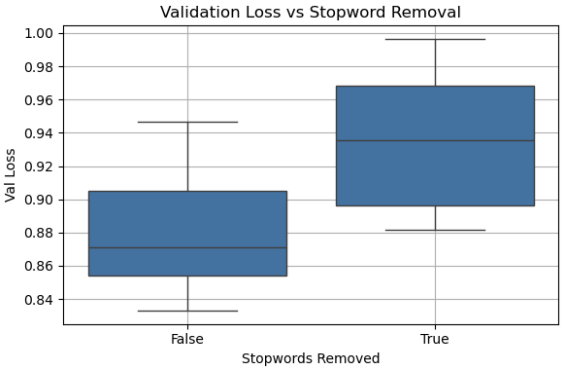


Figure A.46: Validation Loss Vs Stopword Removal

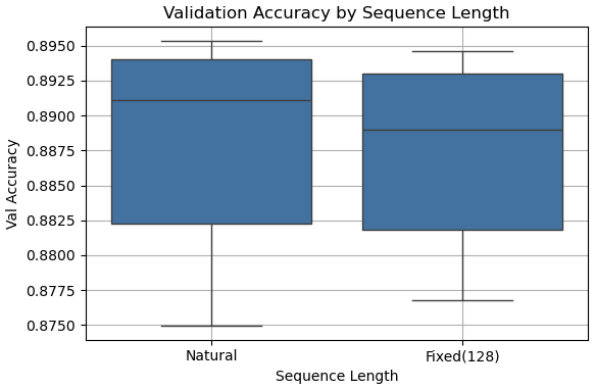


Figure A.47: Validation Accuracy Vs Sequence length

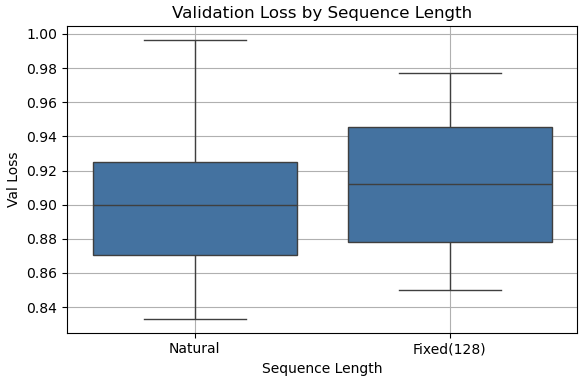


Figure A.48: Validation Loss Vs Sequence length

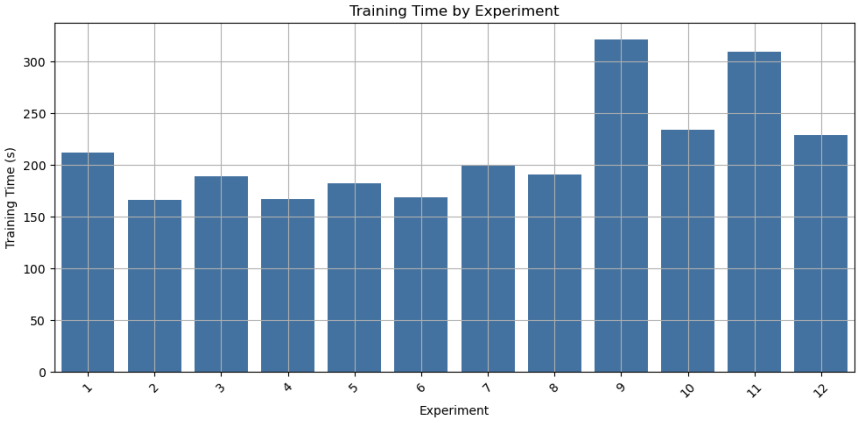


Figure A.49: Training Time by Experiments

APPENDIX B – EXPERIMENT-B SIMPLE RNN (Uni & Bi-Directional)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.25 | 1 | 0.4 | 24000 |
| 1 | 0 | 0 | 0 | 24000 |
| 2 | 0 | 0 | 0 | 24000 |
| 3 | 0.94 | 0 | 0 | 24000 |
| Accuracy | | | 0.25 | 96000 |
| Macro Avg | 0.3 | 0.25 | 0.1 | 96000 |
| Weighted Avg | 0.3 | 0.25 | 0.1 | 96000 |

Table B.1: Classification Report B1 Simple RNN Uni- Directional (Train dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.25 | 1 | 0.4 | 6000 |
| 1 | 0 | 0 | 0 | 6000 |
| 2 | 0 | 0 | 0 | 6000 |
| 3 | 0.8 | 0 | 0 | 6000 |
| Accuracy | | | 0.25 | 24000 |
| Macro Avg | 0.26 | 0.25 | 0.1 | 24000 |
| Weighted Avg | 0.26 | 0.25 | 0.1 | 24000 |

Table B.2: Classification Report B1 Simple RNN Uni- Directional (Validation dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.25 | 1 | 0.4 | 1900 |
| 1 | 0 | 0 | 0 | 1900 |
| 2 | 0 | 0 | 0 | 1900 |
| 3 | 0 | 0 | 0 | 1900 |
| Accuracy | | | 0.25 | 7600 |
| Macro Avg | 0.06 | 0.25 | 0.1 | 7600 |
| Weighted Avg | 0.06 | 0.25 | 0.1 | 7600 |

Table B.3: Classification Report B1 Simple RNN Uni- Directional (Test dataset)

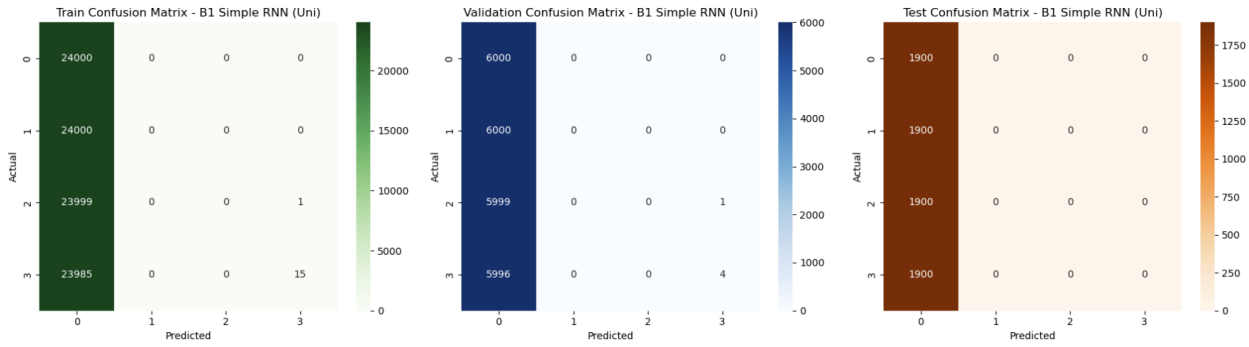


Figure B.1: Confusion Matrix Experiment-B1 -Train (left), Validation (Centre), and Test (Right)

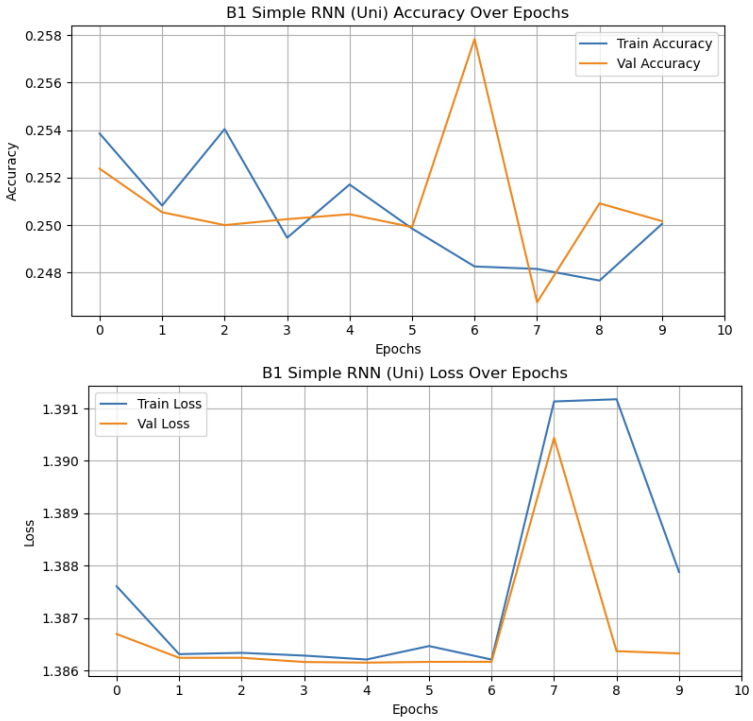


Figure B.2: Model training Exp-B1 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.94 | 0.85 | 0.89 | 24,000 |
| 1 | 0.91 | 0.99 | 0.95 | 24,000 |
| 2 | 0.9 | 0.82 | 0.85 | 24,000 |
| 3 | 0.83 | 0.9 | 0.86 | 24,000 |
| Accuracy | | | 0.89 | 96,000 |
| Macro Avg | 0.89 | 0.89 | 0.89 | 96,000 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 96,000 |

Table B.4: Classification Report B2 Simple RNN Uni- Directional (Train dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.95 | 0.86 | 0.9 | 6,000 |
| 1 | 0.9 | 0.99 | 0.94 | 6,000 |
| 2 | 0.89 | 0.8 | 0.85 | 6,000 |
| 3 | 0.82 | 0.91 | 0.86 | 6,000 |
| Accuracy | | | 0.89 | 24,000 |
| Macro Avg | 0.89 | 0.89 | 0.89 | 24,000 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 24,000 |

Table B.5: Classification Report B2 Simple RNN Uni- Directional (Validation dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.93 | 0.86 | 0.89 | 1,900 |
| 1 | 0.91 | 0.98 | 0.94 | 1,900 |
| 2 | 0.88 | 0.8 | 0.84 | 1,900 |
| 3 | 0.83 | 0.89 | 0.86 | 1,900 |
| Accuracy | | | 0.88 | 7,600 |
| Macro Avg | 0.88 | 0.88 | 0.88 | 7,600 |
| Weighted Avg | 0.88 | 0.88 | 0.88 | 7,600 |

Table B.6: Classification Report B2 Simple RNN Uni- Directional (Test dataset)

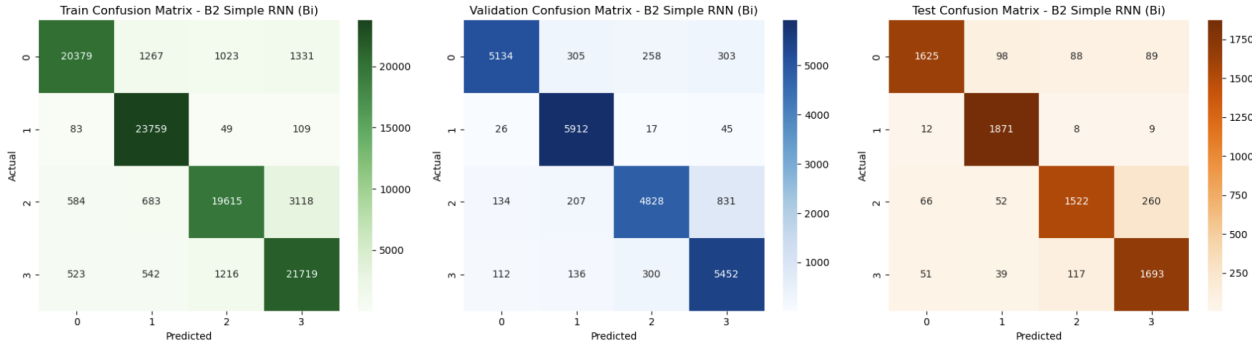


Figure B.3: Confusion Matrix Experiment-B2 -Train (left), Validation (Centre), and Test (Right)

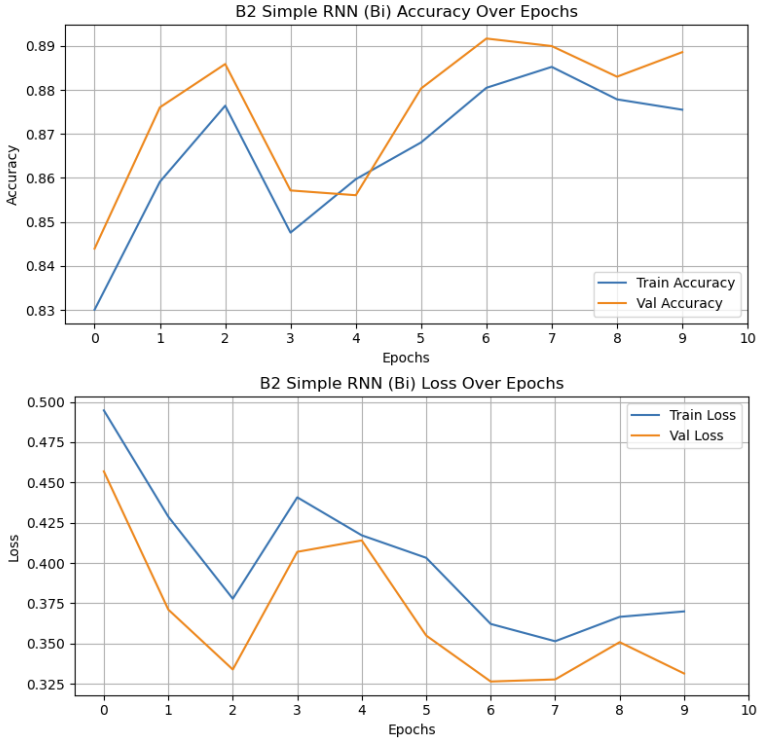


Figure B.4: Model training Exp-B2 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

## APPENDIX C – EXPERIMENT-C LSTM (Uni & Bi-Directional)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.93 | 0.88 | 0.9 | 24,000 |
| 1 | 0.96 | 0.98 | 0.97 | 24,000 |
| 2 | 0.92 | 0.82 | 0.87 | 24,000 |
| 3 | 0.82 | 0.95 | 0.88 | 24,000 |
| Accuracy | | | 0.9 | 96,000 |
| Macro Avg | 0.91 | 0.9 | 0.9 | 96,000 |
| Weighted Avg | 0.91 | 0.9 | 0.9 | 96,000 |

Table C.1: Classification Report C1 Simple RNN Uni- Directional (Train dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.93 | 0.88 | 0.9 | 6,000 |
| 1 | 0.95 | 0.97 | 0.96 | 6,000 |
| 2 | 0.91 | 0.8 | 0.86 | 6,000 |
| 3 | 0.81 | 0.94 | 0.87 | 6,000 |
| Accuracy | | | 0.9 | 24,000 |
| Macro Avg | 0.9 | 0.9 | 0.9 | 24,000 |
| Weighted Avg | 0.9 | 0.9 | 0.9 | 24,000 |

Table C.2: Classification Report C1 Simple RNN Uni- Directional (Validation dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.92 | 0.87 | 0.9 | 1,900 |
| 1 | 0.95 | 0.97 | 0.96 | 1,900 |
| 2 | 0.91 | 0.79 | 0.85 | 1,900 |
| 3 | 0.81 | 0.94 | 0.87 | 1,900 |
| Accuracy | | | 0.89 | 7,600 |
| Macro Avg | 0.9 | 0.89 | 0.89 | 7,600 |
| Weighted Avg | 0.9 | 0.89 | 0.89 | 7,600 |

Table C.3: Classification Report C1 Simple RNN Uni- Directional (Test dataset)

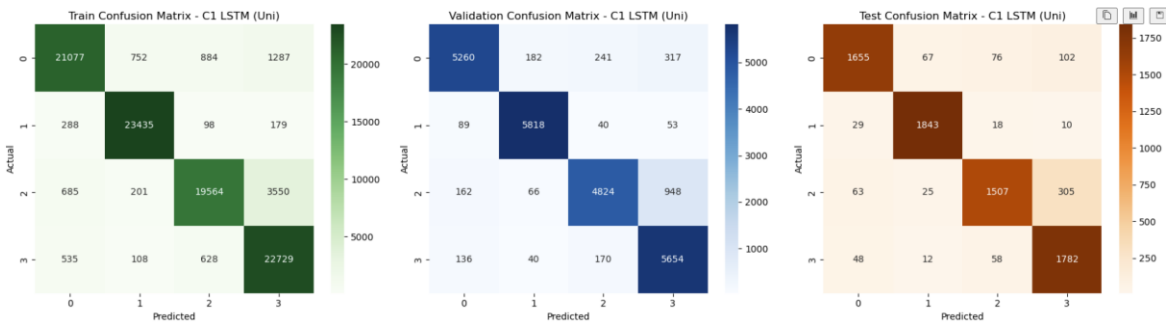


Figure C.1: Confusion Matrix Experiment-C1 -Train (left), Validation (Centre), and Test (Right)

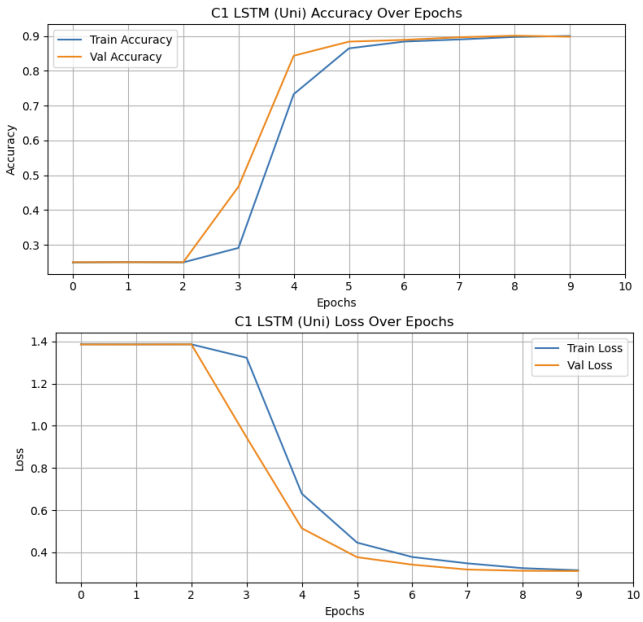


Figure C.2: Model training Exp-C1 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.97 | 0.94 | 0.95 | 24,000 |
| 1 | 0.98 | 0.99 | 0.99 | 24,000 |
| 2 | 0.94 | 0.91 | 0.93 | 24,000 |
| 3 | 0.91 | 0.95 | 0.93 | 24,000 |
| Accuracy | | | 0.95 | 96,000 |
| Macro Avg | 0.95 | 0.95 | 0.95 | 96,000 |
| Weighted Avg | 0.95 | 0.95 | 0.95 | 96,000 |

Table C.4: Classification Report C2 Simple RNN Uni- Directional (Train dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.94 | 0.92 | 0.93 | 6,000 |
| 1 | 0.96 | 0.97 | 0.97 | 6,000 |
| 2 | 0.9 | 0.87 | 0.89 | 6,000 |
| 3 | 0.87 | 0.92 | 0.9 | 6,000 |
| Accuracy | | | 0.92 | 24,000 |
| Macro Avg | 0.92 | 0.92 | 0.92 | 24,000 |
| Weighted Avg | 0.92 | 0.92 | 0.92 | 24,000 |

Table C.5: Classification Report C2 Simple RNN Uni- Directional (Validation dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.92 | 0.91 | 0.91 | 1,900 |
| 1 | 0.96 | 0.97 | 0.97 | 1,900 |
| 2 | 0.89 | 0.86 | 0.88 | 1,900 |
| 3 | 0.87 | 0.91 | 0.89 | 1,900 |
| Accuracy | | | 0.91 | 7,600 |
| Macro Avg | 0.91 | 0.91 | 0.91 | 7,600 |
| Weighted Avg | 0.91 | 0.91 | 0.91 | 7,600 |

Table C.6: Classification Report C2 Simple RNN Uni- Directional (Test dataset)

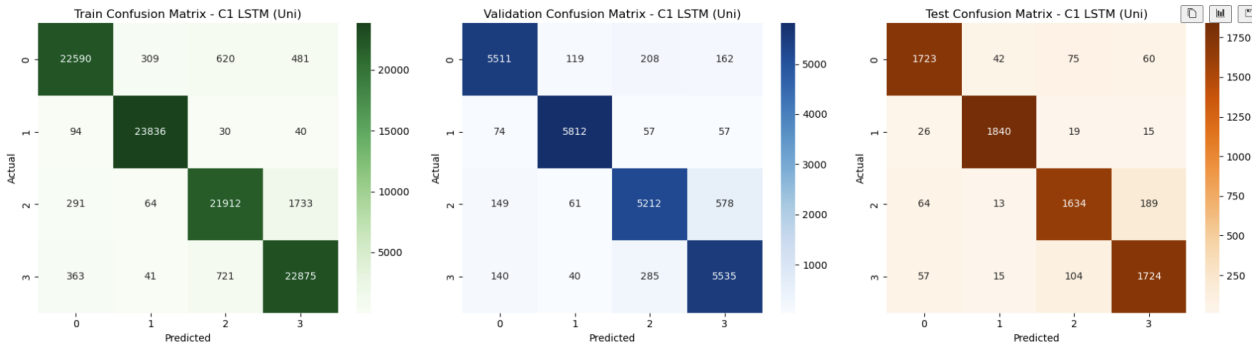


Figure C.3: Confusion Matrix Experiment-C2 -Train (left), Validation (Centre), and Test (Right)

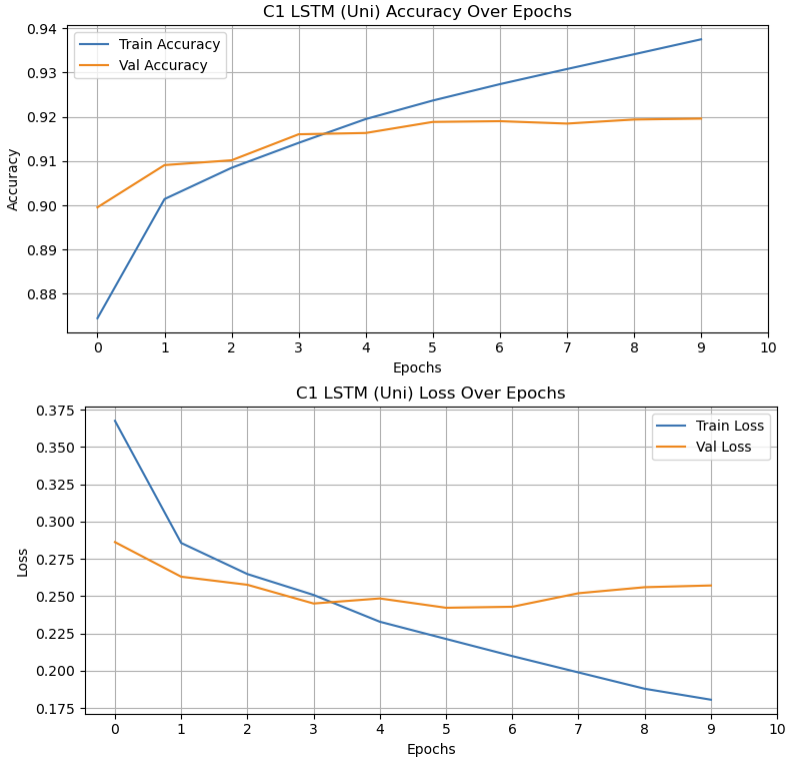


Figure C.4: Model training Exp-C2 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

APPENDIX D – EXPERIMENT-D 1D CNN:

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.97 | 0.96 | 0.97 | 24,000 |
| 1 | 0.99 | 0.99 | 0.99 | 24,000 |
| 2 | 0.96 | 0.92 | 0.94 | 24,000 |
| 3 | 0.92 | 0.97 | 0.95 | 24,000 |
| Accuracy | | | 0.96 | 96,000 |
| Macro Avg | 0.96 | 0.96 | 0.96 | 96,000 |
| Weighted Avg | 0.96 | 0.96 | 0.96 | 96,000 |

Table D.1: Classification Report C1 Simple RNN Uni- Directional (Train dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.91 | 0.92 | 0.92 | 6,000 |
| 1 | 0.96 | 0.97 | 0.97 | 6,000 |
| 2 | 0.9 | 0.84 | 0.87 | 6,000 |
| 3 | 0.86 | 0.91 | 0.88 | 6,000 |
| Accuracy | | | 0.91 | 24,000 |
| Macro Avg | 0.91 | 0.91 | 0.91 | 24,000 |
| Weighted Avg | 0.91 | 0.91 | 0.91 | 24,000 |

Table D.2: Classification Report C1 Simple RNN Uni- Directional (Validation dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.92 | 0.92 | 0.92 | 1,900 |
| 1 | 0.96 | 0.97 | 0.97 | 1,900 |
| 2 | 0.9 | 0.85 | 0.87 | 1,900 |
| 3 | 0.87 | 0.91 | 0.89 | 1,900 |
| Accuracy | | | 0.91 | 7,600 |
| Macro Avg | 0.91 | 0.91 | 0.91 | 7,600 |
| Weighted Avg | 0.91 | 0.91 | 0.91 | 7,600 |

Table D.3: Classification Report C1 Simple RNN Uni- Directional (Test dataset)

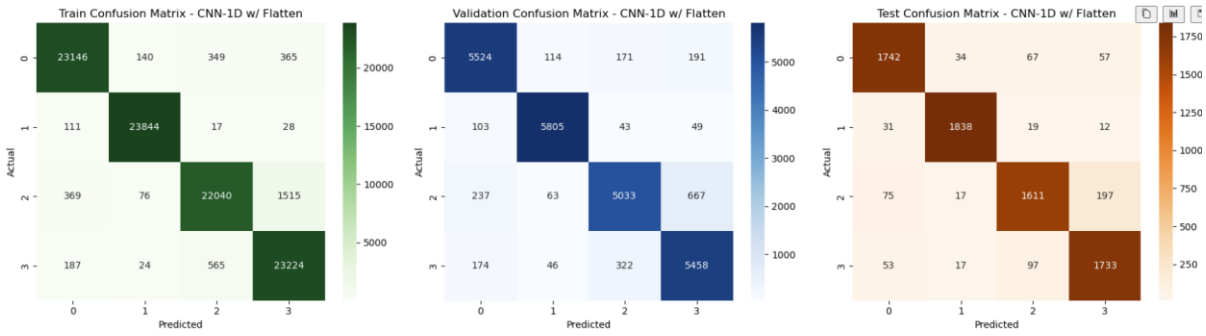


Figure D.1: Confusion Matrix Experiment-C1 -Train (left), Validation (Centre), and Test (Right)

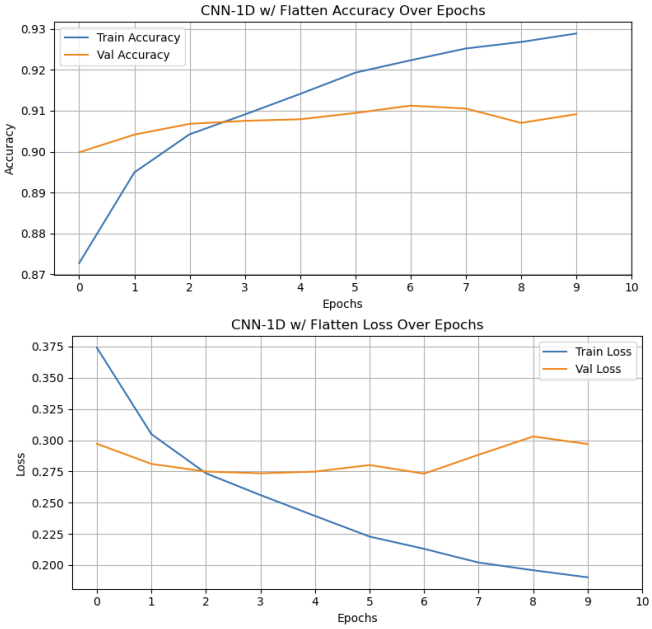


Figure D.2: Model training Exp-C1 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.97 | 0.96 | 0.97 | 24,000 |
| 1 | 0.99 | 0.99 | 0.99 | 24,000 |
| 2 | 0.96 | 0.92 | 0.94 | 24,000 |
| 3 | 0.92 | 0.97 | 0.95 | 24,000 |
| Accuracy | | | 0.96 | 96,000 |
| Macro Avg | 0.96 | 0.96 | 0.96 | 96,000 |
| Weighted Avg | 0.96 | 0.96 | 0.96 | 96,000 |

Table D.4: Classification Report C2 Simple RNN Uni- Directional (Train dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.91 | 0.9 | 0.9 | 6,000 |
| 1 | 0.95 | 0.97 | 0.96 | 6,000 |
| 2 | 0.88 | 0.86 | 0.87 | 6,000 |
| 3 | 0.87 | 0.88 | 0.87 | 6,000 |
| Accuracy | | | 0.9 | 24,000 |
| Macro Avg | 0.9 | 0.9 | 0.9 | 24,000 |
| Weighted Avg | 0.9 | 0.9 | 0.9 | 24,000 |

Table D.5: Classification Report C2 Simple RNN Uni- Directional (Validation dataset)

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.91 | 0.9 | 0.91 | 1,900 |
| 1 | 0.94 | 0.97 | 0.95 | 1,900 |
| 2 | 0.86 | 0.85 | 0.85 | 1,900 |
| 3 | 0.86 | 0.86 | 0.86 | 1,900 |
| Accuracy | | | 0.89 | 7,600 |
| Macro Avg | 0.89 | 0.89 | 0.89 | 7,600 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 7,600 |

Table D.6: Classification Report C2 Simple RNN Uni- Directional (Test dataset)

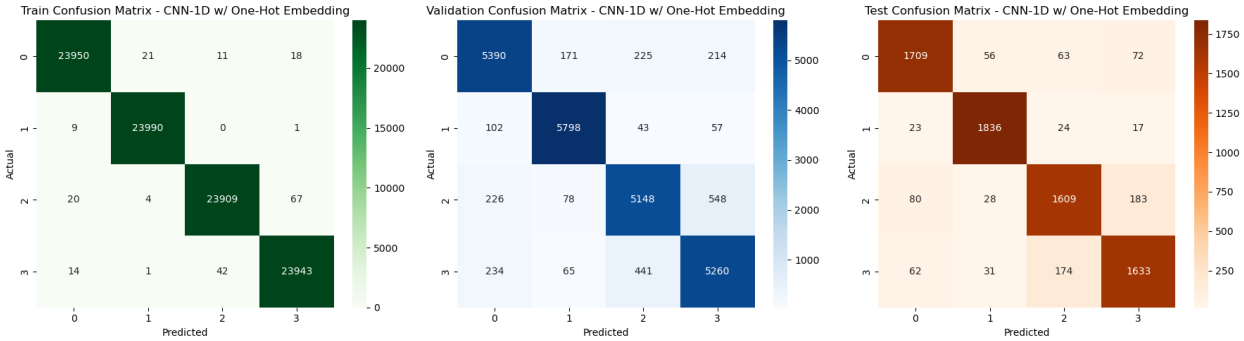


Figure D.3: Confusion Matrix Experiment-C2 -Train (left), Validation (Centre), and Test (Right)

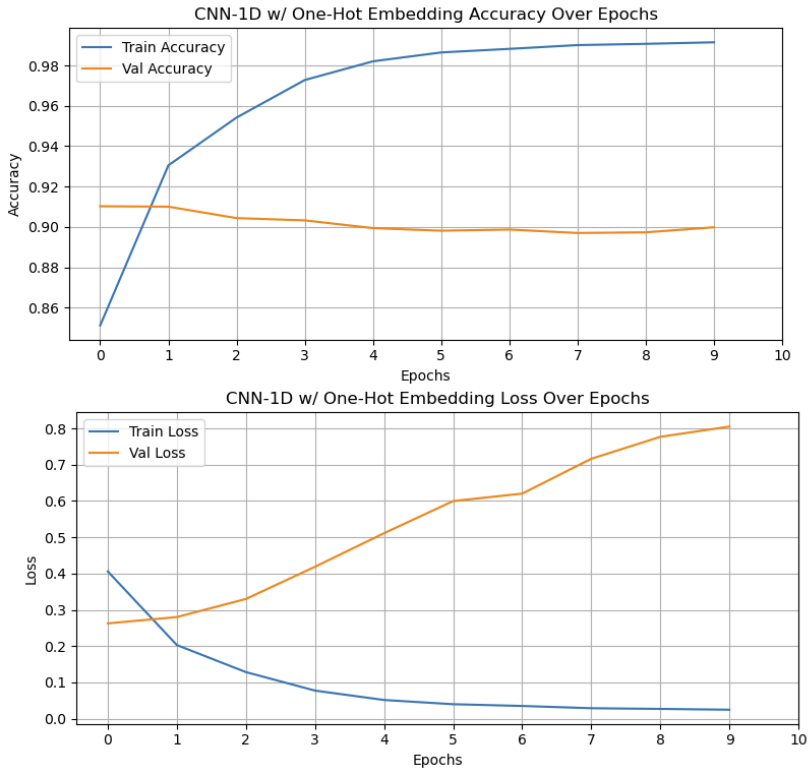


Figure D.4: Model training Exp-C2 (Top: Accuracy vs Epochs; Bottom: Log Loss Vs Epochs)

| Experiment | Train Accuracy | Val Accuracy | Train Loss | Val Loss | Test Accuracy | Training Time (s) |
|---|---|---|---|---|---|---|
| B1 Simple RNN (Uni) | 0.2501 | 0.2502 | 1.3879 | 1.386 | 0.25 | 245.12 |
| B2 Simple RNN (Bi) | 0.8755 | 0.8886 | 0.37 | 0.331 | 0.883 | 312.82 |
| C1 LSTM (Uni) | 0.9001 | 0.8982 | 0.3147 | 0.311 | 0.893 | 625.09 |
| C2 LSTM (Bi) | 0.9375 | 0.9196 | 0.1806 | 0.257 | 0.9107 | 1110.05 |
| CNN-1D w/ Flatten | 0.9289 | 0.9092 | 0.1902 | 0.297 | 0.9111 | 408.59 |
| CNN-1D w/ One-Hot Embedding | 0.9915 | 0.8998 | 0.0249 | 0.805 | 0.893 | 600.11 |

Table D.7: Model Performance Comparison