Dataset Summary

In this assignment, the IMDB Movie Review Dataset served as the foundation for this comparative analysis of Recurrent Neural Network (RNN) architectures in regard to sentiment classification. This dataset comprised of 50,000 movie reviews evenly distributed between both positive and negative sentiments, providing a balanced binary classification task that is well suited for evaluating different neural network architectures.

Preprocessing began with a comprehensive text cleaning procedure. All of the reviews were converted to lowercase to ensure consistency, and HTML tags, punctuation, and special characters were systematically removed to focus the model's attention on semantic content rather than formatting artifacts. The text was then tokenized using whitespace splitting. Following this, the top 10,000 most frequent words were retained, striking a balance between capturing semantic information and maintaining computational efficiency. Two tokens were incorporated into the vocabulary: a padding token (PAD) at index 0 for the sequence length normalization, and an unknown token (UNK) at index 1 for handling OOV words.

The dataset was split evenly into training and test sets, with 25,000 reviews allocated to each. This stratified splitting procedure ensured that both positive and negative sentiments were equally represented in both sets, preventing any class imbalance issues.

Model Configuration

All of the configurations in this assignment shared a common foundation to ensure the fair comparison across different model types. The embedding layer, which transformed discrete token indices into continuous vector representations, was configured with a dimensionality of 100. This embedding size proved to be a sufficient capacity for capturing semantic relationships while maintaining reasonable computational requirements.

The recurrent architectures all utilized two hidden layers with 64 units per layer. This depth was chosen to provide the models with adequate capacity for learning hierarchical representations while avoiding the computational burden of deeper networks. The two-layer configuration allows the first layer to capture the local patterns while the second layer can learn more abstract, sentiment-relevant representations. The dropout regularization was set to 0.4 to mitigate any overfitting.

Three recurrent architectures were evaluated in the assignment. The basic RNN architecture utilized simple recurrent units with either “tanh” or “ReLU” activation functions. The basic RNN struggled with long-term dependencies due to the vanishing gradient problem. The LSTM was able to address this limitation through its gating mechanism, incorporating forget gates, input gates, and output gates that allow the model to selectively maintain or disregard information over longer sequences. The bidirectional LSTM extended this even further by processing sequences in both forward and backward directions, enabling the model to leverage both past and future context when making predictions at any position.

Training was conducted across five epochs with a batch size of 32 samples, which was a good balance between gradient estimate quality and memory efficiency. The three different optimizers were evaluated: Adam with its adaptive learning rates and momentum, stochastic gradient descent (SGD) with fixed learning rate and momentum term, and RMSProp with its adaptive learning rate scaling. All of the optimizers used an initial learning rate of 0.001.

Parameter counts varied across all of the architectures. The basic RNN contained the fewest parameters. LSTM was in between and contained approximately four times as many parameters as RNN. Bidirectional LSTM doubled the parameter count of the LSTM.

Comparative Analysis

The results revealed substantial performance differences across the fifteen experiments tested in the study. The optimal configuration achieved an accuracy of 79.89% with an F1 score of 0.7985, using an LSTM architecture with sigmoid activation, RMSProp optimizer, sequence length of 100, and gradient clipping enabled. This represented a significant improvement over the baseline LSTM configuration, which achieved 76.13% accuracy. The average epoch time for this configuration was 19.52 seconds, which is reasonable given the longer sequence length and the computational overhead of gradient clipping.

The comparison of the architectures revealed a clear performance hierarchy amongst the three evaluated. The basic RNN achieved only 64.90% accuracy with an F1 score of 0.6485. This relatively poor performance can be attributed to the vanishing gradient problem inherent in basic recurrent networks, limiting their ability to capture long-term dependencies that are crucial for sentiment analysis.

The LSTM demonstrated superior performance under the same baseline configuration, achieving 76.13% accuracy and an F1 score of 0.7609. This validates the effectiveness of LSTM’s gating mechanisms in preserving relevant information over longer sequences.

Bidirectional LSTMs achieved the highest performance with 76.48% accuracy with an F1 score of 0.7647. This was a marginal improvement over the standard LSTM. However it does suggest that there is some benefit by allowing the model to consider future context while making predictions. The computational cost of bidirectionality was substantial, with training time increasing to 20.48 seconds per epoch. This suggests that bidirectional LSTM might be of less value compared to standard LSTM.

Optimizer selection proved to be one of the most impactful hyperparameter choices. Under the baseline LSTM, Adam achieved 76.13% accuracy, RMSProp achieved 75.87% and SGD managed only 50.66% accuracy. Adam’s adaptive learning rates and bias correction make it particularly well-suited for the non-stationary optimization landscape of training RNNs.

Discussion

The LSTM architecture with sigmoid activation, RMSProp optimizer, sequence length 100, and gradient clipping enabled achieved the highest accuracy of 79.89% with an F1 core of 0.7985. The sigmoid activation’s bounded output range provided stability in the recurrent connections, while RMSProp’s adaptive learning rate scaling enabled effective optimization of the longer sequence representations. The sequence length of 100 allowed the model to access more complete review content compared to shorter sequences. Finally, gradient clipping may have contributed to a more stable optimization when combined with the longer sequences and sigmoid activation.

With an epoch time of 19.52 seconds, this configuration requires approximately 1.5 minutes to complete just five training epochs. For times when training time is critical, the baseline LSTM with tanh activation might be more well suited, as it completed five epochs in just 10.71 seconds, representing a 45% reduction in training time.

The impact of sequence length on performance also revealed crucial insights about the information requirements of sentiment classification. The dramatic increase from sequence length 25 to 50, represented a gain of 3.92 percentage points, demonstrating that movie reviews contain sentiment-relevant information beyond the first 25 tokens. The slight performance decrease from sequence length 50 to 100 suggests that there are diminishing returns. The optimal sequence length of 50 tokens appeared to capture sufficient semantic context while avoiding pitfalls of excessive length.

Different optimizers on model performance also highlighted the importance of adaptive learning rate mechanisms for recurrent architectures. SGD achieved only 50.66% accuracy, which demonstrated that fixed learning rates are unsuited for recurrent networks. The loss surface of an RNN exhibits extreme curvature and non-stationarity, with gradient magnitudes varying dramatically. SGD cannot adapt to this variability. However, Adam and RMSProp both address this issue through adaptive learning rates. The near equivalent performance of Adam at 76.13% and RMSProp at 75.87% suggests that adaptive learning rate mechanisms are more important than specific implementation details.

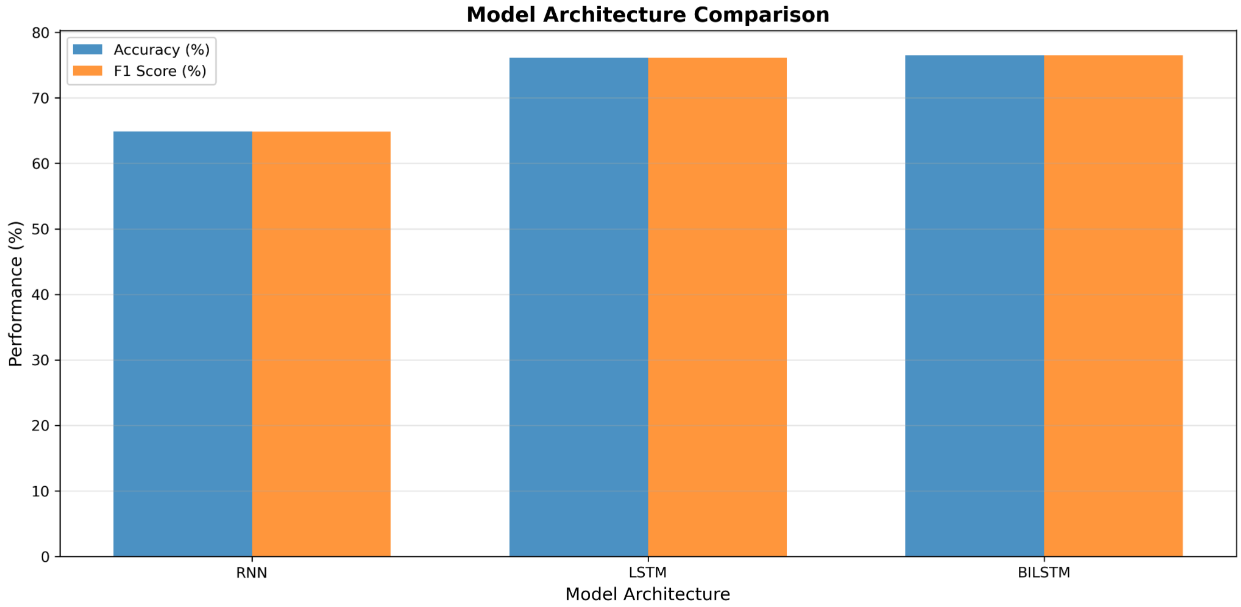
The success of the optimal configuration with sigmoid activation and RMSProp optimizer, rather than the more commonly used tanh and Adam combination, suggest that careful hyperparameter optimization can yield substantial benefits beyond traditional default choices. However, it should be noted that this optimal configuration emerged from a limited set of 15 experiments only, and a more comprehensive search might reveal better configurations.

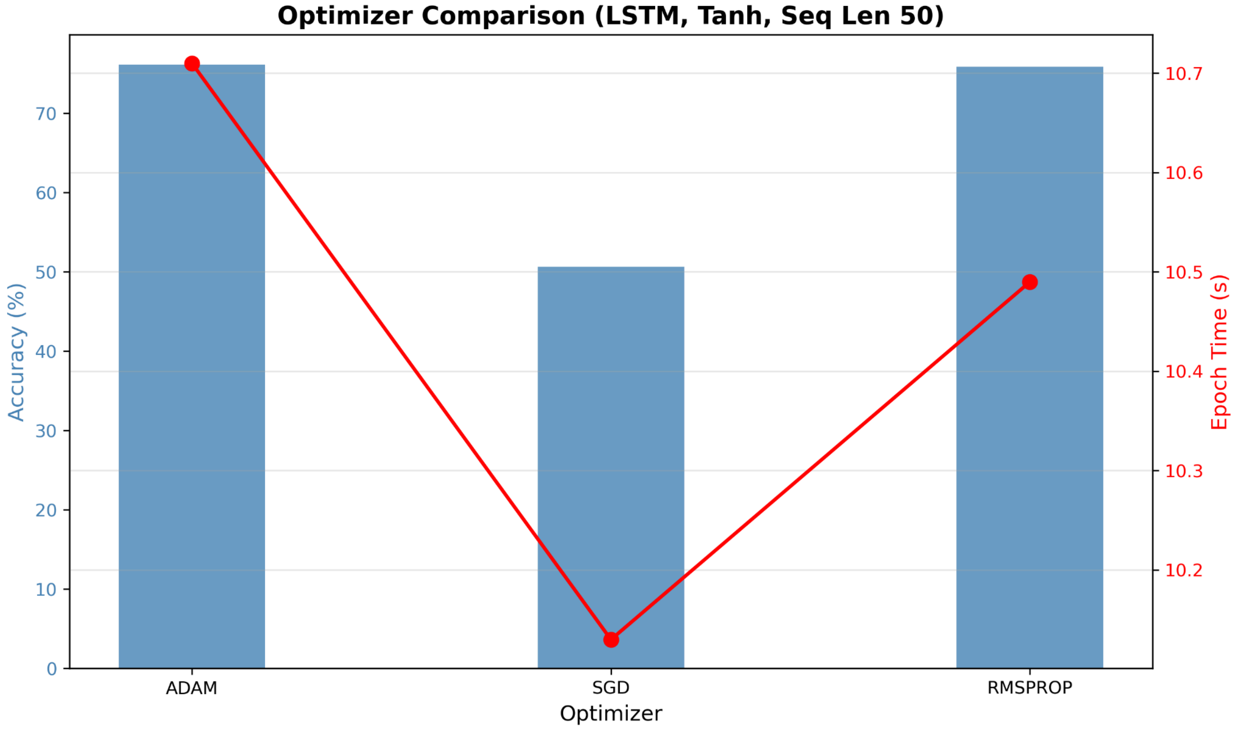
Conclusion

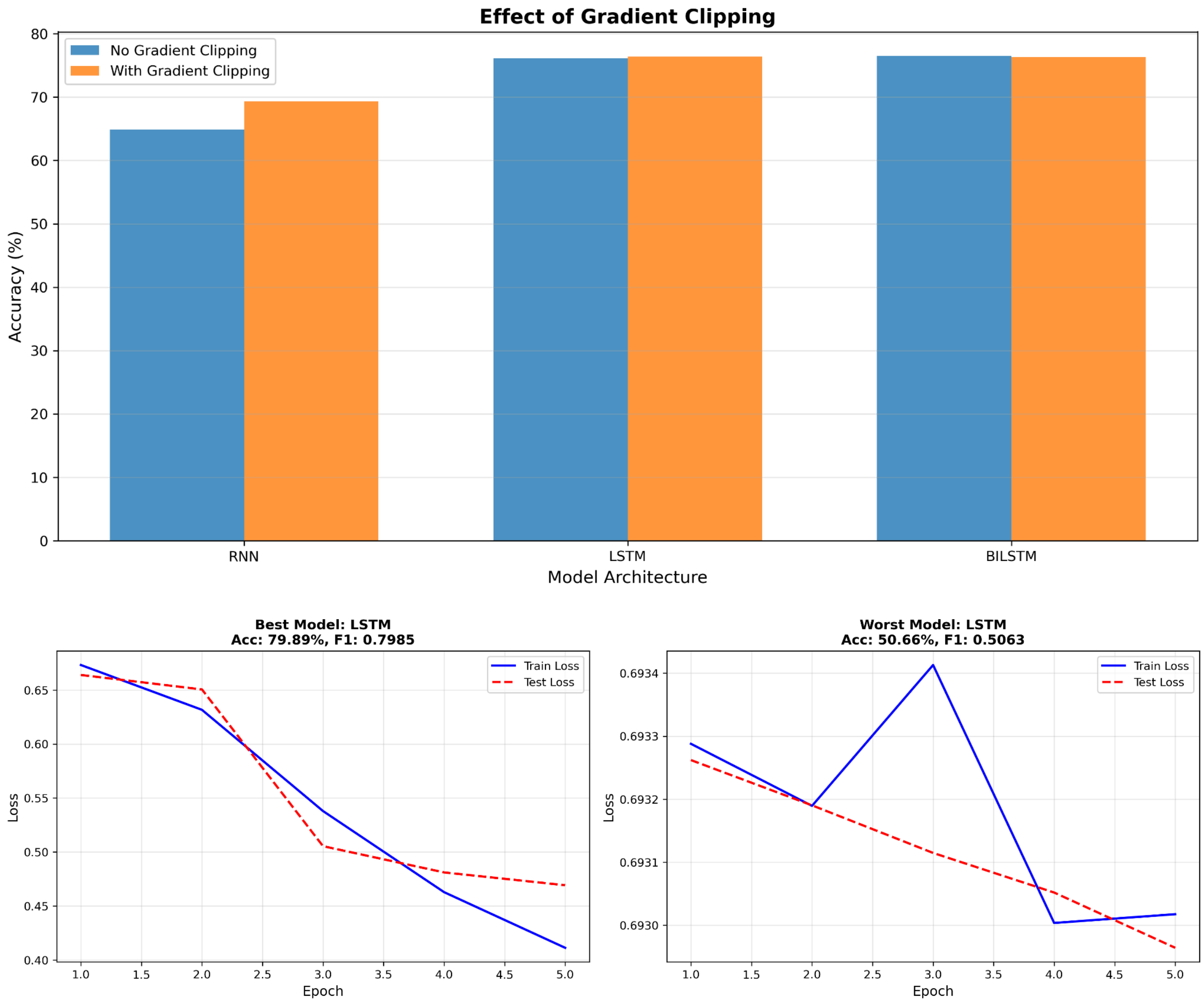
Under CPU constraints and the requirements of this assignment, the optimal configuration for IMDB movie review sentiment classification is an LSTM architecture with sigmoid activation function, RMSProp optimizer, sequence length of 100 tokens, and gradient clipping enabled. This configuration achieved 79.89% test accuracy with an F1 score of 0.7985, representing the best performance across all fifteen experimental configurations. The training efficiency of 19.52 seconds per epoch is acceptable for CPU-based training, requiring about 1.5 minutes to complete five epochs of training.

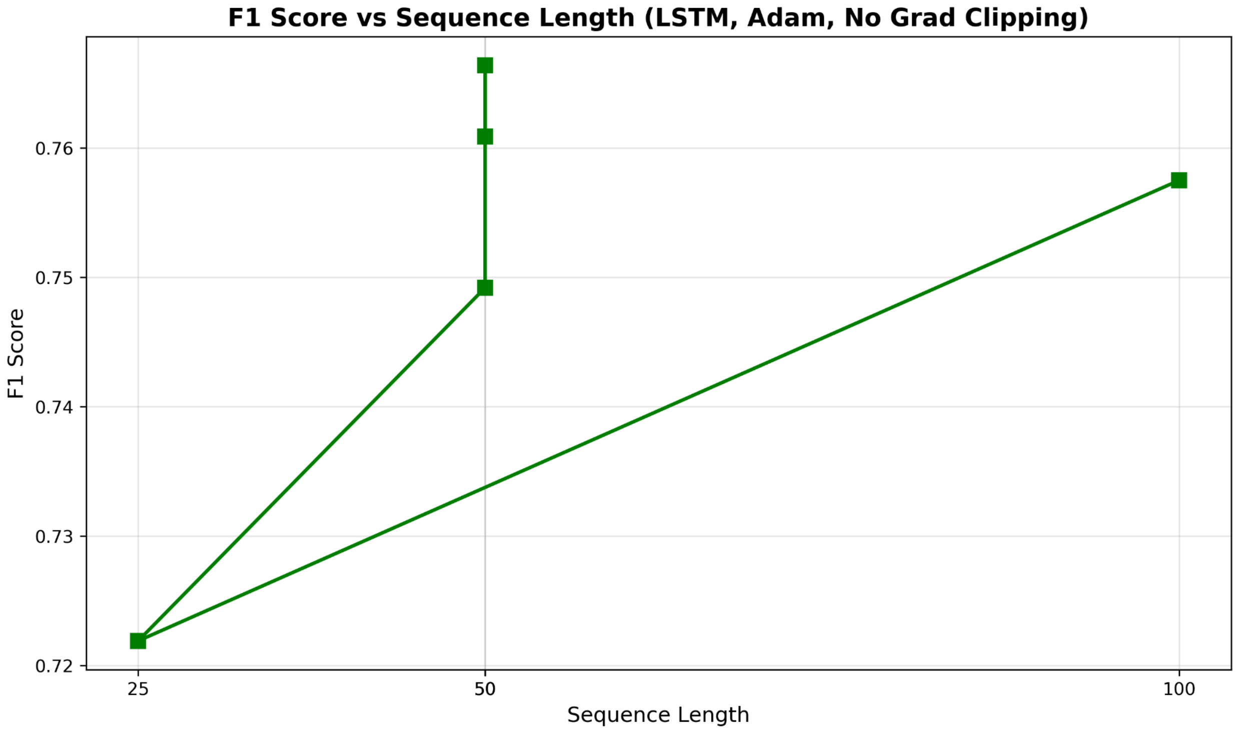
In conclusion, this assignment demonstrates that careful selection of architectures and hyperparameters can yield substantial performance improvements for sentiment classification tasks. The optimal LSTM-based configuration achieved nearly 80% accuracy on the challenging IMDB movie review dataset, providing a robust foundation for practical sentiment analysis applications. The insights gained regarding the relative importance of different architectural choice and hyperparameters provide valuable guidance to further implement effective sentiment classification systems under computational constraints.

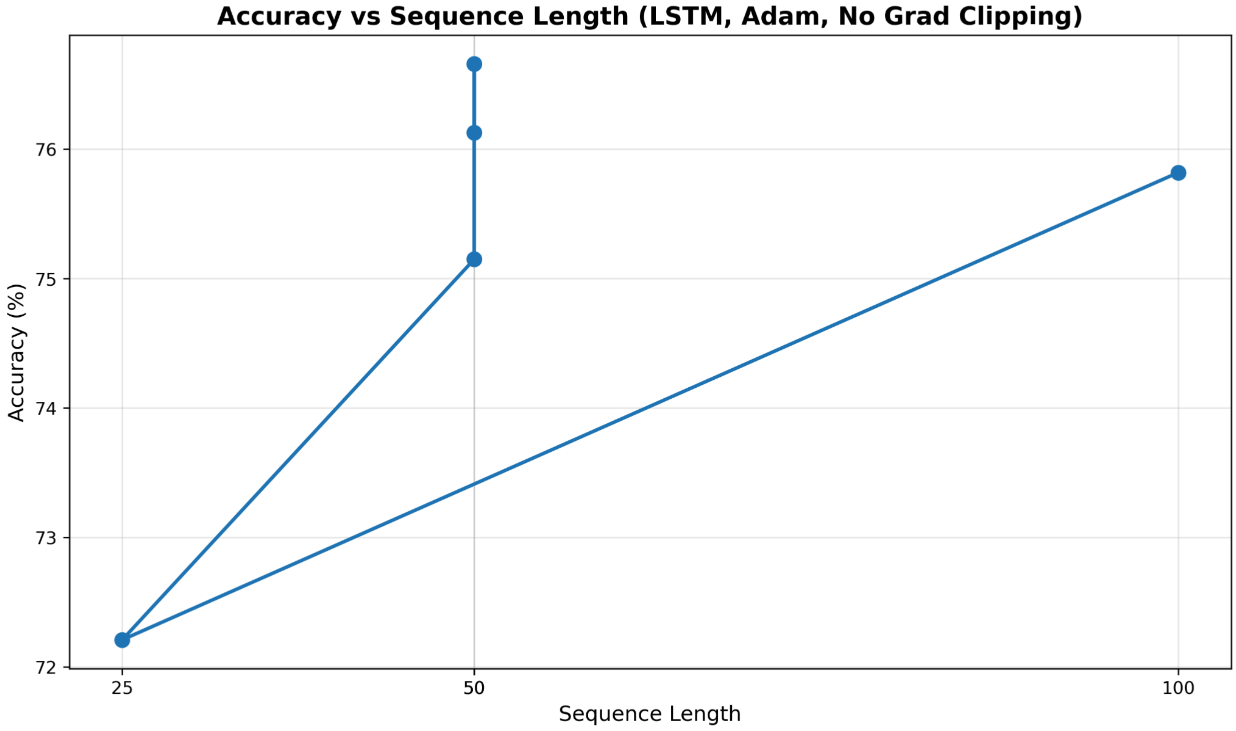
Plots

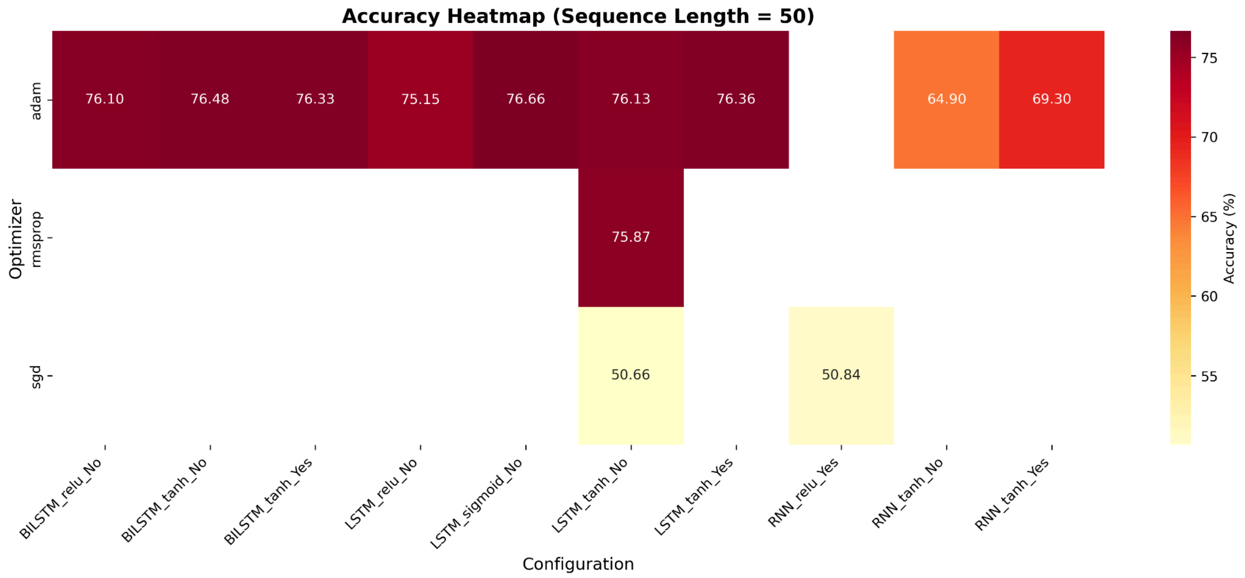

Model Architecture Comparison
Accuracy (%)
F1 Score (%)
Performance (%)
0
10
20
30
40
50
60
70
80
RNN
LSTM
BILSTM
Model Architecture


Optimizer Comparison (LSTM, Tanh, Seq Len 50)
Accuracy (%)
0
10
20
30
40
50
60
70
Epoch Time (s)
10.2
10.3
10.4
10.5
10.6
10.7
ADAM
SGD
RMSPROP
Optimizer


Effect of Gradient Clipping
No Gradient Clipping
With Gradient Clipping
Accuracy (%)
0
10
20
30
40
50
60
70
80
RNN
LSTM
BILSTM
Model Architecture
Best Model: LSTM
Acc: 79.89%, F1: 0.7985
Train Loss
Test Loss
Loss
0.40
0.45
0.50
0.55
0.60
0.65
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Epoch
Worst Model: LSTM
Acc: 50.66%, F1: 0.5063
Train Loss
Test Loss
Loss
0.6930
0.6931
0.6932
0.6933
0.6934
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Epoch


F1 Score vs Sequence Length (LSTM, Adam, No Grad Clipping)
F1 Score
0.72
0.73
0.74
0.75
0.76
25
50
100
Sequence Length


Accuracy vs Sequence Length (LSTM, Adam, No Grad Clipping)
Accuracy (%)
72
73
74
75
76
25
50
100
Sequence Length


Accuracy Heatmap (Sequence Length = 50)
Optimizer
adam
rmsprop
sgd
76.10
76.48
76.33
75.15
76.66
76.13
76.36
64.90
69.30
75.87
50.66
50.84
BILSTM_relu_No
BILSTM_tanh_No
BILSTM_tanh_Yes
LSTM_relu_No
LSTM_sigmoid_No
LSTM_tanh_No
LSTM_tanh_Yes
RNN_relu_Yes
RNN_tanh_No
RNN_tanh_Yes
Configuration
Accuracy (%)
75
70
65
60
55