Name :- Shubham Kathiriya
Roll No:- 2023201050
Subject :- Intro to NLP

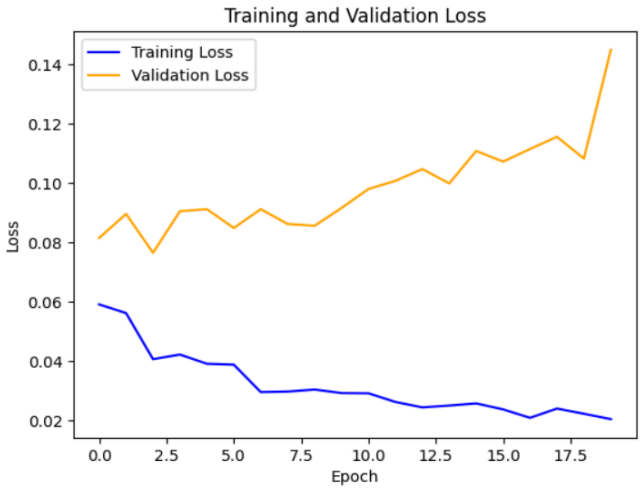
Assignment 2:- POS Tagging

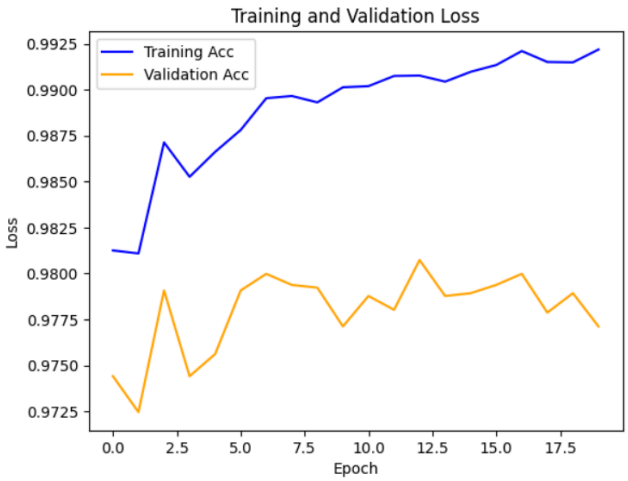
❖ Hyperparameter tunning result:-

1. FFNN Result

- **Configuration 1:-**

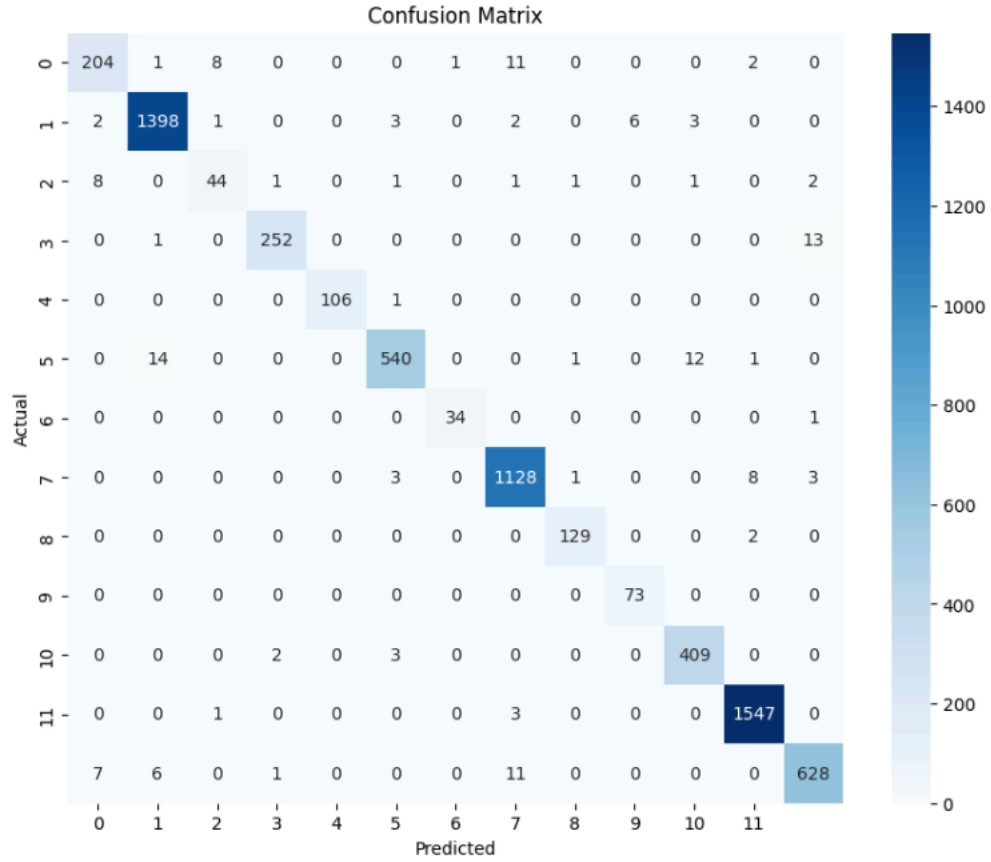
```
p = 3
s = 2
BATCH_SIZE = 64
EPOCHS = 10
embedding_dim = 100
HIDDEN_LAYER_DIMENSION = 200

def forward(self, x):
      x = x.to(self.l1.weight.dtype)
      out = self.l1(x)
      out = self.relu(out)
      out = self.l2(out)
      out = self.relu(out)
      out = self.l3(out)
      return out
```





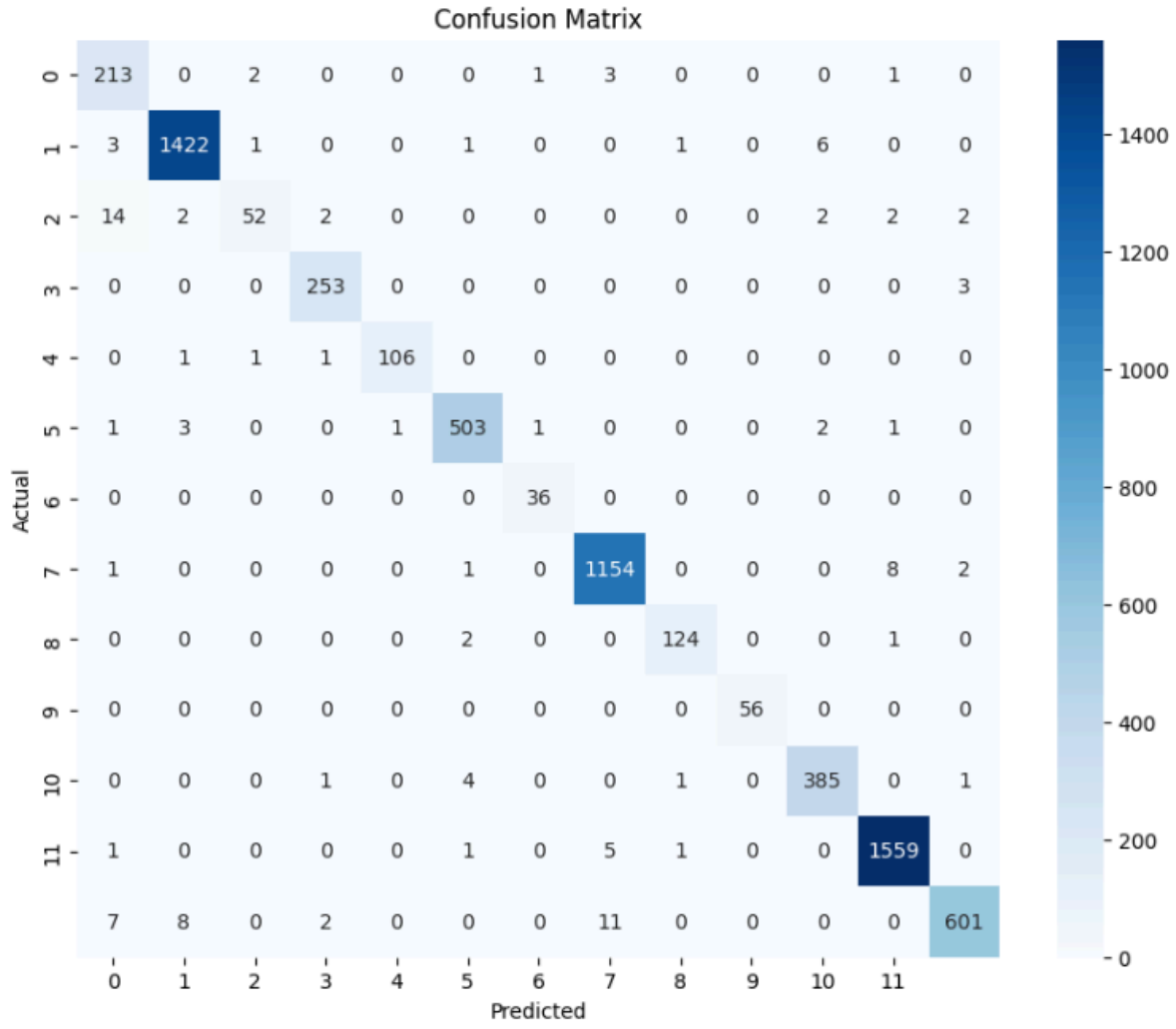
For Validation:-

```
========================== Validation Data ==========================
Validation Loss:  0.1448
Validation Accuracy: 0.9771
Validation Precision: 0.9993
Validation Recall: 0.9882
Validation F1 Score: 0.9937
======================================================================
```



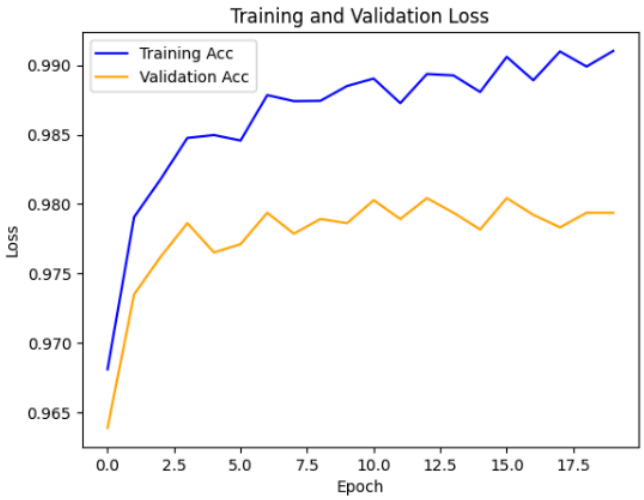
For Test:-

```
========================== Test Data ==========================
Test Loss:  0.1043
Test Accuracy: 0.9824
Test Precision: 1.0000
Test Recall: 0.9815
Test F1 Score: 0.9907
================================================================================
```



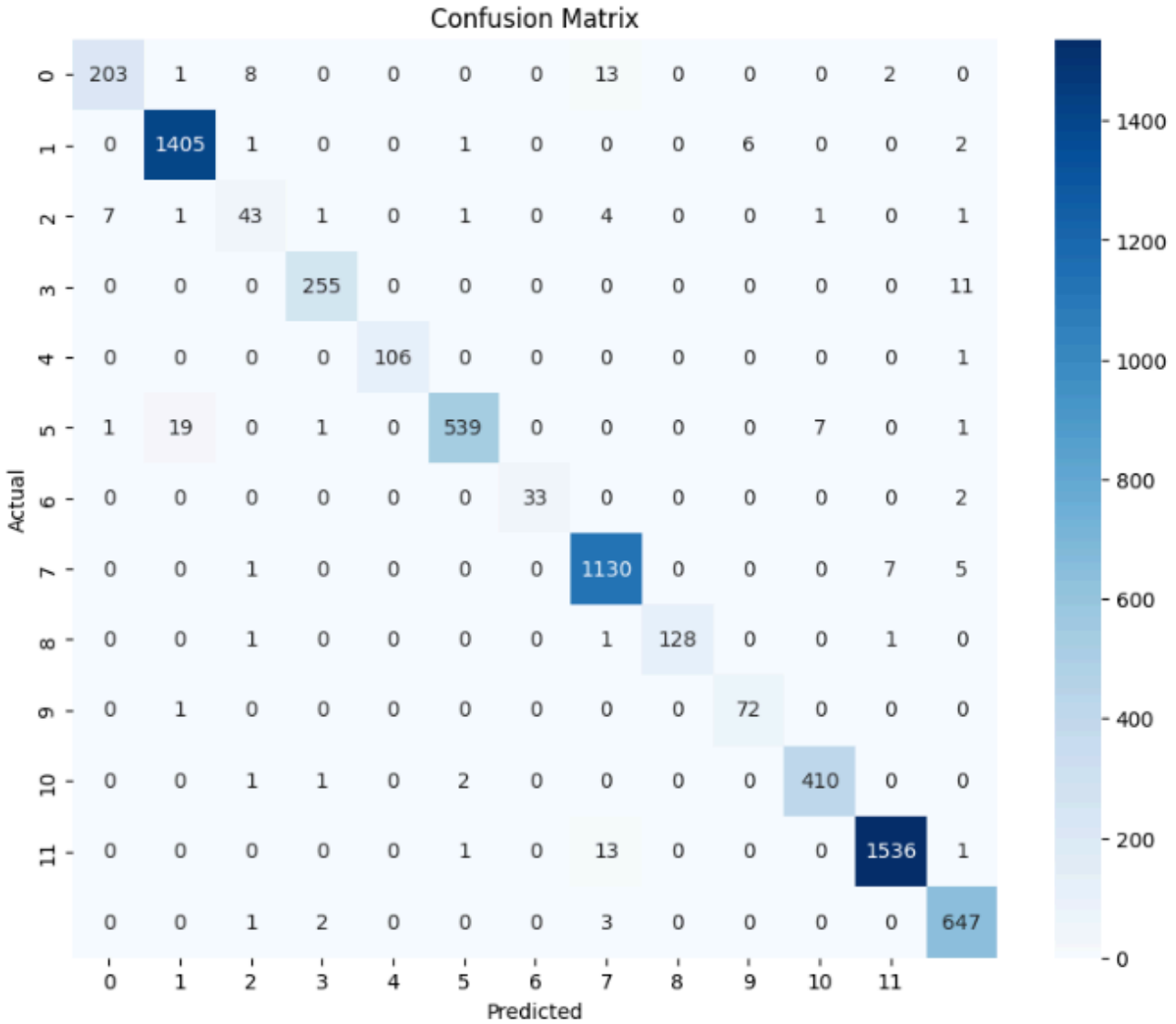
- **Configuration 2:-**

```
p = 2
s = 3
BATCH_SIZE = 64
EPOCHS = 20
embedding_dim = 100
HIDDEN_LAYER_DIMENSION = 100

def forward(self, x):
    x = x.to(self.l1.weight.dtype)
    out = self.l1(x)
    out = self.relu(out)
    out = self.l2(out)
    out = self.relu(out)
    out = self.l3(out)
    out = self.relu(out)
    out = self.l4(out)
    out = self.relu(out)
    out = self.l5(out)

    return out
```



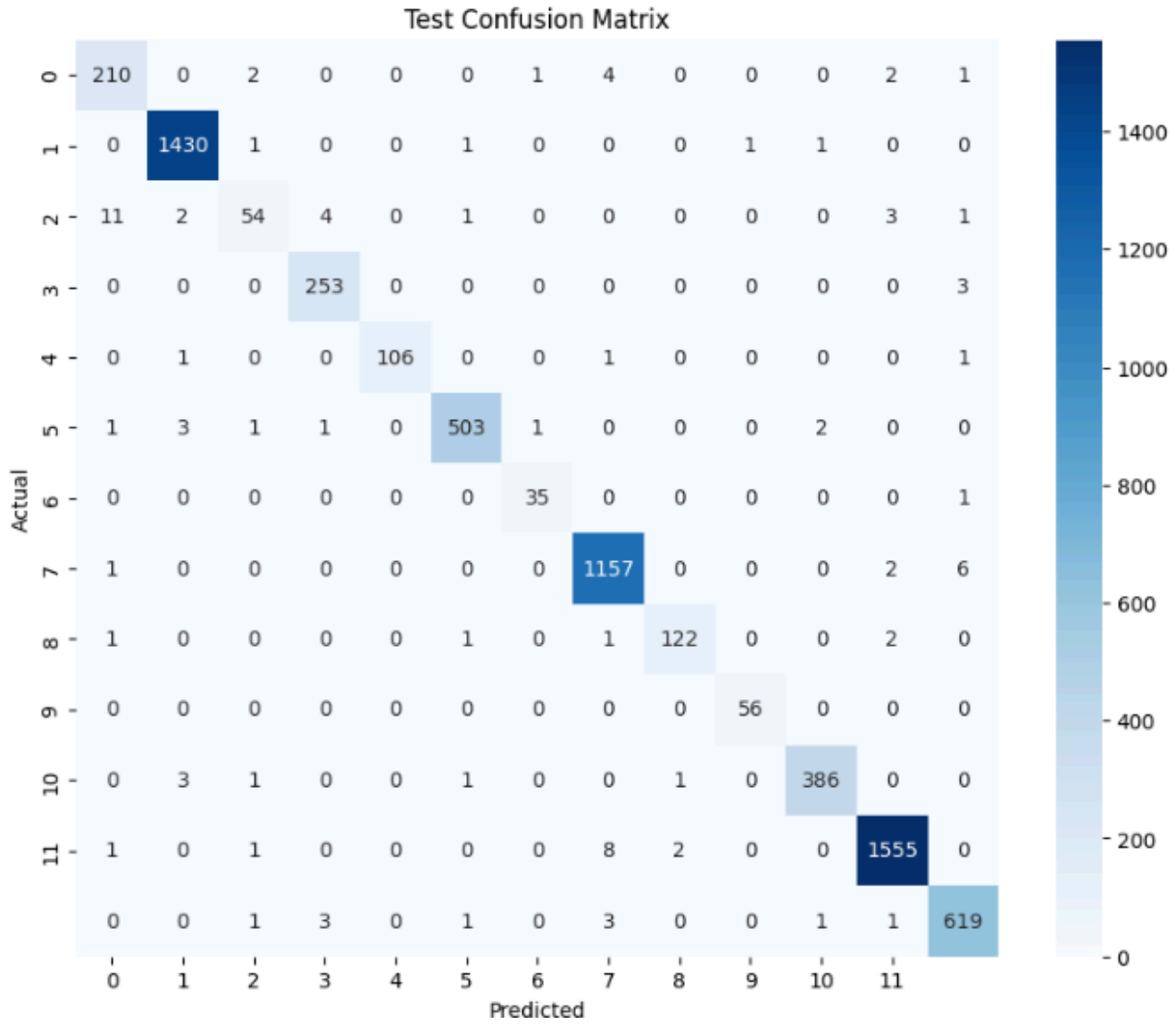
For Validation Data:-

```
============================ Validation Data ============================
Validation Loss:  0.0993
Validation Accuracy: 0.9794
Validation Precision: 0.9993
Validation Recall: 0.9944
Validation F1 Score: 0.9969
=========================================================================
```



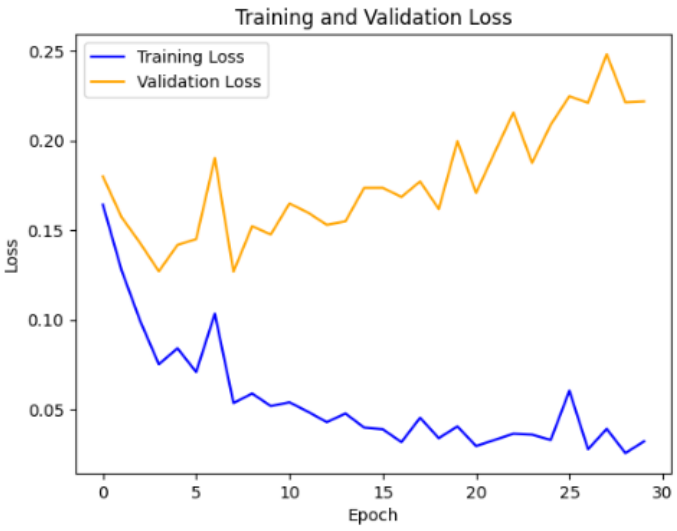
For Test Data:-

```
============================ Test Data ============================
Test Loss:  0.0858
Test Accuracy: 0.9857
Test Precision: 1.0000
Test Recall: 0.9897
Test F1 Score: 0.9948
===================================================================
```



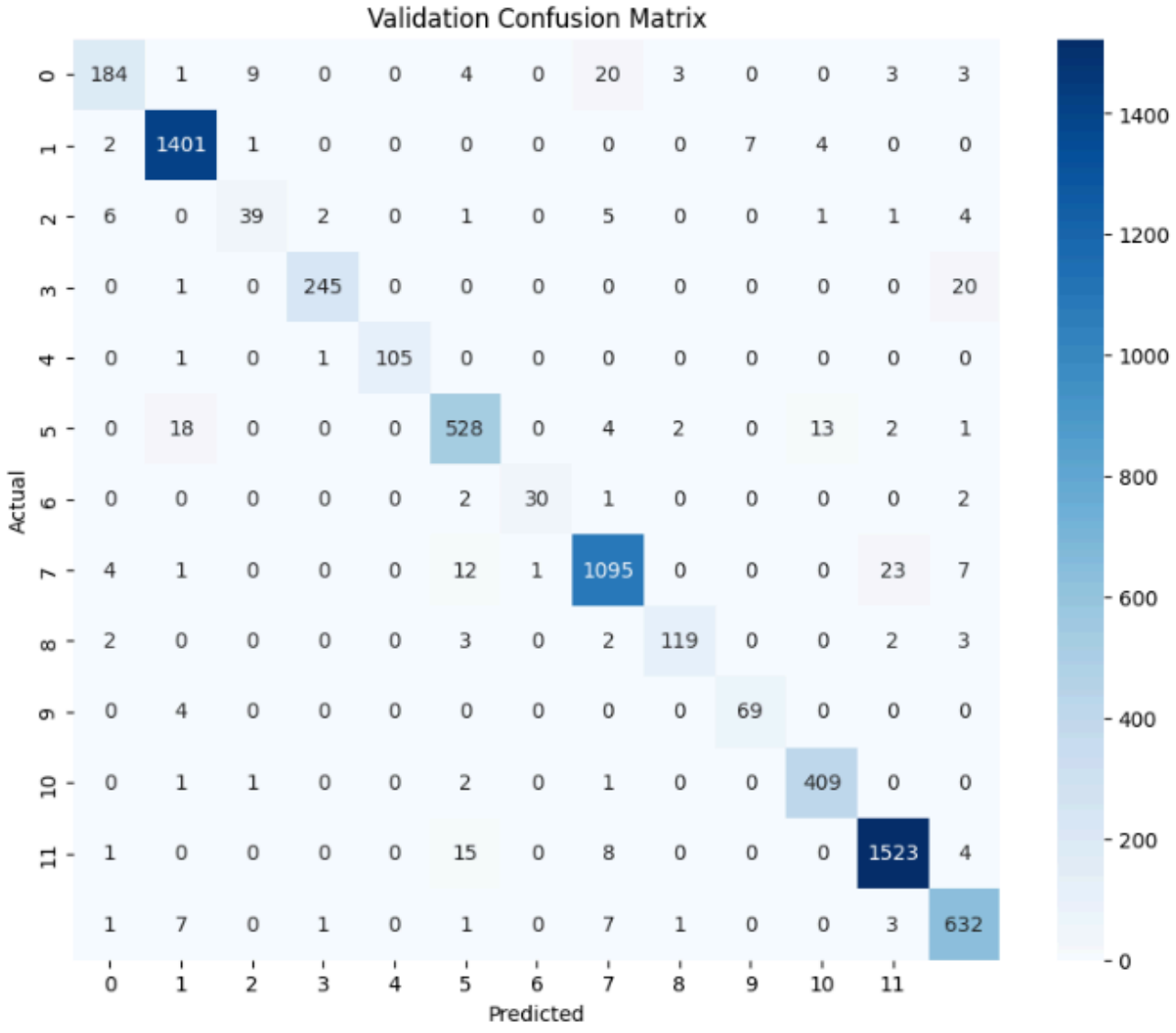
- **Configuration 3:-**

```
p = 4
s = 5
BATCH_SIZE = 64
round_precision = 5
EPOCHS = 30
embedding_dim = 100
HIDDEN_LAYER_DIMENSION = 200

def forward(self, x):
        x = x.to(self.l1.weight.dtype)
        out = self.l1(x)
        out = self.leaky_relu(out)
        out = self.l2(out)
        out = self.leaky_relu(out)
        out = self.l3(out)
        out = self.leaky_relu(out)
        out = self.l4(out)

        return out
```



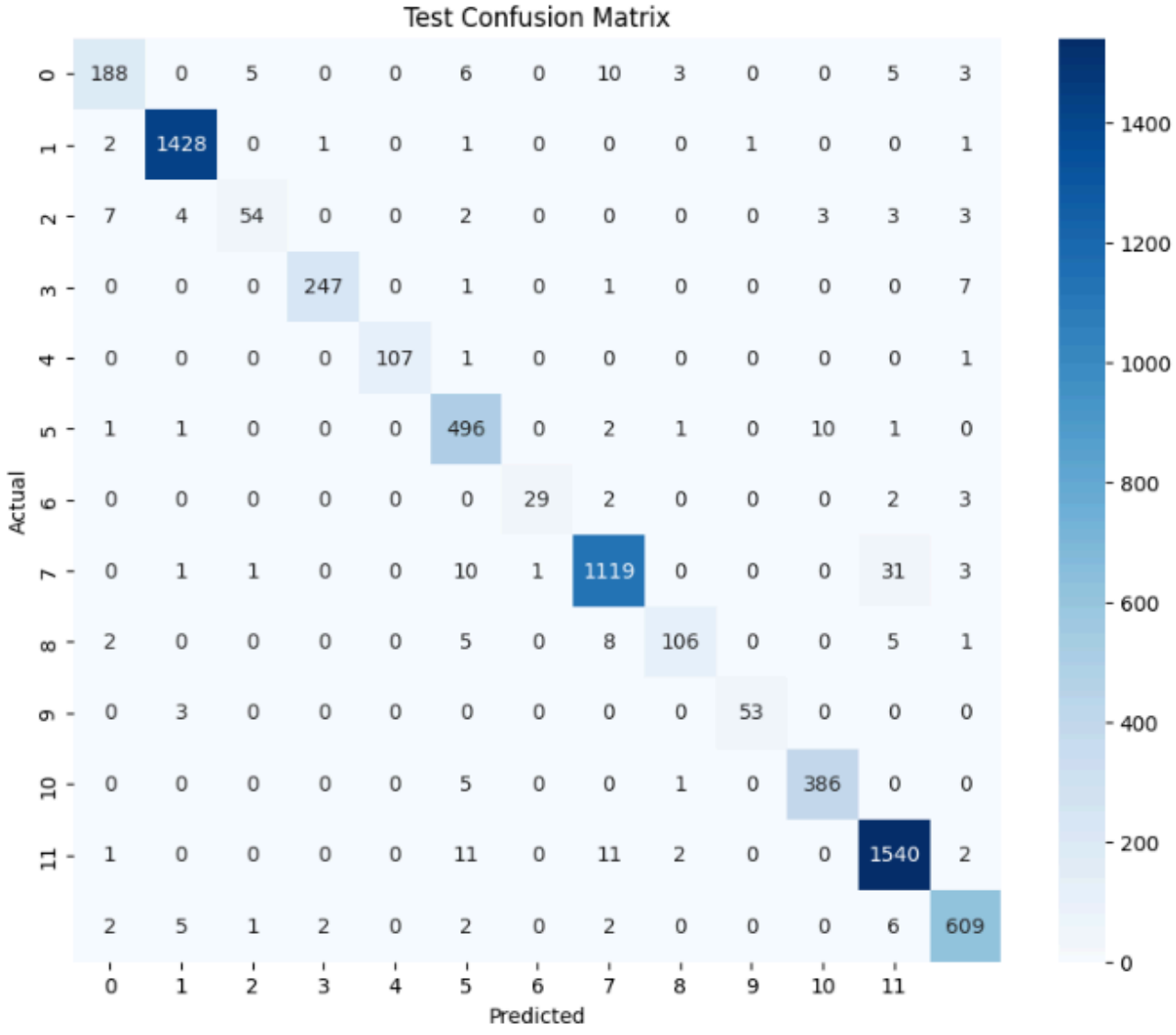
For Validation Data:-

```
============================ Validation Data ============================
Validation Loss:  0.2217
Validation Accuracy: 0.9601
Validation Precision: 0.9993
Validation Recall: 0.9890
Validation F1 Score: 0.9941
=========================================================================
```

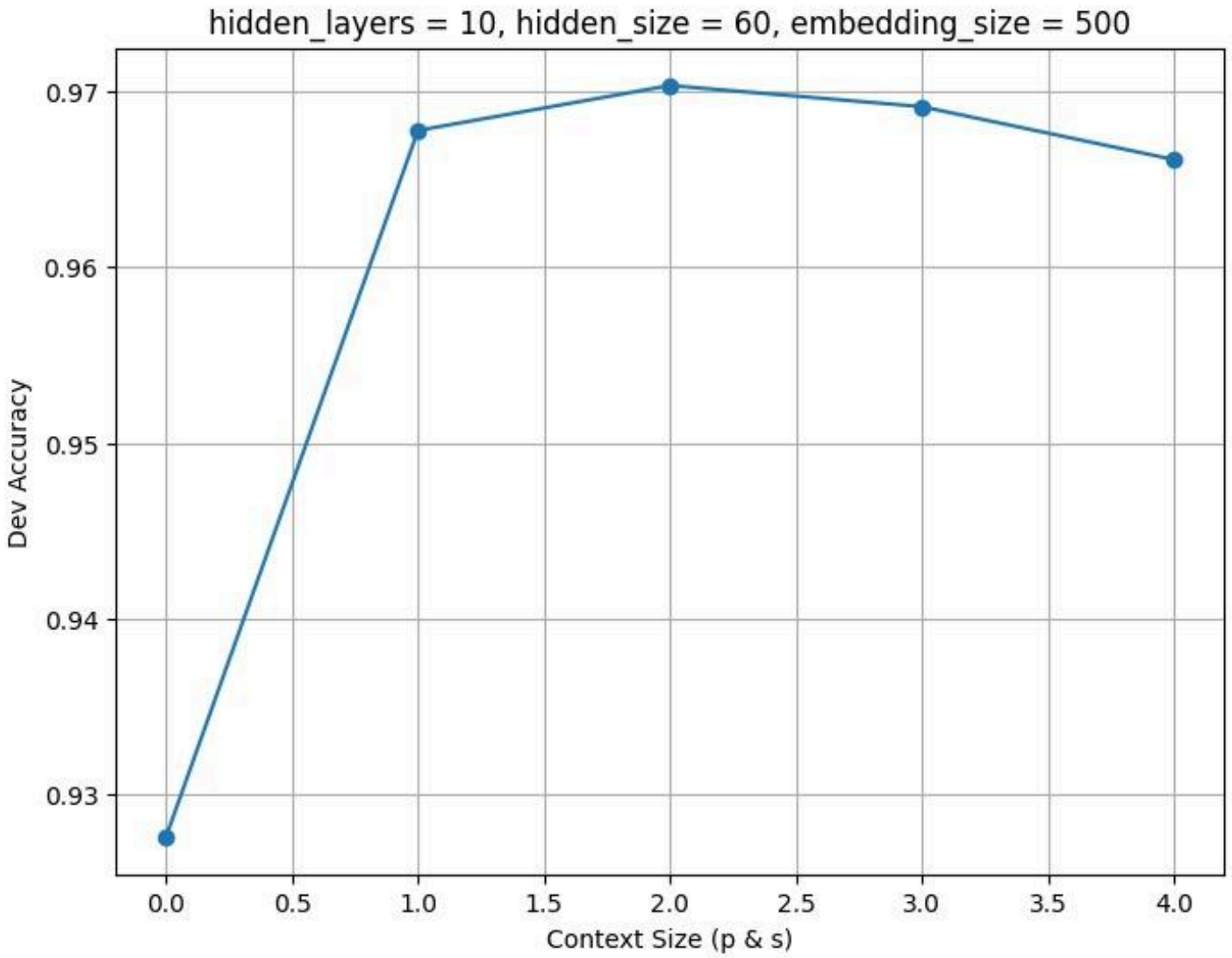


For test data:-

```
========================== Test Data ==========================
Test Loss:  0.2062
Test Accuracy: 0.9669
Test Precision: 1.0000
Test Recall: 0.9897
Test F1 Score: 0.9948
================================================================================
```



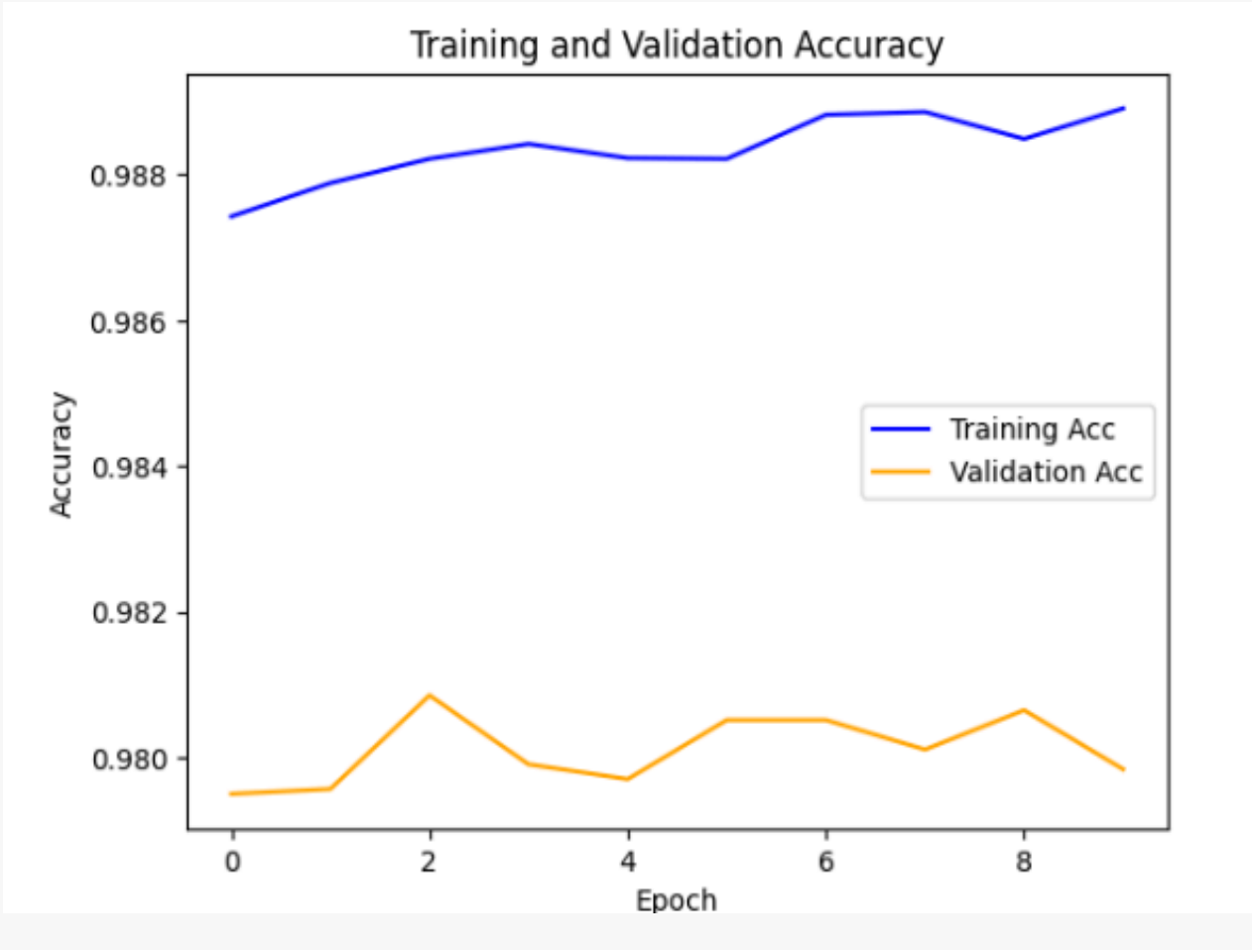
For Various P and S for configuration:-



2. LSTM

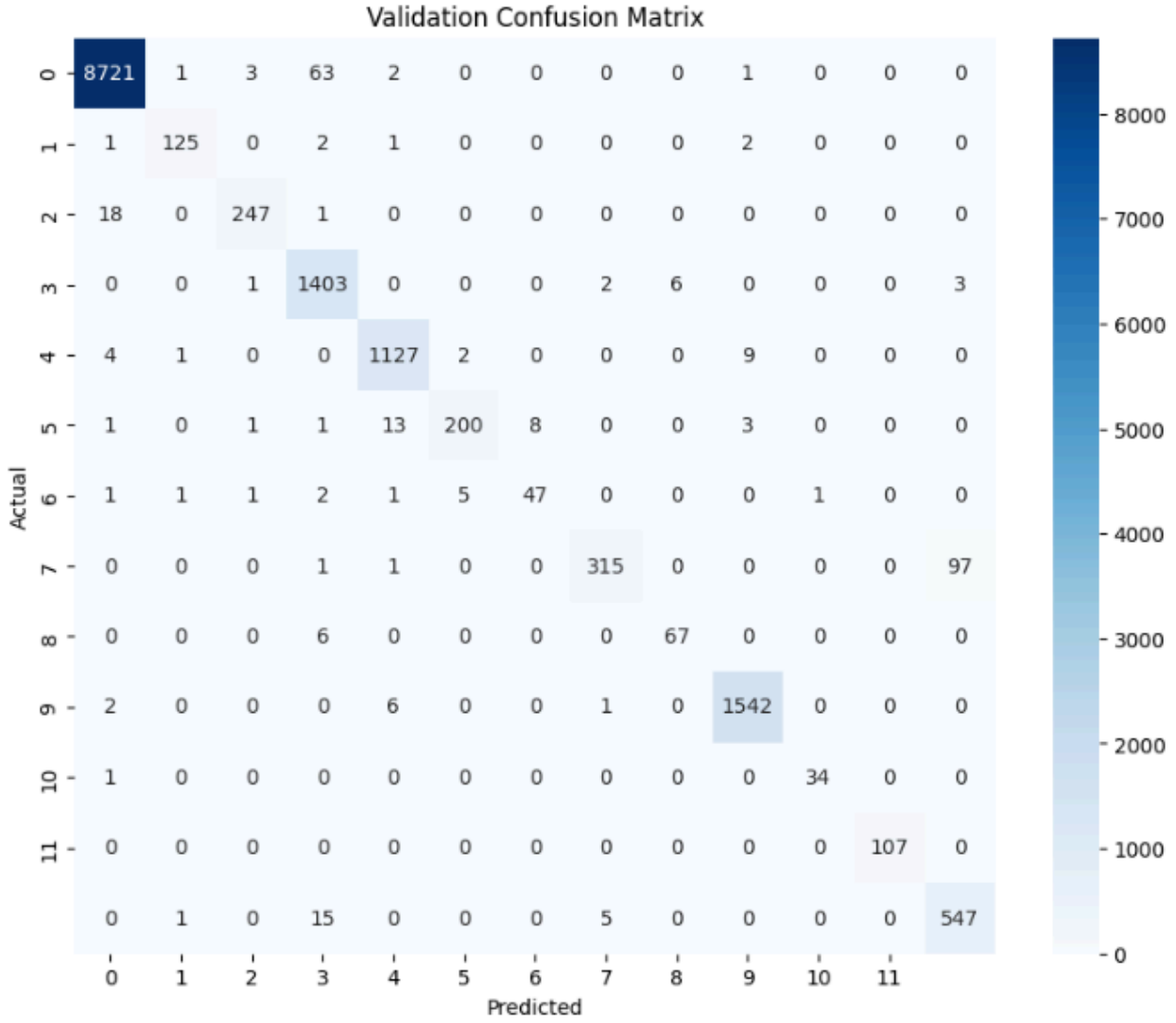
- **Confriguation 1:-**

```
BATCH_SIZE = 64
round_precision = 5
EPOCHS = 50
embedding_dim = 100
hidden_layer_dim = 200
learning_rate = 1e-3
epochs = 10
Bidirectional = False
Num_layer = 1
```



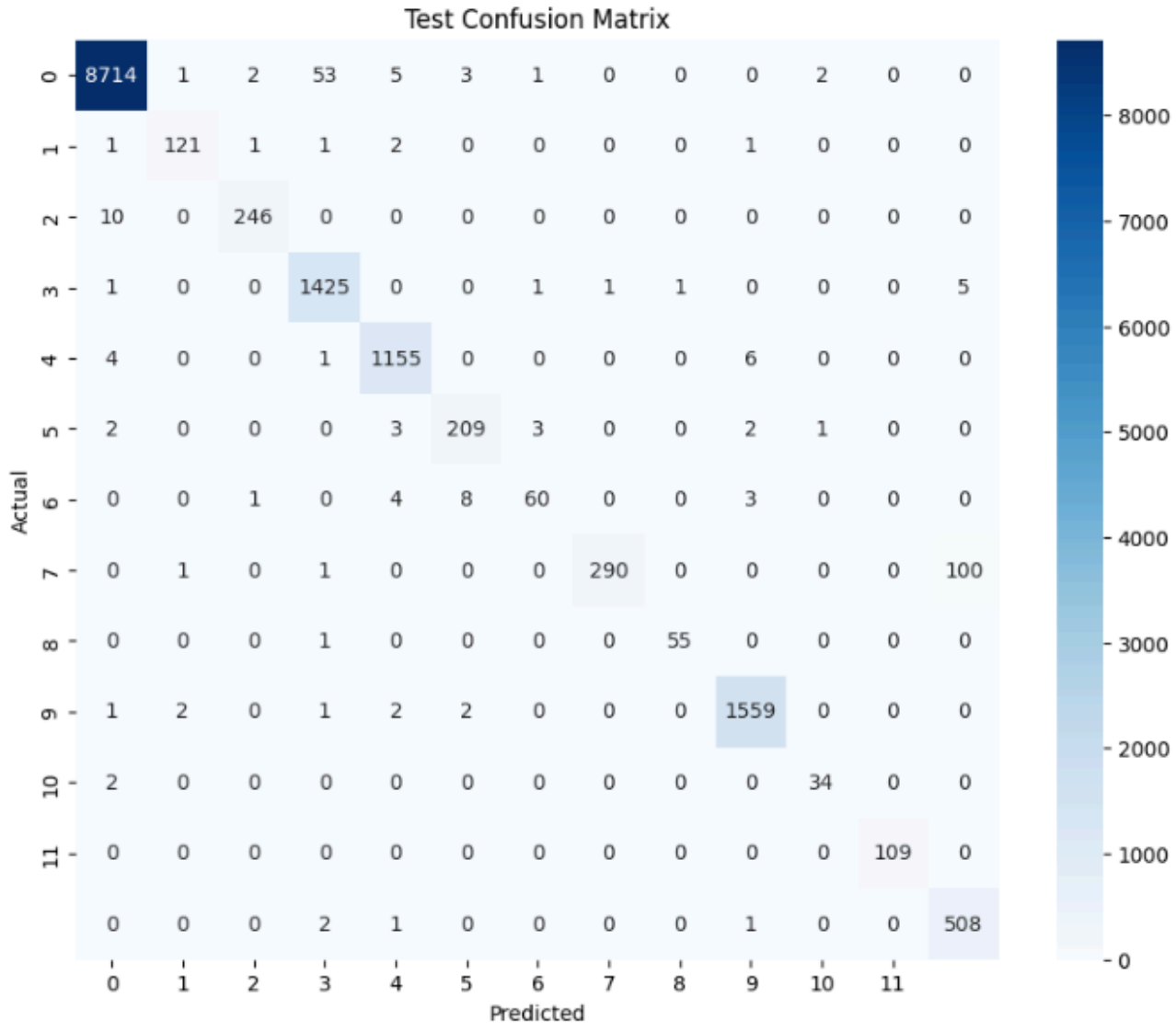
For Validation Data :-

```
========================== Validation Data ==========================
Validation Accuracy: 0.9798
Validation Precision: 0.9921
Validation Recall: 0.9921
Validation F1 Score: 0.9921
=====================================================================
```



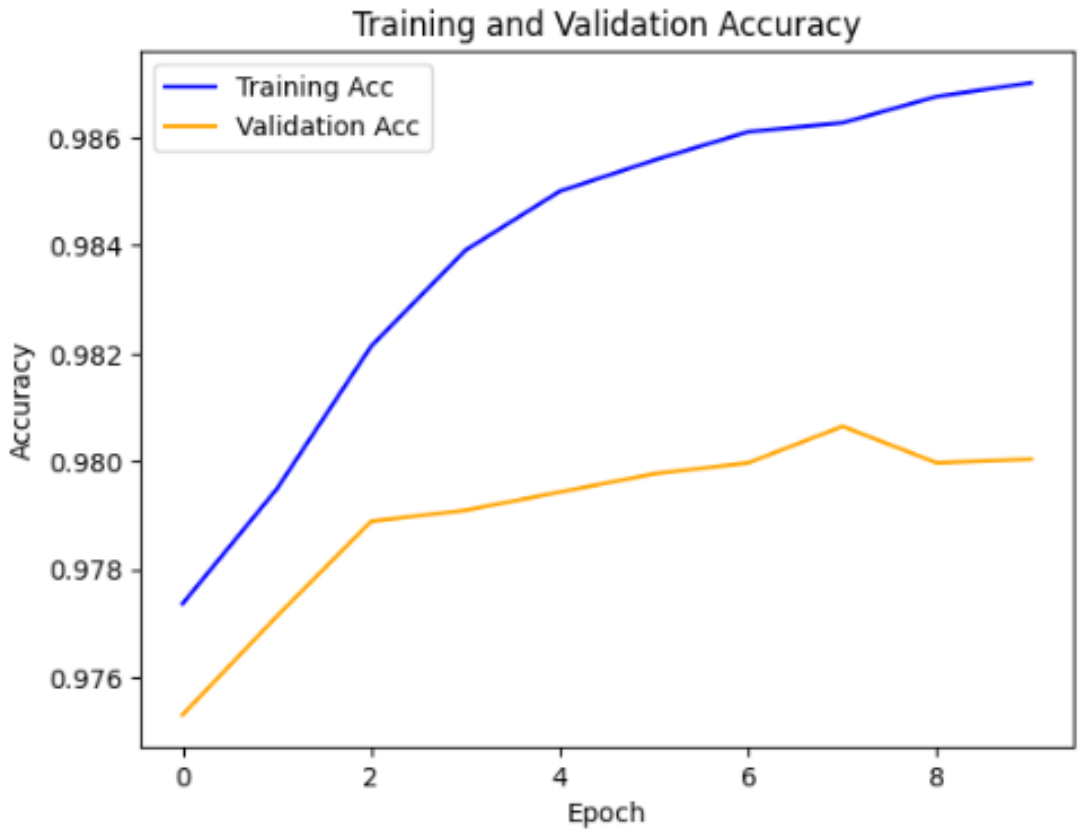
For Test Data:-

```
============================ Test Data ============================
Test Accuracy: 0.9832
Test Precision: 0.9918
Test Recall: 0.9918
Test F1 Score: 0.9918
===================================================================
```



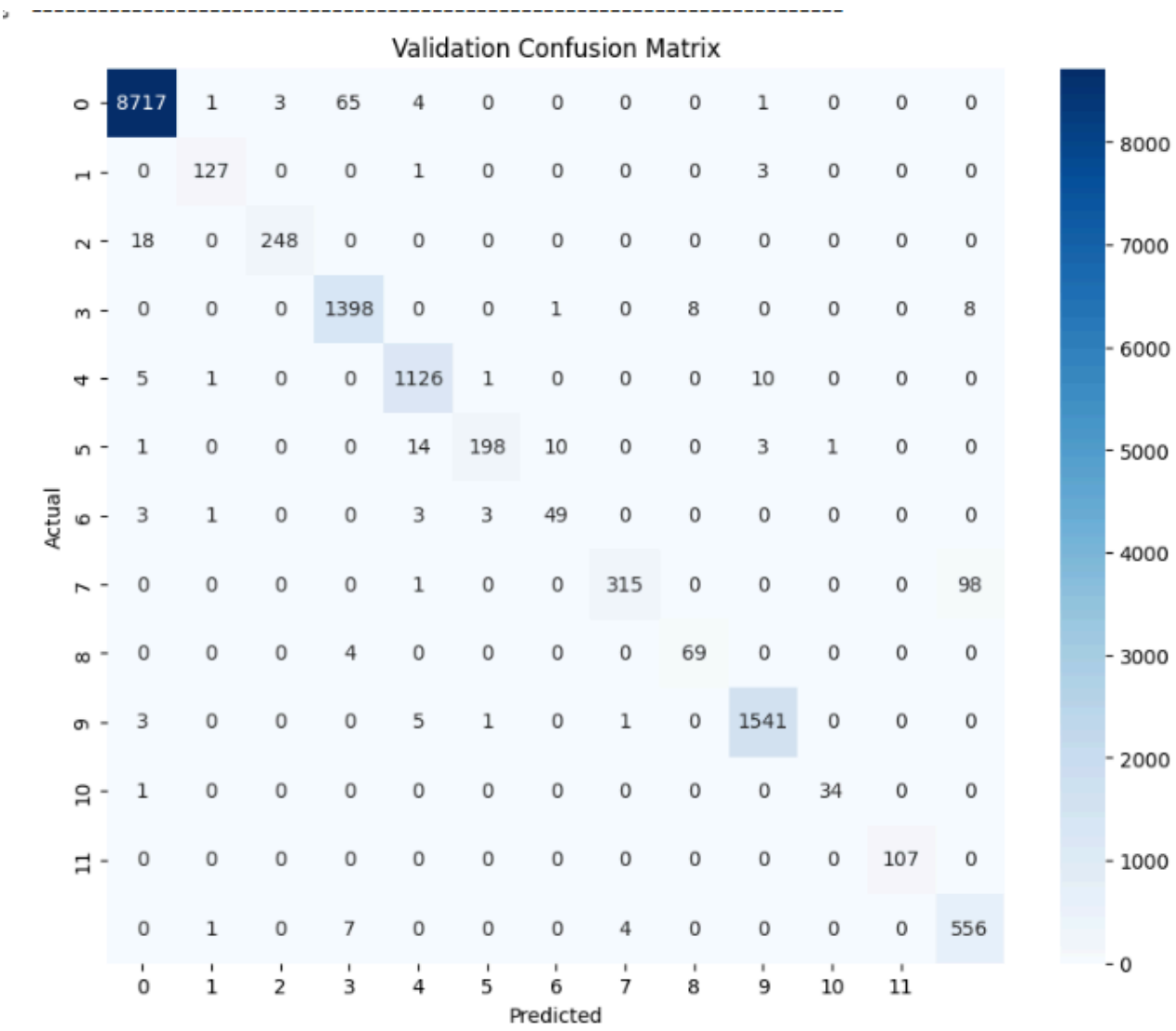
- **Confriguation 2:-**

```
BATCH_SIZE = 64
round_precision = 5
EPOCHS = 50
embedding_dim = 100
hidden_layer_dim = 200
learning_rate = 1e-3
epochs = 10
bidirectional = True
num_layer = 2
```



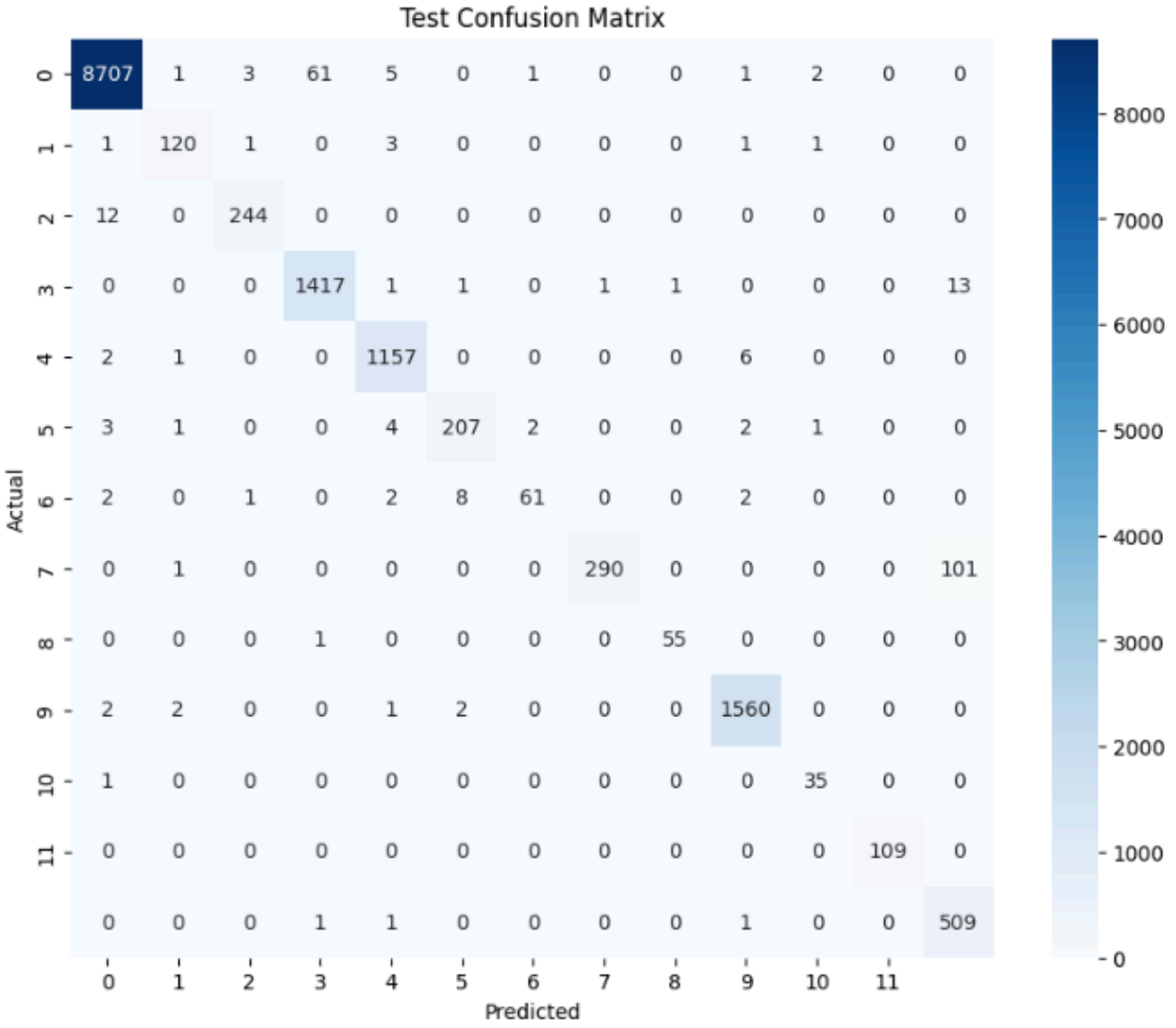
For Validation Data:-

```
============================ Validation Data ============================
Validation Accuracy: 0.9800
Validation Precision: 0.9922
Validation Recall: 1.0000
Validation F1 Score: 0.9961
=========================================================================
```



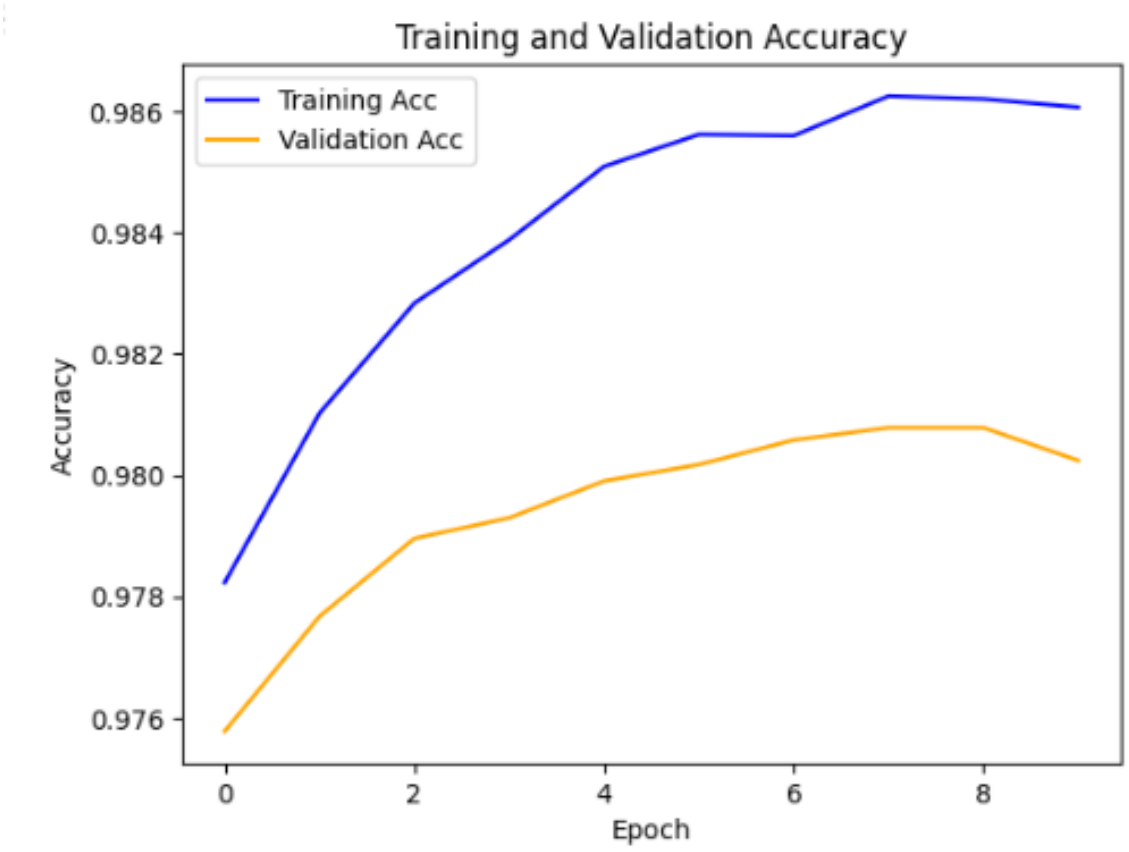
For Test Data:-

```
============================ Test Data ============================
Test Accuracy: 0.9823
Test Precision: 0.9917
Test Recall: 0.9917
Test F1 Score: 0.9917
================================================================================
```



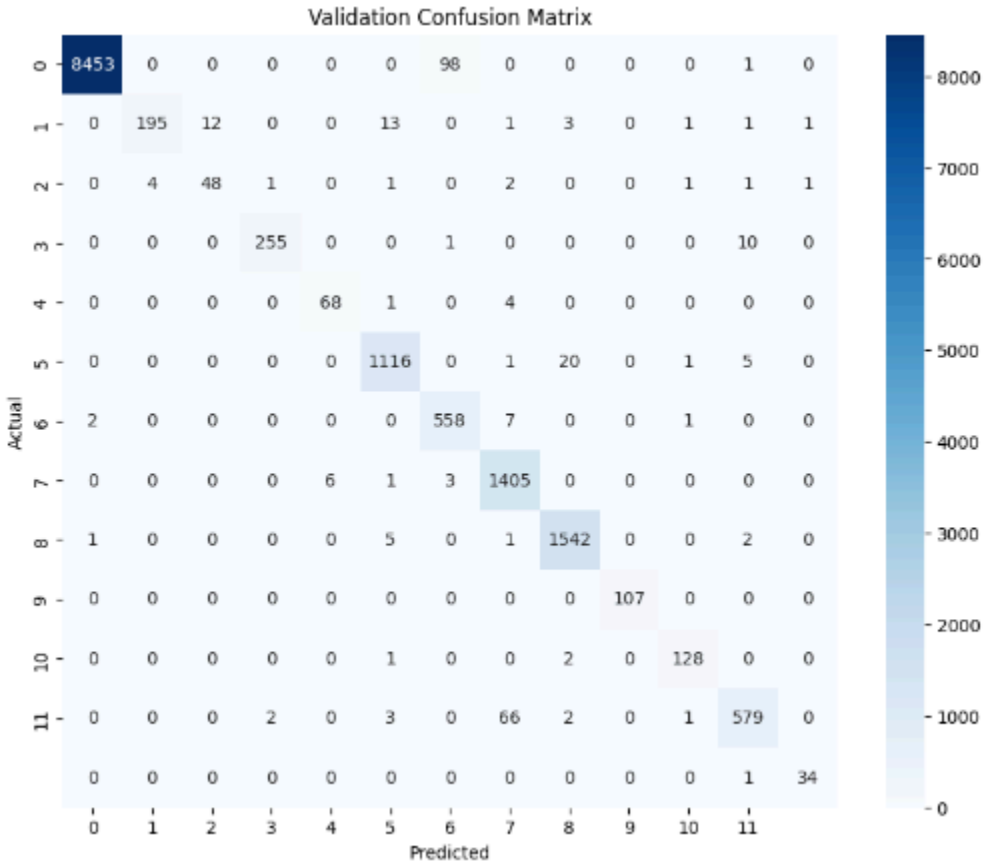
- **Confriguation 3:-**

```
BATCH_SIZE = 64
round_precision = 5
EPOCHS = 10
embedding_dim = 300
hidden_layer_dim = 400
learning_rate = 1e-3
epochs = 10
bidirectional = True
num_layer = 4
```



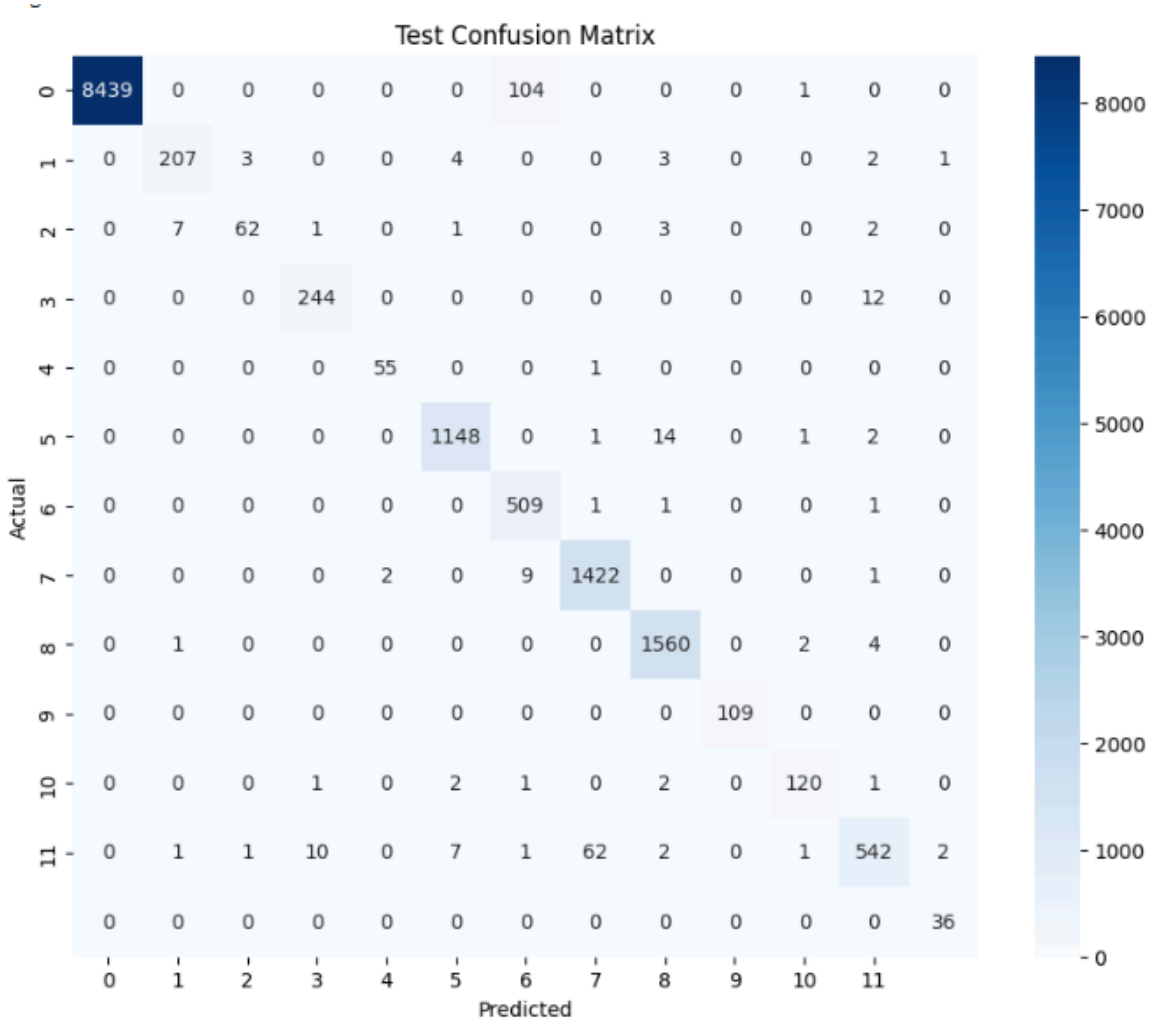
For Validation Accuracy:-

```
========================== Validation Data ==========================
Validation Accuracy: 0.9802
Validation Precision: 1.0000
Validation Recall: 1.0000
Validation F1 Score: 1.0000
======================================================================
```



For Test Data :-

```
============================ Test Data ============================
Test Accuracy: 0.9811
Test Precision: 1.0000
Test Recall: 1.0000
Test F1 Score: 1.0000
================================================================================
```



❖ Analysis:-

● Analytical Result of Feed Forward Neural Network POS Tagger:

- Advancing Model Complexity: By increasing parameters such as embedding dimension and hidden layer sizes, there's a slight uptick in development accuracy, both in macro and weighted averages.
- Test Accuracy Enhancement: Augmenting model complexity also leads to a noticeable boost in test accuracy.
- Influence of Activation Functions: Switching from ReLU to Tanh activation function causes a slight dip in accuracy, favoring ReLU for better performance.
- Context Window Impact:
    - The context window size significantly affects model accuracy.
    - The lowest accuracy is observed when both preceding and succeeding window sizes are 0, while the highest accuracy is achieved when both are 1.
    - Further increase in window size results in a minor decrease in accuracy.
- Activation function: ReLu gives the best performance and leaky ReLu gives slightly less good performance than ReLu and softmax gives even more noticeable lesser accuracy

● Analytical Result of LSTM (RNN) POS Tagger:

- Effect of Layer Count: Increasing the number of layers extends training time without notable accuracy improvement.
- Bidirectionality Effect: Activating bidirectionality notably boosts accuracy, showcasing its importance in capturing context effectively.

- Epoch Count Impact: Despite increasing epochs, accuracy doesn't improve significantly due to bidirectionality's substantial performance enhancement.
- Activation Functions: Tanh function is utilized internally within LSTM architecture, while SoftMax is applied at the output layer as the final activation function.