```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive",
force_remount=True).

==== Training LSTM Models for Each Dataset ====


🔹 Training on NSLKDD dataset...

✅ NSLKDD dataset limited to 100000 rows.
Epoch 1/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 100s 44ms/step - accuracy: 0.8923 - loss: 0.2586
Epoch 2/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 129s 38ms/step - accuracy: 0.9514 - loss: 0.1203
Epoch 3/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 143s 38ms/step - accuracy: 0.9605 - loss: 0.0990
Epoch 4/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 142s 38ms/step - accuracy: 0.9636 - loss: 0.0901
Epoch 5/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 84s 38ms/step - accuracy: 0.9683 - loss: 0.0795
Epoch 6/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 143s 39ms/step - accuracy: 0.9723 - loss: 0.0734
Epoch 7/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 140s 38ms/step - accuracy: 0.9734 - loss: 0.0690
Epoch 8/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 143s 39ms/step - accuracy: 0.9764 - loss: 0.0647
Epoch 9/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 83s 38ms/step - accuracy: 0.9772 - loss: 0.0620
Epoch 10/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 142s 38ms/step - accuracy: 0.9712 - loss: 0.0741
938/938 ━━━━━━━━━━━━━━━━━━━━ 12s 12ms/step
🎯 LSTM Accuracy for NSLKDD: 0.9772
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     15055
           1       0.98      0.98      0.98     14945

    accuracy                           0.98     30000
   macro avg       0.98      0.98      0.98     30000
weighted avg       0.98      0.98      0.98     30000


🔹 Training on UNSW_NB15 dataset...

✅ UNSW_NB15 dataset limited to 100000 rows.
Epoch 1/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 98s 43ms/step - accuracy: 0.8998 - loss: 0.2237
Epoch 2/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 136s 40ms/step - accuracy: 0.9996 - loss: 0.0016
Epoch 3/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 145s 42ms/step - accuracy: 0.9997 - loss: 0.0010
Epoch 4/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 140s 41ms/step - accuracy: 0.9999 - loss:
4.6008e-04
```

```
Epoch 5/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 143s 41ms/step - accuracy: 1.0000 - loss:
6.2611e-05
Epoch 6/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 142s 41ms/step - accuracy: 0.9999 - loss:
3.6120e-04
Epoch 7/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 142s 41ms/step - accuracy: 1.0000 - loss:
2.4038e-05
Epoch 8/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 140s 41ms/step - accuracy: 1.0000 - loss:
1.7639e-05
Epoch 9/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 146s 42ms/step - accuracy: 0.9998 - loss:
8.2605e-04
Epoch 10/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 89s 41ms/step - accuracy: 1.0000 - loss:
5.6749e-06
938/938 ━━━━━━━━━━━━━━━━━━━━ 12s 12ms/step
🎯 LSTM Accuracy for UNSW_NB15: 1.0000
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     10663
           1       1.00      1.00      1.00     19337

    accuracy                           1.00     30000
   macro avg       1.00      1.00      1.00     30000
weighted avg       1.00      1.00      1.00     30000


🔹 Training on KDDCup dataset...

✅ KDDCup dataset limited to 100000 rows.
Epoch 1/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 95s 40ms/step - accuracy: 0.9172 - loss: 0.3511
Epoch 2/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 86s 39ms/step - accuracy: 0.9950 - loss: 0.0260
Epoch 3/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 84s 39ms/step - accuracy: 0.9963 - loss: 0.0193
Epoch 4/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 145s 40ms/step - accuracy: 0.9966 - loss: 0.0161
Epoch 5/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 141s 39ms/step - accuracy: 0.9969 - loss: 0.0137
Epoch 6/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 148s 42ms/step - accuracy: 0.9978 - loss: 0.0090
Epoch 7/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 86s 39ms/step - accuracy: 0.9978 - loss: 0.0089
Epoch 8/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 85s 39ms/step - accuracy: 0.9981 - loss: 0.0074
Epoch 9/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 143s 40ms/step - accuracy: 0.9985 - loss: 0.0063
Epoch 10/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 141s 39ms/step - accuracy: 0.9981 - loss: 0.0080
```

```
938/938 ━━━━━━━━━━━━━━━━━━━━ 12s 12ms/step
🎯 LSTM Accuracy for KDDCup: 0.9986
              precision    recall  f1-score   support

           0       0.91      0.91      0.91        11
           3       0.96      0.97      0.97        72
           4       0.00      0.00      0.00         2
           6       1.00      1.00      1.00      3792
           7       1.00      0.79      0.88        19
           8       1.00      1.00      1.00      6457
           9       0.00      0.00      0.00         1
          10       1.00      0.89      0.94        63
          11       0.98      0.96      0.97       164
          12       1.00      1.00      1.00     19409
          13       0.00      0.00      0.00         1
          14       0.00      0.00      0.00         9

    accuracy                           1.00     30000
   macro avg       0.65      0.63      0.64     30000
weighted avg       1.00      1.00      1.00     30000


🔹 Training on CICIDS2017 dataset...

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
✅ CICIDS2017 dataset limited to 100000 rows.
Epoch 1/10
/usr/local/lib/python3.11/dist-packages/keras/src/ops/nn.py:907: UserWarning: You are using a softmax over axis -1 of a tensor of shape (None, 1). This axis has size 1. The softmax operation will always return the value 1, which is likely not what you intended. Did you mean to use a sigmoid instead?
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/keras/src/losses/losses.py:33: SyntaxWarning: In loss categorical_crossentropy, expected y_pred.shape to be (batch_size, num_classes) with num_classes > 1. Received: y_pred.shape=(None, 1). Consider using 'binary_crossentropy' if you only have 2 classes.
  return self.fn(y_true, y_pred, **self._fn_kwargs)
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 168s 74ms/step - accuracy: 1.0000 - loss: 0.0000e+00
Epoch 2/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 195s 71ms/step - accuracy: 1.0000 - loss: 0.0000e+00
Epoch 3/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 199s 70ms/step - accuracy: 1.0000 - loss: 0.0000e+00
```

```
Epoch 4/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 206s 72ms/step - accuracy: 1.0000 - loss:
0.0000e+00
Epoch 5/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 157s 72ms/step - accuracy: 1.0000 - loss:
0.0000e+00
Epoch 6/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 155s 71ms/step - accuracy: 1.0000 - loss:
0.0000e+00
Epoch 7/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 202s 71ms/step - accuracy: 1.0000 - loss:
0.0000e+00
Epoch 8/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 153s 70ms/step - accuracy: 1.0000 - loss:
0.0000e+00
Epoch 9/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 203s 71ms/step - accuracy: 1.0000 - loss:
0.0000e+00
Epoch 10/10
2188/2188 ━━━━━━━━━━━━━━━━━━━━ 200s 70ms/step - accuracy: 1.0000 - loss:
0.0000e+00
  4/938 ━━━━━━━━━━━━━━━━━━━━ 18s 20ms/step
/usr/local/lib/python3.11/dist-packages/keras/src/ops/nn.py:907: UserWarning: You are using a softmax
over axis -1 of a tensor of shape (32, 1). This axis has size 1. The softmax operation will always return the
value 1, which is likely not what you intended. Did you mean to use a sigmoid instead?
  warnings.warn(
938/938 ━━━━━━━━━━━━━━━━━━━━ 19s 20ms/step
🎯 LSTM Accuracy for CICIDS2017: 1.0000
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     30000

    accuracy                           1.00     30000
   macro avg       1.00      1.00      1.00     30000
weighted avg       1.00      1.00      1.00     30000
```