Mounted at /content/drive

==== Training CNN Models for Each Dataset ====

🔷 Training on NSLKDD dataset...

✅ NSLKDD dataset limited to 100000 rows.

/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

super().__init__(activity_regularizer=activity_regularizer, **kwargs)

Epoch 1/10

2188/2188 ━━ 25s 10ms/step - accuracy: 0.9298 - loss: 0.1756

Epoch 2/10

2188/2188 ━━ 20s 9ms/step - accuracy: 0.9695 - loss: 0.0807

Epoch 3/10

2188/2188 ━━ 22s 10ms/step - accuracy: 0.9762 - loss: 0.0660

Epoch 4/10

2188/2188 ━━ 19s 9ms/step - accuracy: 0.9787 - loss: 0.0566

Epoch 5/10

2188/2188 ━━ 22s 10ms/step - accuracy: 0.9792 - loss: 0.0549

Epoch 6/10

2188/2188 ━━ 19s 9ms/step - accuracy: 0.9815 - loss: 0.0508

Epoch 7/10

2188/2188 ━━ 22s 9ms/step - accuracy: 0.9828 - loss: 0.0497

Epoch 8/10

2188/2188 ━━ 19s 9ms/step - accuracy: 0.9826 - loss: 0.0459

Epoch 9/10

2188/2188 ━━ 22s 9ms/step - accuracy: 0.9844 - loss: 0.0439

Epoch 10/10

2188/2188 ━━ 42s 10ms/step - accuracy: 0.9842 - loss: 0.0432

938/938 ━━ 4s 4ms/step

🎯 CNN Accuracy for NSLKDD: 0.9845

	precision	recall	f1-score	support
0	0.98	0.99	0.98	14887
1	0.99	0.98	0.98	15113
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

/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

super().__init__(activity_regularizer=activity_regularizer, **kwargs)

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23s 10ms/step - accuracy: 0.9013 - loss: 0.2237

Epoch 2/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 9ms/step - accuracy: 0.9399 - loss: 0.1346

Epoch 3/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 9ms/step - accuracy: 0.9504 - loss: 0.1151

Epoch 4/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 40s 9ms/step - accuracy: 0.9601 - loss: 0.0977

Epoch 5/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 22s 10ms/step - accuracy: 0.9603 - loss: 0.0931

Epoch 6/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 9ms/step - accuracy: 0.9635 - loss: 0.0895

Epoch 7/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 22s 10ms/step - accuracy: 0.9641 - loss: 0.0844

Epoch 8/10

2188/2188 ━━ 20s 9ms/step - accuracy: 0.9652 - loss: 0.0849

Epoch 9/10

2188/2188 ━━ 22s 10ms/step - accuracy: 0.9664 - loss: 0.0796

Epoch 10/10

2188/2188 ━━ 22s 10ms/step - accuracy: 0.9684 - loss: 0.0759

938/938 ━━ 3s 3ms/step

🎯 CNN Accuracy for UNSW_NB15: 0.9662

	precision	recall	f1-score	support
0	0.96	0.95	0.95	10805
1	0.97	0.98	0.97	19195
accuracy			0.97	30000
macro avg	0.96	0.96	0.96	30000
weighted avg	0.97	0.97	0.97	30000

🔷 Training on KDDCup dataset...

✅ KDDCup dataset limited to 100000 rows.

Epoch 1/10

/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using

Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

super().__init__(activity_regularizer=activity_regularizer, **kwargs)

2188/2188 ━━ 23s 9ms/step - accuracy: 6.1964e-07 - loss: -2996931584.0000

Epoch 2/10

2188/2188 ━━ 20s 9ms/step - accuracy: 2.9520e-06 - loss: -153115262976.0000

Epoch 3/10

2188/2188 ━━ 21s 9ms/step - accuracy: 1.4755e-06 - loss: -949813116928.0000

Epoch 4/10

2188/2188 ━━ 19s 9ms/step - accuracy: 3.7009e-06 - loss: -2968876482560.0000

Epoch 5/10

2188/2188 ━━ 21s 9ms/step - accuracy: 1.2386e-05 - loss: -6823767703552.0000

Epoch 6/10

2188/2188 ━━ 19s 9ms/step - accuracy: 1.6980e-05 - loss: -13302665052160.0000

Epoch 7/10

2188/2188 ━━ 22s 9ms/step - accuracy: 1.3449e-05 - loss: -23135480971264.0000

Epoch 8/10

2188/2188 ━━ 19s 9ms/step - accuracy: 1.4616e-05 - loss: -37200456581120.0000

Epoch 9/10

2188/2188 ━━ 23s 10ms/step - accuracy: 3.7178e-06 - loss: -56775705362432.0000

Epoch 10/10

2188/2188 ━━ 40s 9ms/step - accuracy: 1.4418e-07 - loss: -82653306945536.0000

938/938 ━━ 3s 3ms/step

🎯 CNN Accuracy for KDDCup: 0.0000

	precision	recall	f1-score	support
0	0.00	0.00	0.00	21
1	0.00	1.00	0.00	1
2	0.00	0.00	0.00	1
3	0.00	0.00	0.00	84
6	0.00	0.00	0.00	3795
7	0.00	0.00	0.00	26
8	0.00	0.00	0.00	6384
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	73
11	0.00	0.00	0.00	135
12	0.00	0.00	0.00	19466
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	12
accuracy			0.00	30000
macro avg	0.00	0.08	0.00	30000
weighted avg	0.00	0.00	0.00	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/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

super().__init__(activity_regularizer=activity_regularizer, **kwargs)

2188/2188 ━━━━━━━━━━━━━━━━━━━━ 33s 14ms/step - accuracy: 0.9974 - loss: 0.0070

Epoch 2/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━ 31s 14ms/step - accuracy: 1.0000 - loss: 2.4812e-08

Epoch 3/10

2188/2188 ━━━━━━━━━━━━━━━━━━━━ 31s 14ms/step - accuracy: 1.0000 - loss: 6.6633e-09

Epoch 4/10

2188/2188 ━━ 44s 16ms/step - accuracy: 1.0000 - loss: 5.2692e-09

Epoch 5/10

2188/2188 ━━ 38s 14ms/step - accuracy: 1.0000 - loss: 2.0006e-09

Epoch 6/10

2188/2188 ━━ 32s 15ms/step - accuracy: 1.0000 - loss: 4.6004e-10

Epoch 7/10

2188/2188 ━━ 31s 14ms/step - accuracy: 1.0000 - loss: 3.3216e-10

Epoch 8/10

2188/2188 ━━ 31s 14ms/step - accuracy: 1.0000 - loss: 9.6069e-10

Epoch 9/10

2188/2188 ━━ 31s 14ms/step - accuracy: 1.0000 - loss: 7.6048e-11

Epoch 10/10

2188/2188 ━━ 32s 15ms/step - accuracy: 1.0000 - loss: 8.9653e-11

938/938 ━━ 5s 5ms/step

🎯 CNN 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