cse-487-projeckt

November 26, 2025

[1]:
```
!pip install -q transformers
!pip install -q torch torchvision torchaudio
!pip install -q torch-geometric
!pip install -q torch-scatter torch-sparse torch-cluster -f https://data.pyg.
 ↪org/whl/torch-2.1.0+cu118.html
!pip install -q scikit-learn
!pip install -q xgboost
!pip install -q lightgbm
!pip install -q tensorflow
!pip install -q gensim
!pip install -q tensorflow-hub
!pip install -q nltk
!pip install -q lime
!pip install -q networkx
!pip install -q matplotlib seaborn plotly
!pip install -q pandas numpy scipy
!pip install -q imbalanced-learn
```

```
                         363.4/363.4 MB
4.7 MB/s eta 0:00:00
                         13.8/13.8 MB
101.4 MB/s eta 0:00:00
                         24.6/24.6 MB
80.6 MB/s eta 0:00:00
                         883.7/883.7 kB
48.5 MB/s eta 0:00:00
                         664.8/664.8 MB
2.5 MB/s eta 0:00:00
                         211.5/211.5 MB
8.0 MB/s eta 0:00:00
                         56.3/56.3 MB
30.9 MB/s eta 0:00:00
                         127.9/127.9 MB
13.5 MB/s eta 0:00:00
                         207.5/207.5 MB
8.2 MB/s eta 0:00:00
                         21.1/21.1 MB
87.8 MB/s eta 0:00:00
```

```
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of the
following dependency conflicts.
libcugraph-cu12 25.6.0 requires libraft-cu12==25.6.*, but you have libraft-cu12
25.2.0 which is incompatible.
pylibcugraph-cu12 25.6.0 requires pylibraft-cu12==25.6.*, but you have
pylibraft-cu12 25.2.0 which is incompatible.
pylibcugraph-cu12 25.6.0 requires rmm-cu12==25.6.*, but you have rmm-cu12 25.2.0
which is incompatible.
                           63.7/63.7 kB
2.1 MB/s eta 0:00:00
                          1.3/1.3 MB
22.2 MB/s eta 0:00:00
                           10.2/10.2 MB
71.8 MB/s eta 0:00:00
                           4.9/4.9 MB
89.5 MB/s eta 0:00:00
                           3.3/3.3 MB
82.0 MB/s eta 0:00:00
                          319.9/319.9 kB
7.2 MB/s eta 0:00:00
```

```
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of the
following dependency conflicts.
bigframes 2.12.0 requires google-cloud-bigquery-storage<3.0.0,>=2.30.0, which is
not installed.
google-cloud-translate 3.12.1 requires protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21
.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.19.5, but you have protobuf
5.29.5 which is incompatible.
ray 2.51.1 requires click!=8.3.0,>=7.0, but you have click 8.3.0 which is
incompatible.
bigframes 2.12.0 requires rich<14,>=12.4.4, but you have rich 14.2.0 which is
incompatible.
pydrive2 1.21.3 requires cryptography<44, but you have cryptography 46.0.3 which
is incompatible.
pydrive2 1.21.3 requires pyOpenSSL<=24.2.1,>=19.1.0, but you have pyopenssl
25.3.0 which is incompatible.
gcsfs 2025.3.0 requires fsspec==2025.3.0, but you have fsspec 2025.10.0 which is
incompatible.
                         13.5/13.5 MB
100.7 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of the
following dependency conflicts.
category-encoders 2.7.0 requires scikit-learn<1.6.0,>=1.0.0, but you have
scikit-learn 1.6.1 which is incompatible.
cesium 0.12.4 requires numpy<3.0,>=2.0, but you have numpy 1.26.4 which is
incompatible.
```

[2]:
```python
import nltk
nltk.download('punkt')
nltk.download('stopwords')

print("All packages installed successfully!")
```

```
All packages installed successfully!
```

```
[nltk_data] Downloading package punkt to /usr/share/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /usr/share/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[3]:
```
import shutil
import os

src_dir = '/kaggle/input/cse-487-porjeckt'
dest_dir = '/kaggle/working/cse-487-porjeckt'  # You can change this if you want

# Copy the whole folder tree (preserves directory structure)
if os.path.exists(dest_dir):
    shutil.rmtree(dest_dir)  # Remove existing copy if it exists
shutil.copytree(src_dir, dest_dir)
print(f'Copied all files to: {dest_dir}')
```

```
Copied all files to: /kaggle/working/cse-487-porjeckt
```

[4]:
```
import sys
sys.path.append('/kaggle/working/cse-487-porjeckt')
from master_runner import MasterPipeline, GLOBAL_CONFIG

pipeline = MasterPipeline(GLOBAL_CONFIG)
pipeline.run_full_pipeline()
```

```
2025-11-20 01:23:48.689537: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to
STDERR
E0000 00:00:1763601828.897048      19 cuda_dnn.cc:8310] Unable to register cuDNN
factory: Attempting to register factory for plugin cuDNN when one has already
been registered
E0000 00:00:1763601828.956531      19 cuda_blas.cc:1418] Unable to register
cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has
already been registered
```

```
======================================================================
    COMPREHENSIVE WEB ATTACK DETECTION RESEARCH PIPELINE
======================================================================

Start: 2025-11-20 01:24:05
Random seed: 42
Output: /kaggle/working/web_attack_detection
```

```
[STEP 1/10] SETUP AND CONFIGURATION
----------------------------------------------------------------------
Directory structure created successfully!

======================================================================
GPU AVAILABILITY CHECK
======================================================================

 TensorFlow:
   No GPU found, using CPU

 PyTorch:
   CUDA not available, using CPU
======================================================================

Current memory usage: 1.11 GB
Setup complete!

[STEP 2/10] DATA PREPROCESSING
----------------------------------------------------------------------

======================================================================
LOADING ALL DATASETS
======================================================================

Loading XSS_dataset.csv…
  Loaded 13686 samples
  Label distribution: {1: 7373, 0: 6313}
Loading Train.csv…
2025-11-20 01:24:05.337547: E
external/local_xla/xla/stream_executor/cuda/cuda_driver.cc:152] failed call to
cuInit: INTERNAL: CUDA error: Failed call to cuInit: CUDA_ERROR_NO_DEVICE: no
CUDA-capable device is detected
  Loaded 98062 samples
  Label distribution: {0: 51377, 1: 46685}
Loading Test.csv…
  Loaded 32688 samples
  Label distribution: {0: 17194, 1: 15494}
Loading Validation.csv…
  Loaded 32687 samples
  Label distribution: {0: 17115, 1: 15572}
Loading Modified_SQL_Dataset.csv…
  Loaded 30919 samples
  Label distribution: {0: 19537, 1: 11382}

======================================================================
```

```
MERGING DATASETS
======================================================================

Total samples: 208042
Total features: 3
Label distribution:
label
0    111536
1     96506
Name: count, dtype: int64

Dataset sources:
dataset_source
sql_train       98062
sql_test        32688
sql_val         32687
sql_modified    30919
xss             13686
Name: count, dtype: int64

Checking for duplicates…
  Found 33689 duplicate rows
  Removed duplicates, 174353 samples remaining

======================================================================
PREPROCESSING DATA (Content Matching)
======================================================================

Applying content matching preprocessing…
Preprocessing complete!

Example transformations:

  Original: <li><a href="/wiki/File:Socrates.png" class="image"><img
alt="Socrates.png" src="//upload.wikimedia…
  Processed:  <LT <GT>  li <GT>  <LT <GT>  a href  <LT <GT>  EQUALS <GT>    <LT
<GT>  DQUOTE <GT>  /wiki/file:soc…

  Original: <tt onmouseover="alert(1)">test</tt>…
  Processed:  <LT <GT>  tt  <LT <GT>  xss_onmouseover <GT>   <LT <GT>  EQUALS
<GT>    <LT <GT>  DQUOTE <GT>   <LT…

  Original:      </span> <span class="reference-text">Steering for the 1995 "<a
href="/wiki/History_of_autonomous_c…
  Processed:  <LT <GT>  /span <GT>   <LT <GT>  span class  <LT <GT>  EQUALS <GT>
<LT <GT>  DQUOTE <GT>  refere…

======================================================================
```

```
CREATING DATA SPLITS
======================================================================

Train set: 122047 samples (70.0%)
   Class distribution: [62496 59551]

Validation set: 26153 samples (15.0%)
   Class distribution: [13392 12761]

Test set: 26153 samples (15.0%)
   Class distribution: [13392 12761]
Data preprocessing complete!

[STEP 3/10] FEATURE EXTRACTION
----------------------------------------------------------------------

======================================================================
STAGE 1: UNIEMBED FEATURES
======================================================================
Training Word2Vec model…
  Word2Vec trained: 628443 tokens
Training FastText model…
  FastText trained: 628443 tokens
Loading Universal Sentence Encoder…
  USE loaded successfully

======================================================================
EXTRACTING UNIEMBED FEATURES (612D)
  Word2Vec: 50D
  FastText: 50D
  USE: 512D
======================================================================

Extracting Word2Vec features…
  Processed 10000/122047 samples
  Processed 20000/122047 samples
  Processed 30000/122047 samples
  Processed 40000/122047 samples
  Processed 50000/122047 samples
  Processed 60000/122047 samples
  Processed 70000/122047 samples
  Processed 80000/122047 samples
  Processed 90000/122047 samples
  Processed 100000/122047 samples
  Processed 110000/122047 samples
  Processed 120000/122047 samples

Extracting FastText features…
```

```
  Processed 10000/122047 samples
  Processed 20000/122047 samples
  Processed 30000/122047 samples
  Processed 40000/122047 samples
  Processed 50000/122047 samples
  Processed 60000/122047 samples
  Processed 70000/122047 samples
  Processed 80000/122047 samples
  Processed 90000/122047 samples
  Processed 100000/122047 samples
  Processed 110000/122047 samples
  Processed 120000/122047 samples

Extracting USE features…
  Processed 10000/122047 samples
  Processed 20000/122047 samples
  Processed 30000/122047 samples
  Processed 40000/122047 samples
  Processed 50000/122047 samples
  Processed 60000/122047 samples
  Processed 70000/122047 samples
  Processed 80000/122047 samples
  Processed 90000/122047 samples
  Processed 100000/122047 samples
  Processed 110000/122047 samples
  Processed 120000/122047 samples
  Processed 122047/122047 samples

UniEmbed features extracted: (122047, 612)

======================================================================
EXTRACTING UNIEMBED FEATURES (612D)
  Word2Vec: 50D
  FastText: 50D
  USE: 512D
======================================================================

Extracting Word2Vec features…
  Processed 10000/26153 samples
  Processed 20000/26153 samples

Extracting FastText features…
  Processed 10000/26153 samples
  Processed 20000/26153 samples

Extracting USE features…
  Processed 10000/26153 samples
  Processed 20000/26153 samples
```

```
  Processed 26153/26153 samples

UniEmbed features extracted: (26153, 612)

======================================================================
EXTRACTING UNIEMBED FEATURES (612D)
  Word2Vec: 50D
  FastText: 50D
  USE: 512D
======================================================================

Extracting Word2Vec features…
  Processed 10000/26153 samples
  Processed 20000/26153 samples

Extracting FastText features…
  Processed 10000/26153 samples
  Processed 20000/26153 samples

Extracting USE features…
  Processed 10000/26153 samples
  Processed 20000/26153 samples
  Processed 26153/26153 samples

UniEmbed features extracted: (26153, 612)

Saving UniEmbed models to /kaggle/working/web_attack_detection/features…
Models saved successfully!

======================================================================
STAGE 2: TF-IDF FEATURES
======================================================================

======================================================================
EXTRACTING TF-IDF FEATURES (1000D)
======================================================================

Fitting TF-IDF vectorizer…
Transforming validation set…
Transforming test set…

TF-IDF features extracted:
   Train: (122047, 1000)
   Val: (26153, 1000)
   Test: (26153, 1000)
TF-IDF vectorizer saved!

======================================================================
```

```
ALL FEATURES EXTRACTED AND SAVED!
======================================================================
Feature extraction complete!

[STEP 4/10] TRAINING CLASSICAL ML MODELS
----------------------------------------------------------------------

======================================================================
TRAINING ALL CLASSICAL ML MODELS
Feature Type: TFIDF
======================================================================


======================================================================
TRAINING: Logistic_Regression
======================================================================

 Training Logistic_Regression…
   Training samples: 122047
   Features: 1000
 Training complete in 4.49 seconds
 Model saved:
/kaggle/working/web_attack_detection/models/classical_ml/Logistic_Regression.pkl
 Metadata saved: /kaggle/working/web_attack_detection/models/classical_ml/Logist
ic_Regression_metadata.json
 Logistic_Regression complete!


======================================================================
TRAINING: SVM
======================================================================

 Training SVM…
   Training samples: 122047
   Features: 1000
 Training complete in 566.32 seconds
 Model saved: /kaggle/working/web_attack_detection/models/classical_ml/SVM.pkl
 Metadata saved:
/kaggle/working/web_attack_detection/models/classical_ml/SVM_metadata.json
 SVM complete!


======================================================================
TRAINING: Gaussian_Naive_Bayes
======================================================================

 Training Gaussian_Naive_Bayes…
   Training samples: 122047
```

```
   Features: 1000
 Training complete in 1.59 seconds
 Model saved: /kaggle/working/web_attack_detection/models/classical_ml/Gaussian_
Naive_Bayes.pkl
 Metadata saved: /kaggle/working/web_attack_detection/models/classical_ml/Gaussi
an_Naive_Bayes_metadata.json
 Gaussian_Naive_Bayes complete!


======================================================================
TRAINING: Decision_Tree
======================================================================

 Training Decision_Tree…
   Training samples: 122047
   Features: 1000
 Training complete in 18.23 seconds
 Model saved:
/kaggle/working/web_attack_detection/models/classical_ml/Decision_Tree.pkl
 Metadata saved: /kaggle/working/web_attack_detection/models/classical_ml/Decisi
on_Tree_metadata.json
 Decision_Tree complete!


======================================================================
TRAINING: KNN
======================================================================

 Training KNN…
   Training samples: 122047
   Features: 1000
 Training complete in 0.09 seconds
 Model saved: /kaggle/working/web_attack_detection/models/classical_ml/KNN.pkl
 Metadata saved:
/kaggle/working/web_attack_detection/models/classical_ml/KNN_metadata.json
 KNN complete!


======================================================================
TRAINING: Random_Forest
======================================================================

 Training Random_Forest…
   Training samples: 122047
   Features: 1000
 Training complete in 27.08 seconds
 Model saved:
/kaggle/working/web_attack_detection/models/classical_ml/Random_Forest.pkl
```

```
 Metadata saved: /kaggle/working/web_attack_detection/models/classical_ml/Random
_Forest_metadata.json
 Random_Forest complete!


======================================================================
TRAINING: XGBoost
======================================================================

 Training XGBoost…
   Training samples: 122047
   Features: 1000
 Training complete in 97.24 seconds
 Model saved:
/kaggle/working/web_attack_detection/models/classical_ml/XGBoost.pkl
 Metadata saved:
/kaggle/working/web_attack_detection/models/classical_ml/XGBoost_metadata.json
 XGBoost complete!


======================================================================
TRAINING: Gradient_Boosting
======================================================================

 Training Gradient_Boosting…
   Training samples: 122047
   Features: 1000
 Training complete in 1555.61 seconds
 Model saved:
/kaggle/working/web_attack_detection/models/classical_ml/Gradient_Boosting.pkl
 Metadata saved: /kaggle/working/web_attack_detection/models/classical_ml/Gradie
nt_Boosting_metadata.json
 Gradient_Boosting complete!


======================================================================
TRAINING: Extra_Trees
======================================================================

 Training Extra_Trees…
   Training samples: 122047
   Features: 1000
 Training complete in 31.77 seconds
 Model saved:
/kaggle/working/web_attack_detection/models/classical_ml/Extra_Trees.pkl
 Metadata saved: /kaggle/working/web_attack_detection/models/classical_ml/Extra_
Trees_metadata.json
 Extra_Trees complete!
```

```
======================================================================
 ALL CLASSICAL ML MODELS TRAINED: 9/9
======================================================================

Classical ML training complete! Trained 9 models.

[STEP 5/10] TRAINING DEEP LEARNING MODELS
----------------------------------------------------------------------

======================================================================
TRAINING ALL DEEP LEARNING MODELS
Feature Type: UNIEMBED
======================================================================


======================================================================
TRAINING: MLP
======================================================================

Preparing data…
   Train shape: (122047, 612)
   Val shape: (26153, 612)

Model Architecture:
```

Model: "MLP"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	156,928
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 198,145 (774.00 KB)

Trainable params: 198,145 (774.00 KB)

Non-trainable params: 0 (0.00 B)

```
Training MLP…
Epoch 1/20
3814/3814             17s 4ms/step -
accuracy: 0.9241 - loss: 0.7379 - precision: 0.9184 - recall: 0.9285 -
val_accuracy: 0.9698 - val_loss: 0.1730 - val_precision: 0.9754 - val_recall:
0.9625 - learning_rate: 0.0010
Epoch 2/20
3814/3814             15s 4ms/step -
accuracy: 0.9581 - loss: 0.2038 - precision: 0.9574 - recall: 0.9567 -
val_accuracy: 0.9709 - val_loss: 0.1654 - val_precision: 0.9885 - val_recall:
0.9514 - learning_rate: 0.0010
Epoch 3/20
3814/3814             15s 4ms/step -
accuracy: 0.9586 - loss: 0.1980 - precision: 0.9551 - recall: 0.9603 -
val_accuracy: 0.9671 - val_loss: 0.1693 - val_precision: 0.9890 - val_recall:
0.9430 - learning_rate: 0.0010
Epoch 4/20
3814/3814             15s 4ms/step -
accuracy: 0.9605 - loss: 0.1855 - precision: 0.9582 - recall: 0.9610 -
val_accuracy: 0.9695 - val_loss: 0.1520 - val_precision: 0.9875 - val_recall:
0.9495 - learning_rate: 0.0010
Epoch 5/20
3814/3814             15s 4ms/step -
accuracy: 0.9599 - loss: 0.1868 - precision: 0.9572 - recall: 0.9609 -
val_accuracy: 0.9763 - val_loss: 0.1380 - val_precision: 0.9913 - val_recall:
0.9598 - learning_rate: 0.0010
Epoch 6/20
3814/3814             15s 4ms/step -
accuracy: 0.9601 - loss: 0.1853 - precision: 0.9571 - recall: 0.9614 -
val_accuracy: 0.9736 - val_loss: 0.1542 - val_precision: 0.9943 - val_recall:
0.9513 - learning_rate: 0.0010
Epoch 7/20
3814/3814             15s 4ms/step -
accuracy: 0.9617 - loss: 0.1822 - precision: 0.9592 - recall: 0.9624 -
val_accuracy: 0.9789 - val_loss: 0.1357 - val_precision: 0.9924 - val_recall:
0.9642 - learning_rate: 0.0010
Epoch 8/20
3814/3814             15s 4ms/step -
accuracy: 0.9624 - loss: 0.1816 - precision: 0.9614 - recall: 0.9616 -
```

```
val_accuracy: 0.9755 - val_loss: 0.1415 - val_precision: 0.9942 - val_recall:
0.9554 - learning_rate: 0.0010
Epoch 9/20
3814/3814            15s 4ms/step -
accuracy: 0.9604 - loss: 0.1830 - precision: 0.9588 - recall: 0.9601 -
val_accuracy: 0.9787 - val_loss: 0.1198 - val_precision: 0.9841 - val_recall:
0.9721 - learning_rate: 0.0010
Epoch 10/20
3814/3814            15s 4ms/step -
accuracy: 0.9619 - loss: 0.1776 - precision: 0.9610 - recall: 0.9610 -
val_accuracy: 0.9704 - val_loss: 0.1457 - val_precision: 0.9941 - val_recall:
0.9450 - learning_rate: 0.0010
Epoch 11/20
3814/3814            15s 4ms/step -
accuracy: 0.9610 - loss: 0.1793 - precision: 0.9596 - recall: 0.9605 -
val_accuracy: 0.9685 - val_loss: 0.1501 - val_precision: 0.9957 - val_recall:
0.9395 - learning_rate: 0.0010
Epoch 12/20
3813/3814            0s 4ms/step -
accuracy: 0.9612 - loss: 0.1798 - precision: 0.9612 - recall: 0.9592
Epoch 12: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
3814/3814            15s 4ms/step -
accuracy: 0.9612 - loss: 0.1798 - precision: 0.9612 - recall: 0.9592 -
val_accuracy: 0.9781 - val_loss: 0.1276 - val_precision: 0.9928 - val_recall:
0.9621 - learning_rate: 0.0010
Epoch 13/20
3814/3814            15s 4ms/step -
accuracy: 0.9675 - loss: 0.1585 - precision: 0.9679 - recall: 0.9655 -
val_accuracy: 0.9768 - val_loss: 0.1208 - val_precision: 0.9906 - val_recall:
0.9615 - learning_rate: 5.0000e-04
Epoch 14/20
3814/3814            15s 4ms/step -
accuracy: 0.9677 - loss: 0.1527 - precision: 0.9673 - recall: 0.9665 -
val_accuracy: 0.9753 - val_loss: 0.1210 - val_precision: 0.9929 - val_recall:
0.9562 - learning_rate: 5.0000e-04
Epoch 14: early stopping
Restoring model weights from the end of the best epoch: 9.

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.


Training complete in 212.35 seconds
Model saved: /kaggle/working/web_attack_detection/models/deep_learning/MLP.h5
Error training MLP: The filename must end in `.weights.h5`. Received: filepath=/
kaggle/working/web_attack_detection/models/deep_learning/MLP_weights.h5
```

```
======================================================================
TRAINING: CNN
======================================================================

Preparing data…
   Train shape: (122047, 612, 1)
   Val shape: (26153, 612, 1)

Model Architecture:
```

Model: "CNN"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 612, 1)	0
conv1d (Conv1D)	(None, 612, 16)	64
max_pooling1d (MaxPooling1D)	(None, 306, 16)	0
conv1d_1 (Conv1D)	(None, 306, 32)	1,568
max_pooling1d_1 (MaxPooling1D)	(None, 153, 32)	0
flatten (Flatten)	(None, 4896)	0
dense (Dense)	(None, 64)	313,408
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 315,105 (1.20 MB)

Trainable params: 315,105 (1.20 MB)

Non-trainable params: 0 (0.00 B)

```
Training CNN…
Epoch 1/20
3814/3814             44s 11ms/step -
```

```
accuracy: 0.9311 - loss: 0.2647 - precision: 0.9260 - recall: 0.9342 -
val_accuracy: 0.9695 - val_loss: 0.1263 - val_precision: 0.9906 - val_recall:
0.9465 - learning_rate: 0.0010
Epoch 2/20
3814/3814            41s 11ms/step -
accuracy: 0.9749 - loss: 0.1091 - precision: 0.9775 - recall: 0.9709 -
val_accuracy: 0.9738 - val_loss: 0.1042 - val_precision: 0.9946 - val_recall:
0.9514 - learning_rate: 0.0010
Epoch 3/20
3814/3814            41s 11ms/step -
accuracy: 0.9779 - loss: 0.0926 - precision: 0.9816 - recall: 0.9730 -
val_accuracy: 0.9734 - val_loss: 0.0946 - val_precision: 0.9964 - val_recall:
0.9490 - learning_rate: 0.0010
Epoch 4/20
3814/3814            41s 11ms/step -
accuracy: 0.9796 - loss: 0.0862 - precision: 0.9832 - recall: 0.9747 -
val_accuracy: 0.9760 - val_loss: 0.0921 - val_precision: 0.9943 - val_recall:
0.9564 - learning_rate: 0.0010
Epoch 5/20
3814/3814            42s 11ms/step -
accuracy: 0.9809 - loss: 0.0800 - precision: 0.9853 - recall: 0.9754 -
val_accuracy: 0.9815 - val_loss: 0.0706 - val_precision: 0.9933 - val_recall:
0.9687 - learning_rate: 0.0010
Epoch 6/20
3814/3814            41s 11ms/step -
accuracy: 0.9812 - loss: 0.0773 - precision: 0.9854 - recall: 0.9760 -
val_accuracy: 0.9806 - val_loss: 0.0720 - val_precision: 0.9936 - val_recall:
0.9665 - learning_rate: 0.0010
Epoch 7/20
3814/3814            42s 11ms/step -
accuracy: 0.9819 - loss: 0.0762 - precision: 0.9863 - recall: 0.9765 -
val_accuracy: 0.9830 - val_loss: 0.0657 - val_precision: 0.9909 - val_recall:
0.9741 - learning_rate: 0.0010
Epoch 8/20
3814/3814            42s 11ms/step -
accuracy: 0.9824 - loss: 0.0725 - precision: 0.9871 - recall: 0.9767 -
val_accuracy: 0.9824 - val_loss: 0.0668 - val_precision: 0.9912 - val_recall:
0.9727 - learning_rate: 0.0010
Epoch 9/20
3814/3814            42s 11ms/step -
accuracy: 0.9823 - loss: 0.0717 - precision: 0.9869 - recall: 0.9766 -
val_accuracy: 0.9834 - val_loss: 0.0643 - val_precision: 0.9948 - val_recall:
0.9711 - learning_rate: 0.0010
Epoch 10/20
3814/3814            42s 11ms/step -
accuracy: 0.9829 - loss: 0.0689 - precision: 0.9876 - recall: 0.9772 -
val_accuracy: 0.9844 - val_loss: 0.0615 - val_precision: 0.9946 - val_recall:
0.9733 - learning_rate: 0.0010
```

```
Epoch 11/20
3814/3814            42s 11ms/step -
accuracy: 0.9831 - loss: 0.0704 - precision: 0.9875 - recall: 0.9777 -
val_accuracy: 0.9844 - val_loss: 0.0604 - val_precision: 0.9942 - val_recall:
0.9737 - learning_rate: 0.0010
Epoch 12/20
3814/3814            42s 11ms/step -
accuracy: 0.9834 - loss: 0.0677 - precision: 0.9878 - recall: 0.9782 -
val_accuracy: 0.9849 - val_loss: 0.0604 - val_precision: 0.9961 - val_recall:
0.9730 - learning_rate: 0.0010
Epoch 13/20
3814/3814            42s 11ms/step -
accuracy: 0.9839 - loss: 0.0668 - precision: 0.9886 - recall: 0.9784 -
val_accuracy: 0.9861 - val_loss: 0.0568 - val_precision: 0.9949 - val_recall:
0.9765 - learning_rate: 0.0010
Epoch 14/20
3814/3814            42s 11ms/step -
accuracy: 0.9837 - loss: 0.0663 - precision: 0.9880 - recall: 0.9784 -
val_accuracy: 0.9865 - val_loss: 0.0550 - val_precision: 0.9901 - val_recall:
0.9821 - learning_rate: 0.0010
Epoch 15/20
3814/3814            43s 11ms/step -
accuracy: 0.9838 - loss: 0.0664 - precision: 0.9887 - recall: 0.9779 -
val_accuracy: 0.9863 - val_loss: 0.0547 - val_precision: 0.9921 - val_recall:
0.9798 - learning_rate: 0.0010
Epoch 16/20
3814/3814            43s 11ms/step -
accuracy: 0.9836 - loss: 0.0654 - precision: 0.9886 - recall: 0.9778 -
val_accuracy: 0.9866 - val_loss: 0.0548 - val_precision: 0.9946 - val_recall:
0.9778 - learning_rate: 0.0010
Epoch 17/20
3814/3814            42s 11ms/step -
accuracy: 0.9840 - loss: 0.0664 - precision: 0.9891 - recall: 0.9780 -
val_accuracy: 0.9869 - val_loss: 0.0554 - val_precision: 0.9955 - val_recall:
0.9777 - learning_rate: 0.0010
Epoch 18/20
3810/3814            0s 11ms/step -
accuracy: 0.9842 - loss: 0.0651 - precision: 0.9886 - recall: 0.9789
Epoch 18: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
3814/3814            43s 11ms/step -
accuracy: 0.9842 - loss: 0.0651 - precision: 0.9886 - recall: 0.9789 -
val_accuracy: 0.9865 - val_loss: 0.0547 - val_precision: 0.9942 - val_recall:
0.9781 - learning_rate: 0.0010
Epoch 19/20
3814/3814            43s 11ms/step -
accuracy: 0.9861 - loss: 0.0557 - precision: 0.9910 - recall: 0.9803 -
val_accuracy: 0.9871 - val_loss: 0.0490 - val_precision: 0.9951 - val_recall:
0.9784 - learning_rate: 5.0000e-04
```

```
Epoch 20/20
3814/3814             43s 11ms/step -
accuracy: 0.9863 - loss: 0.0533 - precision: 0.9912 - recall: 0.9806 -
val_accuracy: 0.9870 - val_loss: 0.0503 - val_precision: 0.9960 - val_recall:
0.9773 - learning_rate: 5.0000e-04
Restoring model weights from the end of the best epoch: 19.

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.


Training complete in 845.31 seconds
Model saved: /kaggle/working/web_attack_detection/models/deep_learning/CNN.h5
Error training CNN: The filename must end in `.weights.h5`. Received: filepath=/
kaggle/working/web_attack_detection/models/deep_learning/CNN_weights.h5


======================================================================
TRAINING: LSTM
======================================================================

Preparing data…
   Train shape: (122047, 50, 12)
   Val shape: (26153, 50, 12)

Model Architecture:

Model: "LSTM"


 Layer (type)                    Output Shape                  Param #

 input_layer (InputLayer)        (None, 50, 12)                      0

 lstm (LSTM)                     (None, 50, 64)                 19,712

 dropout (Dropout)               (None, 50, 64)                      0

 lstm_1 (LSTM)                   (None, 32)                     12,416

 dropout_1 (Dropout)             (None, 32)                          0

 dense (Dense)                   (None, 64)                      2,112

 dropout_2 (Dropout)             (None, 64)                          0
```

```
 dense_1 (Dense)                 (None, 1)                      65

Total params: 34,305 (134.00 KB)

Trainable params: 34,305 (134.00 KB)

Non-trainable params: 0 (0.00 B)


Training LSTM…
Epoch 1/20
3814/3814             144s 37ms/step
- accuracy: 0.9053 - loss: 0.2366 - precision: 0.8756 - recall: 0.9508 -
val_accuracy: 0.9768 - val_loss: 0.0891 - val_precision: 0.9924 - val_recall:
0.9597 - learning_rate: 0.0010
Epoch 2/20
3814/3814             136s 36ms/step
- accuracy: 0.9786 - loss: 0.0714 - precision: 0.9835 - recall: 0.9726 -
val_accuracy: 0.9798 - val_loss: 0.0705 - val_precision: 0.9956 - val_recall:
0.9629 - learning_rate: 0.0010
Epoch 3/20
3814/3814             135s 35ms/step
- accuracy: 0.9825 - loss: 0.0571 - precision: 0.9866 - recall: 0.9774 -
val_accuracy: 0.9821 - val_loss: 0.0604 - val_precision: 0.9907 - val_recall:
0.9725 - learning_rate: 0.0010
Epoch 4/20
3814/3814             135s 35ms/step
- accuracy: 0.9848 - loss: 0.0505 - precision: 0.9884 - recall: 0.9803 -
val_accuracy: 0.9858 - val_loss: 0.0435 - val_precision: 0.9922 - val_recall:
0.9785 - learning_rate: 0.0010
Epoch 5/20
3814/3814             136s 36ms/step
- accuracy: 0.9864 - loss: 0.0451 - precision: 0.9900 - recall: 0.9822 -
val_accuracy: 0.9875 - val_loss: 0.0393 - val_precision: 0.9910 - val_recall:
0.9833 - learning_rate: 0.0010
Epoch 6/20
3814/3814             136s 36ms/step
- accuracy: 0.9874 - loss: 0.0423 - precision: 0.9904 - recall: 0.9838 -
val_accuracy: 0.9864 - val_loss: 0.0483 - val_precision: 0.9954 - val_recall:
0.9766 - learning_rate: 0.0010
Epoch 7/20
3814/3814             136s 36ms/step
- accuracy: 0.9880 - loss: 0.0399 - precision: 0.9910 - recall: 0.9844 -
val_accuracy: 0.9874 - val_loss: 0.0395 - val_precision: 0.9935 - val_recall:
0.9806 - learning_rate: 0.0010
```

```
Epoch 8/20
3813/3814            0s 34ms/step -
accuracy: 0.9891 - loss: 0.0364 - precision: 0.9918 - recall: 0.9858
Epoch 8: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
3814/3814            137s 36ms/step
- accuracy: 0.9891 - loss: 0.0364 - precision: 0.9918 - recall: 0.9858 -
val_accuracy: 0.9867 - val_loss: 0.0412 - val_precision: 0.9972 - val_recall:
0.9755 - learning_rate: 0.0010
Epoch 9/20
3814/3814            133s 35ms/step
- accuracy: 0.9904 - loss: 0.0305 - precision: 0.9930 - recall: 0.9874 -
val_accuracy: 0.9895 - val_loss: 0.0308 - val_precision: 0.9948 - val_recall:
0.9836 - learning_rate: 5.0000e-04
Epoch 10/20
3814/3814            132s 35ms/step
- accuracy: 0.9910 - loss: 0.0291 - precision: 0.9934 - recall: 0.9881 -
val_accuracy: 0.9881 - val_loss: 0.0375 - val_precision: 0.9964 - val_recall:
0.9792 - learning_rate: 5.0000e-04
Epoch 11/20
3814/3814            132s 35ms/step
- accuracy: 0.9913 - loss: 0.0278 - precision: 0.9938 - recall: 0.9884 -
val_accuracy: 0.9901 - val_loss: 0.0316 - val_precision: 0.9927 - val_recall:
0.9870 - learning_rate: 5.0000e-04
Epoch 12/20
3814/3814            132s 35ms/step
- accuracy: 0.9914 - loss: 0.0273 - precision: 0.9936 - recall: 0.9887 -
val_accuracy: 0.9902 - val_loss: 0.0289 - val_precision: 0.9948 - val_recall:
0.9850 - learning_rate: 5.0000e-04
Epoch 13/20
3814/3814            132s 35ms/step
- accuracy: 0.9919 - loss: 0.0256 - precision: 0.9942 - recall: 0.9891 -
val_accuracy: 0.9905 - val_loss: 0.0281 - val_precision: 0.9946 - val_recall:
0.9858 - learning_rate: 5.0000e-04
Epoch 14/20
3814/3814            136s 36ms/step
- accuracy: 0.9922 - loss: 0.0257 - precision: 0.9940 - recall: 0.9899 -
val_accuracy: 0.9894 - val_loss: 0.0301 - val_precision: 0.9959 - val_recall:
0.9824 - learning_rate: 5.0000e-04
Epoch 15/20
3814/3814            137s 36ms/step
- accuracy: 0.9920 - loss: 0.0250 - precision: 0.9940 - recall: 0.9895 -
val_accuracy: 0.9904 - val_loss: 0.0288 - val_precision: 0.9946 - val_recall:
0.9857 - learning_rate: 5.0000e-04
Epoch 16/20
3814/3814            0s 34ms/step -
accuracy: 0.9922 - loss: 0.0244 - precision: 0.9943 - recall: 0.9897
Epoch 16: ReduceLROnPlateau reducing learning rate to 0.0002502000118743628.
3814/3814            137s 36ms/step
```

```
- accuracy: 0.9922 - loss: 0.0244 - precision: 0.9943 - recall: 0.9897 -
val_accuracy: 0.9905 - val_loss: 0.0296 - val_precision: 0.9950 - val_recall:
0.9854 - learning_rate: 5.0000e-04
Epoch 17/20
3814/3814             136s 36ms/step
- accuracy: 0.9932 - loss: 0.0222 - precision: 0.9949 - recall: 0.9911 -
val_accuracy: 0.9918 - val_loss: 0.0268 - val_precision: 0.9926 - val_recall:
0.9905 - learning_rate: 2.5000e-04
Epoch 18/20
3814/3814             136s 36ms/step
- accuracy: 0.9936 - loss: 0.0208 - precision: 0.9951 - recall: 0.9917 -
val_accuracy: 0.9919 - val_loss: 0.0259 - val_precision: 0.9929 - val_recall:
0.9904 - learning_rate: 2.5000e-04
Epoch 19/20
3814/3814             135s 35ms/step
- accuracy: 0.9938 - loss: 0.0205 - precision: 0.9957 - recall: 0.9915 -
val_accuracy: 0.9919 - val_loss: 0.0261 - val_precision: 0.9939 - val_recall:
0.9894 - learning_rate: 2.5000e-04
Epoch 20/20
3814/3814             135s 35ms/step
- accuracy: 0.9939 - loss: 0.0201 - precision: 0.9956 - recall: 0.9919 -
val_accuracy: 0.9919 - val_loss: 0.0263 - val_precision: 0.9942 - val_recall:
0.9893 - learning_rate: 2.5000e-04
Restoring model weights from the end of the best epoch: 18.

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.


Training complete in 2709.92 seconds
Model saved: /kaggle/working/web_attack_detection/models/deep_learning/LSTM.h5
Error training LSTM: The filename must end in `.weights.h5`. Received: filepath=
/kaggle/working/web_attack_detection/models/deep_learning/LSTM_weights.h5


======================================================================
TRAINING: BiLSTM
======================================================================

Preparing data…
   Train shape: (122047, 50, 12)
   Val shape: (26153, 50, 12)

Model Architecture:

Model: "BiLSTM"
```

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 50, 12)	0
bidirectional (Bidirectional)	(None, 50, 128)	39,424
dropout (Dropout)	(None, 50, 128)	0
bidirectional_1 (Bidirectional)	(None, 64)	41,216
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 64)	4,160
dropout_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 84,865 (331.50 KB)

Trainable params: 84,865 (331.50 KB)

Non-trainable params: 0 (0.00 B)

```
Training BiLSTM…
Epoch 1/20
3814/3814             185s 47ms/step
- accuracy: 0.9348 - loss: 0.1729 - precision: 0.9308 - recall: 0.9380 -
val_accuracy: 0.9734 - val_loss: 0.0754 - val_precision: 0.9915 - val_recall:
0.9536 - learning_rate: 0.0010
Epoch 2/20
3814/3814             203s 47ms/step
- accuracy: 0.9803 - loss: 0.0615 - precision: 0.9851 - recall: 0.9745 -
val_accuracy: 0.9816 - val_loss: 0.0530 - val_precision: 0.9915 - val_recall:
0.9706 - learning_rate: 0.0010
Epoch 3/20
3814/3814             180s 47ms/step
- accuracy: 0.9834 - loss: 0.0517 - precision: 0.9873 - recall: 0.9787 -
val_accuracy: 0.9817 - val_loss: 0.0568 - val_precision: 0.9915 - val_recall:
0.9708 - learning_rate: 0.0010
Epoch 4/20
```

```
3814/3814            178s 47ms/step
- accuracy: 0.9853 - loss: 0.0454 - precision: 0.9889 - recall: 0.9808 -
val_accuracy: 0.9866 - val_loss: 0.0396 - val_precision: 0.9929 - val_recall:
0.9795 - learning_rate: 0.0010
Epoch 5/20
3814/3814            179s 47ms/step
- accuracy: 0.9870 - loss: 0.0412 - precision: 0.9906 - recall: 0.9827 -
val_accuracy: 0.9850 - val_loss: 0.0407 - val_precision: 0.9838 - val_recall:
0.9856 - learning_rate: 0.0010
Epoch 6/20
3814/3814            181s 47ms/step
- accuracy: 0.9880 - loss: 0.0378 - precision: 0.9912 - recall: 0.9842 -
val_accuracy: 0.9864 - val_loss: 0.0386 - val_precision: 0.9942 - val_recall:
0.9779 - learning_rate: 0.0010
Epoch 7/20
3814/3814            180s 47ms/step
- accuracy: 0.9893 - loss: 0.0345 - precision: 0.9926 - recall: 0.9855 -
val_accuracy: 0.9873 - val_loss: 0.0347 - val_precision: 0.9961 - val_recall:
0.9778 - learning_rate: 0.0010
Epoch 8/20
3814/3814            178s 47ms/step
- accuracy: 0.9894 - loss: 0.0332 - precision: 0.9923 - recall: 0.9860 -
val_accuracy: 0.9866 - val_loss: 0.0357 - val_precision: 0.9971 - val_recall:
0.9754 - learning_rate: 0.0010
Epoch 9/20
3814/3814            180s 47ms/step
- accuracy: 0.9901 - loss: 0.0317 - precision: 0.9930 - recall: 0.9866 -
val_accuracy: 0.9840 - val_loss: 0.0441 - val_precision: 0.9823 - val_recall:
0.9850 - learning_rate: 0.0010
Epoch 10/20
3814/3814            0s 45ms/step -
accuracy: 0.9904 - loss: 0.0303 - precision: 0.9932 - recall: 0.9871
Epoch 10: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
3814/3814            181s 47ms/step
- accuracy: 0.9904 - loss: 0.0303 - precision: 0.9932 - recall: 0.9871 -
val_accuracy: 0.9875 - val_loss: 0.0363 - val_precision: 0.9972 - val_recall:
0.9770 - learning_rate: 0.0010
Epoch 11/20
3814/3814            179s 47ms/step
- accuracy: 0.9919 - loss: 0.0257 - precision: 0.9944 - recall: 0.9890 -
val_accuracy: 0.9908 - val_loss: 0.0267 - val_precision: 0.9956 - val_recall:
0.9855 - learning_rate: 5.0000e-04
Epoch 12/20
3814/3814            173s 45ms/step
- accuracy: 0.9919 - loss: 0.0247 - precision: 0.9942 - recall: 0.9892 -
val_accuracy: 0.9914 - val_loss: 0.0254 - val_precision: 0.9967 - val_recall:
0.9857 - learning_rate: 5.0000e-04
Epoch 13/20
```

```
3814/3814             175s 46ms/step
- accuracy: 0.9928 - loss: 0.0232 - precision: 0.9951 - recall: 0.9901 -
val_accuracy: 0.9917 - val_loss: 0.0253 - val_precision: 0.9964 - val_recall:
0.9866 - learning_rate: 5.0000e-04
Epoch 14/20
3814/3814             172s 45ms/step
- accuracy: 0.9929 - loss: 0.0222 - precision: 0.9948 - recall: 0.9905 -
val_accuracy: 0.9915 - val_loss: 0.0252 - val_precision: 0.9969 - val_recall:
0.9856 - learning_rate: 5.0000e-04
Epoch 15/20
3814/3814             175s 46ms/step
- accuracy: 0.9929 - loss: 0.0215 - precision: 0.9949 - recall: 0.9906 -
val_accuracy: 0.9915 - val_loss: 0.0246 - val_precision: 0.9957 - val_recall:
0.9868 - learning_rate: 5.0000e-04
Epoch 16/20
3814/3814             176s 46ms/step
- accuracy: 0.9932 - loss: 0.0209 - precision: 0.9951 - recall: 0.9910 -
val_accuracy: 0.9919 - val_loss: 0.0248 - val_precision: 0.9960 - val_recall:
0.9873 - learning_rate: 5.0000e-04
Epoch 17/20
3814/3814             178s 47ms/step
- accuracy: 0.9933 - loss: 0.0208 - precision: 0.9955 - recall: 0.9908 -
val_accuracy: 0.9911 - val_loss: 0.0261 - val_precision: 0.9964 - val_recall:
0.9852 - learning_rate: 5.0000e-04
Epoch 18/20
3814/3814             0s 44ms/step -
accuracy: 0.9934 - loss: 0.0207 - precision: 0.9957 - recall: 0.9907
Epoch 18: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
3814/3814             178s 47ms/step
- accuracy: 0.9934 - loss: 0.0207 - precision: 0.9957 - recall: 0.9907 -
val_accuracy: 0.9903 - val_loss: 0.0298 - val_precision: 0.9974 - val_recall:
0.9827 - learning_rate: 5.0000e-04
Epoch 19/20
3814/3814             177s 46ms/step
- accuracy: 0.9938 - loss: 0.0187 - precision: 0.9960 - recall: 0.9914 -
val_accuracy: 0.9927 - val_loss: 0.0229 - val_precision: 0.9944 - val_recall:
0.9907 - learning_rate: 2.5000e-04
Epoch 20/20
3814/3814             180s 47ms/step
- accuracy: 0.9944 - loss: 0.0173 - precision: 0.9964 - recall: 0.9921 -
val_accuracy: 0.9927 - val_loss: 0.0239 - val_precision: 0.9931 - val_recall:
0.9920 - learning_rate: 2.5000e-04
Restoring model weights from the end of the best epoch: 19.

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
```

'my_model.keras')`.

Training complete in 3587.26 seconds
Model saved: /kaggle/working/web_attack_detection/models/deep_learning/BiLSTM.h5
Error training BiLSTM: The filename must end in `.weights.h5`. Received: filepat
h=/kaggle/working/web_attack_detection/models/deep_learning/BiLSTM_weights.h5

==========
TRAINING: CNN_LSTM
==========

Preparing data…
Train shape: (122047, 50, 12)
Val shape: (26153, 50, 12)

Model Architecture:

Model: "CNN_LSTM"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 50, 12)	0
conv1d (Conv1D)	(None, 50, 32)	1,184
max_pooling1d (MaxPooling1D)	(None, 25, 32)	0
conv1d_1 (Conv1D)	(None, 25, 64)	6,208
max_pooling1d_1 (MaxPooling1D)	(None, 12, 64)	0
lstm (LSTM)	(None, 64)	33,024
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33

Total params: 42,529 (166.13 KB)

Trainable params: 42,529 (166.13 KB)

Non-trainable params: 0 (0.00 B)

```
Training CNN_LSTM…
Epoch 1/20
3814/3814            39s 9ms/step -
accuracy: 0.9378 - loss: 0.1665 - precision: 0.9296 - recall: 0.9485 -
val_accuracy: 0.9758 - val_loss: 0.0920 - val_precision: 0.9957 - val_recall:
0.9545 - learning_rate: 0.0010
Epoch 2/20
3814/3814            35s 9ms/step -
accuracy: 0.9802 - loss: 0.0647 - precision: 0.9853 - recall: 0.9739 -
val_accuracy: 0.9668 - val_loss: 0.1091 - val_precision: 0.9992 - val_recall:
0.9328 - learning_rate: 0.0010
Epoch 3/20
3814/3814            34s 9ms/step -
accuracy: 0.9834 - loss: 0.0537 - precision: 0.9879 - recall: 0.9781 -
val_accuracy: 0.9787 - val_loss: 0.0628 - val_precision: 0.9989 - val_recall:
0.9575 - learning_rate: 0.0010
Epoch 4/20
3814/3814            35s 9ms/step -
accuracy: 0.9851 - loss: 0.0477 - precision: 0.9895 - recall: 0.9798 -
val_accuracy: 0.9810 - val_loss: 0.0587 - val_precision: 0.9930 - val_recall:
0.9679 - learning_rate: 0.0010
Epoch 5/20
3814/3814            35s 9ms/step -
accuracy: 0.9864 - loss: 0.0444 - precision: 0.9906 - recall: 0.9813 -
val_accuracy: 0.9886 - val_loss: 0.0353 - val_precision: 0.9953 - val_recall:
0.9813 - learning_rate: 0.0010
Epoch 6/20
3814/3814            35s 9ms/step -
accuracy: 0.9871 - loss: 0.0401 - precision: 0.9910 - recall: 0.9826 -
val_accuracy: 0.9872 - val_loss: 0.0367 - val_precision: 0.9951 - val_recall:
0.9786 - learning_rate: 0.0010
Epoch 7/20
3814/3814            34s 9ms/step -
accuracy: 0.9887 - loss: 0.0366 - precision: 0.9922 - recall: 0.9846 -
val_accuracy: 0.9879 - val_loss: 0.0348 - val_precision: 0.9979 - val_recall:
0.9772 - learning_rate: 0.0010
Epoch 8/20
3814/3814            35s 9ms/step -
accuracy: 0.9890 - loss: 0.0349 - precision: 0.9927 - recall: 0.9847 -
val_accuracy: 0.9848 - val_loss: 0.0437 - val_precision: 0.9884 - val_recall:
0.9803 - learning_rate: 0.0010
Epoch 9/20
```

```
3814/3814            34s 9ms/step -
accuracy: 0.9893 - loss: 0.0341 - precision: 0.9930 - recall: 0.9849 -
val_accuracy: 0.9904 - val_loss: 0.0289 - val_precision: 0.9935 - val_recall:
0.9869 - learning_rate: 0.0010
Epoch 10/20
3814/3814            35s 9ms/step -
accuracy: 0.9903 - loss: 0.0305 - precision: 0.9941 - recall: 0.9860 -
val_accuracy: 0.9894 - val_loss: 0.0311 - val_precision: 0.9952 - val_recall:
0.9832 - learning_rate: 0.0010
Epoch 11/20
3814/3814            34s 9ms/step -
accuracy: 0.9905 - loss: 0.0295 - precision: 0.9939 - recall: 0.9867 -
val_accuracy: 0.9881 - val_loss: 0.0340 - val_precision: 0.9951 - val_recall:
0.9804 - learning_rate: 0.0010
Epoch 12/20
3809/3814            0s 9ms/step -
accuracy: 0.9911 - loss: 0.0283 - precision: 0.9941 - recall: 0.9876
Epoch 12: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
3814/3814            36s 9ms/step -
accuracy: 0.9911 - loss: 0.0283 - precision: 0.9941 - recall: 0.9876 -
val_accuracy: 0.9881 - val_loss: 0.0350 - val_precision: 0.9970 - val_recall:
0.9787 - learning_rate: 0.0010
Epoch 13/20
3814/3814            34s 9ms/step -
accuracy: 0.9924 - loss: 0.0234 - precision: 0.9949 - recall: 0.9896 -
val_accuracy: 0.9901 - val_loss: 0.0302 - val_precision: 0.9977 - val_recall:
0.9820 - learning_rate: 5.0000e-04
Epoch 14/20
3814/3814            35s 9ms/step -
accuracy: 0.9931 - loss: 0.0218 - precision: 0.9956 - recall: 0.9901 -
val_accuracy: 0.9902 - val_loss: 0.0298 - val_precision: 0.9972 - val_recall:
0.9826 - learning_rate: 5.0000e-04
Epoch 14: early stopping
Restoring model weights from the end of the best epoch: 9.

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.


Training complete in 492.57 seconds
Model saved:
/kaggle/working/web_attack_detection/models/deep_learning/CNN_LSTM.h5
Error training CNN_LSTM: The filename must end in `.weights.h5`. Received: filep
ath=/kaggle/working/web_attack_detection/models/deep_learning/CNN_LSTM_weights.h
5
```

```
======================================================================
ALL DEEP LEARNING MODELS TRAINED: 0/5
======================================================================

Deep learning training complete! Trained 0 models.

[STEP 6/10] FINE-TUNING TRANSFORMER MODELS
----------------------------------------------------------------------

======================================================================
FINE-TUNING ALL TRANSFORMER MODELS
======================================================================


======================================================================
FINE-TUNING: DistilBERT
======================================================================

Using 12204 samples (10.0% of training data)
Loading distilbert-base-uncased…

tokenizer_config.json:   0%|          | 0.00/48.0 [00:00<?, ?B/s]

vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]

config.json:   0%|          | 0.00/483 [00:00<?, ?B/s]

model.safetensors:   0%|          | 0.00/268M [00:00<?, ?B/s]

Some weights of DistilBertForSequenceClassification were not initialized from
the model checkpoint at distilbert-base-uncased and are newly initialized:
['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.

DistilBERT loaded successfully
Creating data loaders…
   Train batches: 12204
   Val batches: 13077
   Effective batch size: 32

Fine-tuning DistilBERT for 1 epochs…

Epoch 1/1

Training:   0%|          | 0/12204 [00:00<?, ?it/s]

Evaluating:   0%|          | 0/13077 [00:00<?, ?it/s]
```

```
   Train Loss: 0.1732, Train Acc: 0.9412
   Val Loss: 0.0985, Val Acc: 0.9736
   New best model!

Fine-tuning complete in 4773.44 seconds
Model and tokenizer saved:
/kaggle/working/web_attack_detection/models/transformers/DistilBERT
Metadata saved: /kaggle/working/web_attack_detection/models/transformers/DistilB
ERT/metadata.json
History saved:
/kaggle/working/web_attack_detection/models/transformers/DistilBERT/history.json
DistilBERT complete!


======================================================================
ALL TRANSFORMER MODELS TRAINED: 1/1
======================================================================

Transformer training complete! Trained 1 models.

[STEP 7/10] TRAINING GRAPH NEURAL NETWORK MODELS
----------------------------------------------------------------------

Applying aggressive sampling for GNN…

GNN Dataset sizes:
  Train: 6000
  Val: 1200
  Test: 1200

Constructing training graphs…
Converting texts to graphs…
  Processed 1000/6000 graphs
  Processed 2000/6000 graphs
  Processed 3000/6000 graphs
  Processed 4000/6000 graphs
  Processed 5000/6000 graphs
  Processed 6000/6000 graphs
Created 6000 graphs
Constructing validation graphs…
Converting texts to graphs…
  Processed 1000/1200 graphs
Created 1200 graphs
Constructing test graphs…
Converting texts to graphs…
  Processed 1000/1200 graphs
Created 1200 graphs
```

```
Saving graphs…
Graphs saved!

======================================================================
TRAINING ALL GNN MODELS
======================================================================


======================================================================
TRAINING: GCN
======================================================================

   Train graphs: 6000
   Val graphs: 1200
   Node feature dim: 57
Creating GCN with input_dim=57

Model: GCN
   Parameters: 17826

Training for 20 epochs…
Epoch 5/20: Train Loss: 0.1099, Train Acc: 0.9608, Val Acc: 0.9742
Epoch 10/20: Train Loss: 0.0826, Train Acc: 0.9740, Val Acc: 0.9758
Epoch 15/20: Train Loss: 0.0703, Train Acc: 0.9787, Val Acc: 0.9767
Epoch 20/20: Train Loss: 0.0619, Train Acc: 0.9797, Val Acc: 0.9775

Training complete in 1045.25 seconds
Model saved: /kaggle/working/web_attack_detection/models/gnn/GCN.pt
Metadata saved:
/kaggle/working/web_attack_detection/models/gnn/GCN_metadata.json
History saved: /kaggle/working/web_attack_detection/models/gnn/GCN_history.json
GCN complete!


======================================================================
TRAINING: GAT
======================================================================

   Train graphs: 6000
   Val graphs: 1200
   Node feature dim: 57
Creating GAT with input_dim=57

Model: GAT
   Parameters: 603810

Training for 20 epochs…
Epoch 5/20: Train Loss: 0.0800, Train Acc: 0.9783, Val Acc: 0.9858
```

```
Epoch 10/20: Train Loss: 0.0417, Train Acc: 0.9897, Val Acc: 0.9892
Epoch 15/20: Train Loss: 0.0416, Train Acc: 0.9888, Val Acc: 0.9875
Epoch 20/20: Train Loss: 0.0326, Train Acc: 0.9913, Val Acc: 0.9875

Training complete in 11189.05 seconds
Model saved: /kaggle/working/web_attack_detection/models/gnn/GAT.pt
Metadata saved:
/kaggle/working/web_attack_detection/models/gnn/GAT_metadata.json
History saved: /kaggle/working/web_attack_detection/models/gnn/GAT_history.json
GAT complete!


======================================================================
 ALL GNN MODELS TRAINED: 2/2
======================================================================

GNN training complete! Trained 2 models.

[STEP 8/10] TRAINING HYBRID ENSEMBLE MODELS
----------------------------------------------------------------------
BERT embeddings not found, skipping BERT_XGBoost hybrid.

======================================================================
TRAINING ALL HYBRID MODELS
======================================================================


======================================================================
TRAINING: Stacking_Ensemble
======================================================================

Training stacking ensemble (this may take a while)…
Error training Stacking_Ensemble: 'super' object has no attribute
'__sklearn_tags__'


======================================================================
TRAINING: Soft_Voting
======================================================================

Training soft voting ensemble…
Error training Soft_Voting: 'super' object has no attribute '__sklearn_tags__'


======================================================================
TRAINING: Hard_Voting
======================================================================
```

```
Training hard voting ensemble…
Error training Hard_Voting: 'super' object has no attribute '__sklearn_tags__'


======================================================================
ALL HYBRID MODELS TRAINED: 0/3
======================================================================

Hybrid training complete! Trained 0 models.

[STEP 9/10] EVALUATING ALL MODELS
----------------------------------------------------------------------

======================================================================
COMPREHENSIVE MODEL EVALUATION
======================================================================

----------------------------------------------------------------------
EVALUATING CLASSICAL ML MODELS
----------------------------------------------------------------------

 Evaluating Logistic_Regression…
   Accuracy: 0.9845
   F1-Score: 0.9840
   Inference time: 0.00 ms/sample

 Evaluating SVM…
   Accuracy: 0.8711
   F1-Score: 0.8820
   Inference time: 0.77 ms/sample

 Evaluating Gaussian_Naive_Bayes…
   Accuracy: 0.9004
   F1-Score: 0.8869
   Inference time: 0.01 ms/sample

 Evaluating Decision_Tree…
   Accuracy: 0.9898
   F1-Score: 0.9895
   Inference time: 0.00 ms/sample

 Evaluating KNN…
   Accuracy: 0.9674
   F1-Score: 0.9674
   Inference time: 2.93 ms/sample

 Evaluating Random_Forest…
   Accuracy: 0.9929
```

```
   F1-Score: 0.9927
   Inference time: 0.01 ms/sample

 Evaluating XGBoost…
   Accuracy: 0.9951
   F1-Score: 0.9949
   Inference time: 0.01 ms/sample

 Evaluating Gradient_Boosting…
   Accuracy: 0.9938
   F1-Score: 0.9937
   Inference time: 0.01 ms/sample

 Evaluating Extra_Trees…

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or evaluate the
model.

   Accuracy: 0.9862
   F1-Score: 0.9858
   Inference time: 0.01 ms/sample

----------------------------------------------------------------------
EVALUATING DEEP LEARNING MODELS
----------------------------------------------------------------------

 Evaluating MLP…

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or evaluate the
model.

   Accuracy: 0.9776
   F1-Score: 0.9769
   Parameters: 198,145

 Evaluating CNN…

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or evaluate the
model.

   Accuracy: 0.9878
   F1-Score: 0.9875
   Parameters: 315,105

 Evaluating LSTM…

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or evaluate the
```

```
model.

   Accuracy: 0.9911
   F1-Score: 0.9909
   Parameters: 34,305

 Evaluating BiLSTM…

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or evaluate the
model.

   Accuracy: 0.9932
   F1-Score: 0.9931
   Parameters: 84,865

 Evaluating CNN_LSTM…
   Accuracy: 0.9907
   F1-Score: 0.9904
   Parameters: 42,529

----------------------------------------------------------------------
EVALUATING TRANSFORMER MODELS
----------------------------------------------------------------------

 Evaluating DistilBERT…
   Accuracy: 0.9739
   F1-Score: 0.9727

 Evaluating BERT…
 Error evaluating BERT: Repo id must be in the form 'repo_name' or
'namespace/repo_name':
'/kaggle/working/web_attack_detection/models/transformers/BERT'. Use `repo_type`
argument if needed.

----------------------------------------------------------------------
EVALUATING GNN MODELS
----------------------------------------------------------------------
 Loaded 1200 test graphs for GNN evaluation

 Evaluating GCN…
   Accuracy: 0.9808
   F1-Score: 0.9796

 Evaluating GAT…
   Accuracy: 0.9917
   F1-Score: 0.9912

Results saved:
```

```
/kaggle/working/web_attack_detection/results/metrics/all_models_metrics.csv

======================================================================
EVALUATION SUMMARY - ALL MODELS
======================================================================
          model_name    model_type  accuracy  precision   recall  f1_score
roc_auc
             XGBoost  Classical ML  0.995067   0.997636 0.992242  0.994932
0.999069
   Gradient_Boosting  Classical ML  0.993844   0.997002 0.990361  0.993671
0.998813
              BiLSTM Deep Learning  0.993232   0.994576 0.991537  0.993054
0.999192
       Random_Forest  Classical ML  0.992888   0.997941 0.987462  0.992674
0.998309
                 GAT           GNN  0.991667   0.992933 0.989437  0.991182
NaN
                LSTM Deep Learning  0.991129   0.991449 0.990361  0.990905
0.998902
            CNN_LSTM Deep Learning  0.990670   0.993145 0.987697  0.990413
0.998806
       Decision_Tree  Classical ML  0.989791   0.995243 0.983779  0.989478
0.991352
                 CNN Deep Learning  0.987841   0.994201 0.980801  0.987456
0.997933
         Extra_Trees  Classical ML  0.986235   0.995287 0.976413  0.985759
0.996699
 Logistic_Regression  Classical ML  0.984514   0.991175 0.976961  0.984017
0.996411
                 GCN           GNN  0.980833   0.987478 0.971831  0.979592
NaN
                 MLP Deep Learning  0.977555   0.980806 0.973043  0.976909
0.995954
          DistilBERT   Transformer  0.973884   0.992497 0.953687  0.972705
0.987310
                 KNN  Classical ML  0.967422   0.945729 0.990048  0.967381
0.991863
Gaussian_Naive_Bayes  Classical ML  0.900394   0.994161 0.800564  0.886921
0.973109
                 SVM  Classical ML  0.871143   0.797202 0.986992  0.882003
0.989092
Evaluation complete! Evaluated 17 models.

[STEP 10/10] GENERATING VISUALIZATIONS
----------------------------------------------------------------------

======================================================================
GENERATING COMPARATIVE VISUALIZATIONS
```

```
======================================================================

Saved accuracy comparison: /kaggle/working/web_attack_detection/visualizations/c
omparative/accuracy_comparison.png
Saved F1-score comparison: /kaggle/working/web_attack_detection/visualizations/c
omparative/f1_score_comparison.png
Saved training time comparison: /kaggle/working/web_attack_detection/visualizati
ons/comparative/training_time_comparison.png
Saved precision-recall scatter: /kaggle/working/web_attack_detection/visualizati
ons/comparative/precision_recall_scatter.png
Saved radar chart: /kaggle/working/web_attack_detection/visualizations/comparati
ve/radar_chart_top_models.png
Saved error analysis heatmap: /kaggle/working/web_attack_detection/visualization
s/comparative/error_analysis_heatmap.png

All comparative visualizations generated!
Visualization complete!

======================================================================
    PIPELINE COMPLETE!
======================================================================
End: 2025-11-20 09:50:33
```