**Intrusion Detection System (IDS) with Wireshark Integration**

A comprehensive machine learning-based Intrusion Detection System that supports real-time network traffic analysis, PCAP file processing, and live packet capture integration with Wireshark.

**Features**

* **Multiple ML Models**: CNN, LSTM, and Transformer architectures
* **Live Packet Capture**: Real-time network traffic monitoring
* **PCAP File Analysis**: Process Wireshark capture files
* **Interactive Web UI**: User-friendly Streamlit interface
* **Explainable AI**: LIME and SHAP model explanations
* **Simulation Mode**: Generate realistic network traffic for testing

**Project Structure**

project/

├── main.py # Model training script

├── ui\_with\_wireshark.py # Enhanced UI with Wireshark integration

├── models.py # CNN, LSTM, Transformer model definitions

├── utils.py # Data preprocessing utilities

├── explain.py # XAI explanations (LIME/SHAP)

├── nsl\_kdd\_cleaned.csv # NSL KDD Dataset

├── saved\_model\_\*.pt # Trained model files (generated)

├── label\_encoder\_\*.pkl # Label encoders (generated)

└── reports/ # XAI explanation outputs (generated)

**Installation & Setup**

**1. Install Python Dependencies**

pip install torch torchvision torchaudio

pip install streamlit pandas numpy scikit-learn

pip install scapy matplotlib seaborn

pip install lime shap

pip install joblib cryptography

**2. Windows Users - Install Packet Capture Support**

For real packet capture on Windows, install **Npcap**:

* Download from: https://nmap.org/npcap/
* Install with default settings
* Restart your computer after installation

*Note: If you skip this step, you can still use the simulation mode for testing.*

**3. Dataset Preparation**

Ensure your dataset CSV file has the following columns (KDD Cup 99 format):

duration, protocol\_type, service, flag, src\_bytes, dst\_bytes, land, wrong\_fragment,

urgent, hot, num\_failed\_logins, logged\_in, num\_compromised, root\_shell, su\_attempted,

num\_root, num\_file\_creations, num\_shells, num\_access\_files, num\_outbound\_cmds,

is\_host\_login, is\_guest\_login, count, srv\_count, serror\_rate, srv\_serror\_rate,

rerror\_rate, srv\_rerror\_rate, same\_srv\_rate, diff\_srv\_rate, srv\_diff\_host\_rate,

dst\_host\_count, dst\_host\_srv\_count, dst\_host\_same\_srv\_rate, dst\_host\_diff\_srv\_rate,

dst\_host\_same\_src\_port\_rate, dst\_host\_srv\_diff\_host\_rate, dst\_host\_serror\_rate,

dst\_host\_srv\_serror\_rate, dst\_host\_rerror\_rate, dst\_host\_srv\_rerror\_rate, label

**Usage Instructions**

**Step 1: Train the Models – Provided the trained models**

Train all three models (CNN, LSTM, Transformer):

# Train CNN model

python main.py --data dataset.csv --model CNN --epochs 20

# Train LSTM model

python main.py --data dataset.csv --model LSTM --epochs 20

# Train Transformer model

python main.py --data dataset.csv --model Transformer --epochs 20

**What this does:**

* Preprocesses your dataset
* Trains the selected model
* Saves model weights (saved\_model\_\*.pt)
* Saves label encoder (label\_encoder\_\*.pkl)
* Generates XAI explanations in reports/ folder
* Displays training metrics and evaluation results

**Step 2: Launch the Web Interface**

streamlit run ui\_with\_wireshark.py

**For better packet capture permissions:**

# Windows (as Administrator)

streamlit run ui\_with\_wireshark.py

# Linux/Mac (with sudo)

sudo streamlit run ui\_with\_wireshark.py

**Step 3: Use the Web Interface**

The web interface provides 4 main tabs:

**Upload CSV Tab**

* Upload CSV datasets for batch prediction
* View data preview and prediction results
* Download results and visualizations

**Single Record Tab**

* Manual entry of network connection features
* Dropdown menus for categorical data (protocol\_type, service, flag)
* Real-time single prediction results

**Live Capture Tab**

* **Real Packet Capture**:
  + Select network interface (try "any" first)
  + Set capture duration
  + Analyze captured traffic in real-time
* **Simulation Mode** (recommended for testing):
  + Generate realistic network traffic data
  + Include both normal and attack patterns
  + Perfect for testing without real network access

**PCAP File Tab**

* Upload PCAP files from Wireshark
* Automatic packet processing and feature extraction
* Batch analysis of captured network sessions

**Quick Start Guide**

**Option A: Using Real Dataset**

1. Prepare your CSV dataset with network flow features
2. Train models: python main.py --data dataset.csv --model CNN
3. Launch UI: streamlit run ui\_with\_wireshark.py
4. Test with CSV upload or single record entry

**Option B: Testing with Simulated Data**

1. Train with your dataset (or use a sample dataset)
2. Launch UI: streamlit run ui\_with\_wireshark.py
3. Go to "Live Capture" → "Simulation Mode"
4. Generate 20-50 packets and analyze them
5. See real-time intrusion detection results!

**Model Training Details**

**Command Line Arguments:**

* --data: Path to your CSV dataset
* --model: Choose from CNN, LSTM, or Transformer
* --epochs: Number of training epochs (default: 10)

**Training Output:**

* **Model files**: saved\_model\_[MODEL].pt
* **Label encoders**: label\_encoder\_[MODEL].pkl
* **Evaluation metrics**: Accuracy, Precision, Recall, F1-Score
* **XAI reports**: LIME and SHAP explanations in reports/

**Network Interfaces Guide**

**Interface Selection:**

* **"any"**: Captures on all interfaces (recommended first try)
* **Windows**: Try "WiFi", "Ethernet", or specific adapter names
* **Linux**: Try "eth0", "wlan0", "ens33"
* **Mac**: Try "en0", "en1"

**Troubleshooting Packet Capture:**

1. **Permission Denied**: Run as administrator/root
2. **No Packets Captured**: Try different interface or use simulation mode
3. **Npcap Error**: Install Npcap from https://nmap.org/npcap/
4. **Testing**: Use simulation mode to generate test data

**Supported Attack Types**

The system can detect various network intrusions:

**DOS Attacks**: neptune, smurf, back, land, pod, teardrop **Probe Attacks**: ipsweep, portsweep, nmap, satan **R2L Attacks**: guess\_passwd, ftp\_write, imap, phf, multihop, warezmaster, warezclient, spy **U2R Attacks**: buffer\_overflow, loadmodule, rootkit

**Demo**

# 1. Train a model

python main.py --data nsl\_kdd\_cleaned.csv --model CNN --epochs 10

# 2. Launch the interface

streamlit run ui\_with\_wireshark.py

# 3. In the web interface:

# - Go to "Live Capture" tab

# - Click "Generate Traffic" (simulation mode)

# - Click "Analyze Captured Traffic"

# - See real-time intrusion detection results!