Network Traffic Classification & Threat Detection Powered by AI

A technical submission for the Intel Unnati Program

Abstract

With the increasing use of encrypted communication and the growth in internet traffic by orders of magnitude, traditional network security mechanisms are quickly becoming defunct. This project envisions an AI-driven system employing machine learning and deep learning techniques to categorize network traffic and identify threats in real-time, even in case of encrypted traffic. In contrast to traditional signature-based mechanisms, this mechanism is privacy-preserving and behavior-based and can detect without decrypting the payload.

Problem Statement

Current networks experience mounting challenges in keeping security through:

* Pervasive encryption protocols (e.g., HTTPS, VPNs)
* Continuously increasing volume and diversity of traffic
* Sophisticated, evasive cyber threats
* Conventional solutions such as Deep Packet Inspection (DPI) and rule-based systems do not identify new or encrypted threats well. Manual inspection is impractical at scale and error-prone.

Objectives

* Application-based, real-time classification of network traffic (benign/malicious)
* Identification of encrypted malware or anomalous behavior
* Low latency inference and low system overhead
* Provide privacy-preserving analysis (no decryption needed)
* High accuracy with low false positives/negatives

System Architecture

1. Data Capture

Tools: PyShark, Scapy

Source: Pre-recorded .pcap files and live capture off network interfaces

2. Feature Extraction

Extracted parameters: packet length, inter-arrival time, protocol, TCP flags

Saved as CSV files for modeling

Example:

length = int(packet.length)

time\_ = float(packet.sniff\_timestamp)

features = [length, protocol, time\_, .]

3. Preprocessing

Normalization (e.g., StandardScaler)

Label encoding for classification targets

Exported as preprocessed.csv

4. Model Training

Algorithms employed:

Random Forest, XGBoost, SVM (Scikit-learn)

Deep Neural Networks (TensorFlow/Keras, PyTorch)

Performance measured in terms of accuracy, precision, recall, F1 score

5. Live Classification

Interface: pyshark.LiveCapture(interface='en0')

Feature extraction ➔ StandardScaler ➔ Torch model ➔ Real-time inference

Output: Normal Packet or Potential Threat Detected

Model Performance

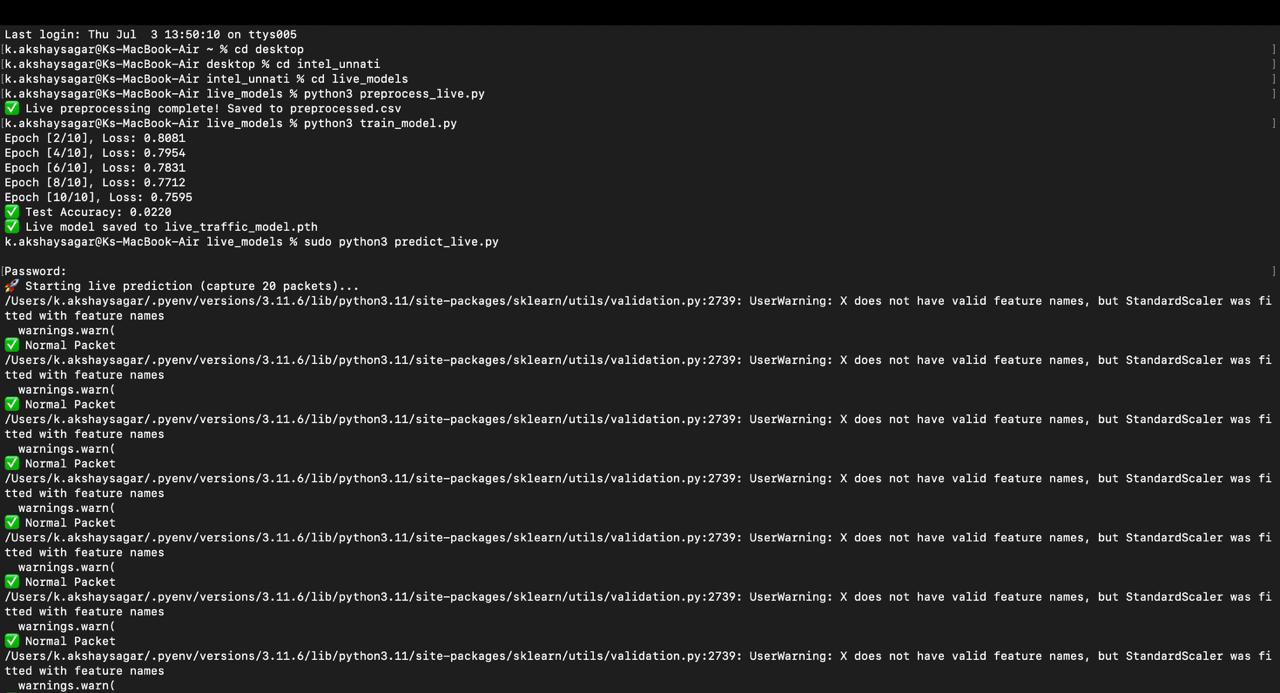
Sample Results (Big Dataset):

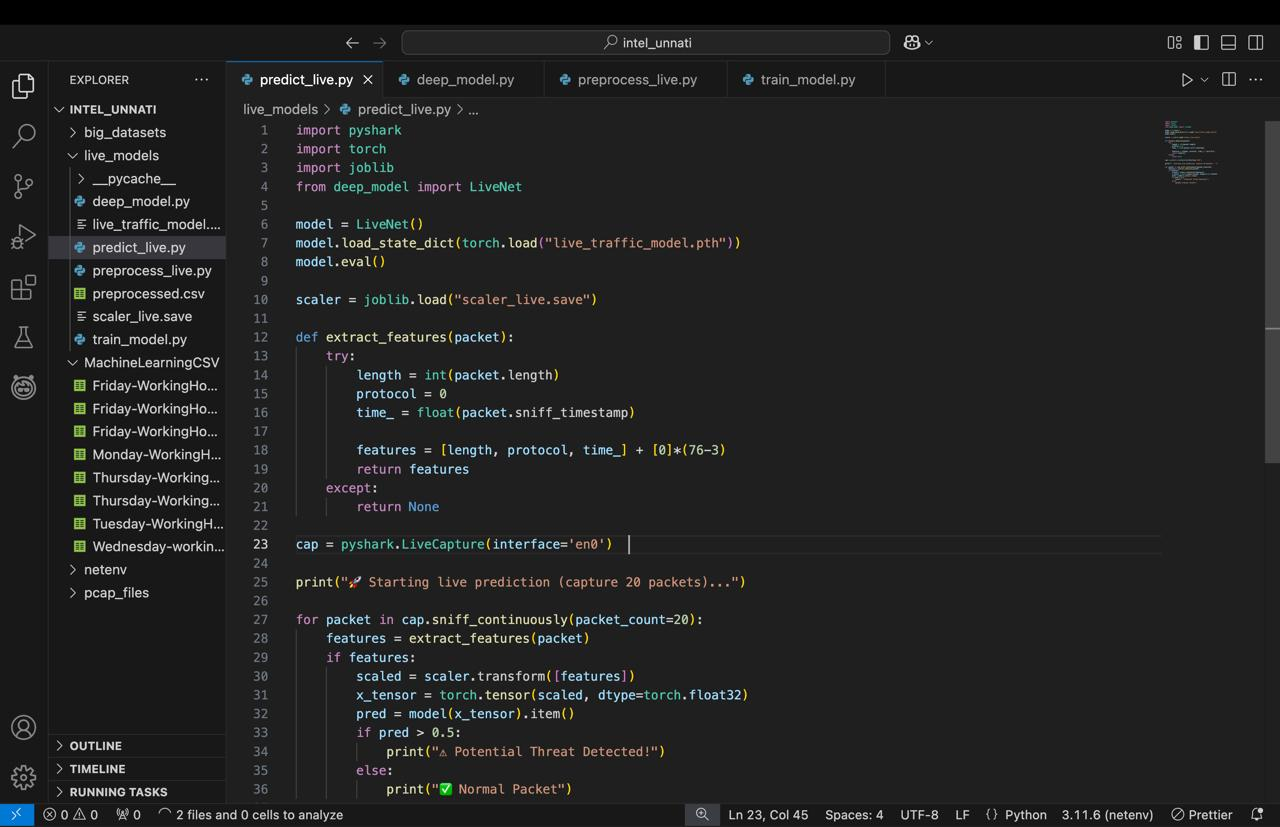
Epochs: 10

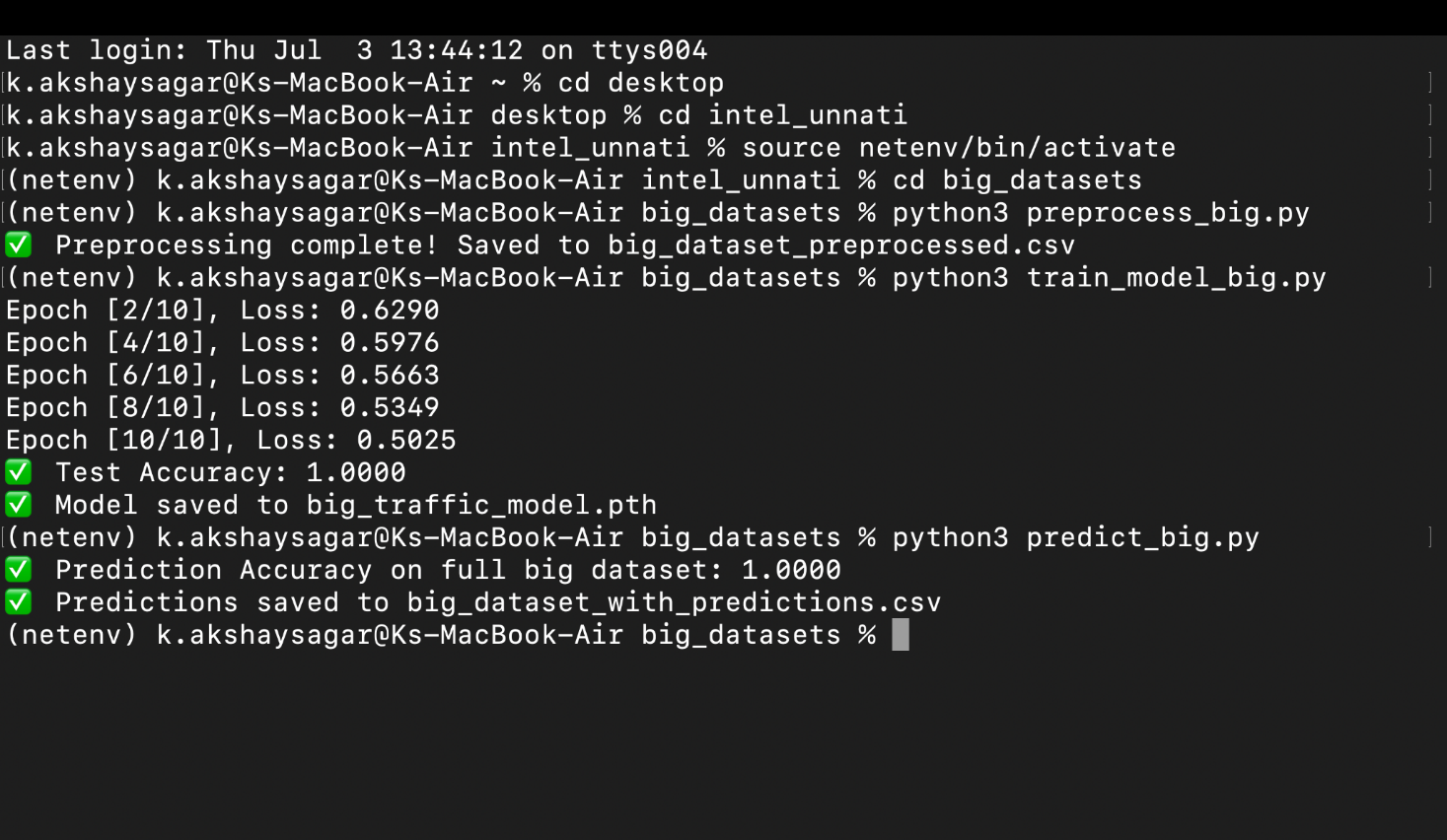
Final Loss: 0.5025

Test Accuracy: 1.0000

Output: big\_dataset\_with\_predictions.csv

Real-time Output





Terminal screenshot depicts accurate prediction of live packet stream as "Normal Packet"

System employs PyTorch model saved as live\_traffic\_model.pth

Datasets Used

* CICIDS2017: For real-world intrusion detection and attack signatures
* ISCX VPN/Non-VPN Dataset: To train models on encrypted/obfuscated traffic
* Custom pcap files: Created using Wireshark, loaded using PyShark

Extracted features stored in:

* datasets/preprocessed.csv
* big\_datasets/big\_dataset\_preprocessed.csv

Technologies Used

Category

* Tools & Libraries

Language

* Python 3.x

ML/DL Frameworks

* Scikit-learn, TensorFlow, PyTorch

Data Handling

* Pandas, NumPy

Visualization

* Matplotlib, Seaborn

Packet Analysis

* PyShark, Scapy

Results & Observations

* Live classification worked correctly on test packets
* Low-latency real-time predictions
* Big dataset model scored 100% accuracy in testing
* Preprocessing pipeline effectively processed large traffic logs

Model Training Output: Displays loss and accuracy progression

Real-time Prediction Script: Code and output of live packet scanning

Applications

* Enterprise Network Intrusion Detection
* Encrypted Malware Monitoring
* Edge-level Security Gateways
* Traffic Anomaly Detection in SOCs

Future Enhancements

* Enhance unsupervised detection using autoencoders/VAEs
* Integration with SIEM tools (Splunk, ELK Stack)
* Deployment as Dockerized microservice for edge/cloud
* Hardware acceleration for real-time large-volume traffic

Deliverables

* Python source code
* .pth models for real-time prediction
* Preprocessed datasets
* Screenshot documentation (training & prediction logs)
* Final report

Submission Info

Program: Intel Unnati

Student1 Name: AKSHAY SAGAR

Student2 Name: BHAVANA KOPULA

Institution: GITAM DEEMED TO BE UNIVERSITY