AI/ML-BASED CYBER THREAT DETECTION IN NETWORK TRAFFIC

Intel Unnati Industrial Training – AI/ML for Networking

Team Members:

1.Mohana Priya

2.Jahnavi

3.Inbaa

4.Sangeetha

5. Mowriya

Problem Statement

OBJECTIVE:
To build a machine learning pipeline that can detect malicious network activity using real-time traffic data.

KEY GOALS:

- Analyze labeled packet capture data (CICIDS 2017).
- Train an ML model to classify threats.
- Create a real-time prediction system for new network activity.

Solution Summary

- Developed a machine learning model to classify network threats in real time
- Built a Flask web application for interactive predictions
- Processes uploaded CSV files and predicts threat labels using a pre-trained model
- Displays the prediction results on a user-friendly web interface

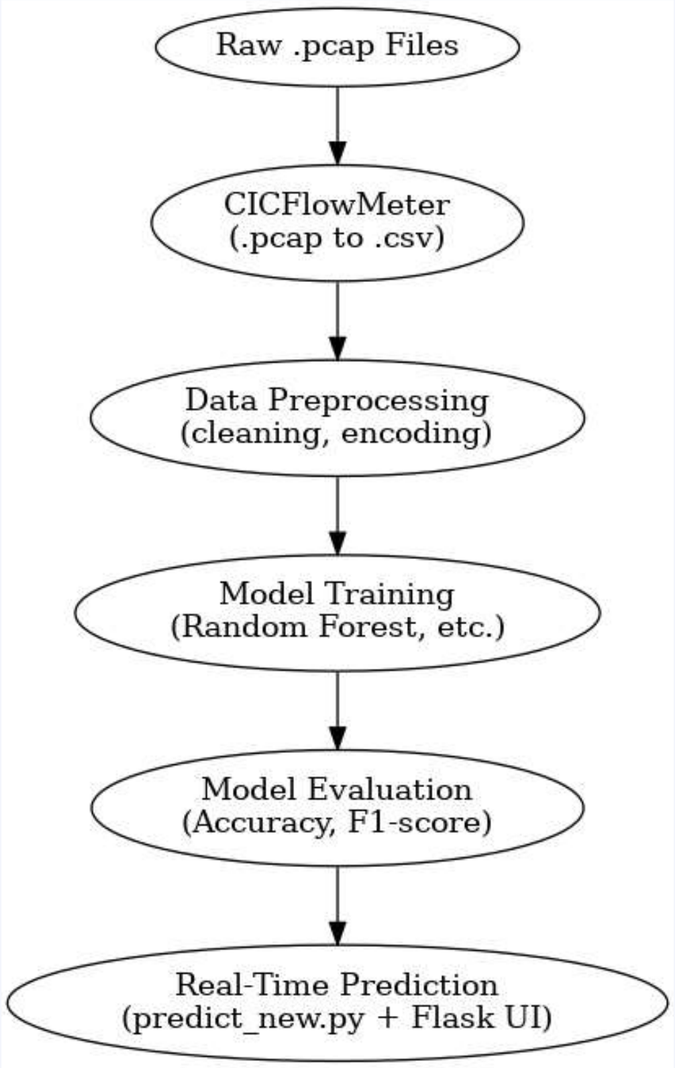
Architecture Flow

The architecture begins with raw network traffic (.pcap) files, which are converted into structured CSV format using CICFlowMeter. This data is then preprocessed, used to train a machine learning model, evaluated, and finally integrated into a real-time Flask-based prediction system.

Tools Used

- Python, Scikit-learn
- Flask (for real-time prediction UI)
- CICFlowMeter (for CSV conversion)
- Pandas, NumPy, Matplotlib

Flow Chart



Dataset Description

Dataset Used: CICIDS 2017

Source: Canadian Institute for Cybersecurity

Size: ~1.6 million records

Classes:

- Benign
- PortScan, DDoS, Bot, Web Attack, Infiltration, etc.

Features:

- Flow-based metrics like Flow Duration, Packet Length, Flow Bytes/s, etc.

Tools Used: CICFlowMeter to convert .pcap to .csv format

Issues & Fixes

S.No	Issue Faced	Fix Applied
1	Prediction shape error	Used .ravel()
2	Missing features	Column alignment
3	No result display	Defined decoded_labels
4	File not found	Corrected file path
5	Missing imports	Added libraries
6	File undefined	Fixed variable usage

Model Design & Features

Preprocessing:

- Removed null/NaN/infinite values
- Label Encoding of threat types
- Feature scaling using StandardScaler

Model Trained: Random Forest (best performance)

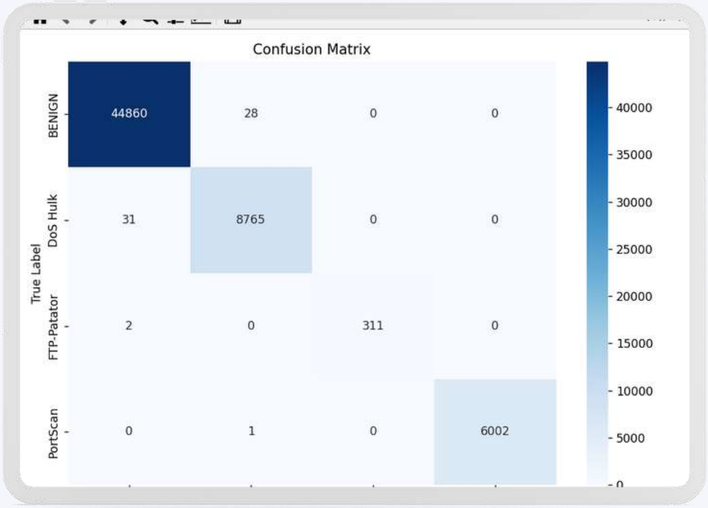
Others Tried: Logistic Regression, Decision Trees, Gradient Boosting

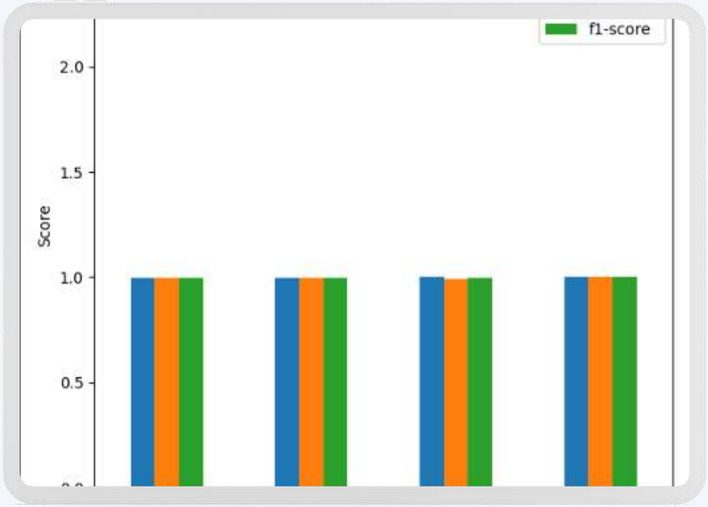
Selected Features:

- Flow Duration
- Fwd Packet Length Mean
- Flow Bytes/s
- Init_Win_bytes_forward
- and others…

Results & Visuals

- **Accuracy:** ~99.3%
- **Precision/Recall/F1:** High across classes
- **Confusion Matrix:** Shows strong classification ability





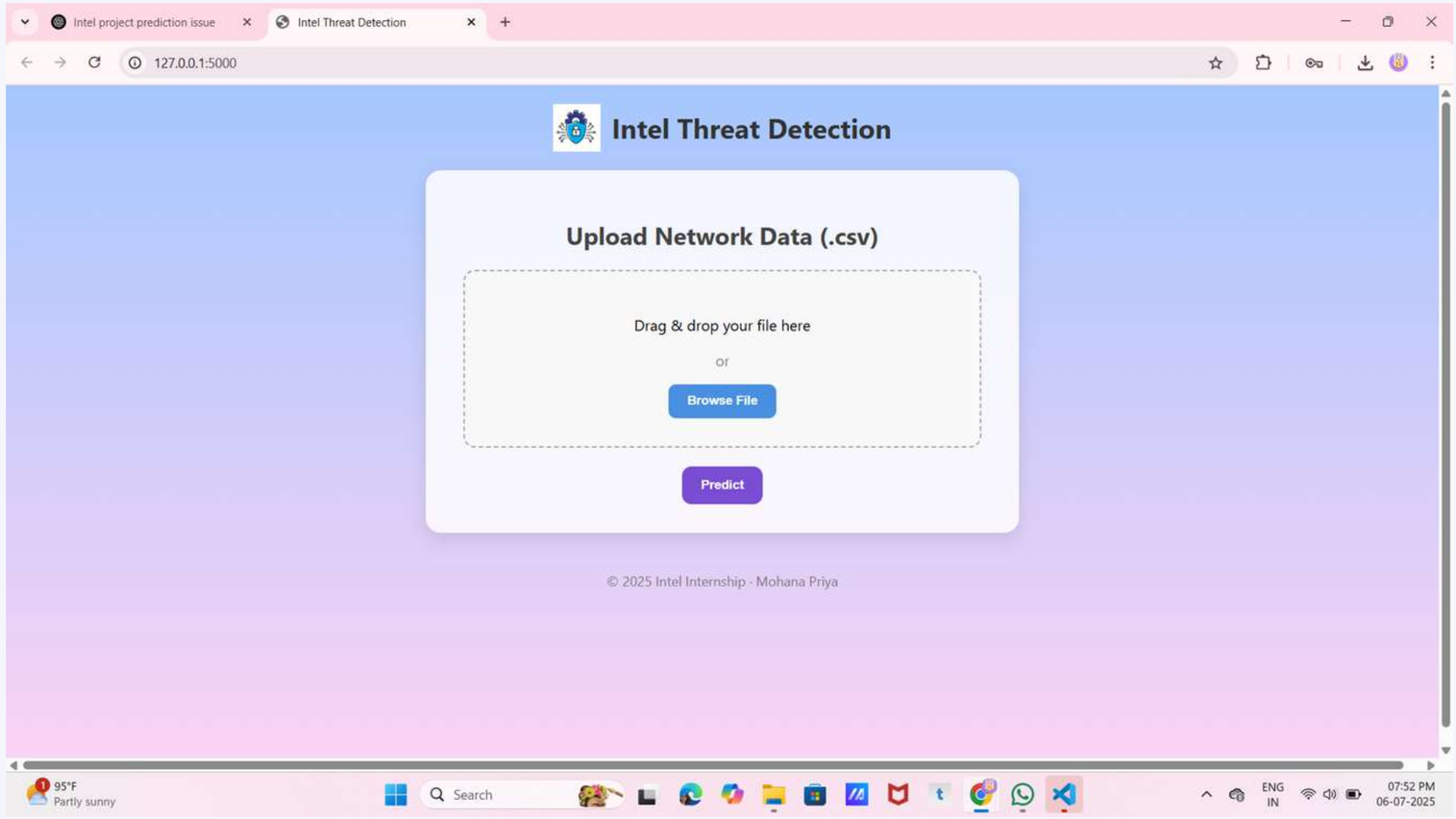
Real-Time Prediction Demo

Backend: predict_new.py script

- Loads trained model + encoder
- Accepts new flow data (.csv)
- Predicts threat type instantly

Frontend: Flask Web UI

- Upload new network file
- Get instant threat classification
- Simple and responsive design



Team Roles & Contributions

Name	Contribution
Mohana Priya	Real-time prediction module, Flask UI, model integration
Inbaa	Data cleaning, preprocessing, feature engineering
Mowriya	Model training, hyperparameter tuning
Jahnavi	Evaluation metrics, graph plotting
Sangeetha	Documentation, PPT

Learnings & Future Scope

Learnings:

- Applied ML to a real cybersecurity dataset
- Understood end-to-end ML pipeline
- Built and deployed a real-time Flask app

Future Scope:

- Deploy as background network monitor agent
- Use deep learning for more complex traffic
- Handle encrypted packets and evolving threats

THANK YOU!

COLLEGE NAME
J N N Institute of Engineering

PROJECT REPOSITORY
Link

SUBMITTED TO
Intel Unnati, July 2025