Project Title

AI-Driven Network Traffic Anomaly Detection

Author

Kaan Sulkalar

Date

October 24, 2025 (Europe/Istanbul)

1. Objective

Detect anomalies (e.g., DDoS, port scan, brute-force) from network traffic using ML.
Provide a reproducible pipeline: data → preprocessing → modeling → dashboard → reporting.

2. Dataset

- Recommended: UNSW-NB15 (labeled, modern traffic) - Alternatives: CICIDS2017, KDDCup99
Notes: Use CSV version; ensure 'label' column or map attack types into 0/1.

3. Methodology

• Preprocessing: missing-value imputation, label encoding for categoricals, scaling numerics. • Models: RandomForest (supervised baseline), IsolationForest (unsupervised), optional Autoencoder. • Split: stratified train/test; evaluate with Confusion Matrix, ROC-AUC, PR AUC, F1. • Real-time (optional): pyshark/scapy for PCAP → features → model scores.

4. Results Template

Insert after running on your dataset: - Accuracy: ____ - Precision: ____ - Recall: ____ - F1-Score: ____ - ROC-AUC: ____ - Confusion Matrix snapshot - Feature importance (top 10)

5. Dashboard

Streamlit app with tabs: EDA, Model, Predict. Upload CSV, pick label column, train, save artifacts, score new traffic. Visuals: Confusion Matrix, ROC/PR Curves, Feature Importances, Score Histograms.

6. Reproducibility

Environment: Python 3.11+, packages in requirements.txt. Steps: 1) pip install -r requirements.txt 2) streamlit run dashboard/app.py 3) Load CSV and follow tabs.

7. Limitations & Future Work

• Class imbalance; consider SMOTE or threshold tuning. • Concept drift in live networks → periodic retraining. • Try Autoencoder/One-Class SVM; deploy with Docker; integrate Azure Monitor.

8. References

UNSW-NB15, CICIDS2017 datasets; scikit-learn docs; Streamlit docs; Plotly docs.