Network Intrusion Detection System (NIDS) using Machine Learning

Name: Shaike Adila

College: HKBK College of Engineering

Submission Date: 26-06-2025

Abstract

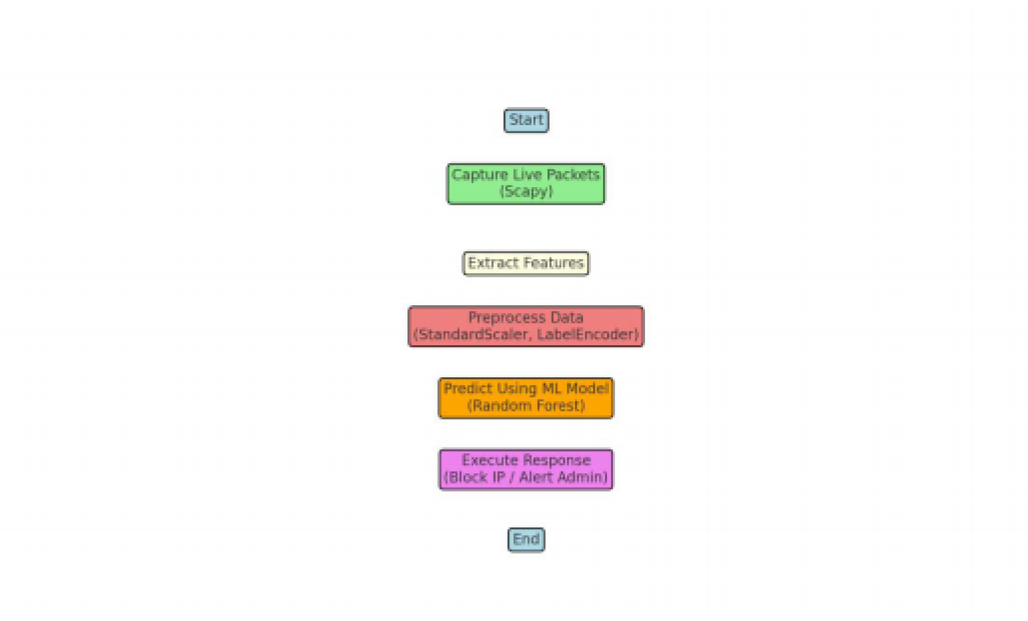
This project presents a Network Intrusion Detection System (NIDS) developed using Python and Machine Learning. The system uses real-time network packet monitoring and classification to detect potential threats such as DDoS, port scans, brute force attacks, and more. It includes components for data preprocessing, model training, real-time monitoring, and automated responses.

1. Introduction

In the modern digital era, securing network infrastructures is crucial. Intrusion Detection Systems (IDS) help in identifying unauthorized access or anomalies in network traffic. This project utilizes machine learning to train a model on labeled network data and monitor real-time packets to detect potential threats with high accuracy.

2. System Architecture

Below is the flowchart that explains how the system works:


Start
Capture Live Packets (Scapy)
Extract Features
Preprocess Data (StandardScaler, LabelEncoder)
Predict Using ML Model (Random Forest)
Execute Response (Block IP / Alert Admin)
End

The system is divided into modular components:
- Data Processing Module
- Machine Learning Models
- Network Monitoring
- Response System
- Visualization and Evaluation

Each module handles a specific task for intrusion detection and automated mitigation.

3. Module Descriptions

3.1 Data Processing

The DataProcessor class handles feature selection, encoding protocol types, and scaling numerical values. It ensures that both training data and live packets are processed in the same format.

3.2 Machine Learning Models

The IntrusionDetectionModels class contains three ML models: Random Forest, Isolation Forest, and Neural Network. During training, the system evaluates model performance and selects the best-performing supervised model.

3.3 Network Monitoring

This module captures real-time network packets using Scapy and extracts important features. Based on characteristics like payload size and flags, it attempts to label the type of network activity (normal, ddos, port scan, etc.).

3.4 Response System

When an attack is detected, the response system executes actions such as blocking IPs, alerting admins, or logging the activity. It supports multiple types of responses based on the attack category.

3.5 Visualization

After model training, confusion matrix and classification reports are generated to assess model performance. These visuals help in understanding precision, recall, and overall accuracy.

4. Model Training and Output

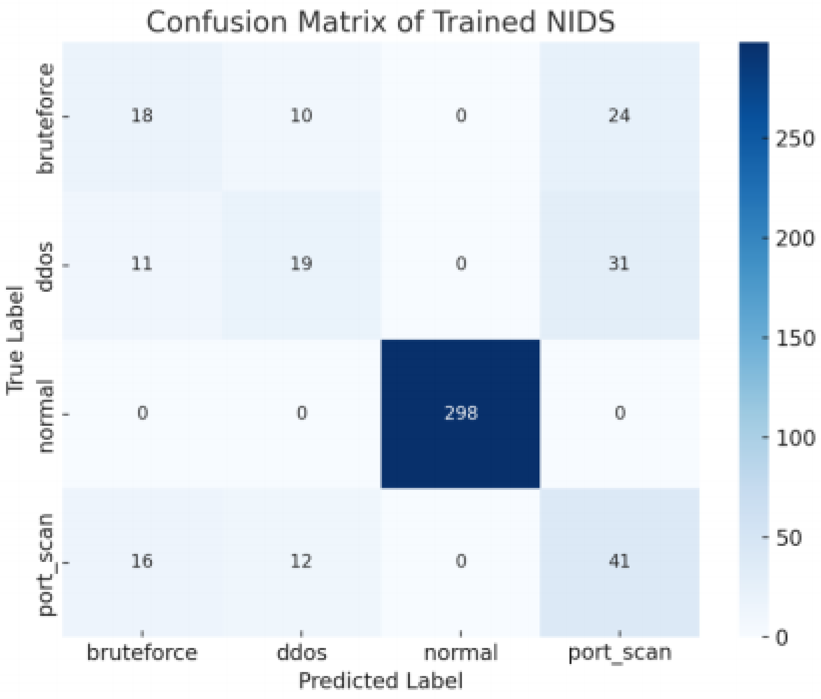
The system was trained on a demo dataset with labels for normal and attack traffic. During training:

- Random Forest achieved 100% accuracy
- Neural Network reached 93%
- Isolation Forest (unsupervised) was also trained

Random Forest was selected as the best model based on performance.

The evaluation on test data showed 72% overall accuracy with perfect detection of normal traffic. Detection of brute force, ddos, and port scans was fair but could be improved with more balanced data.

The confusion matrix of the model performance is shown below:



5. Real-Time Monitoring

The system captures live packets for 30 seconds and processes them in real time. If a threat is identified, it executes an appropriate response (e.g., blocking the IP for DDoS). The monitor runs in a separate thread to avoid blocking the main program.

6. Conclusion

This project demonstrates the successful implementation of a machine learning-based NIDS system using Python. It integrates data processing, model training, live monitoring, and automated responses in a modular format. While it performs well in simulation, real-world deployment would require deeper traffic analysis and continuous learning.

Advantages, Disadvantages, and Future Scope of Network Intrusion Detection System (NIDS)

1. Advantages of NIDS using Machine Learning - Real-time Monitoring: Detects attacks as they occur, allowing immediate action. **- High Accuracy**: Supervised models like Random Forest provide reliable classification results. **- Automated Response**: Automatically blocks IPs, alerts administrators, or logs events upon detection. **- Scalable Architecture**: Modular system allows integration with larger enterprise security frameworks. **- Adaptability**: Learns from evolving attack patterns through continuous training. **- Reduced Human Intervention**: Filters out common threats, freeing security personnel for critical analysis.

2. Disadvantages of NIDS using Machine Learning - Data Dependency: Requires large, well-labeled datasets for effective training. **- False Positives/Negatives**: Imbalanced or noisy data can reduce detection reliability. **- Resource Intensive**: Training and real-time analysis can consume significant computational power. **- Frequent Maintenance**: Models need to be retrained regularly to stay effective. **- Encrypted Traffic Challenges**: Cannot inspect or analyze payloads of encrypted traffic easily. **- Integration Complexity**: Combining real-time sniffing tools with ML can be challenging for beginners.

3. Future Scope of NIDS with Machine Learning - Deep Learning Integration: Use of CNNs and RNNs for better temporal and spatial attack detection. **- Federated Learning**: Enables collaborative training across multiple devices without sharing raw data. **- Self-Healing Systems**: Incorporation of AI systems that detect and autonomously mitigate threats. **- Hybrid Approaches**: Integration with Host-based Intrusion Detection Systems (HIDS) for comprehensive security. **- Real-Time Threat Intelligence**: Use of live blacklists and reputation data for dynamic decision-making. **- IoT and Cloud Compatibility**: Expanding NIDS to support diverse environments like smart devices and cloud infrastructure. **- Automated ML (AutoML)**: Leveraging AutoML to simplify model training and tuning, reducing human effort.