An Intelligent Intrusion Detection System for Network Security Using Machine Learning

Rajabboyeva Surayyo
PhD student of TUIT named
after Muhammad al-Khwarizmi
"information security" department
surayyobaxromqizi@gmail.com

Samarov Xusnutdin
associate professor of TUIT named
after Muhammad al-Khwarizmi
"information security" department
samarov07@gmail.com

Azizbek Xaitbayev Pirnazarovich
Assistant of TUIT named after
Muhammad al-Khwarizmi, Urganch
branch "information security" department
azizbekxaitbayev93@gmail.com

Allanazarov Asadbek Azatovich
Student of TATU named after
Muhammad al-Khwarizmi
"computer engineering" department
asadbekallanazarov974@gmail.com

Abstract: As cyber threats become increasingly sophisticated, the need for intelligent and adaptive security mechanisms has become more crucial than ever. Traditional intrusion detection systems (IDS) struggle to identify novel or evolving attacks due to their reliance on static rules and signatures. This research proposes a machine learning-based IDS designed to detect anomalous behavior in network traffic. Using supervised and unsupervised learning models such as Random Forest, Support Vector Machine (SVM), and Autoencoders, the system is trained and tested on benchmark datasets including NSL-KDD and CICIDS2017. The proposed IDS demonstrates high accuracy and robustness in identifying multiple types of attacks and offers a real-time detection interface for practical deployment.

Introduction

The increasing dependency on networked systems has led to a parallel increase in cyberattacks such as Denial of Service (DoS), probing, data theft, and botnet intrusions. Intrusion Detection Systems (IDS) are critical components in the defense against such threats. However, signature-based IDS solutions are often limited to previously known attacks. This paper explores the development of a network-based IDS using machine learning techniques capable of detecting both known and unknown threats by identifying anomalies in network behavior.

Background and Related Work

IDS are generally classified into:

- **Host-Based IDS (HIDS)** – monitors activities on individual devices.

A **Host-Based Intrusion Detection System (HIDS)** is a type of IDS that monitors and analyzes the internals of a computing system rather than network traffic. It is installed on individual hosts (e.g., servers, workstations) and provides visibility into activities occurring at the operating system and application level.

- **Network-Based IDS (NIDS)** – analyzes traffic across a network.

A **Network-Based Intrusion Detection System (NIDS)** is a security solution that monitors and analyzes network traffic in real time to detect malicious activities, unauthorized access, or policy violations. Unlike Host-Based IDS, NIDS operates at the network level and observes traffic flowing across a network segment.

Furthermore, IDS can be:

- **Signature-based** – detects known threats based on predefined patterns.

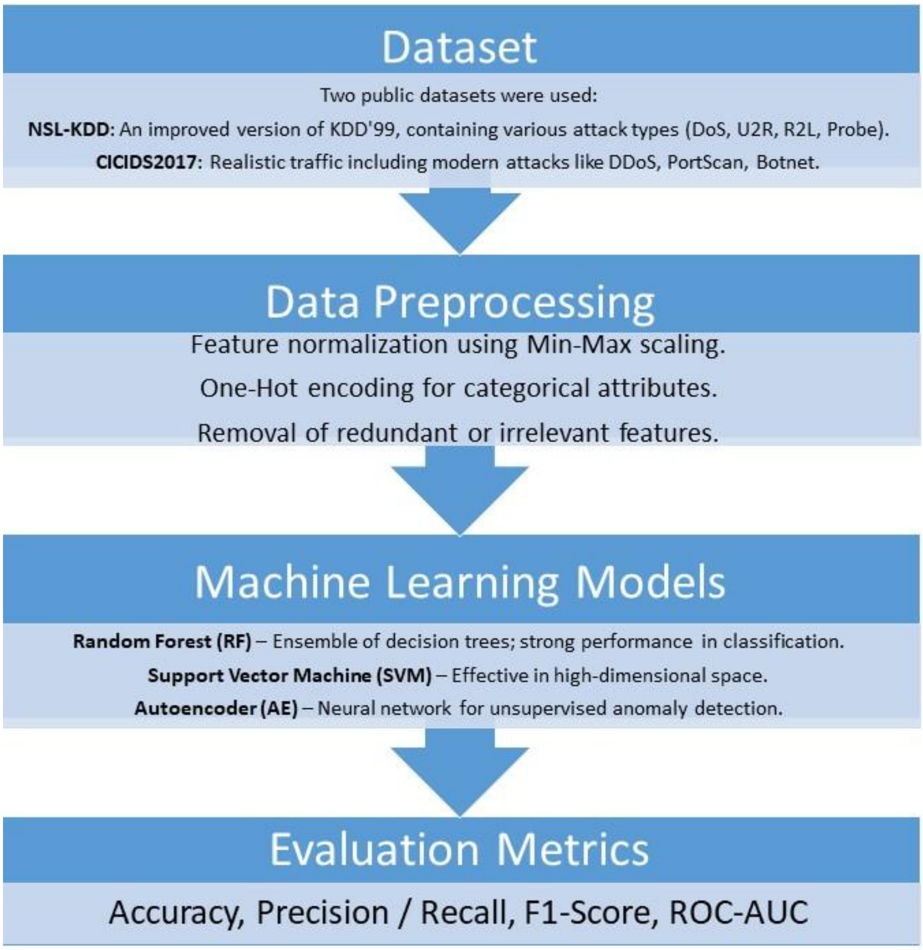
A **Signature-Based Intrusion Detection System (IDS)** is a type of security system that detects threats by looking for **known patterns of malicious activity**, called **signatures**. These signatures are like fingerprints of previously identified attacks—specific sequences of bytes, behaviors, or rule patterns.

- **Anomaly-based** – detects deviations from normal behavior.

An **Anomaly-Based Intrusion Detection System (AIDS)** identifies potential threats by **detecting deviations from the normal behavior** of a system, user, or network. Unlike signature-based IDS, which looks for known attack patterns, anomaly-based IDS focuses on what is unusual or unexpected.

Machine learning enhances anomaly-based IDS by learning patterns from data. Past studies have demonstrated the effectiveness of algorithms such as Random Forest, SVM, and deep learning models in improving detection rates and reducing false positives.

Methodology



Dataset

Two public datasets were used:

NSL-KDD: An improved version of KDD'99, containing various attack types (DoS, U2R, R2L, Probe).

CICIDS2017: Realistic traffic including modern attacks like DDoS, PortScan, Botnet.

Data Preprocessing

Feature normalization using Min-Max scaling.

One-Hot encoding for categorical attributes.

Removal of redundant or irrelevant features.

Machine Learning Models

Random Forest (RF) – Ensemble of decision trees; strong performance in classification.

Support Vector Machine (SVM) – Effective in high-dimensional space.

Autoencoder (AE) – Neural network for unsupervised anomaly detection.

Evaluation Metrics

Accuracy, Precision / Recall, F1-Score, ROC-AUC

Experimental Results

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	97.4%	96.8%	97.6%	97.2%
SVM	94.1%	92.5%	94.8%	93.6%
Autoencoder	91.2%	90.4%	89.6%	90.0%

Random Forest outperformed other models in terms of overall classification metrics. Autoencoders showed potential for detecting unknown anomalies, particularly in zero-day attack scenarios.

System Implementation

The IDS prototype consists of the following components:

- **Traffic Capturing**: Using pyshark or scapy to read pcap files or real-time traffic.
- **Model Inference Engine**: Trained models are serialized using joblib and used to classify live traffic.
- **User Interface**: A real-time dashboard is built using **Streamlit** to visualize traffic classification and trigger alerts for detected threats.

Conclusion and Future Work

This study demonstrates that machine learning can significantly enhance the effectiveness of intrusion detection systems. The proposed IDS achieved high accuracy across multiple types of attacks and was integrated into a lightweight, interactive interface. In future work, we plan to integrate deep learning techniques such as LSTM for sequence-aware traffic analysis and develop automated mitigation features turning the IDS into an Intrusion Prevention System (IPS).

References:

1. Tavallaee, M., et al. (2009). A detailed analysis of the KDD CUP 99 dataset. Computational Intelligence for Security.

2. Canadian Institute for Cybersecurity. CICIDS2017 Dataset. https://www.unb.ca/cic/datasets/ids-2017.html

3. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

4. Scikit-learn documentation. https://scikit-learn.org/

5. Keras Autoencoder Examples. https://keras.io/examples/autoencoder/