Intrusion Detection System Using Machine Learning for Real-Time Networks

[1]Muruganandham.L, [2]Malararasan.M, [3]Malararasi.M, [4]Madhan.P,[5]V.Hemalatha

[1,2,3,4]*N.S.N College of Engineering and Technology, Karur, India.*

[5]*HOD-CSE, N.S.N College of Engineering and Technology, Karur, India.*

[1]muruganandhamlakshmanan223@gmail.com [2]arasan0413@gmail.com

Abstract

With the rapid growth of network-based systems, ensuring security against cyber threats has become a critical challenge. Traditional intrusion detection systems often fail to detect new or evolving attack patterns in real time. To address this issue, this project presents a Machine Learning-based Intrusion Detection System (IDS) that leverages the Random Forest algorithm for the accurate and efficient detection of network attacks. The system is developed using the NSL-KDD dataset, from which essential features such as duration, protocol type, service, flag, source bytes, destination bytes, and traffic-based metrics are extracted. Categorical features are transformed into numerical values to make them suitable for machine learning processing.

The trained Random Forest classifier is capable of identifying multiple types of network traffic, including Normal, Denial of Service (DoS), Probe, and Unauthorized Access (R2L) attacks. For real-time intrusion monitoring, the system integrates Scapy to capture live network packets, extracts relevant features, and passes them to the trained model to predict traffic types. If an intrusion is detected, an alert message is instantly generated, displaying critical details such as attack type, source IP, destination IP, and timestamp. Furthermore, a user-friendly web interface provides functionalities to view training datasets, analyze attack records, perform manual detection, and monitor traffic via an analytics dashboard. This approach provides scalability, adaptability, and significant improvements in detection accuracy for modern network environments.

I. Introduction

A. Background

In today's digitally connected world, the security of network systems is of paramount importance. With the increasing dependence on the internet for communication, commerce, and critical infrastructure, networks have become prime targets for cyber-attacks. Threats such as Distributed Denial of Service (DDoS) attacks, malware, ransomware, phishing, and other forms of cyberintrusions cause significant disruptions, financial losses, and data breaches.

Traditional security mechanisms, such as firewalls and signature-based antivirus software, are no longer sufficient to address the sophisticated and evolving nature of these threats. Signature-based detection methods depend heavily on pre-defined attack signatures, making them incapable of identifying new or zero-day attacks. Anomaly-based systems often suffer from high false-positive rates, overwhelming security teams with inaccurate alerts.

B. Machine Learning in Cybersecurity

Predicting network attacks before they occur is a crucial step in enhancing network security. Machine learning (ML) has emerged as a powerful tool in this context, offering the capability to analyze vast amounts of data, detect patterns, and predict future events based on historical information. ML-based models can effectively learn from past attack patterns and behaviors, enabling them to identify anomalies and flag potential security threats in real time. This project utilizes Supervised Learning, where the algorithm is fed labeled data (input features paired with known outcomes) to predict and classify future network traffic.

C. Problem Statement & Motivation

The increasing dependence on real-time networks for critical applications has led to a surge in sophisticated cyber-attacks. The core problem addressed by this research is the development of an effective intrusion detection system using ML techniques to identify, classify, and mitigate both known and unknown cyber threats in real-time. The system aims to minimize false positives and false negatives by optimizing feature selection , while ensuring continuous monitoring of live traffic without affecting network performance.

II. Literature Survey

The application of machine learning to intrusion detection has been the subject of extensive academic research:

- Anomaly-Based IDS: Yang et al. (2022) conducted a comprehensive review of anomaly-based IDS, analyzing 119 research papers covering statistical, ML, and hybrid methods. Their study emphasized the critical role of preprocessing steps, feature selection, and normalization techniques on benchmark datasets like KDD99, NSL-KDD, and CICIDS2017.
- IoT Network Security: Rahman et al. (2025) and Kantharaju et al. (2023) investigated IDS designed specifically for the Internet of Things (IoT). These studies highlighted the unique challenges in
  IoT, including limited computational power and heterogeneous devices, and evaluated ML techniques for their suitability in securing IoT communication.
- Critical Infrastructure Protection: Pinto et al. (2023) surveyed MLbased IDS tailored to protect critical infrastructures (CI) such as power grids and industrial control networks. The authors examined challenges in deploying IDS in realworld CIs, including strict lowlatency requirements, high reliability, and robustness against adversarial attacks.
- General ML Techniques for IDS: Khraisat et al. (2019) and Hamid (2016) provided broad surveys evaluating supervised, unsupervised, and semi-supervised learning approaches. These studies consistently demonstrate the necessity of feature selection and data preprocessing to improve detection rates and reduce computational overhead.
- Deep Learning Advancements: Oliveira (2024) explored deep learning applications for IDS, reviewing architectures like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models for detecting anomalies.

III. Proposed System Architecture

A. Concept and Design Philosophy

The proposed system is designed to provide an intelligent, adaptive, and highly efficient security framework. The architecture follows a modular and layered design approach to ensure flexibility, scalability, and maintainability. It consists of interconnected components: data collection modules, preprocessing units, feature extraction layers, machine learning models, and alert systems. The design philosophy emphasizes the separation of concerns, allowing specific modules (such as the web interface or the prediction engine) to operate and update independently.

B. System Workflow

1. Data Acquisition: The workflow begins with the continuous collection of network traffic data, capturing packet-level and flowlevel information.
2. Data Preprocessing & Feature Extraction: Raw data is unstructured and contains noise, which is addressed through cleaning, normalization, and transformation. The system extracts key characteristics such as packet size, connection duration, and protocol type.
3. Machine Learning Engine: Features are fed into trained ML models (Random Forest) that distinguish between normal and malicious behavior.
4. Real-Time Prediction & Alerting: The system operates in real time, analyzing incoming data streams with minimal delay. If an intrusion is detected, the system activates the alert module to notify administrators and logs the event for future analysis.

IV. Hardware and Software Description

A. Core Software Stack

- Python: Serving as the foundational programming language, Python is utilized for its simplicity, powerful data handling capabilities, and extensive ML libraries.
- Pandas & NumPy: Pandas provides high-performance data structures (DataFrames) for cleaning, transforming, and aggregating structured network data. NumPy supports memory-efficient, multidimensional array processing necessary for mathematical computations.
- Scikit-Learn: This popular opensource ML library is used to implement the Random Forest classifier. It provides essential tools for model selection, feature scaling, and preprocessing (such as
  LabelEncoder).
- Scapy: An interactive packet manipulation program used in this project to sniff live network traffic, decode packets (TCP, UDP, ICMP), and extract the necessary real-time features.
- Flask & SQLite: Flask, a lightweight web framework, is used to develop the intuitive user interface. SQLite serves as the embedded, serverless database storing historical intrusion logs.

V. Project Description & Modules

The web-based control center consists of several integrated modules designed to facilitate system monitoring and analysis:

A. Dataset Viewers

1. View Training Dataset: This module provides access to the foundational NSL-KDD dataset used for building the model. It enables the visualization of 11 critical features, helping administrators understand data distribution and ensure consistency.
2. View Attack Dataset: Focusing specifically on malicious traffic, this module isolates records of DoS, Probe, and Unauthorized Access attacks. It aids in studying attack characteristics and validating the detection model's accuracy against purely malicious sets.

B. Live Operations

1. Start Real-Time Detection: As the core functionality, this module continuously monitors network traffic. It uses Scapy to process network packets in real-time, extracts relevant features, and leverages the ML model to classify traffic as "SAFE" or "ALERT".
2. Analytics Dashboard: A centralized interface providing visual representations of system performance via interactive charts. It displays metrics such as total monitored events, safe vs. malicious traffic distribution, and top attacking IP addresses.

C. Manual Testing & Logging

1. Manual Attack Detection: This module allows administrators to input custom packet parameters (e.g., protocol type, source bytes, destination bytes) manually. It evaluates the custom inputs against the trained model, serving as an excellent tool for hypothesis testing and investigating ambiguous traffic cases.
2. View Intrusion Logs: This module presents a comprehensive SQLitebacked record of all security events.
   Logs include timestamps, source/destination IPs, protocols, and the specific attack type predicted.

VI. System Implementation & Source Code Logic

A. Data Preprocessing and Model Training

The ML model is trained exclusively using the nsl_kdd.csv dataset. The system narrows down the 43 standard features to 10 highly relevant variables plus the target label: duration, protocol_type, service, flag, src_bytes, dst_bytes, count, srv_count, serror_rate, and same_srv_rate.

The target labels are mapped into four distinct classes to simplify detection:

- Normal: Normal traffic.
- DoS: 'neptune', 'smurf', 'pod', 'teardrop', 'land', etc.
- Probe: 'portsweep', 'ipsweep', 'nmap', 'satan', etc.
- Unauthorized Access: Acts as a catch-all for R2L and U2R attacks.

Categorical text columns (protocol_type, service, flag) are converted to numerical values using Scikit-Learn's LabelEncoder.
The system then initializes a RandomForestClassifier with 100 estimators (n_estimators=100), fits it to the dataset, and exports the serialized .pkl files (ids_model.pkl and encoders.pkl) for realtime deployment.

B. Real-Time Sniffing Engine

The live packet sniffer operates on a background thread utilizing Scapy. When an IP packet is intercepted:

1. Noise Filtering: Broadcast and loopback addresses (e.g., 255.255.255.255, 127.0.0.1) are ignored to optimize processing overhead.
2. Feature Extraction: The system determines the protocol (TCP, UDP, or ICMP) and infers the active service (e.g., HTTP for port 80/443, domain for port 53). Packet lengths map to src_bytes.
3. Encoding & Prediction: The extracted parameters are mapped to the 10 expected features, encoded using the pre-loaded encoders.pkl, and passed to the Random Forest model for prediction .
4. Database Commit: If the prediction yields anything other than "Normal", the status is flagged as "ALERT", and the details are immediately inserted into the logs table in the SQLite ids.db database .

VII. Conclusion and Future Enhancements

A. Conclusion

The intrusion detection system using machine learning for real-time network environments provides a highly efficient and intelligent solution for safeguarding modern digital infrastructures. By integrating the robust data analysis capabilities of Random Forest models with Scapy's real-time packet processing, the system successfully identifies known and unknown threats with remarkable speed. The system minimizes false alarms through optimized feature selection and provides a scalable, user-friendly interface for continuous network monitoring. By combining automated ML predictions with manual oversight capabilities, this approach represents a powerful, proactive advancement in addressing modern cybersecurity challenges.

B. Future Enhancements

The future scope of this project focuses on improving adaptability and scaling the architecture for next-generation threats:

- Deep Learning Integration:
  Incorporating architectures like Recurrent Neural Networks (RNNs) and transformer-based models will enable the system to capture complex temporal patterns across sequential network traffic.
- Big Data & Edge Computing: Integrating distributed stream processing frameworks will allow the system to handle massive enterprise-level data volumes. Furthermore, moving the detection nodes to edge computing environments will reduce latency and enable faster detection closer to IoT data sources.

- Federated Learning: Utilizing federated learning will allow the ML model to train collaboratively across multiple network nodes without sharing sensitive packet data, significantly enhancing privacy.
- Active Mitigation & Threat Intelligence: Future versions should evolve from passive detection to active prevention (Intrusion Prevention Systems), automatically blocking malicious IPs without human intervention while actively updating attack signatures via global threat intelligence feeds.

References

1. Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of Big Data*.
2. Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things. *Electronics (MDPI)*.
3. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*.
4. Jakotiya, K. S., Shirsath, V. S., & Mishra, R. G. (2023). Intrusion Detection System Using Deep Learning and Machine Learning: Review. *IEEE Conference*.
5. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*.
6. Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomalybased intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*.
7. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*.
8. Verma, A., & Ranga, V. (2021). Machine learning based intrusion detection systems: A comprehensive review. *Computer Science Review*.
9. Sharma, S., & Mukherjee, S. (2021). Intrusion detection using machine learning techniques in computer networks. *Internet of Things Journal*.
10. Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*.