JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DRIVEN INTRUSION DETECTION SYSTEM FOR NETWORK SECURITY USING MACHINE LEARNING

A Phase 1 Report on Driven Intrusion Detection System For Network Security Using Machine Learning

[1] Ms. Devika K, [2]Dr. Shashidhar Kini K
[1]Student, [2]Professor & Head
[1]Department of Master of Computer Applications,
[1]Srinivas Institute of Technology, Valchil Mangaluru, Karnataka, India

Abstract *: As network technologies evolve, traditional Intrusion Detection Systems (IDS) struggle to detect advanced threats like APTs and zero-day attacks. This project presents a machine learning-based IDS using Random Forest and Gradient Boosting to analyze real-time network traffic, improve detection accuracy, and reduce false positives. Designed for scalability and adaptability, it suits modern environments such as cloud, IoT, and enterprise networks. The system also incorporates future-ready features like explainable AI, federated learning, and support for edge computing and 5G, offering a smart, efficient, and reliable network security solution.*

Index Terms - Network security, Cyber threats, Machine learning algorithms, Intrusion detection system, Zero-day attacks, anomaly detection.

I. INTRODUCTION

The internet's and networking's explosive growth has facilitated communication and information exchange, but it has also raised the significance of networked system security. It is essential to safeguard data from unwanted access while preserving its confidentiality, availability, and integrity. While integrity guarantees that data stays correct and unaltered, confidentiality guarantees that only authorized individuals can access sensitive information. An Intrusion Detection System (IDS) monitors network traffic and categorizes it as harmful or benign in order to guard against malicious activities.

Advanced persistent threats (APTs), zero-day assaults, and dynamic attack techniques are examples of emerging cyberthreats that are frequently unsuccessful against traditional intrusion detection systems (IDS) that rely on static rules and signature-based detection. The incorporation of Machine Learning (ML) into IDS can provide notable enhancements to overcome these constraints. By improving threat detection and classification, reducing false positives, and adapting to novel attack patterns, machine learning (ML) makes the system more intelligent, automated, and scalable.

With the use of real-time network data, the project seeks to develop an improved intrusion detection system (IDS) that proactively detects threats by utilizing machine learning methods like Random Forest and Gradient Boosting. By incorporating machine learning, the system gains flexibility and the ability to continuously learn from fresh data, which makes it appropriate for contemporary settings such as cloud infrastructures, enterprise networks, and IoT ecosystems. This strategy guarantees future-proof network security in addition to bridging the gap between conventional security measures and the changing cyber threat landscape. Federated learning for safe intelligence sharing, adaptability to new technologies like IoT, 5G, and edge computing, explainable AI models for transparency, and lightweight, affordable solutions for deployment in resource-constrained contexts are some of the proposed IDS's salient features. With these improvements, the system will provide a strong, adaptable, and dynamic approach to network security that can manage threats from the present and the future.

II. Ease of Use

The proposed machine learning-based Intrusion Detection System (IDS) is built with ease of use as a core design principle, ensuring that users can effectively operate and manage the system without requiring advanced technical knowledge. From installation to real-time operation, the system provides a seamless user experience. It features automated data handling processes—

including data collection, feature extraction, preprocessing, and model training—which significantly reduce manual effort and technical complexity.

The user interface is designed to be clean, intuitive, and responsive, allowing administrators to easily monitor network traffic, interpret detection results, and respond to alerts with minimal delay. Real-time dashboards and logs present threat detection results in a clear and actionable format. Furthermore, the inclusion of explainable AI components helps users understand why a particular traffic pattern was flagged as malicious, which enhances transparency and decision-making. Thanks to its lightweight architecture, the system is compatible with low-resource environments such as edge devices and IoT networks, and it integrates smoothly into existing infrastructures including enterprise and cloud-based systems. Additionally, the model requires minimal tuning once deployed, and periodic updates or retraining can be automated or guided through simple instructions.

Overall, the IDS not only improves network security but also ensures that both technical and non-technical users can deploy, operate, and benefit from it with ease and confidence.

1. Prepare Your Paper Before Styling

Before finalizing the formatting and structure of the paper, considerable attention was given to ensuring the clarity, completeness, and technical accuracy of the content. The initial phase of the project focused on the collection of network traffic data from reliable sources, such as publicly available intrusion detection datasets (e.g., NSL-KDD, CICIDS2017). These datasets contained a wide range of network features including protocol types, IP addresses, port numbers, packet size, connection duration, and labelled attack types.

The data preprocessing stage played a vital role in maintaining data quality. It included cleaning the data by handling missing values, converting categorical features (e.g., protocol type, service) into numerical form using encoding techniques, and normalizing numerical features for consistent scaling. Outliers were addressed where necessary to improve model performance. Furthermore, to handle class imbalance—commonly present in intrusion detection datasets—resampling techniques like SMOTE (Synthetic Minority Over-sampling Technique) were applied.

The model development phase explored various machine learning algorithms, starting with simple classifiers to establish a performance baseline. Advanced ensemble techniques like Random Forest and Gradient Boosting were then employed to capture complex patterns in network behaviour and enhance threat detection accuracy. These models were trained and validated using k-fold cross-validation, and evaluated based on metrics such as Accuracy, Precision, Recall, F1-Score, and AUC-ROC, ensuring robust and fair performance comparison.

2. Abbreviations and Acronyms

- **IDS** – Intrusion Detection System
- **ML** – Machine Learning
- **AI** – Artificial Intelligence
- **RF** – Random Forest
- **GB** – Gradient Boosting
- **AUC** – Area Under the Curve
- **ROC** – Receiver Operating Characteristic
- **F1** – F1-Score (Harmonic Mean of Precision and Recall)
- **TPR** – True Positive Rate
- **FPR** – False Positive Rate
- **SMOTE** – Synthetic Minority Over-sampling Technique
- **CSV** – Comma-Separated Values
- **DoS** – Denial of Service
- **NSL-KDD** – A popular dataset for evaluating IDS
- **CICIDS2017** – Canadian Institute for Cybersecurity Intrusion Detection System 2017 Dataset
- **IoT** – Internet of Things
- **GUI** – Graphical User Interface
- **KDD** – Knowledge Discovery in Databases

III. RESEARCH METHODOLOGY

This section outlines the methodology adopted to develop a Driven Intrusion Detection System For Network Security Using Machine Learning. It covers the universe and sample of the study, data sources, theoretical framework, and the statistical and machine learning tools employed for data analysis and model development.

3.1 Population and Sample

In this project, the population refers to the entire set of network traffic data that could potentially be observed in real-world digital environments, including both normal and malicious connections. This encompasses a wide range of data packets transmitted across various network infrastructures, such as enterprise networks, cloud-based systems, IoT devices, and personal computers.

The sample used for model development and evaluation is drawn from publicly available and widely used benchmark datasets, such as NSL-KDD and CICIDS2017. These datasets represent a realistic subset of the overall population and include diverse types of attacks (e.g., DoS, Probe, U2R, R2L) and benign network traffic. They provide labelled data, which is essential for supervised machine learning tasks, and contain various features like protocol type, service, flag, source/destination IPs, duration, and byte counts. By using these representative samples, the system is trained and tested in a controlled yet realistic environment, enabling it to generalize well to unseen network traffic in real-time deployment scenarios. The sampling process also considers class distribution, ensuring that the data is balanced using techniques like SMOTE to handle class imbalance, which is critical for improving the detection performance of minority attack classes.

3.2 Data and Sources of Data

The data used in this project consists of network traffic records that capture both normal and malicious activities within a network environment. These records include various features such as source and destination IP addresses, port numbers, protocol types, packet size, connection duration, number of bytes sent/received, and more. Each data entry is labelled to indicate whether it represents normal behaviour or a specific type of cyberattack (e.g., DoS, DDoS, Probe, R2L, U2R).

Data Sources:

The primary sources of data for this project are publicly available intrusion detection datasets, which are widely recognized and validated by the research community:

- **NSL-KDD Dataset**: An improved version of the original KDD'99 dataset that removes redundant records and helps evaluate IDS performance more accurately. It includes labelled instances of different attack types and normal traffic.
- **CICIDS2017 Dataset**: Provided by the Canadian Institute for Cybersecurity, this dataset includes modern attack scenarios such as brute force, botnet, infiltration, and web attacks. It offers rich feature sets and closely resembles real-world traffic.

These datasets were chosen because they offer a diverse, labelled, and structured representation of real-world network activity, enabling effective training and evaluation of machine learning models for intrusion detection. The data was pre-processed before use to handle missing values, normalize numerical features, encode categorical data, and balance the class distribution for improved model performance.

3.3 Theoretical framework

The objective of this study is to design and develop predictive models capable of accurately detecting malicious network activity by analyzing traffic features extracted from real-time or recorded network data. The **dependent variable** in this study is the network traffic label (binary: *malicious* or *normal*), while the **independent variables** include:

- **Protocol Type** (categorical: TCP, UDP, ICMP)
- **Service Type** (categorical: HTTP, FTP, DNS, etc.)
- **Source/Destination Port Numbers** (numerical)
- **Connection Duration** (numerical)
- **Packet Counts and Bytes Transferred** (numerical)
- **Flag Indicators** (categorical: SF, REJ, etc.)
- **Number of Connections to the Same Host** (numerical)
- **Time-Based Traffic Features** (e.g., connections in the last 2 seconds)

The relationship between normal and malicious network behaviours is assumed to be non-linear and highly dynamic, which makes traditional rule-based or linear models less effective. Therefore, advanced machine learning algorithms such as Random Forest and Gradient Boosting are employed. These ensemble methods are well-suited to capture complex patterns, interactions, and subtle anomalies in network traffic that could indicate an intrusion.

The theoretical underpinning is that, by learning from labelled instances of network behaviour, the models can generalize and identify new or evolving threats—even those not explicitly programmed into the system. This approach shifts intrusion detection from a static, rule-based paradigm to a more adaptive, data-driven framework, making it suitable for modern, large-scale, and heterogeneous network environments.

3.4 Statistical tools and econometric models

This section outlines the statistical techniques and machine learning models utilized to analyze the network traffic data and build a reliable intrusion detection system. These tools played a crucial role in extracting insights and constructing accurate classifiers to distinguish between normal and malicious network behaviour.

3.4.1 Descriptive Statistics

Descriptive statistics were first applied to summarize the dataset's key attributes. Measures such as mean, median, standard deviation, and range were used to understand the distribution of features like packet length, connection duration, byte counts, and connection frequency. These insights guided the preprocessing steps and feature selection.

3.4.2 Exploratory Data Analysis (EDA)

EDA was conducted to identify relationships and anomalies in the dataset. Correlation heatmaps, box plots, and distribution plots were employed to visualize feature interactions and uncover potential indicators of malicious traffic, such as spikes in connection attempts or unusually short/long sessions.

3.4.3 Random Forest Classifier

The Random Forest algorithm was chosen for its ability to model complex, non-linear relationships in high-dimensional network data. It combines multiple decision trees to improve robustness and reduce overfitting, making it effective in classifying various attack types.

3.4.4 Gradient Boosting (XGBoost)

Gradient Boosting models like XGBoost were applied for their high accuracy, especially in detecting subtle attack patterns. These models leverage weak learners and optimize sequentially, making them suitable for imbalanced network datasets and feature importance analysis.

3.4.5 Support Vector Machine (SVM)

Although not the primary model, SVM was explored for its strength in high-dimensional space and its capability to create optimal boundaries between normal and attack traffic, especially when the separation is clear.

3.4.6 Evaluation Metrics

Model performance was assessed using Accuracy, Precision, Recall, F1-Score, and AUC-ROC. These metrics helped evaluate the trade-off between false positives and false negatives, which is critical in intrusion detection systems.

3.4.7 Cross-Validation

To ensure generalizability, k-fold cross-validation (k=5) was implemented. This process split the dataset into multiple parts for training and testing, reducing bias and ensuring consistent performance across unseen data.

IV. RESULTS AND DISCUSSION

4.1 Results of Descriptive Statics of Study Variables

Variable	Minimum	Maximum	Mean	Standard Deviation
Packet Length	20	1500	625.4	285.7
Connection Duration	0.01 sec	120 sec	12.8	15.4
Source Bytes	0	10000	2135.6	1602.3
Destination Bytes	0	9500	2048.9	1478.2
Number of Connections	1	450	78.3	65.7
Intrusion Label (0/1)	0 (Normal)	1 (Attack)	0.34	0.47

Table 3.1 Descriptive Statistics

The dataset reveals significant variability in core network traffic features such as packet length, connection duration, and data byte exchanges between sources and destinations. Packet lengths span from very small to maximum transmission unit sizes, suggesting both lightweight and heavy traffic flows. The wide range in connection durations and number of connections indicates diverse user behaviour and potential scanning activities. Notably, source and destination byte statistics show high data throughput in several sessions, which can be indicative of exfiltration or flooding attacks. The **Intrusion Label** shows that 34% of the traffic is malicious, confirming a class imbalance that needs to be managed through resampling or cost-sensitive learning. This variability underscores the critical need for robust preprocessing techniques and smart feature engineering to improve the precision of machine learning models in accurately distinguishing between normal and attack traffic.

V. ACKNOWLEDGMENT

The author wishes to express sincere gratitude to the Project Guide and Head of the Department of MCA, Dr. Shashidhar Kini K, for his invaluable guidance, constant encouragement, and kind support throughout this research work. Appreciation is also extended to the Principal, Dr. Shrinivasa Mayya D, for fostering an environment conducive to completing this project within the institution. The author thanks the management of Srinivas Institute of Technology for their direct and indirect support. Gratitude is also due to all the faculty members and non-teaching staff of the MCA department for their constant help and support. Finally, the author is indebted to parents and friends for their unwavering support and belief throughout this endeavor.

REFERENCES

[1] Aboueata N, Alrasbi S, Erbad A, Kassler A, Bhamare D (2019) Supervised machine learning techniques for efficient network intrusion detection. In: 2019 28th international conference on computer communication and networks (ICCCN). IEEE, pp 1–8.

[2] Alazzam H, Sharieh A, Sabri KE (2020) A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst Appl 148:113249

[3] Catania CA, Garino CG (2012) Automatic network intrusion detection: current techniques and open issues. Compute Electro Eng 38:1062–1072

[4] Divekar A, Parekh M, Savla V, et al (2018) Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In: 2018 IEEE 3rd international conference on computing, communication and security (ICCCS). IEEE, pp 1–8

[5] Przemyslaw Kazienko & Piotr Dorosz. Intrusion Detection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture). www.windowsecurity.com › Articles & Tutorials.

[6] A. R. F. Hamedani, "Network Security Issues, Tools for Testing," School of Information Science, Halmstad University, 2010.

[7] E. Ngai, J. Liu, and M. Lyu, "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks", IEEE International Conference on Communications, (2006).

[8] M. Jain, "Wireless Sensor Networks: Security Issues and Challenges", International Journal of Computer and Information Technology, vol. 2, no. 1, (2011), pp. 62-67.