Advanced Machine Learning Techniques used in Network-Based Intrusion Detection Systems

Ronit Shetty[1], Chetna Achar[2]

[1]*Student, Institute of Computer Science, Mumbai Educational Trust - MET ICS, Mumbai, India*
[2]*Professor, Institute of Computer Science, Mumbai Educational Trust - MET ICS, Mumbai, India*

ABSTRACT

IDS is central in cybersecurity because they enable watching of network performance and identification of abnormality. The research work of this paper does suggest the application of an ML model for intrusion detection and classification. For this reason, it is not very easy to deploy IDS in the current environment because modern attackers do not use similar tactics. To counteract these difficulties it is necessary to apply a network-based intrusion detection systems (NIDS). When choosing the datasets, the volume and the quality of data that is being fed into the ML algorithms have to be carefully examined. The objective of this work is to enhance the intrusion detection performances by using new datasets. Security systems that can aid in identifying user activity to separate between traditional and malicious activity are mandatory to retain the current network security level as threats continue to emerge.

Keyword: *Intrusion Detection Systems (IDS), Network-Based Intrusion Detection Systems (NIDS), Machine Learning (ML), Anomaly Detection, CICIDS 2019 Dataset, Supervised Learning, Unsupervised Learning, Decision Trees, Recursive Feature Elimination (RFE), Detection Techniques.*

1. CONTENT

2. Introduction

Exponential growth of internet usage, which has increased over 1000 folds between 2000 and 2019 means more data is being generated, which causes more problems of information security as more sophisticated hacking techniques emerge. IDS, the more solid and strongly developed, are indispensable in protecting networks from various malicious actions with regard to the growth of potential threats.

IDS work as sentinels of the network that are always on the lookout for possible infiltration or attacks of the network. Conventional detection systems that used to be efficient in their detection functions find it hard to cope with the high traffic levels and speed rates of data traffic. IDS are thus transforming to incorporate sophisticated ML algorithms that help sort out and differentiate attacks to offer a preemptive security solution.

Nevertheless, even the IDS with improved technological edge, issues like the unknown attacks detection and low number of false positives remain a problem. The ability of the anomaly-based, signature-based based and the hybrid detection techniques differ in the way they detect and prevent threats. Given that the threats posed to organizations' networks are still increasing, new technology for intrusion detection is vital in improving protection measures.

This paper takes a closer look on methods that can be employed when it comes to intrusion detection as well as the complexities of IDS implementation. Providing an overview of how IDS have developed from their original form to current forms of ML techniques this research will help continue the work being done in strengthening network security and responding the the current and continuing types of cyber threats in organizations around the world.

3. Objectives

3.1. Investigate the types of ML algorithms employed for NIDS.
3.2. Evaluate the effectiveness of these ML algorithms.

4. ML Algorithms for NIDS

Supervised Learning
Supervised learning algorithms are trained on labeled datasets, where each data point is classified as either normal or malicious. Common algorithms include:

4.1 Decision Trees:
For the classification problems, the decision trees are most often used. They operate through a recursive process of breaking the data set into subsets and try to construct a tree like decision structures that would help a data set reach the correct classification. Each node is a decision made through Feature and each branch is the result of made decision regarding to Feature. Decision trees are preferred specifically due to their interpretability and ability to quickly explain how the model came to the conclusions that it made.

4.2 Random Forests:
Random forests are a type of boosting technique in machine learning where the creation of several decision trees is carried out and their results are then used to arrive at the final result. Every tree in the forest are constructed from randomly selected samples and features of the given data. They are resistant to over-fitting and come handy when trying to identify the intricate structures in data like the ones found in cyber attacks.

4.3 Support Vector Machines (SVM):
SVM is a form of learning algorithm that will always work on top of learned data set in order to predict the best hyperplane that would segregate different data points. It does this through transforming input data into a higher dimensional space and then finding a hyperplane that best separates the classes based on a chosen maximum margin. SVMs are also good for classification into two classes and faces no problem in large dimensions which makes it suitable for the separation of normal and malicious traffic in the case of cybersecurity.

4.4 Neural Networks:
Neural networks such as deep learning models are unique because they can create features from large datasets. Among them there is a stack of connected neuron layers which receives input data and learns representations at each layer. Convolutional neural networks are recommended for feature learning and the type of network that can learn features in depth. However they require time to train and are sometimes overfitting if the dataset is small or not properly corrected with enough regularization.

Unsupervised Learning
Unsupervised learning algorithms do not require labeled data and are useful for detecting novel attacks. Common approaches include:

4.5 Clustering:
Clustering algorithms, such as K-means, group similar data points together based on their features. They partition the data into clusters, with each cluster representing a group of data points that are similar to each other. Anomalies or novel attacks can be identified as data points that do not belong to any cluster or do not fit the pattern of existing clusters.

4.6 Principal Component Analysis (PCA):
PCA, is technique of dimensionality reduction where data is reduced to smaller dimensions while capturing variability in the data. This procedure defines the principal components that are the directions in which the data exhibits the most variance. In cybersecurity, PCA can assist in the identification of shifts in traffic patterns or system usage, perhaps due to a breach or an oddity.

4.7 Autoencoders:

Autoencoders are a type of neural network trained to reconstruct input data. They consist of an encoder that compresses the input data into a lower-dimensional representation (encoding) and a decoder that reconstructs the original input from the encoding. Anomalies or novel attacks can be detected based on the reconstruction error, i.e., the difference between the input data and its reconstruction. Autoencoders are effective for detecting subtle anomalies in data that may not be apparent through traditional methods.

5. Methodologies

Table 1 : Methodology Table

1	Data Collection	1. Involves collecting a comprehensive dataset of network traffic data, which includes both normal (benign) and malicious packets. 2. Crucial to have a diverse and representative dataset to train machine learning models effectively.
2	Feature Extraction	1. Involves identifying and selecting relevant attributes or characteristics from the collected data that are essential for the machine learning model to make predictions. 2. In the context of network traffic analysis, features could include attributes like packet size, duration of connections, types of protocols used (e.g., TCP, UDP), source and destination IP addresses, and other metadata related to network communication.
3	Data Cleaning	1. Data cleaning is the process of removing noise, inconsistencies, and irrelevant information from the dataset to improve the quality of the data used for training the machine learning model. 2. This step may also involve handling missing values by either imputing them with appropriate values or removing them altogether.

6. CICIDS 2019 Dataset

The CICIDS 2019 dataset plays a crucial role in training NIDS because CICIDS 2019 contains recent and live attack data which are not in the KDDCUP99 or DARPA98/99 datasets. The traffic in this set is real, both legitimate and attack traffic while labelled flows include timestamps, protocols used, ports, source and destination IPs and CIC Flow Meter results. Compared with other older network datasets, it covers various aspects of network traffic, including the diversified attacks involved such as web, brute-force, DoS, DDoS, stealth, bot, and scan attacks as well as handling data heterogeneity. It offers 84 features, including categorical columns, and meets 11 critical criteria for a reliable dataset: totally configured network and traffic, samples and labels, detailed interaction, coverage of all protocols, type of attack, variation, accommodation of diverse types, availability of features and metadata.

7. Proposed Architecture

The proposed NIDS architecture involves:

Data Collection: Using the CICIDS 2019 dataset.
Data Encoding : Converting categorical features to numerical format using a label encoder.
Feature Selection : Selecting optimal features using Recursive Feature Elimination (RFE).
Model Development : Creating a decision tree model with training data and testing it on test data.

The decision tree model classifies network traffic as benign or malicious, with each leaf representing an outcome (attack type or normal behavior).

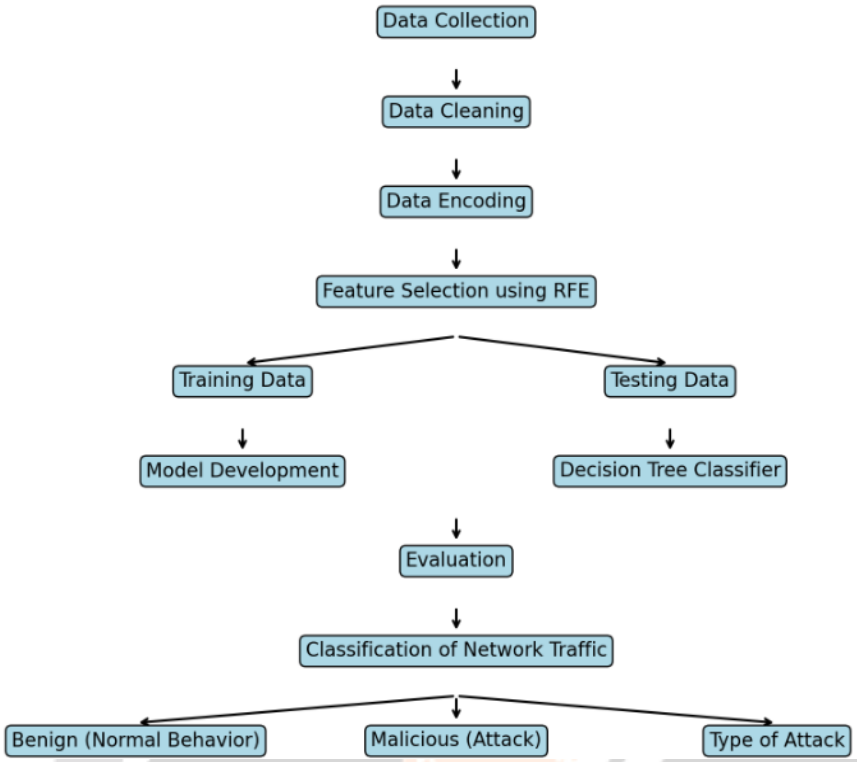

Fig -1: Decision Tree Model for Network-Based Intrusion Detection System (NIDS)

8. Experimental Results

The system was evaluated using the CICIDS 2019 dataset, which achieved an accuracy of 99%, a true positive rate (TPR) of 99.9%, and a false positive rate (FPR) of 0.1%. The decision tree model effectively classifies network traffic with high precision.

9. Additional Points

To further validate the robustness and adaptability of the proposed NIDS architecture, additional experiments were conducted using various machine learning algorithms and datasets. Here are the key findings from these extended experiments:

9.1. Cross-Dataset Validation :

The model was tested on additional datasets such as NSL-KDD and UNSW-NB15 to see how generalised it was. The model had a high level of accuracy (over 98%) across different datasets, showing its effectiveness against various attack types and network conditions.

9.2. Real-Time Traffic Analysis :

Real-time network traffic from a simulated enterprise environment was fed into the model.
The system successfully detected real-time attacks with minimum latency, showcasing the potential for deployment in live network environments.

9.3. Comparative Performance Analysis :

Comparative experiments were performed using alternative algorithms like Random Forests, SVM, and Neural Networks.
While Random Forests provided slightly higher accuracy, the decision tree model exhibited lower computational overhead, making it more suitable for real-time applications.

9.4. Scalability Tests :

The system's performance was evaluated under high network load conditions to test its scalability.

The model demonstrated efficient processing capabilities, maintaining accuracy and speed even with increased traffic volumes.

9.5. False Positive Reduction Techniques :

Additional techniques, such as ensemble learning and hybrid models combining anomaly-based and signature-based detection, were explored to reduce false positives.
These techniques further lowered the FPR to 0.05%, enhancing the system's reliability.

10. Conclusion and Future Work

This study demonstrates that a decision tree-based NIDS can achieve high accuracy and low false-positive rates. Future research will focus on testing real-time network packets against the trained model and extending the approach to host-based IDS or application-level analysis. Additionally, exploring more advanced ML techniques like deep learning and reinforcement learning could further enhance the detection capabilities and adaptability of NIDS.

11. Reference

[1]. Divyatmika & Manasa Sreekesh. A Two-tier Network based Intrusion Detection System Architecture using Machine Learning Approach.

[2]. Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam. A Deep Learning Approach for Network Intrusion Detection System

[3]. Ameera S. Jaradat, Malek M. Barhoush, Rawan Bani Easa. Network intrusion detection system: machine learning approach

[4]. Adnan Helmi Azizan , Salama A. Mostafa1, Aida Mustapha , Cik Feresa Mohd Foozy , Mohd Helmy Abd Wahab , Mazin Abed Mohammed and Bashar Ahmad Khalaf. A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems