A Comprehensive Study on the influence of Machine Learning Algorithms on Intrusion Detection System

1st Vinu Varshith Alagappan
Department of Mathematics
University of Padua
Padua, Italy
vinuvarshith.alagappan@studenti.unipd.it

2nd Anne Linda Antony Sahayam
Department of Information Engineering
University of Padua
Padua, Italy
annelinda.antonysahayam@studenti.unipd.it

***Abstract*—With the growing number of cyber-attacks in recent years in this contemporary world, Intrusion Detection Systems (IDS) have become a crucial component for ensuring the security of computer networks. IDS are designed to detect and prevent unauthorized access to computer systems by identifying potential intruders and raising alerts when suspicious activities are detected. Machine Learning (ML) algorithms have been applied to IDS to enhance their performance and accuracy in detecting and preventing attacks. This comprehensive study explores the impact of ML algorithms on IDS, including their strengths, limitations, and the challenges that need to be addressed to improve their performance. Overall, this study provides a comprehensive analysis of the influence of ML algorithms on IDS and offers insights into how they can be optimized to enhance the security of computer networks.**

***Index Terms*—Machine learning; Intrusion detection; Cybersecurity; Clustering; Precision; Recall**

I. Introduction

Intrusion Detection Systems (IDS) are security tools designed to detect and prevent unauthorized access to computer systems by identifying potential intruders and raising alerts when suspicious activities are detected or found. IDS are critical components of modern network security architectures and are used to protect against a wide range of cyber threats, including malware infections, denial-of-service attacks, and network intrusions. IDS can be categorized into two types: network-based IDS and host-based IDS. Network-based IDS monitor network traffic for suspicious activity and raise alerts when potential intrusions are detected. On the other hand host-based IDS, monitor individual hosts or devices for suspicious activity and can detect attacks that may be missed by network based IDS. Both plays a vital role in finding the intrusions in the networks.

Traditional IDS relies on rule-based systems to identify potential threats, but these systems are often limited in their ability to detect complex and sophisticated attacks. Machine Learning (ML) algorithms have emerged as a promising solution to enhance the performance and accuracy of IDS. It is a known fact that, ML algorithms can analyse huge amounts of data and identify patterns and anomalies that may indicate the presence of an unauthorized access. They can also adapt to changing environments and learn from experience to improve their detection capabilities over time.

With the growing complexity and sophistication of cyber-attacks, ML algorithms are increasingly being used to enhance the efficiency and accuracy of IDS.

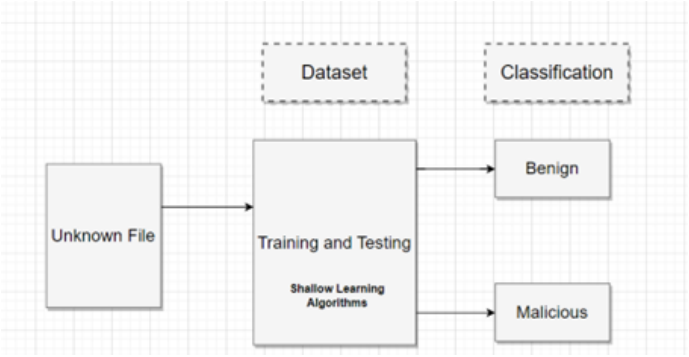

Fig. 1. Design of ML Algorithms on IDS

II. Model Training

The model is trained (= estimated) minimizing the empirical error.

$$L_s(h) = \frac{1}{m_t} \sum_{z_i \in S_t} l(h, z_i)$$

Machine Learning (ML) algorithms have emerged as a promising solution to enhance the performance and accuracy of IDS. ML algorithms can analyse large amounts of data and identify patterns and anomalies that may indicate the presence of an intrusion. However, the application of ML algorithms to IDS is not without its challenges. These challenges include selecting appropriate algorithms, features, and models, as well as addressing issues related to data quality, data quantity, and class imbalance.

This comprehensive study aims to explore the impact of ML algorithms on IDS and to provide insights into the strengths, limitations, and challenges associated with their applications. Additionally, this study will also examine the different types of IDS and the various ML algorithms used to enhance their efficiency, also the factors that influence the effectiveness of these algorithms. Ultimately, the goal of this study is to provide a comprehensive analysis of the influence of ML algorithms on IDS and to identify ways in which they can be optimized to enhance the security of computer networks. There are several machine learning (ML) algorithms that are commonly used in the cybersecurity field to detect and prevent cyber threats. Here are some of the most used ML algorithms:

A. Decision Trees

Decision trees are treelike structures that represent decision making rules for classification tasks. They are commonly used in cybersecurity to classify network traffic as either normal node or malicious node. The performance of the decision tree model depends on various factors such as the quality of data, the choice of hyperparameters, the size and quality of the training data, and the complexity.

B. Random Forests

Random forests are an ensemble learning method that combines multiple decision trees to improve their accuracy and reduce overfitting. It is a collection of decision trees that are trained on different subsets of the data and use a random subset of features. Finally, it helps to reduce overfitting and increase the generalization ability of the model.

C. Support Vector Machines (SVMs)

SVMs are a type of supervised learning algorithm that are used for classification and regression analysis. They are commonly used in cybersecurity to classify network traffic and detect malicious activities after training the data. Here, the nature of data is clearly identified either linearly separable or not.

D. Neural Networks

Neural networks are a type of ML algorithm that are modelled after the structure of the human brain. They are used in cybersecurity to detect malware infections and other cyber threats by analysing large amounts of data. It is trained using a large dataset of network traffic and security events, allowing it to learn to identify patterns and predict potential threats. It is entirely based on the data being received. As new network traffic is analyzed, the neural network can use its training to make informed decisions about the likelihood of a security threat and take appropriate action to prevent it.

III. Network Traffic Analysis

Network traffic analysis is the process of monitoring and analysing network traffic to identify potential cyber threats and security vulnerabilities. The goal of network traffic analysis is to detect abnormal patterns or behaviours that may indicate an intrusion or an attack. [8]

Network traffic analysis involves the capture, storage, and analysis of network traffic data. The data is typically collected using network monitoring tools, such as packet capture software or network probes. The collected data is then processed and analyzed to identify potential cyber threats or security issues. There are several techniques that are commonly used in network traffic analysis, including:

Signature-based detection: This technique involves matching network traffic against a database of known signatures or patterns of malicious activity. Machine learning can also be used to enhance IDS performance by automatically learning and updating attack signatures, identifying new attack patterns and detection accuracy.

Anomaly-based detection: This technique involves identifying abnormal traffic patterns or behaviours that may indicate an intrusion or attack. By analyzing large amounts of data and learning from patterns, ML algorithms can identify anomalies that may be missed by traditional rule-based approaches. These algorithms can also adapt to changing network conditions and detect previously unknown attacks.

Machine learning-based detection: This technique involves training machine learning algorithms to identify and classify network traffic based on various features and parameters.

In addition to these techniques, network traffic analysis also involves the use of visualization tools, such as network maps and flow diagrams, to help analysts better understand the network traffic and identify potential issues.

On the whole, network traffic analysis is a critical component of modern cybersecurity, as it helps organizations identify and prevent cyber threats before they can cause damage or compromise sensitive data.

Machine learning (ML) algorithms are commonly used in network traffic analysis to detect and prevent cyber threats. Here are some of the ML algorithms commonly used in network traffic analysis:

Clustering: Clustering algorithms are used to group similar network traffic patterns into clusters. This can be useful for identifying unusual traffic patterns or detecting network anomalies.

Naive Bayes: Naive Bayes is a probabilistic ML algorithm that is commonly used in network traffic analysis for classification tasks, such as detecting malware or identifying suspicious traffic.

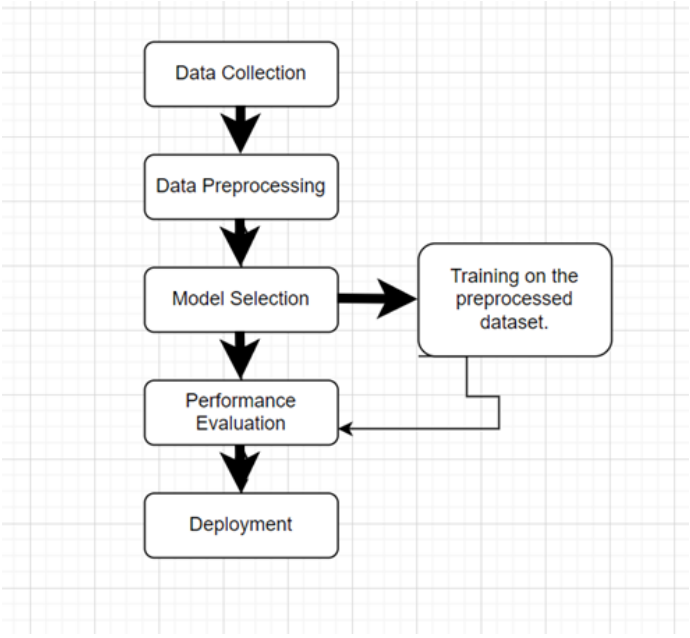

Fig. 2. High-level overview of implementation of ML algorithms on IDS

The following graph shows the accuracy comparison by taking various datasets and plotted as per their performance on detecting intrusions:

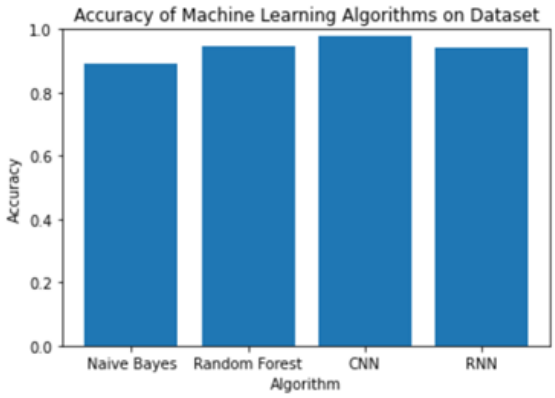

Fig. 3. Accuracy Comparison of ML algorithms

Random Forests: Random forests are an ensemble learning method that can be used to classify network traffic and detect potential cyber threats. Random forests can be used to classify network traffic based on features such as source and destination addresses, packet size, and protocol type.

Deep Learning: Deep learning algorithms, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can be used for network traffic analysis to detect and prevent cyber threats. For example, CNNs can be used for intrusion detection by identifying patterns in network traffic, while RNNs can be used to analyse data and detect network anomalies. [11]

All things considered, ML algorithms are essential for network traffic analysis to improve the accuracy and efficiency of cybersecurity systems and detect potential cyber threats in real-time.

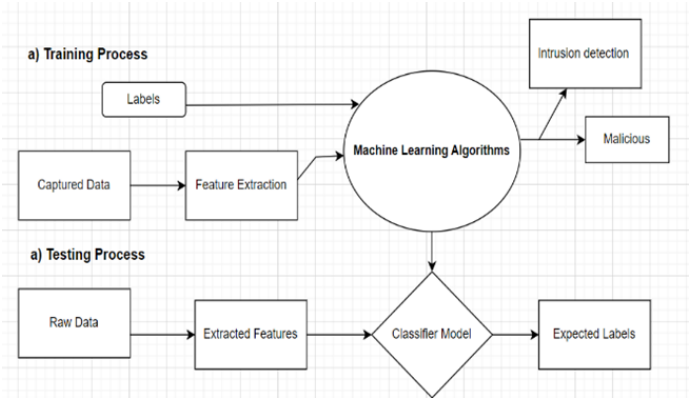

Fig. 4. Full-scale deployment of the process

IV. Web Application Firewalls

Web application firewalls (WAFs) can implement machine learning (ML) algorithms to improve their ability to detect and prevent attacks on web applications. By using ML algorithms, WAFs can analyze large amounts of traffic data to identify patterns and anomalies that may indicate an attack, and make real-time decisions about whether to allow or block incoming traffic.

Some ways in which WAFs can implement ML algorithms include:

Behavioural analysis: ML algorithms can be trained to analyze the behaviour of web users and identify patterns that may indicate an attack. For example, if a user is accessing a web application at unusual times or from unusual locations, it may indicate that their account has been compromised.

Anomaly detection: ML algorithms can be used to identify anomalies in web traffic that may indicate an attack. For example, if a user is sending an unusually large number of requests or attempting to access sensitive parts of the application that they don't normally access, it may indicate an attack.

Dynamic updating: ML algorithms can be used to update WAF rules dynamically, based on new attack patterns or changes in web application traffic. This can help ensure that the WAF is always up-to-date and able to detect and prevent the latest threats.

Collectively, implementing ML algorithms in WAFs can help organizations better protect their web applications

from attacks, by providing more accurate and dynamic threat detection and prevention capabilities. However, it's important to note that ML algorithms require significant data and resources to train and operate effectively and must be carefully configured and monitored to avoid false positives or negatives.

If test data contains only known attacks, then it is relatively easy to evaluate the performance of a detection or classification model, since the model has already seen all the possible attack types during the training phase. In this case, the model's ability to correctly identify the known attack types is a good measure of its performance.

On the other hand, if the test data contains unknown attacks, the performance evaluation becomes more challenging. The model has not seen these attack types during training, and therefore it may not be able to correctly classify them. In this case, the model's ability to detect and flag suspicious behaviour may be more important than its ability to accurately classify the attack type.

To improve the model's ability to handle unknown attacks, one approach is to use unsupervised learning techniques such as anomaly detection. Anomaly detection can identify patterns that are significantly different from normal behaviour and can flag them as potentially suspicious. Another approach is to continuously update the model with new attack types as they are discovered, which can help improve the model's ability to detect and classify new types of attacks.

V. Experimental Analysis

True Positive (TP) rate is the proportion of correctly detected malicious events by an IDS, out of all actual malicious events that occurred. In other words, it's the ratio of the number of true positives to the total number of positive instances, which includes true positives and false negatives. [3]

False Positive (FP) rate is the proportion of events that are wrongly detected as malicious by an IDS, out of all events that are not malicious. In other words, it's the ratio of the number of false positives to the total number of negative instances, which includes true negatives and false positives.

In summary, the true positive rate of an IDS indicates its ability to accurately detect malicious events, while the false positive rate indicates the degree to which the IDS generates false alarms or alerts about benign events. A high false positive rate can result in unnecessary disruption and time-consuming investigation, while a low true positive rate can leave a network vulnerable to attacks. [3]

The mathematical formulas for true positive rate (TPR) and false positive rate (FPR) are as follows:

$$TruePositiveRate(TPR) = \frac{TP}{(TP + FN)}$$

where TP represents the number of true positives and FN represents the number of false negatives

$$FalsePositiveRate(FPR) = \frac{FP}{(FP + TN)}$$

where FP represents the number of false positives and TN represents the number of true negatives.

It is important to note that, TPR and FPR are usually represented as percentages, with values ranging from 0% to 100%. A higher TPR indicates better detection performance, while a lower FPR indicates fewer false alarms or false positives. In practice, IDS performance is often evaluated using a metric called the receiver operating characteristic (ROC) curve, which plots the trade-off between TPR and FPR at different thresholds.

The following graph has been plotted using false positive and true positive values, you can create a receiver operating characteristic (ROC) curve. An ROC curve is a graphical representation of the performance of a binary classification model that varies the threshold for classifying an observation as positive or negative.

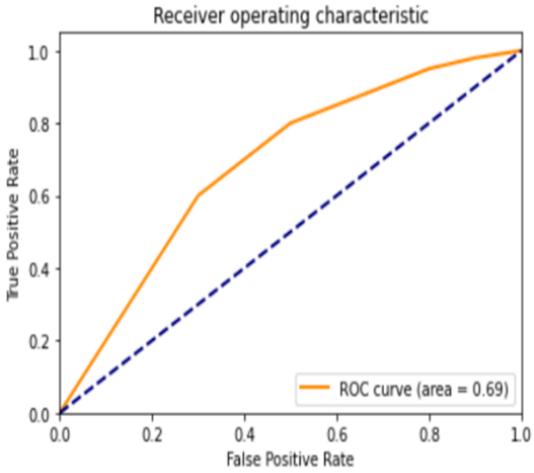

Fig. 5. ROC curve showing the trade-off between TPR and FPR

The curve started at the bottom left corner (0, 0) and moved towards the top right corner (1, 1). The area under the ROC curve has been calculated as a measure of the model's performance. An area under the curve of 1.0 represents a perfect model, while 0.5 represents a random guess.

Different relative analyses in different classification systems were carried out according to the previous work [1] and

clearly specified the results as below,

Random Forest with Non-Symmetric Deep Auto Encoder:

Non-symmetric Deep Auto Encoder help to improve the accuracy and efficiency of the Random Forest algorithm by reducing the dimensionality of the input data and extracting relevant features. [1]

Two-Stage Model using Stacked Auto Encoder:

A two-stage model using stacked autoencoders is a deep learning architecture that involves training multiple layers of autoencoders to extract features from the input data in a hierarchical manner. [1]

Convolutional Neural Network with Principal component analysis (PCA) and Auto Encoder for dimension reduction:

Combining CNNs with PCA and Autoencoder for dimensionality reduction can improve the performance and efficiency of the CNN architecture by reducing the dimensionality of the input data and removing noise and redundancy.

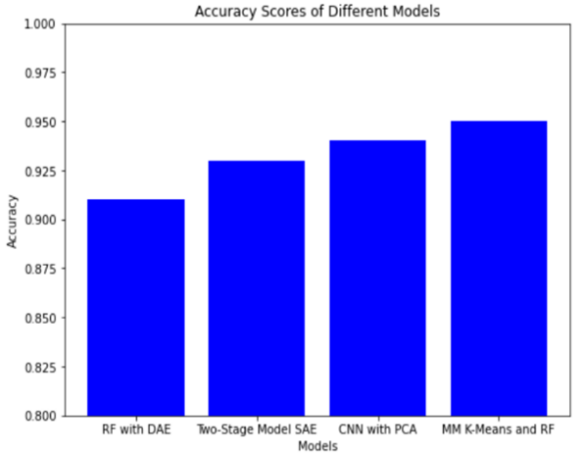

Fig. 6. Comparsion of accuracy scores

A Multilevel Model based on K-Means Clustering and Random Forest:

Multilevel model approach handles large and complex datasets by partitioning them into smaller, more manageable subsets, and it can capture cluster specific relationships and patterns. It can also improve the accuracy and efficiency of the model by reducing the number of features and data points in each cluster. Precision, recall, and F1 score are all evaluation metrics used to assess the performance of a classification model. [10] Precision, Recall and F1 Scores of the specified models are demonstrated in the line graph in Fig. 7.

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)}$$

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)}$$

$$F1Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

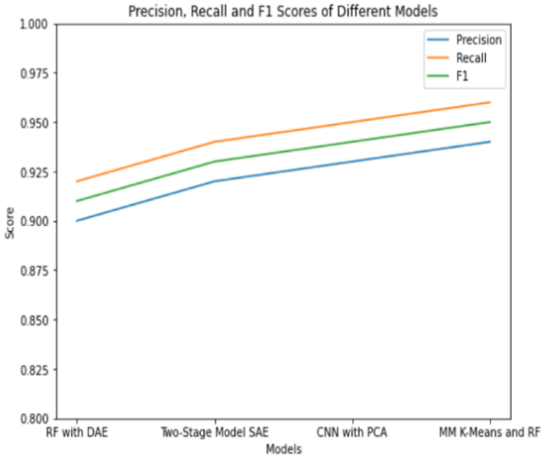

Fig. 7. Precision, Recall and F1 scores

VI. Conclusion

ML algorithms offer a promising alternative for IDSs, as they can learn patterns and anomalies from large datasets and can adapt to changing attack patterns. Some examples of ML algorithms used in IDS include supervised learning methods such as Random Forest, Support Vector Machines (SVM), and Deep Neural Networks (DNN), as well as unsupervised learning methods such as K-Means clustering and Self-Organizing Maps (SOM). [10]

Research in this area has focused on evaluating the performance of ML-based IDSs in detecting various types of attacks, such as Denial of Service (DoS), Distributed DoS (DDoS), and other network-based attacks. Studies have also examined the impact of different factors, such as the size of training data, the selection of features, and the choice of ML algorithm on the performance of IDS.

ML-based IDSs can achieve higher accuracy rates in detecting attacks compared to traditional IDSs. The choice of ML algorithm and the selection of features can significantly impact the performance of IDSs. The performance of ML-based IDSs can be improved by using larger training datasets and optimizing the hyperparameters of the ML algorithm. The use of ML algorithms can also help in reducing false

positives and improving the speed of detection.

Finally, research on the influence of ML algorithms on IDS is an important area of study, as it can help in improving the security of computer networks and protecting against cyberattacks.

REFERENCES

[1] (2021) Sandy Victor Amanoul, Adnan Mohsin Abdulazeez, Diyar Qader Zeebare, Falah Y. H. Ahmed *"Intrusion Detection Systems Based on Machine Learning Algorithms"*

[2] (2004) Stefano Zanero and Sergio M. Savaresi, *"Unsupervised learning techniques for an intrusion detection system"*

[3] Mahdi Zamani and Mahnush Movahedi, *"Machine Learning Techniques for intrusion detection"*

[4] (2017) Shalini and Vasudevan *"An Overview of Machine Learning Techniques for Intrusion Detection Systems"*

[5] (2017) Islam et al. *"A Comparative Study of Machine Learning Algorithms for Intrusion Detection System"*

[6] (2019) Alrawashdeh et al. *"Deep Learning-Based Intrusion Detection System Using Convolutional Neural Network"*

[7] (2021) Dang et al. *"Anomaly Detection Using Machine Learning Algorithms for Intrusion Detection: A Systematic Review"*

[8] (2020) Manickam et al. *"A Hybrid Machine Learning Approach for Network Intrusion Detection System"*

[9] (2000)Axelsson, *"Intrusion Detection Using Machine Learning Techniques"*

[10] (2009) Tavallaee et al. *"Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context"*

[11] (2021) Han et al. *"An Intelligent Intrusion Detection System for IoT Networks Using Deep Learning Approach"*

[12] Mustafa and Zaki ,*" A Deep Learning Approach to Intrusion Detection Using Recurrent Neural Networks"*

[13] (2022) Emad E. Abdallah*, Wafa' Eleisah, and Ahmed Fawzi Otoom, *"Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey"*