Analysis of the SNORT intrusion detection system using machine learning

Ouafae El Aeraj[1], Cherkaoui Leghris[2]
Laboratory of Mathematics, Computer Science and Applications, Faculty of Sciences and Techniques
Mohammedia, Hassan II University of Casablanca
Mohammedia, Morocco
ouafaeelaeraj@gmail.com[1], cherkaoui.leghris@fstm.ac.ma[2]

***Abstract*— Today, cyber-attacks that exploit networks and systems vulnerabilities are becoming more and more effective, reflecting the malicious intentions of certain Internet users. These attacks harm both individuals, through loss or theft of personal data and invasion of privacy, and businesses, through loss of know-how, damage to reputation and financial loss. Against this backdrop, it is essential that network operators adopt robust security measures. Intrusion Detection Systems (IDS) are emerging as promising solutions for strengthening network security. An IDS discreetly monitors network traffic for abnormal or suspicious behavior, enabling proactive accessibility measures to be taken against intrusion attempts. This article focuses on intrusion detection technologies, and more specifically on SNORT, a tool capable of identifying network intrusions in real time. We will explore the vulnerabilities associated with this technology and look at research that applies machine learning methods to overcome these shortcomings.**

***Keywords*—SNORT, Intrusion detection System, Support vector machines, and Machine Learning.**

I. INTRODUCTION

In an increasingly connected world, where the digitization of business processes and dependence on information systems are growing, network security has become a major concern for organizations of all sizes. As computer networks become increasingly complex and extensive, they also attract a variety of malicious actors, from individual hackers to criminal organizations and even state agencies, looking to exploit vulnerabilities for their own gain. These intrusions can have disastrous consequences, including the loss of sensitive data, disruption of business operations and significant damage to corporate reputations. In the face of this constant threat, Intrusion Detection Systems (IDS) are an essential line of defense in an organization's network security arsenal.

IDSs are designed to monitor network traffic for signs of suspicious or malicious activity, enabling network administrators to intervene quickly before attackers can cause real damage. It functions as a software tool that monitors activity on systems and networks, identifying and reporting attacks and other forms of malicious behavior occurring in network environments. Upon detection, it generates reports which are sent to the system's security administrator, as illustrated in Figure 1 [1].

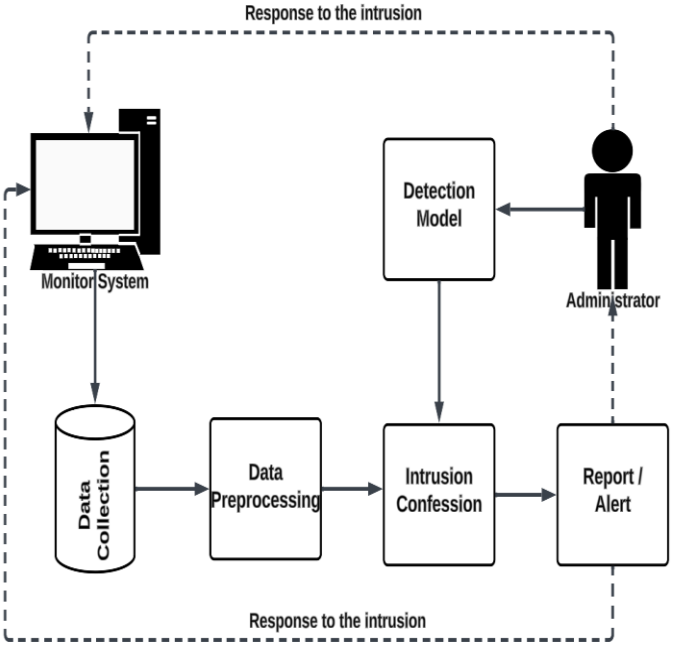

Fig. 1. Overview of intrusion detection system.

For more explanation of the operational flow within an Intrusion Detection System (IDS), starting with the "monitoring system", meaning the continuous monitoring of network traffic, the process then moves on to "data collection", where data is systematically collected. Subsequently, "data pre-processing" is represented, indicating the refinement and organization of data, essential for accurate analysis. At the heart of the system is the "Detection Model", a critical decision-making stage in which pre-processed data is examined for anomalies indicating security flaws. The next stage, called "Intrusion Confession", involves the confirmation of any security incident. Upon detection, the process culminates in the "Report/Alert" phase, during which the system generates a notification for the "administrator", prompting review and appropriate response to the identified intrusion. This entire sequence is encapsulated in a feedback loop, "Intrusion Response", highlighting the cyclical nature of the IDS process, geared towards continuous monitoring and improved response to detected threats.

An Intrusion Detection System (IDS) is an essential security device that monitors abnormal or malicious activity within a network or host system. This tool is fundamental to developing prevention tactics and implementing appropriate responses to potential incursions. It analyzes the flow of data

passing through the system and triggers alerts in the presence of suspicious behavior or data, thus contributing to the proactive detection of attacks against the computer system. IDSs fall into three main categories:

- Network Intrusion Detection Systems (NIDS) work by examining incoming and outgoing traffic through a key location in the network. They have the ability to scan traffic across an entire subnet and identify suspicious activity without straining the resources of the systems they protect. These systems offer real-time monitoring and intervention, as their analysis often focuses on small volumes of data, such as packets or data streams as illustrated in Figure 2. Nevertheless, their ability to detect attacks occurring directly on individual devices or in network segments that do not cross the control point is limited. Furthermore, they cannot examine the contents of encrypted data packets.

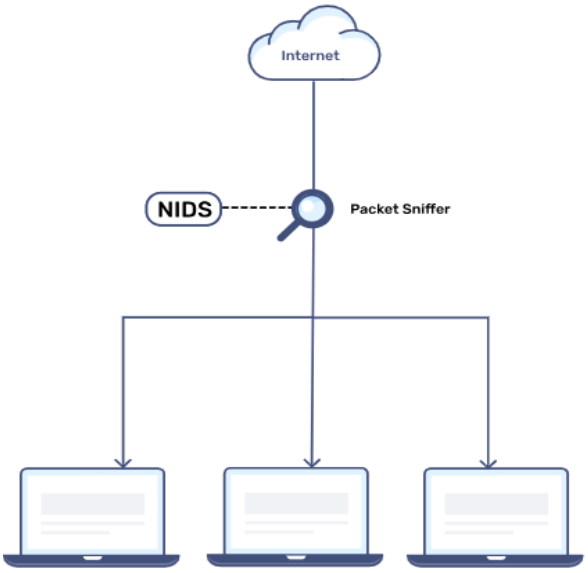

Fig. 2. Deployment of a network intrusion detection system.

- Host Intrusion Detection Systems (HIDS) are designed to monitor activity at the level of a specific device, monitoring both internal activity and incoming and outgoing data traffic as illustrated in Figure 3. These systems are capable of performing in-depth analysis of internal operations and the content of data packets, offering immediate reaction to incidents. However, their operation requires the use of the resources of the system they monitor, and their perspective on network activity is restricted to the device to which they are attached.

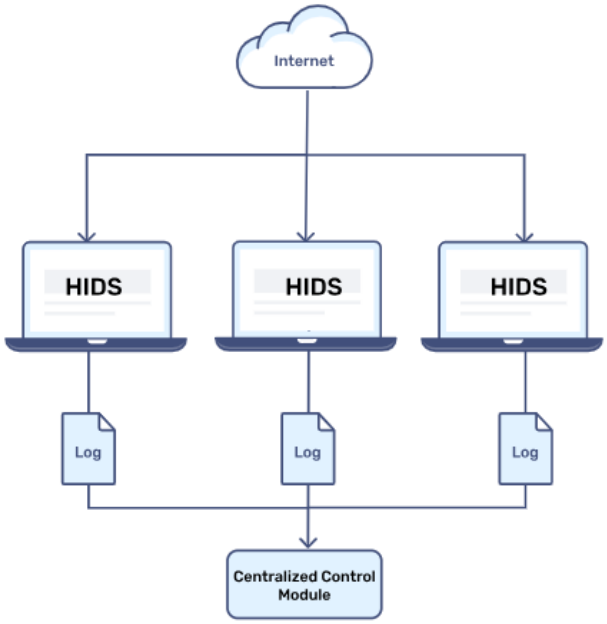

Fig. 3. Deploying the host intrusion detection system.

- Hybrid Intrusion Detection Systems (Hybrid IDS) merge the functionality of NIDS and HIDS systems to provide more comprehensive security monitoring and generate more relevant alerts, as illustrated in Figure 4. By integrating the ability of NIDS to inspect global network traffic with the precision of HIDS in monitoring events at the level of individual systems, Hybrid IDS offer a more nuanced perspective of potential threats. This combined approach enables information to be correlated between the network and hosts, improving intrusion detection and reducing the number of false alarms, leading to more effective management of security resources.

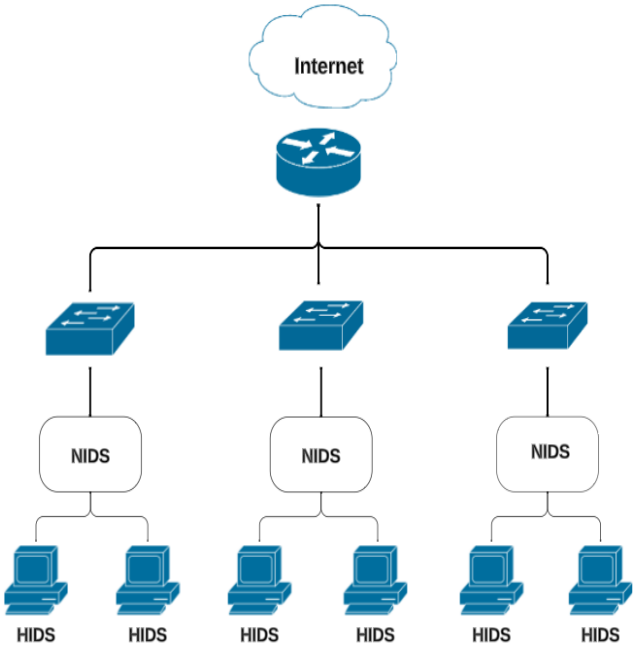

Fig. 4. Deployment of a hybrid intrusion detection system.

Unlike HIDSs, which are limited to a single host, NIDSs can monitor an entire network. HIDS are particularly effective at determining whether a host is infected.

Network intrusion detectors, which monitor and analyze network traffic, look for signs of attack and transmit alerts, make up NIDS. A NIDS consists of three main elements: capture, signatures and alerts (Figure 5).

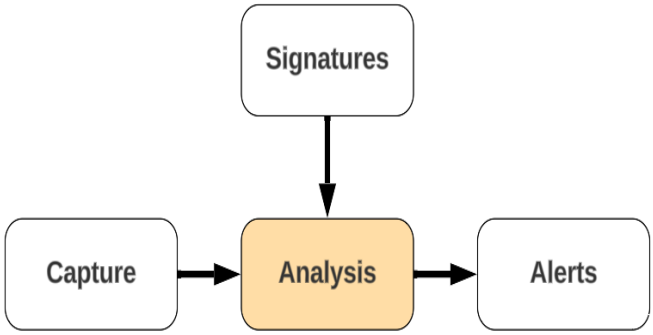

Fig. 5. NIDS functioning.

- *Capture:* the capture component of a network intrusion detection system (NIDS) is responsible for collecting network traffic, which serves as a data source for security analysis. Typically, this data capture is carried out in real time, enabling the NIDS to analyze network packets as they pass over the network.
- *Signatures:* the signature component works in the same way as antivirus software, using a database of known threat patterns or "signatures" to identify malicious activity. During analysis, incoming network data is compared with these signatures using a scenario-based approach. If the traffic matches a known threat signature, NIDS recognizes the activity as a potential attack.
- *Alerts:* Alerts are generally stored in a syslog server. However, there is a standard that allows the content to be formalized, so that many security components can work together. According to RFC4765, this format is known as IDMEF (Intrusion Detection Message Exchange Format). IDMEF has been popularized by the Prelude project, which provides an infrastructure that allows IDSs not to have to worry about issuing alarms. This enables IDSs to simply describe the information they know, and Prelude takes care of storing it so that it can be consulted by humans at a later date.

SNORT is an open-source network intrusion detection system (IDS) and prevention system [2], renowned for its ability to inspect IP traffic in real time, making it one of the most widely used NIDS.It is a lightweight monitoring solution that captures and examines data packets transiting the network. SNORT uses user-defined rules to filter incoming packets, and issues alerts when a match is found with known threat signatures [3]. Able to identify a multitude of attacks, such as attempted port scans, SNORT is based on a detection engine with a modular architecture, allowing the integration of plug-ins to extend its functionality.

SNORT has three main uses. It can be used as a network traffic logger for later analysis, act as a packet sniffer like tcpdump, or operate as a complete network intrusion detection system.

SNORT architecture consists of four main modules, detailed below, each playing a crucial role in monitoring and analyzing network traffic for intrusion detection as illustrated in Figure 6:

1) *Packet Decoder*
   This module serves as an entry point for network traffic. It captures packets in transit on the network and breaks them down into different protocols (IP, TCP, UDP, ICMP, etc.), facilitating subsequent analysis. This enables SNORT to understand the structure of the data it is inspecting.
2) *Preprocessors*
   Preprocessors modify and prepare packet data before it reaches the detection engine. This can include reassembling packet fragments, normalizing traffic to counter evasion techniques, and identifying and managing specific communication sessions. Pre-processors improve detection efficiency and accuracy by pre-processing data to solve complex problems.
3) *Detection motor*
   At the heart of SNORT, the detection engine analyzes pre-processed traffic for patterns or signatures corresponding to known malicious behavior, exploits or vulnerabilities. Based on a set of defined and regularly updated rules, this module decides whether a specific packet represents a potential threat.
4) *Alert and Logging System*
   When suspicious activity is detected, this module is responsible for alert management and event logging. Actions can range from simple logging to real-time alerts, enabling administrators to take immediate action. This module also ensures that evidence of detected activities is retained for further analysis.

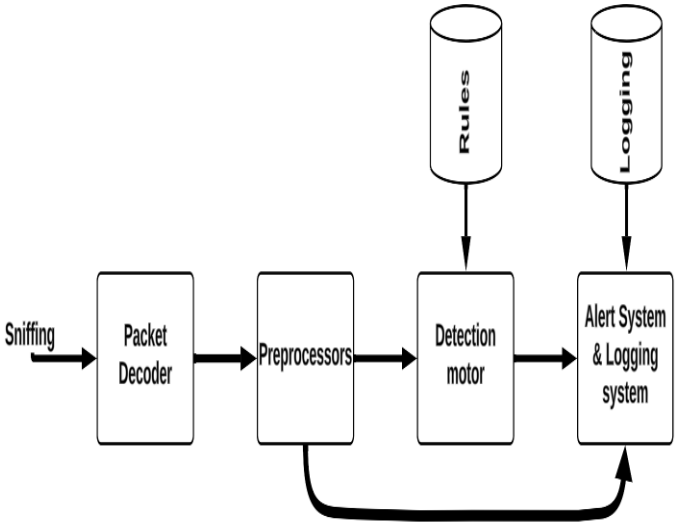

Fig. 6. SNORT architecture.

SNORT systems, in their role as network guardians, are designed to intercept and block what they identify as potential threats. However, their accuracy is not infallible, which can lead to unintentional blocking of legitimate network traffic or authorized applications. Although powerful, these systems have

a greater propensity to generate inaccurate alerts, with an estimate that around 70% of alert notifications turn out to be false positives. This tendency to over-report incidents that are not actually threats can hamper operational efficiency, and calls for careful alert management.

Faced with this limitation, the scientific community and cybersecurity professionals have been exploring ways of improving SNORT, notably through the integration of machine learning techniques. This research aims to refine SNORT's detection capabilities, enabling it to better discern genuine threats from benign activities. The application of machine learning offers significant promise, equipping SNORT with increased intelligence to analyze traffic patterns, learn from past incidents and substantially reduce the rate of false alarms.

By integrating Machine Learning (ML) with SNORT, a widely used network intrusion detection system (NIDS), it is possible to create a more sophisticated and adaptable security mechanism. SNORT, in its original form, relies primarily on manually defined rules to identify network threats. These rules are based on signatures specific to known attacks, which can limit its ability to detect new threats or variants of existing attacks. However, by integrating machine learning techniques, this capability can be significantly enhanced.

Machine learning enables systems to learn and adapt to new information without being explicitly programmed for each situation. In the context of network security, this means that the system can learn patterns of normal and abnormal traffic over time, fine-tuning itself to better distinguish legitimate threats from false alarms. This ability to learn and adapt is crucial in the face of rapidly evolving computer threats.

Integrating Machine Learning into SNORT can be done in several ways, including:

1) Traffic classification: Use supervised learning to classify traffic as normal or malicious on the basis of characteristics extracted from network traffic. This can help identify zero-day attacks or variations on known attacks that do not exactly match existing signatures;

2) Behavioral analysis: Apply unsupervised learning techniques to identify anomalies in network traffic that could indicate an intrusion or attack attempt, without relying on known attack signatures;

3) Rule optimization: Use reinforcement learning to dynamically adjust detection rules according to their effectiveness in the current network context, thus reducing false positives while maintaining effective threat detection;

4) Resource management: Optimize the allocation of computing resources for threat detection by anticipating periods of heavy traffic and adapting resources accordingly, to minimize the impact on network performance.

In conclusion, the addition of machine learning capabilities to SNORT promises to transform network intrusion detection into a more dynamic, accurate and efficient process. This represents a major advance in the fight against cybercrime, offering better protection against known and emerging threats, while reducing disruption and the impact on system resources. As cyber-attacks become increasingly sophisticated, the importance of such advances in security technology cannot be underestimated, marking a potential turning point in our ability to defend critical IT infrastructures.

II. Intrusion detection systems overview

The relevance and effectiveness of the machine learning-enhanced network intrusion detection system presented in [4] highlights the vital importance of further research in this constantly evolving field, in order to meet the security challenges posed by contemporary computer networks. The study also proposes strategies for strengthening and developing the intrusion detection system, aimed at securing networks more effectively and sustainably in the face of emerging threats.

Ensuring the security of DNS infrastructures is essential to maintaining the integrity and functionality of the Internet, a fact well documented in [5]. This study describes the vulnerabilities exposed by DNS amplification attacks - which amplify the volume of unwanted traffic by exploiting DNS servers - and DNS tunneling attacks, which cleverly bypass established security measures by disguising malicious traffic as legitimate DNS requests. The comprehensive analysis provided reveals that the deployment of an Intrusion Detection and Defense (IDD) system, incorporating SNORT's advanced detection capabilities, offers a robust solution against these sophisticated threats. Notably, this SNORT-enhanced DID framework demonstrates high effectiveness in detecting the nuanced mechanisms of DNS amplification and tunnelling activities, with the added benefit of significantly reducing the rate of false-positive alerts. This double success underlines the essential role of continuous innovation and adaptation of cybersecurity technologies to protect against the evolving landscape of Internet-based threats, in particular to safeguard vital DNS infrastructures.

The development of intrusion detection systems suitable for smart homes is crucial, given the growing security vulnerabilities. The architecture suggested in [6], using an anomaly-based strategy, offers a promising way to counter imminent threats and protect individual privacy and security in smart home environments.

As networks continue to evolve rapidly, and Software Defined Networks (SDN) become more widespread, it is essential to evaluate the performance and effectiveness of Intrusion Detection Systems (IDS) adapted to these modern environments. The benchmarking efforts detailed in [7] have produced relevant data, proving to be a crucial resource for network security professionals aiming to strengthen the defenses of their SDN infrastructures. Nevertheless, the ever-changing landscape of cyber threats, coupled with advances in SDN technology, requires relentless monitoring and perpetual improvement of IDS capabilities. To ensure the continued security of networks, it is imperative that these systems are not only adjusted to meet current vulnerabilities, but are also

adaptable to counter future threats. This approach underlines the importance of a proactive, dynamic cybersecurity management strategy to protect against increasingly sophisticated attacks within the evolving SDN ecosystem.

Over the past decade, an extensive review [8] has been conducted into the integration of machine learning (ML) techniques to enhance cybersecurity measures. This review covered key application areas including intrusion detection, malware classification, log analysis, anomaly detection and distributed denial of service (DDoS) attack prevention. The results of this study demonstrated that ML methods have the potential to significantly improve the identification and combating of cyber threats. Despite this progress, there are still critical areas of research that require attention to further strengthen cybersecurity defenses. These include securing access to high-quality data, improving the adaptability of systems to counter new and evolving threats, and deepening understanding of patterns indicative of malicious activity.

Graphical neural network (GNN) machine learning models excel at processing graphically structured data. The research described in [9] is devoted to exploring their use in the detection of malicious activity, in particular to identify network intrusions. This research highlights the unique attributes of graphs that make them an advantageous approach in cybersecurity, delving into various GNN techniques applied for intrusion detection purposes. In addition, it addresses the challenges inherent in using GNNs in this scenario and discusses strategies for mitigating these issues. The findings provide an in-depth look at the latest advances in the application of graphical neural networks for network intrusion detection.

The proposed method for intrusion detection, as described in [10], is based on a self-organizing intelligent learning framework that dynamically adapts to changes in the network and new threats. This system uses specific constraints to guide the learning process, thus improving the accuracy of malicious activity detection. In addition, the paper details the benefits of this innovative approach, highlighting its exceptional adaptability to the ever-changing landscapes of software-defined network (SDN) environments. It also highlights the system's ability to identify complex cyber-attacks and its effectiveness in minimizing the incidence of false positives. By incorporating advanced machine learning techniques, this approach not only meets the current state of network security, but also offers a forward-looking solution capable of evolving with future cybersecurity challenges. This adaptability, combined with enhanced detection capabilities and reduced error rates, makes this framework a significant advance in network security, offering robust protection against a wide range of cyber threats.

Machine learning, which refers to the study of artificial intelligence, is gaining increasing acceptance in the field of computer science. The term describes a process of development, analysis and implementation that results in the establishment of systematic processes. In simple terms, it's a type of software that enables a machine or computer to learn automatically, so that, it can perform a variety of extremely difficult tasks. This is why researchers are looking to use it to enhance network security.

In the study presented in [11], the authors propose the integration of SNORT with a backpropagation neural network (BPNN) to develop a hybrid and cooperative network intrusion detection system (CH-NIDS) designed specifically to identify network threats in cloud environments. The system begins by extracting crucial features from raw network traffic, which are then transformed into a format suitable for neural network processing during the pre-processing phase. This step ensures that the data fed into the BPNN is both concise and reflects potential intrusion signatures. The architecture of the proposed detection system is structured around four key components: a central repository for malicious packet logs, an alert mechanism, an anomaly detection module and a signature-based detection component as illustrated in Figure 7. This framework aims to improve the accuracy and efficiency of threat detection in cloud networks by leveraging the strengths of SNORT and BPNN.

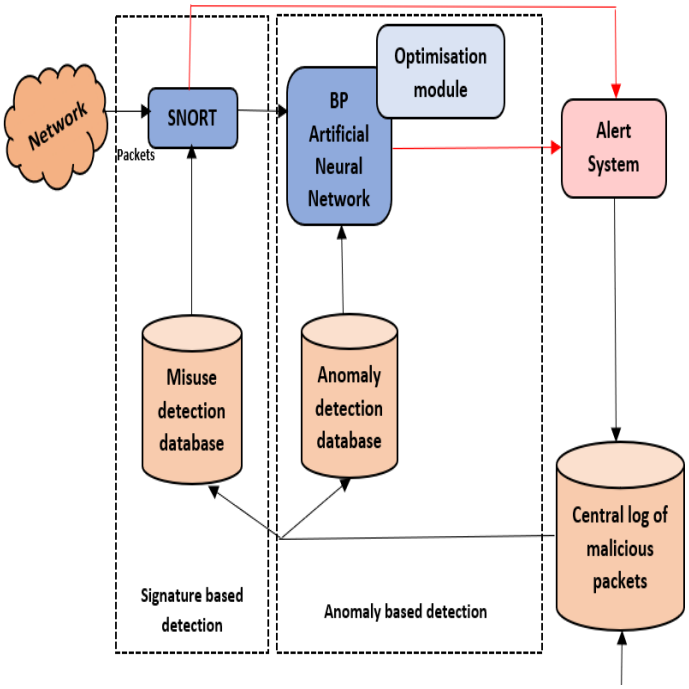

Fig. 7. Architecture of proposed CH-NIDS framework.

The CH-NIDS system combines anomaly-based and signature-based detection methods. Using SNORT for the signature-based approach, it examines captured network packets to identify intrusions. SNORT monitors network data flow, checking it against a collection of rules stored in an attack signature database, and generates alerts when it detects patterns of suspicious activity.

Packets deemed non-intrusive are then routed to an optimized back-propagation neural network (BPN)-based classifier, which assesses whether they belong to normal activity or constitute an intrusion attempt. If malicious activity is detected, the system reports the incident and archives the relevant details in the central database. If the packets are considered legitimate, the BPN classifies them as safe and allows them access to the network.

Article [12] introduces a hybrid approach structured in two phases, tested using the KDD Cup dataset. In the first phase, SNORT's role is to detect and alert on atypical behavior in the data. Data that are identified as normal by SNORT are then passed on to an analysis system for further classification of anomalies. The second phase employs data mining methods, specifically the combination of "k-means + C4.5" and the use of CART (Classification And Regression Trees), to distinguish between normal activities and those considered abnormal. This methodology demonstrated a significant improvement in detection accuracy while reducing the number of false alarms.

Paper [13] presents an innovative Python-based framework for monitoring and diagnosing the SNORT intrusion detection system in distributed firewall contexts. Using Python scripts and essential libraries such as subprocess, requests, and time, this framework performs regular checks on network connectivity, database server availability, resource consumption and SNORT configuration. Thanks to its modular architecture and the use of common Python libraries, it offers a flexible, scalable and easy-to-deploy solution. Extensive testing and simulations confirm the framework's effectiveness in detecting and resolving problems, ensuring robust network security management with SNORT in distributed firewall environments.

Faced with the security challenges posed by the Internet of Things (IoT), which relies on the transmission of data through networks of connected devices, the article [14] proposes the implementation of an Intrusion Detection System to enhance security. It plays a crucial role in protecting computer networks by detecting suspicious activity, and SNORT IDS software, is highlighted for its ability to identify known and unknown threats. By comparing network traffic with a set of predefined rules, SNORT is able to detect abnormal behavior and trigger alerts. The study uses the 1999 MIT-DARPA database to test the effectiveness of SNORT IDS in recognizing abnormal traffic patterns, and assesses the performance of SNORT's rules in mitigating security risks in IoT networks.

As the use of the Internet and related technologies increases, cyberattacks are becoming more frequent and more serious, making it essential to use cybercrime to gather electronic evidence admissible in court. Article [15] discusses the effectiveness of network forensics, which examines evidence of network intrusion to identify suspicious activity. It highlights the use of tools such as SNORT and Wireshark to detect and report attacks, using the example of a local network where an intrusion scenario has been simulated and a honeypot configured. The study revealed that integrating a forensic mechanism into SNORT significantly improved the detection of malicious activity, with a significant increase in the percentage of alerts generated and the volume of traffic analyzed compared to a standard configuration without forensic tools, underlining the importance of such mechanisms in cyber defense.

The article [16] examines port scanning practices and their dual role in network diagnostics and in the conduct of cyberattacks, where they are used to identify vulnerable hosts and execute unauthorized intrusions. Port scanning, increasingly sophisticated and elusive, poses growing challenges to detection. It also explores the use of SNORT, an intrusion detection tool, detailing its operation, installation and detection strategies. The study includes a practical case where SNORT is configured with specific rules to monitor network traffic and detect Denial of Service (DoS) attacks. This work illustrates SNORT's effectiveness in real-time monitoring and detecting the consequences of cyberattacks on the network.

In today's IT security landscape, where the Internet of Things, cloud computing and wireless communications predominate, network intrusion detection systems have become essential for preserving network integrity, confidentiality and availability. Among the many NIDS available, SNORT and Suricata stand out as pre-eminent open source options. The article [17] evaluates their performance, highlighting SNORT 3, which embodies the evolution of SNORT with innovations such as multithreading, functional extensions and improved cross-platform compatibility. Through a quantitative analysis conducted in a virtualized network, the study compares accuracy, memory and CPU efficiency, packet throughput, and packet loss rate between these systems. The results show that SNORT 3 outperforms SNORT 2 in performance, and that although SNORT 3 and Suricata perform well, they have shortcomings that require improvement.

III. SNORT BASED ON MACHINE LEARNING

SNORT traditionally relies on the use of predefined signatures to detect only known attacks. However, the study [18] developed sophisticated rules for SNORT, exploiting the WEKA machine learning tool and the j48 algorithm. This was done after evaluating various machine learning techniques with the CICIDS dataset, with the aim of recognizing new and previously unseen network attacks, while minimizing the number of false alarms. CICIDS was chosen as the reference dataset following a comparative study involving 15 datasets for intrusion detection systems. Based on the classification performance obtained with the j48 algorithm, expert rules were formulated and implemented in SNORT's own rule format. This improved system demonstrated an effective detection rate of 98%.

The study [19] highlights a comparison between SNORT and Suricata, both open source intrusion detection systems (IDS), in terms of their ability to detect malicious traffic. Although Suricata required more system resources than expected, it outperformed SNORT in handling high network traffic volumes, and posted a lower packet loss rate. On the other hand, SNORT showed superior accuracy in intrusion detection, despite a higher number of false alarms. To overcome this shortcoming, an adaptive plug-in for SNORT was developed. A comparative analysis of the performance of different learning algorithms was carried out to identify the most effective option for this plug-in. The support vector machine (SVM) was chosen for its performance, standing out from other methods such as decision tree (DT), fuzzy logic, BayesNet and NaiveBayes.

The study [20] presents a method that combines the SNORT and SVM (Support Vector Machine) algorithms to enhance

intrusion detection. This approach was tested on an artificial dataset, using Python as the programming language to integrate the attacks detected by SNORT into the SVM classifier. The effectiveness of this method was particularly notable for DDOS, DOS and TCP-SYN flooding attacks, with a remarkable accuracy of 99%. Results included 162 true positives, only 1 false positive, 160 true negatives and no false negatives. Compared with existing methods, the proposed system demonstrated significantly superior detection performance, underlining its effectiveness in accurately identifying threats.

TABLE I
SNORT IMPROVEMENT RESULTS THROUGH MACHINE LEARNING

Paper	Algorithm used	Dataset used	Detection type	Precision %	FPR %	Recall %
[11]	Back Propagation Neural Network (BPN)	No performance evaluation done	NIDS	N/A	N/A	N/A
[12]	K-means + C4.5	KDD Cup	NIDS	96,6	3,9	96,4
	K-means + CART			99,4	0,6	99,4
[18]	J48	CICIDS	NIDS	98	N/A	98
[19]	Support Vector Machines	NSA Snort IDS	NIDS	95,6	0,7	96,8
	Decision Trees	DARPA IDS		82	2,9	79,2
	Fuzzy Logic			92,3	0,2	94,5
	Bayes Net	NSL-KDD IDS		73	3,5	65
	Naive Bayes			70	3	62
[20]	Support Vector Machines	Snort IDS Alert	NIDS	99	0,6	100

Table 1 provides a detailed overview of the various research studies that have contributed to the SNORT intrusion detection system, exploiting a variety of machine-learning techniques. These machine learning methods are specifically implemented to overcome the limitations encountered with the SNORT IDS, in particular with regard to significantly reducing false alerts, both false positives (unjustified alarms) and false negatives (undetected incidents). This approach aims to improve the accuracy and effectiveness of SNORT as an intrusion detection system, by leveraging advances in machine learning to deliver more robust and reliable security.

The effectiveness of the models was evaluated using several key parameters: precision, false positive rate (FPR) and recall.

Precision is a critical evaluation criterion in the field of machine learning, assessing the ability of a classification model to accurately identify positive instances from the pool of examples it considers positive. Essentially, it quantifies the proportion of true positives in the predictions that the model classifies as positive. This metric is particularly important as it helps to understand the degree of reliability of a model's positive predictions, indicating the probability that a positively labeled result is actually correct. Accuracy is essential in scenarios where the cost of a false positive is high, ensuring that models are refined to minimize the occurrence of false alarms while correctly identifying true positive cases.

$$\text{Precision} = TP / (TP + FP)$$

TP (True Positives) represents the number of truly positive cases that the model has managed to correctly identify as such. This is a key indicator of the model's ability to recognize the true occurrences of a specific event or condition that it is designed to detect.

FP (False Positives), on the other hand, refers to the number of negative instances that the model has incorrectly classified as positive. In other words, these are situations where the model has predicted a positive result where, in reality, it shouldn't have, reflecting an error of judgment on the part of the model.

FPR (False Positive Rate) is a crucial metric in machine learning for measuring the effectiveness of a binary classification model as a whole. The FPR calculates the ratio between the total number of false positives (FPs) and the total number of true negatives (including FPs and true negatives, TNs). This ratio is essential to assess the extent to which a model is prone to misclassification by falsely marking negative cases as positive. A high RPF indicates a tendency for the model to generate an excessive number of false alerts, which can be particularly problematic in applications where the accuracy of positive predictions is crucial, leading to additional costs, wasted time, or reduced confidence in the model's performance.

$$\text{FPR} = FP / (FP + TN)$$

TN (True Negatives) represents the total count of instances that the model accurately classified as negative.

Recall, is a critical evaluation metric used in the field of machine learning to assess the ability of a classification model to accurately identify positive instances from the set of true positive cases. Specifically, it measures the proportion of true positives detected by the model compared to the sum of true positives and false negatives (cases that are actually positive but wrongly classified as negative by the model). This metric is essential for understanding the extent to which a model is able to capture all relevant instances without missing any positive cases.

Recall becomes particularly important in scenarios where the cost of missing a positive case is high. For example, in medical diagnostics, a high recall rate would mean that the model manages to identify a large majority of patients with a particular disease, minimizing the risk of leaving diseases undetected. Similarly, in fraud detection systems, a high recall value

ensures that most fraudulent transactions are detected, thus protecting against financial loss. In essence, recall provides insight into a model's ability to minimize false negatives, ensuring that as many positive cases as possible are correctly identified.

$$Rappel = TP / (TP + FN)$$

FN (False Negatives) refers to the number of positive occurrences that the model has falsely catalogued as negative.

Analysis of the results obtained from the experiments revealed that five classification algorithms stood out for their accuracy of over 95%. Among these algorithms, two in particular stood out, identified in documents [12] and [20], displaying a remarkable accuracy of 99% or more. These two algorithms have identical False Positive Rates (FPR) of 0.6%, however, document [20] stands out with a perfect Recall rate of 100%, slightly surpassing that of document [12] which is 99.4% as illustrated in Figure 8 . This critical distinction establishes that the SVM (Support Vector Machine) algorithm excels not only in detection accuracy, but also in its ability to minimize unjustified alarms, thus positioning it as the optimal choice among the methods studied. Furthermore, an innovative combination of two algorithms, K-means and CART, proved to be the second-best strategy, achieving a precision and recall rate of 99.4%. This comparative analysis underlines the importance of choosing algorithms adapted to the specificities of the data being processed, in order to optimize performance in terms of detection accuracy and reliability, while minimizing classification errors.

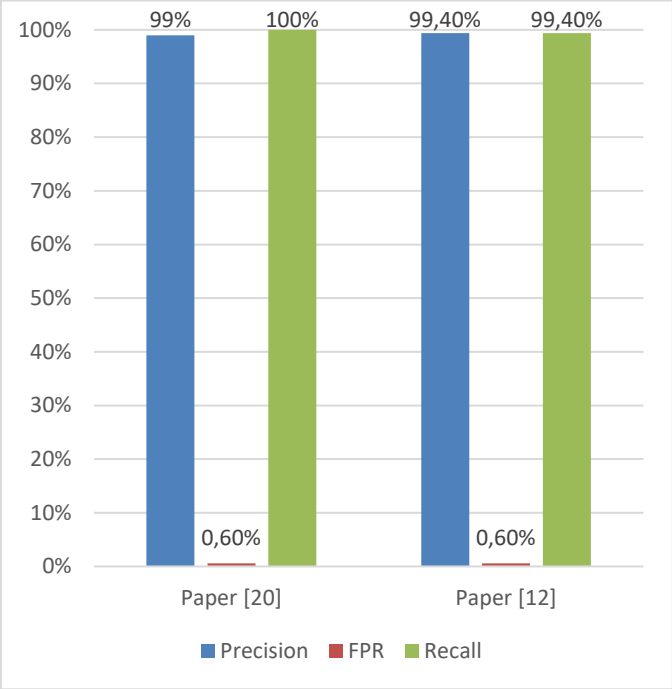

Fig. 8. Comparaison des Performances des Algorithmes de Classification : Précision, FPR et Rappel.

IV. Conclusion and Prospects

In the research presented within this document, we conducted an extensive analysis of various machine learning-based intrusion detection system deployments. Our investigation focused on the efficacy of integrating machine learning strategies into the SNORT IDS framework, specifically aiming to enhance the accuracy of its threat detection capabilities. The empirical data gleaned from our comparative study indicates that the implementation of the Support Vector Machine (SVM) algorithm significantly augments the system's precision in identifying threats, evidenced by a remarkable detection accuracy of 99%. Additionally, the algorithm exhibited a commendable reduction in the rate of false alarms, registering an impressively low false positive rate (FPR) of 0.6%, while maintaining a perfect recall rate of 100%.

The insights derived from this study does not only validate the effectiveness of machine learning techniques in bolstering the SNORT IDS, but also lay a solid foundation for future research endeavors. We postulate that the incorporation of sophisticated deep learning models could potentially yield even more robust and nuanced anomaly detection mechanisms, thereby further refining the precision and reliability of intrusion detection systems. Such advancements could pave the way for creating more resilient and secure cyber environments in an era increasingly threatened by complex and evolving digital intrusions.

REFERENCES

[1] M. Tabash, M. Allah and T. Benbella, “Intrusion Detection Model Using Naive Bayes and Deep Learning Technique,” The International Arab Journal of Information Technology,vol.17,pp.215-224,10.34028/iajit/17/2/9, 2020 .

[2] M. Essid, F. Jemili and O. Korbaa, “Distributed Architecture of Snort IDS in Cloud Environment,” 19th International Conference on Intelligent Systems Design and Applications,pp. 100-111, 0.1007/978-3-030-49342-4-10, 2019.

[3] R. Gupta, S. Singh, S. Verma and S. Singhal,,“Intrusion detection system using SNORT,” International Research Journal of Engineering and Technology , vol. 04, pp. 2100-2104, 2017.

[4] X. Li, "Research and Design of Network Intrusion Detection System," 2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA), pp. 1069-1072, 10.1109/ICPECA53709.2022.9718920, 2022.

[5] S.Adiwal, B.Rajendran, D.Pushparaj and S.Sudarsan, “ DNS Intrusion Detection (DID)—A SNORT-based solution to detect DNS amplification and DNS tunneling attacks,” Franklin Open, vol.2, pp. 100010, 10.1016/j.fraope.2023.100010,2023.

[6] A.Lara, V.Mayor, R.Estepa, A.Estepa and J.Díaz-Verdejo, “ Smart home anomaly-based IDS: Architecture proposal and case study,” Internet of Things, vol.22, pp. 100773, 10.1016/j.iot.2023.100773,2023.

[7] R.Shams, D.Suri, F.Hanif and P.Otero, “ Comparative Analysis Of Intrusion Detection Systems in SDN,” 2023 Global Conference on Wireless and Optical Technologies (GCWOT), pp.1-9, 10.1109/GCWOT57803.2023.10064664,2023.

[8] K.Shaukat Dar, S.Luo ,V.Varadharajan, I.Hameed and M.Xu, “ A Survey on Machine Learning Techniques for Cyber Security in the Last Decade,” IEEE Access,10.1109/ACCESS.2020.3041951,2020.

[9] T.Bilot, N.El Madhoun, K.Agha and A.Zouaou, " Graph Neural Networks for Intrusion Detection: A Survey," IEEE Access, vol.PP, pp. 1-1, 10.1109/ACCESS.2023.3275789,2023.

[10] A.Bhardwaj, R.Tyagi, N.Sharma, Akhilendra, M.Punia and V.Garg, " Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework," Measurement: Sensors, vol.24, pp.100580,10.1016/j.measen.2022.100580,2022.

[11] Z. Chiba, N.Abghour, K.Moussaid , A.omri and M.Rida, "A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort and Optimized Back Propagation Neural Network," Procedia Computer Science,vol.83,pp.1200-1206. 10.1016/j.procs.2016.04.249, 2016.

[12] P.Jaina , P.Krunal, and M.Sameer, "Effective Intrusion Detection System using Data Mining Technique," SSRN Electronic Journal, vol. 2,pp.1869-1879, 2015.

[13] A.-D.Tudosi , "A Python-Based Approach for Monitoring and Troubleshooting Snort IDS in Distributed Firewall Environments," International Conference on Advanced Scientific Computing (ICASC), pp. 1-5, 10.1109/ICASC58845.2023.10328028,2023.

[14] S.Pavithra and P.Durgadevi, "An Approach for Network Based Intrusion Detection System using Snort," Data Science and Intelligent Computing Techniques, pp. 481-49,10.56155/978-81-955020-2-8-44,2023.

[15] M.Stephen , A.Olaniyi and O.Victor , "Analysis of Digital Forensics in the Implementation of Intrusion Detection using Snort," FUOYE Journal of Pure and Applied Sciences (FJPAS), vol. 7, pp.100-107, 10.55518/fjpas.IJMS6335,2022.

[16] K.Nilesh , T.Ritu and D.Joydip, "Network Packet Analysis in Real Time Traffic and Study of Snort IDS During the Variants of DoS Attacks," pp. 362-375 ,10.1007/978-3-030-49336-3_36,2021.

[17] A. A. E. Boukebous, M. I. Fettache, G. Bendiab and S. Shiaeles, "A Comparative Analysis of Snort 3 and Suricata," 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), pp. 1-6, 10.1109/GlobConET56651.2023.10150141,2023.

[18] N. I. Andrey and E. B. Michael, "Realization of Expert Intrusion Detection System Based on the Results of Datasets and Machine Learning Algorithm Analysis," CASPIAN JOURNAL.Management and High Technologies, vol.2, pp.100-107, 2020.

[19] S.Syed and I.Biju, "Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System," Future Generation Computer Systems, vol.80, pp.157-170,10.1016/j.future.2017.10.016, 2018.

[20] O.Elaeraj and L.Cherkaoui, "Toward an Appropriate Approach for Intelligent Intrusion and Detection Systems," Journal of Artificial Intelligence & Cloud Computing, pp. 1-10, 10.47363/JAICC/2022(1)109, 2022.