IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 6, December, 2023, pp. 1629-1635
Journal homepage: http://iieta.org/journals/ria

# Intelligent Intrusion Detection System Snort and SVM

Ouafae El Aeraj*, Cherkaoui Leghris

LMIA, RTM Team, Faculty of Sciences and Techniques Mohammedia, Hassan II University of Casablanca, Mohammedia 28806, Morocco

Corresponding Author Email: ouafaeelaeraj@gmail.com

https://doi.org/10.18280/ria.370627

**Received:** 1 August 2023
**Revised:** 22 October 2023
**Accepted:** 29 November 2023
**Available online:** 27 December 2023

***Keywords:***
*intrusion detection system, machine learning, snort, support vector machines*

## ABSTRACT

Despite significant advances in IT security, current solutions fail to guarantee protection against malicious threats, often consisting of subtle and potentially damaging variants. To counter these risks, it remains essential to adopt robust security policies and devices such as firewalls and intrusion detection systems. However, these systems have their drawbacks, not least the propensity to generate false positives, leading to erroneous alerts and compromising the overall effectiveness of the security system. Faced with these challenges, an innovative approach was adopted, making use of machine learning, in particular support vector machines (SVM) written in Python programming language, in conjunction with the Snort IDS. This approach exploits the Snort IDS traffic training dataset, identifying attacks such as denial of service using alarm-generating rules. The data is then converted to a usable format and used as input for the machine learning model. This model separates the data into training and test sets in order to evaluate performance, using metrics such as F1 score, precision and recall. The results of this study demonstrate exceptional performance, with a precision rate of 99%, a true positive rate of 162, a false positive rate of 1, a true negative rate of 160 and a false negative rate of zero. These results highlight the robustness of the proposed approach, positioning it favorably in relation compared to other intrusion detection techniques.

## 1. INTRODUCTION

Companies are facing a steady increase in security threats due to the diversification and growing sophistication of attacks. Cyber attacks, organized crime and other risks have serious consequences, ranging from the loss of sensitive data to the disruption of business operations, reputational damage and financial losses. Companies need to protect their assets, employees and reputation from these ever-changing threats. Current solutions, such as intrusion detection systems (IDS), advanced firewalls, antivirus, and data encryption solutions, are designed to monitor and detect suspicious activity on computer networks. IDSs rely on predefined signatures, abnormal behavior and anomalies in network traffic to identify possible intrusions in real time when an unauthorized access attempt is made or a security breach is detected.

There are two categories of IDS:

**Network-based IDS (NIDS):** A network-based intrusion detection system is a computer security tool that monitors network traffic to identify and respond to malicious or suspicious activity. NIDS look for patterns of behavior related to known attack signatures in data packets, as shown in Figure 1. They can also use behavioral analysis techniques to identify abnormal activity that could indicate an intrusion attempt. The NIDS issues alerts when a threat is identified, which can trigger automatic responses or require human intervention.

The advantages of NIDS lie in their ability to provide an overview of network traffic and identify potential threats before they reach specific systems. However, attacks, particularly those using encryption, are becoming increasingly complex and require ongoing maintenance to remain effective in the face of evolving threats. Despite these difficulties, NIDS continue to be an essential part of network security strategies to protect organizations against cyber-attacks.

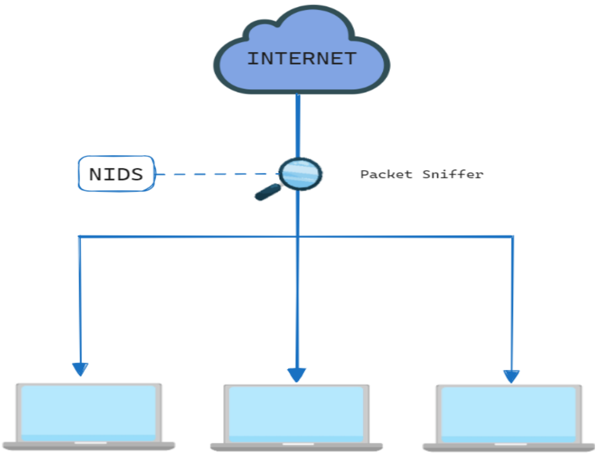

**Figure 1.** Deploying a NIDS on a network

**Host-based IDS (HIDS):** A host-based intrusion detection system is a computer security component that monitors and analyzes activity on an individual computer or device, as shown in Figure 2. These systems aim to detect suspicious behavior, unauthorized modifications or signs of potential intrusion. HIDS trigger alerts to inform security administrators if a threat is detected. Although HIDS offer an in-depth view of activity on a particular device, they can be limited by the need to keep detection rules up to date, and by their inability to monitor threats that do not directly affect the host system. Despite these limitations, HIDS continues to be a crucial element of IT system security, complementing other security measures to strengthen an organization's overall security posture.

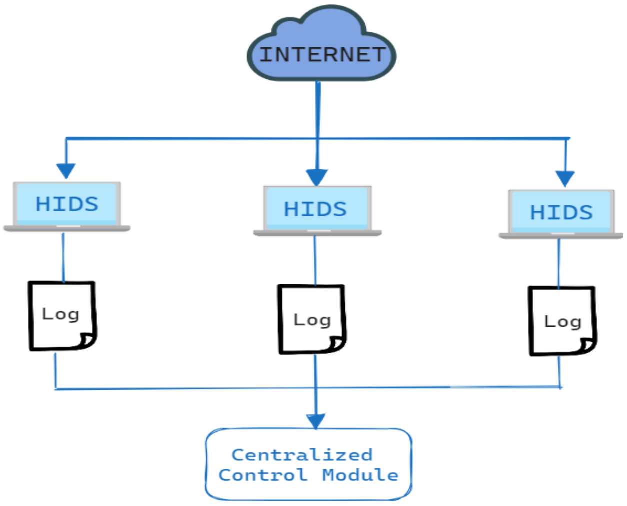

**Figure 2.** Deploying a HIDS on a network

IDS systems use various techniques to detect potential threats. The most common techniques are:

**Signature-Based Detection:** a fundamental technique of intrusion detection systems (IDS), based on the identification of specific patterns of known attacks. In this approach, signatures, or fingerprints, are created by analyzing the distinctive characteristics of previously documented attacks. When network traffic or system activities are inspected, the system compares these signatures with real-time data to detect matches. This method is particularly effective in detecting well-established and widespread attacks. However, it has significant limitations, as it cannot identify new or modified attacks that do not match existing signatures. Attackers can circumvent this type of detection by using innovative tactics or by slightly modifying the characteristics of known attacks.

**Anomaly-Based Detection:** a key technique of intrusion detection systems (IDS), focuses on the identification of novel or deviant activities in relation to the normal behavior patterns of a system or network. Unlike signature-based detection, it does not use pre-established patterns of known attacks, but is based on the creation of a profile of habitual behavior. Anomaly-based IDSs analyze various metrics, such as network traffic and access patterns, to define what is considered normal. When activity becomes atypical, the IDS triggers an alert. While effective in spotting novel attacks and subtle malicious behavior, this method can generate false positives by reporting legitimate activity that is merely unusual. What's more, it requires a learning curve to establish normal behavior profiles, and must be constantly adapted to keep pace with evolving network environments and threats.

**Heuristic-Based Detection:** relies on heuristic rules rather than specific signatures to identify suspicious activity. Unlike signature-based detection, which requires an exact match with known patterns, heuristic detection uses logic rules and suspected malicious behavior to spot anomalous activity. This approach enables heuristic IDSs to detect unknown attacks based on generic models of malicious behavior. However, it can generate false positives, as legitimate activities may correspond to the heuristic rules defined. Heuristic rules, often based on the experience of security experts, offer the flexibility to adapt detection to new or emerging threats.

**Behavior-Based Detection:** Behavior-based detection is an innovative approach to intrusion detection systems (IDS) that focuses on identifying malicious activity by analyzing the behavioral patterns of users, applications and the network. Rather than relying on specific signatures, this method looks for anomalies or variations from normal patterns of behavior. By examining interactions at a deeper level, it enables the detection of sophisticated attacks and insider threats that might otherwise escape detection. The drawback of behavior-based detection lies in the difficulty of quickly distinguishing new, legitimate behavior from malicious activity, which can lead to high false-positive rates or slow adaptation to changes in the threat landscape.

Although IDSs are essential to enterprise security, they have some important limitations in the context of today's security threats. Firstly, they can generate a large number of false positives, which can lead to an additional workload for system administrators. In addition, IDSs can slow down the corporate network by analyzing every packet of data, which can affect performance. Finally, IDSs cannot detect all attacks and can be circumvented by experienced attackers. This is why we decided to add the SVM algorithm to the IDS to improve network security by overcoming the IDS's shortcomings, in other words, by minimizing false positives and improving detection precision.

Support vector machines is a gadget mastering a set of rules used for category and regression analysis. SVM seeks to identify the unusual hyperplane that divides the records into one-of-a-kind classes. The hyperplane is selected to maximize the space among the hyperplane and the closest records factors of every class. These nearest records factors are referred to as aid vectors. SVMs are especially beneficial whilst coping with high-dimensional records. It has packages in numerous fields including photo category, bioinformatics, and textual content category.

The SVM method can be used in network security to classify network traffic data. The goal is to categorize network traffic data into different groups, such as "normal" or "abnormal", "safe" or "dangerous", or "intruder" or "non-intruder".

The system suggested in this paper uses machine learning to improve intrusion detection, and more specifically uses SVM. The rest of the paper is organized as follows: Several recent research that employ SVMs in IDS are briefly mentioned in Section 2. In Section 3, a model of an intelligent system is created using a combination of the Snort intrusion detection approach and SVM machine learning algorithms, and Section 4 is where the outcomes are displayed, while the conclusion and some viewpoints are presented in Section 5.

## 2. RELATED WORKS

To more effectively identify and address security threats, SVMs are interoperable with firewalls, security technologies include intrusion prevention systems (IPS), intrusion detection systems (IDS), and others. Several researchers have proposed various approaches and models for securing the network infrastructure. Including the use of Snort and SVM [1, 2]. According to the findings, both the false positive and false negative rates have significantly decreased.

An optimization approach for IDS selection in the context of network communication security [3] could help users make more informed decisions on IDS selection to protect their networks from malicious attacks; based on the experience, Suricata is the best IDS with high potential. To increase the intrusion detection system's effectiveness, an anomaly detection method known as outlier detection was used, which involved measuring the aberrant dataset by the Neighborhood Outlier Factor (NOF) [4] using huge datasets stored in a distributed storage environment.

A rule-based "Snort" system with machine learning categorization [5] is effective at lowering the rule-based NIDS's false positive and false negative rates, according to experimental data. In their analysis of Snort's design, Shuai and Li [6] suggested a solution to lower the rate of false negatives for high-speed arranging activity. They operate their Snort DAQ engine based on DPDK's high-performance package preparation technology to speed up the execution of Snort's packet capture engine. Test results demonstrate that following optimization, Snort's package capture and harmful activity location rates in high-speed arrange activity are significantly improved.

To fix Snort's performance concerns and enhance the standard Snort intrusion detection system, Zhang and Wang [7] suggested a DPDK-based intrusion detection system. It has been suggested that a new parallel Snort architecture be used along with some adjustments based on Gupta and Sharma [8], which would enhance Snort's performance and reduce the amount of lost packets.

Hadem et al. [9] is an SDN-based intrusion detection system, which, for IP tracing, employed support vector machines (SVMs) and selective logging, on the NSL-KDD dataset, with detection precision of 95.98% and 87.74%, respectively.

Bhati and Rai [10] offered an analytical analysis of intruder detection methods, with support vector machines (SVM) serving as their foundation. Support vector machines (SVM) serve as their foundation. The receiver operating parameters, confusion matrix, and overall detection precision of the results were all discussed. Using the NSL-KDD dataset, the SVM technique's effectiveness is evaluated.

The development of a distributed denial of service (DDoS) assault model utilizing a combination of SVM classification techniques, the extraction of characteristic values from six sets of switched traffic tables, and the construction of a Mininet and SDN environment projection simulation platform is all covered in Ye et al. [11]. The experiments show an average precision rate of 95.24%.

To find the optimal kernel for SVM, Hasan et al. [12] suggest using the different kernels for the NSL-KDD and KDD'99 datasets. By using the RRE-KDD dataset, the superfluous records from KDD'99 are removed. The RRE-KDD and NSL-KDD dataset’s kernels outperformed other kernels in terms of precision, enhancing detection rates while reducing false positive rates.

The three machine learning algorithms Nave Bayes (NB), Support Vector Machine (SVM), and K-nearest neighbor (KNN) utilizing the UNSW-NB15 dataset were examined in Agarwal et al. [13], with the goal of improving algorithm performance and finding the best algorithm for quickly learning the pattern of suspicious network activity. The IDS was used to evaluate the feature sets as input data for the system's training to anticipate and analyze future intrusion behaviors by selecting the top algorithm from the three mentioned above based on performance criteria.

The intrusion detection study described in Kasongo and Sun [14] utilized the UNSW-NB15 dataset, which used the XGBoost algorithm and filter-based feature reduction techniques. The outcomes demonstrate that the feature selection approach based on XGBoost is robust to tools such as decision trees (DT) and improves the test precision of binary classification schemes from 88.13% to 90.85%.

In the study [15], a cutting-edge classification method based on reinforcement vector machines (SVM) and cross-entropy was proposed. The 7 tuples cross-entropy inclusion vectors are used to prepare the multiclass SVM classifier. The findings demonstrate that the recommended classifier is more appropriate for usage in the controlled organization than conventional discovery procedures and can attain critical discovery rates.

Mulay et al. [16] also suggested a decision tree-based method for developing a multiclass intrusion detection system. The final results demonstrate that binary tree-based SVMs can be used to handle multi-class pattern recognition issues, and the resulting intrusion detection system may be quicker than those produced by other techniques. SVMs are used as a potent tool for the classification or genomic subtyping of cancer in the medical area [17] in addition to being exploited in the field of computer security, and it is undeniable that they have also yielded better results in speech emotion recognition [18], and human action recognition [19].

The importance of choosing the right machine learning model for intrusion detection systems was looked at in Mohammed and Hussein [20]. When choosing and implementing machine learning models in intrusion detection systems, the results of performance benchmarking are an invaluable resource. By locating the top-performing models for this crucial network security task, this research helps to increase the precision and effectiveness of intrusion detection systems.

In order to increase the detection of abnormal activity, Runwal [21] emphasizes the significance of utilizing machine learning approaches in intrusion detection systems. By detecting unknown and emerging intrusions, the anomaly-based approach and machine learning present promising potential for enhancing network security. The results show that suspicious behavior and unknown intrusions can be effectively detected using an anomaly-based approach and machine learning.

The importance of choosing the right machine-learning technique was presented in Laqtib et al. [22] for intrusion detection systems in MANETs. Careful selection of learning algorithms can improve intrusion detection precision while taking into account the specific constraints of mobile ad hoc networks. This technical review provides useful information for researchers and practitioners interested in developing effective intrusion detection systems in MANETs.

To improve attack detection, a hybrid intrusion detection system described in Amar and El Ouahidi [23] combines signature-based and machine learning techniques. Experimental results indicate that this combination of approaches offers improved detection precision, particularly for new or unknown attacks. By exploiting the complementary advantages of signature-based approaches and machine learning, this research opens up interesting prospects for improving intrusion detection systems.

The importance of implementing an intrusion detection system in network security [24] and demonstrates the effectiveness of using Snort and Snort community rules to detect different types of network attacks. This research provides a solid basis for implementing intrusion detection systems based on Snort and Snort community rules in real network environments to enhance security and protection against attacks.

The significance of the empirical evaluation of intrusion detection systems was highlighted by a comparison of Snort NIDS with supervised machine learning classifiers [25], which also emphasized the benefits and drawbacks of each method for detecting network intrusions. This comparison was helpful in guiding the choice and implementation of these techniques.

## 3. METHOD

Many researchers are concerned with improving intrusion detection performance, specifically reducing false positives and improving detection precision using machine learning. The aim of the proposed model is to reduce false positives and false negatives and improve detection precision using Snort IDS and the SVM algorithm. Snort was selected on the basis of the conclusions drawn from the article [2]. The latter looks at a comparative evaluation of the effectiveness of two open source intrusion detection systems (IDS), namely Snort and Suricata, in detecting malicious network traffic. Suricata used computing resources beyond the limits specified in the study, demonstrating its ability to handle faster network traffic than Snort. Because of Snort's superior detection precision compared with Suricata, and the lower rate of false positive alerts displayed by the latter, Snort was preferred in this study.

To identify suspicious behavior or known security breaches, Snort continuously monitors network traffic. Snort's operation can be summarized in four main stages: Packet Capture, Packet Inspection, Alert Generation and Logging, where Snort can record a dataset containing alerts and detected events in a log file.

Because of its ability to handle complex, non-linear datasets, the SVM supervised learning algorithm is used in our proposed model. SVMs work by dividing data into classes using an optimal hyperplane, maximizing the margin between different categories. This method is particularly useful for anomaly detection and classification of malicious behavior, as it can efficiently handle high-dimensional data and accommodate the non-linear patterns present in malicious activity. SVMs are also resistant to unbalanced data sets, which are common when intrusions are detected.

Security systems can improve their ability to identify and respond to emerging computer threats more robustly by combining the power of SVM with other advanced techniques, such as feature extraction and behavioral analysis.

The proposed model consists of five stages, as illustrated in Figure 3. The training dataset on Snort IDS traffic and the subsequent creation of the SVM model is to improve network intrusion detection capability by exploiting network traffic features, such as attack signatures, to reduce false positives and false negatives generated by the IDS and identify potential malicious behavior. A detailed explanation of each step and overall operation is provided below.

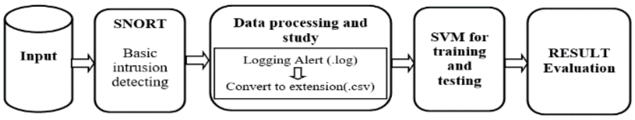
Input
SNORT
Basic intrusion detecting
Data processing and study
Logging Alert (.log)
Convert to extension(.csv)
SVM for training and testing
RESULT Evaluation

**Figure 3.** The architecture of the proposed IDS

### 3.1 Input

To generate both valid and harmful online traffic, this is the initial step, consisting of a mixture of malicious traffic, such as TCP-SYN floods, distributed denial of service (DDOS), and large login records. A DDOS attack is a cybersecurity weapon designed to disrupt services or extort money from a targeted organization. A malicious attempt to overload a Web site with traffic to obstruct it from operating normally is known as a DOS attack, while a TCP-SYN flood is a network saturation (denial of service) attack that uses the handle mechanism in the three stages of the TCP protocol.

### 3.2 Basic intrusion detecting

Snort configuration files are used to set intrusion detection rules, communication protocols, network IP addresses, ports to be monitored, etc. The most important configuration files are:

- **snort.conf**: This is the main configuration file for Snort.
- **threshold.conf**: This file is used to specify alert thresholds for intrusion detection events.
- **sid-msg. map** and **gen-msg.map**: These files are used to map rule identifiers (SIDs) to corresponding alerts.
- **Rules files**: Rules files contain intrusion detection rules that define suspicious behavior to monitor.

These files can be modified to add or remove rules according to the user's needs. When incoming traffic meets the set of rules, Snort inspects it and triggers alarms. We are going to modify the snort.conf file to configure snort. The first section of the file, which discusses network variables, has just been changed.

Below is a list of the changes we made to this file:

- The network interface's IP that monitors traffic is provided by the var HOME_NET any/*.

Any is the default selection. Using the network's or interface's IP address, we can protect it and personalize it.

- The list of external networks to listen to is given by the var EXTERNAL_NET any/*.

Any is the default value, which means that all network traffic is inspected. We substitute !HOME_NET for any to omit the network that requires security. Alternatives include using [network address1, network address2, ...] to indicate the networks.

- **var RULE_PATH:** "/etc/snort/rules" Below is a list of the .rules file directory.

There are two sections to the snort rules:

- The header's core filtering fields protocol, source and destination IP addresses, and ports give you the option to select the type of alert (alarm, log, or pass).
- The options advance the research by dissecting the signature into various values that can be examined from the relevant header or data fields.

When incoming traffic complies with the set of rules, Snort raises alarms after inspecting both legitimate and malicious traffic. We have two guidelines for our approach:

Alert ICMP any any -> any any (msg: "ICMP test"; sid: 100000;)

Alert tcp any any -> any any (msg: "Alert FTP"; sid: 100006927; rev: 005;)

When any of these TCP-SYN floods, Denial of Service (DOS), or Distributed Denial of Service (DDOS) assaults are identified, these rules generate alarms.

### 3.3 Data processing and study

Snort is responsible for generating alerts based on the ruleset. All log files are kept in the second output phase. Preprocessing is a method for converting unstructured data into a usable format. In our case, to provide data for the SVM method, we transform the **.log** output file into a new **.csv** format.

### 3.4 SVM for training and testing

In our proposed system, we use this Snort log file as an input for our machine learning model, which is built on a set of support vectors intended to decrease false alarms.

3.4.1 Data loading and processing

The proposed system is implemented using Python programming language, and then we imported the dataset based on the converted output of Snort. In order to simplify classification and prepare the dataset for our algorithm, we label each column with a date, time, source address, destination address, source port, destination port, and protocol. The source address, destination address, and destination port number serve as the foundation for the study.

3.4.2 Splitting data

We will separate the data into training and test sets in order to evaluate the performance of the model. The training data will be used to create the SVM model, while the test data will be used to evaluate the model's performance.

3.4.3 Generation model

To create a model using a support vector machine. Before building the support vector classifier object, we first introduce the SVM module.

One of the key components is the kernel, a function that converts data into a specific representation. SVMs use a variety of kernel functions, including sigmoid, radial basis function (RBF), polynomial, linear, and non-linear.

The formula (1) represents the modeling of the RBF kernel for the SVM classification algorithm in Sklearn:

$$K(x, x')=e^{-\gamma \|x-x'\|^2} \tag{1}$$

Gamma must be larger than zero and can be manually set. The Sklearn SVM (2) classification method's default value for gamma is:

$$\gamma=\frac{1}{\text{nfeatures}*\sigma^2} \tag{2}$$

$\|x\text{-}x'\|^2$ The Euclidean distance of 2 squared separates the two feature vectors. The influence of a single training sample is represented by a scalar called gamma. The above variables allow us to regulate the amount that each point affects the algorithm as a whole. The Gamma must be greater as the other points approach the model closer.

We will use an RBF kernel for our case using the following parameters: random state=1, gamma=0.05, and C=0.1. Because the decision surface is so straightforward, the parameter C, which is shared by all SVM kernels, corrects for misclassification mistakes in the training instances. A low C guarantees a smooth decision surface, whereas a high C guarantees accurate classification of every training case.

3.4.4 Evolution model

The SVC class fitting method involves training the algorithm with the given parameters of the training data and evaluating the model's effectiveness using factors such as F1 score, precision, and recall.

Precision: It enables us to determine how many accurate, positive forecasts have been made. In other terms, it is the sum of all correctly predicted positive outcomes (True positive + False positive) divided by the number of correctly predicted positive outcomes:

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

Recall: This enables us to determine the proportion of positives that our model accurately anticipated. Otherwise put, it is the number of well-predicted positives (True positives) split by all the positives (True positives + False negatives). In mathematical form, we have:

$$\text{Recall}=TP/(TP+FN)$$

F1 score: A metric used to assess the effectiveness of classification models with two or more classes is the F1 score. It is particularly used for problems using unbalanced data, such as fraud detection or serious incident prediction. The precision and recall values are combined into one metric by the F1 score. The mathematical definition of the following equation is used to represent the F1 score, which is the harmonic mean of recall and precision:

$$\text{F1 score}=2*(\text{Recall}*\text{Precision})/(\text{Recall}+\text{Precision})$$

## 4. RESULTS AND DISCUSSION

False positives, or incorrect alerts, can lead to security operator fatigue and wasted resources investigating false threats, which can impair system operational efficiency. On the other hand, false negatives, or failure to detect a threat, can leave genuine vulnerabilities undetected, exposing the infrastructure to security risks. In a real-life context, false negatives can allow attacks to go undetected, compromising system integrity, confidentiality and availability. Thus, balancing false positive and false negative rates is essential to ensure effective use of the intrusion detection system, and to maintain an appropriate response to threats while minimizing undesirable operational impacts.

The robustness of the model is demonstrated by its high performance when evaluated on the simulated dataset. Using the SVM classifier in the context of intrusion detection shows 99% precision on DDOS, DOS and TCP-SYN attacks, indicating a high ability to distinguish between malicious and normal activity. A true-positive rate of 162 and a false-positive rate of 1 enable attacks to be correctly identified, while minimizing incorrect alerts. False-negative-free performance (false-negative rate of zero) also indicates that the model succeeded in detecting all real attacks in the dataset. These results testify to the model's resilience in the face of noise in the data and the presence of attacks, demonstrating its effectiveness in intrusion detection. Experiments were carried out on the simulated dataset, and the attacks detected by Snort over 4 consecutive days were converted into a file, which served as input to the SVM machine learning classifier using Python programming that is based on various features such as source and destination IP addresses and ports for optimal intrusion detection performance. Performance is evaluated on the basis of validation parameters: precision, recall and F1 score.

Bhati and Rai [10], who evaluated the SVM methodology with NSL-KDD, compared two intrusion detection systems that use support vector machines (SVM). However, our suggested method performs better. The dataset's success results in a maximum detection precision of 98.5%, or compared with the study [26], which proposed an intrusion detection model using machine classifier support vectors on the Apache Spark Big Data platform with 94% precision. Finally, our proposed approach leads us to the conclusion that SVMs offer more precision on the imported Snort dataset, as shown by the comparison in Figure 4.

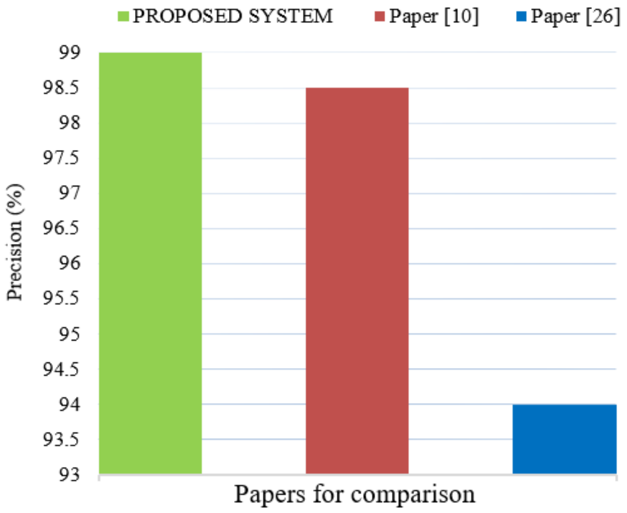

**Figure 4.** Comparison of the work according to the precision

## 5. CONCLUSIONS

The inability to constantly guarantee the proper functioning of computer network security mechanisms, with the emphasis on the importance of an effective intrusion detection system.

The paper suggests the use of all Snort detection system data and the integration of an SVM machine learning classifier, which is realized using the Python programming language. The results of the experiment show that this fusion has a significant effect, with a precision rate of 99%, a true positive rate of 162 and a false positive rate of 1, enabling attacks to be correctly identified while minimizing incorrect alerts and a false negative-free performance. It seems that this method is a promising way of strengthening computer network security.

Using advanced data mining and machine learning techniques, future research prospects are geared towards exploring new and potential data breach attacks. This orientation is in line with the need to stay at the forefront of emerging threats, taking into account the constant evolution of the methods used by attackers.

## REFERENCES

[1] Shah, S., Issac, B., Jacob, S. (2018). Intelligent intrusion detection system through combined and optimized machine learning. International Journal of Computational Intelligence and Applications, 17: 1469-0268. https://doi.org/10.1142/S1469026818500074

[2] Shah, S., Issac, B. (2018). Performance comparison of intrusion detection systems and application of machine learning to snort system. Future Generation Computer Systems, 80: 157-170. https://doi.org/10.1016/j.future.2017.10.016

[3] Abdel-Basset, M., Gamal, A., Sallam, K. , Elgendi, I., Munasinghe, K., Jamalipour, A. (2022). An optimization model for appraising intrusion-detection systems for network security communications: Applications, challenges, and solutions. Sensors, 22(11): 4123. https://doi.org/10.3390/s22114123

[4] Jabez, J., Muthukumar, B. (2015). Intrusion detection system (IDS): Anomaly detection using outlier detection approach. Procedia Computer Science, 48: 338-346, https://doi.org/10.1016/j.procs.2015.04.191

[5] Aslam, U., Batool, E., Ahsan, S., Sultan, A. (2017). Hybrid network intrusion detection system using machine learning classification and rule based learning system. International Journal of Grid and Distributed Computing, 10(2): 51-62. http://dx.doi.org/10.14257/ijgdc.2017.10.2.05

[6] Shuai, L., Li, S. (2021). Performance optimization of Snort based on DPDK and Hyperscan. Procedia Computer Science, 183(2): 837-843. https://doi.org/10.1016/j.procs.2021.03.007

[7] Zhang, D., Wang, S. (2019). Optimization of traditional Snort intrusion detection system. IOP Conference Series: Materials Science and Engineering, 569(4): 042041. http://doi.org/10.1088/1757-899X/569/4/042041

[8] Gupta, A., Sharma, L. (2020). A categorical survey of state-of-the-art intrusion detection system-Snort. International Journal of Information and Computer Security, 13(3/4): 337-356. https://doi.org/10.1504/IJICS.2020.109481

[9] Hadem, P., Saikia, D., Moulik, S. (2021). An SDN-based intrusion detection system using SVM with selective logging for IP Traceback. Computer Networks, 191: 1389-1286. https://doi.org/10.1016/j.comnet.2021.108015

[10] Bhati, B., Rai, C. (2018). Analysis of support vector machine-based intrusion detection techniques. Arabian Journal for Science and Engineering, 45(04): 1–8. http://doi.org/10.1007/s13369-019-03970-z

[11] Ye, J., Cheng, X., Zhu, J., Feng, L., Song, L. (2018). A DDoS attack detection method based on SVM in Software defined network. Security and Communication Networks, 2018(4): 1-8. https://doi.org/10.1155/2018/9804061

[12] Hasan, M.A.M., Xu, S., Kabir, M.M.J., Ahmad, S. (2016). Performance evaluation of different kernels for support vector machine used in intrusion detection system. International Journal of Computer Networks &

Communications, 8(6): 39-53. http://doi.org/10.5121/ijcnc.2016.8604

[13] Agarwal, A., Sharma, P., Alshehri, M., Mohamed, A., Alfarraj, O. (2021). Classification model for accuracy and intrusion detection using machine learning approach. PeerJ Computer Science, 7(3): e437. http://doi.org/10.7717/peerj-cs.437

[14] Kasongo, S.M., Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7: 1-20. https://doi.org/10.1186/s40537-020-00379-6

[15] Han, W., Xue, J., Yan, H. (2019). Detecting anomalous traffic in the controlled network based on cross entropy and support vector machine. IET Information Security, 13: 109-116.http://doi.org/10.1049/iet-ifs.2018.5186

[16] Mulay, S., Devale, P.R., Garje, G. (2010). Intrusion detection system using support vector machine and decision tree. International Journal of Computer Applications, 3(3): 40-43. http://doi.org/10.5120/758-993

[17] Huang, S., Cai, N., Pacheco, P., Narrandes, S., Wang, Y., Xu, W. (2018). Applications of support vector machine (SVM) learning in cancer genomics. Cancer Genomics & Proteomics, 15(1): 41-51. http://doi.org/10.21873/cgp.20063

[18] Lin, Y., Wei, G. (2005). Speech emotion recognition based on HMM and SVM. International Conference on Machine Learning and Cybernetics, 8: 4898-4901. http://doi.org/10.1109/ICMLC.2005.1527805

[19] Schuldt, C., Laptev, I., Caputo, B. (2004). Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, pp. 32-36. http://doi.org/10.1109/ICPR.2004.1334462

[20] Mohammed, S., Hussein, M. (2022). Performance analysis of different machine learning models for intrusion detection systems. Journal of Engineering, 28(5): 61-91. http://doi.org/10.31026/j.eng.2022.05.05

[21] Runwal, A. (2021). Anomaly based intrusion detection system using machine learning. International Journal for Research in Applied Science and Engineering Technology, 9: 255-260. http://doi.org/10.22214/ijraset.2021.37955

[22] Laqtib, S., El Yassini, K., Hasnaoui, M. (2020). A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET. International Journal of Electrical and Computer Engineering (IJECE), 10(3): 2701-2709. http://doi.org/10.11591/ijece.v10i3.pp2701-2709

[23] Amar, M., El Ouahidi, B. (2020). Hybrid intrusion detection system using machine learning. Network Security, 2020(5): 8-19. https://doi.org/10.1016/S1353-4858(20)30056-8

[24] Widodo, T., Aji, A. (2021). Implementation of intrusion detection system (IDS) and snort community rules to detect types of network attacks. International Journal of Computer Applications, 183(42): 30-35. http://doi.org/10.5120/ijca2021921821

[25] Abdulrezzak, S., Sabir, F. (2023). An empirical investigation on snort NIDS versus supervised machine learning classifiers. Journal of Engineering, 29(2): 164-178. http://doi.org/10.31026/j.eng.2023.02.11

[26] Othman, S., Ba-Alwi, F., Nabeel, T., Al-Hashida, A. (2018). Intrusion detection model using machine learning algorithm on big data environment. Journal of Big Data, 5: 1-12. https://doi.org/10.1186/s40537-018-0145-4