Journal of Artificial Intelligence and Technology, 2024, 4, 332-341
https://doi.org/10.37965/jait.2024.0554

ISTP
RESEARCH ARTICLE

Intrusion Detection System Based on an Intelligent Multilayer Model Using Machine Learning

Ouafae El Aeraj and Cherkaoui Leghris
Laboratory of Mathematics, Computer Science and Applications, Faculty of Sciences and Techniques, Hassan II University of Casablanca, Mohammedia 28806, Morocco

(Received 08 May 2024; Revised 11 July 2024; Accepted 21 July 2024; Published online 22 August 2024)

Abstract: With the rapid advent of information technology and social networking, the multiplication of connected devices further exposes users to the vulnerability of their personal data. This growing interconnectedness increases the risk of online attacks, underlining the daily challenge of cybersecurity in the face of increasingly sophisticated attacks. Flaws in automatic software updates and the limited responsiveness of devices underline the need for an innovative approach to detecting intrusions and securing systems. Early detection of intrusions within companies is essential to anticipate threats and respond rapidly to incidents. Researchers recommend the use of several tools and methods to counter malicious activity. This article introduces our innovative development of an automated model called Snort Support Vector Machine (SSVM) based on a hierarchical strategy organized in distinct layers. This model, automated by the joint use of Python and Shell, merges the efficiency of these languages to guarantee autonomous and resilient operation. After examining several intrusion detection and prevention systems, the first layer implements a selected system as the initial basis. The second layer uses machine learning to fill in the gaps in the initial system. Finally, the third layer applies a global evaluation methodology, taking into account execution time, energy consumption, and physical resources in order to orchestrate the entire evaluation process. The approach we propose appears to improve on other conventional intrusion detection systems by making the detection process more efficient. It does this by reducing false positives and false negatives compared with existing models.

Keywords: intrusion detection system; support vector machines; snort; machine learning

I. INTRODUCTION

IT network security is an essential foundation for protecting information and infrastructure in our interconnected digital age. It encompasses various strategies and methodologies designed to prevent, identify, and counter threats to networks. Essential components include firewalls, intrusion detection systems (IDSs), encryption mechanisms, and access control best practices. Even with robust IT security measures, it remains possible for skilled attackers to deploy advanced or refined strategies to compromise a system. Hence, the importance of integrating an intrusion detection layer is to complement existing prevention measures. This is the context in which IDS was conceived, as it keeps track of attacks against the system and prevents the system from being subjected to these attacks [1], which could jeopardize fundamental IT security principles such as confidentiality, integrity, and availability.

IDS design is based on principles established as early as 1980 by James P. Anderson [2], who envisioned the development of security policies, threat models, and the evolution of IDSs to protect computer networks against unauthorized access and cyber threats. An IDS can be defined as a software or hardware device designed to detect malicious activity within computer systems, thereby helping to maintain system security [3]. The aim of IDSs is to identify the various types of threats and malicious uses that cannot be intercepted by a conventional firewall [4].

However, despite their value, the majority of IDSs still come up against two main pitfalls: false positives, which indicate the incorrect alerts generated by the IDS/IPS in relation to the total number of alerts. This gives an idea of how often the system triggers alarms for events that are not real threats [5], and false negatives, which occur when real attacks are not detected, without any alerts being generated. These problems are still present, even in the most modern versions of IDS, indicating that improvements to date have not satisfactorily resolved these issues, despite ongoing research and development efforts in the field of intrusion detection. Cybersecurity research initiatives are now focusing on developing effective IDSs capable of accurately recognizing known and emerging threats, while minimizing false alarms [6].

Recently, there has been growing interest in the use of machine learning (ML) techniques to improve anomaly and abuse detection in IDSs [7, 8], marking a notable trend in security technologies [9]. Various supervised classification algorithms have been tested in this context, including decision trees, Naïve Bayes (NB), K-nearest neighbor (KNN), Tree C4.5, random forest (RF), support vector machine (SVM), and logistic regression (LR) [10]. Thus, fuzzy neural networks inspired by nature have also been explored to improve detection rates and reduce false positives [11].

In a context where cybersecurity is becoming increasingly crucial, this document aims to fill perceived gaps in IDSs. These gaps include the need for proactive detection of emerging threats and the reduction of false positives and false negatives. The objectives of this study are to improve detection precision, reduce threat response time, and optimize resources used. In response to

Corresponding author: Ouafae El Aeraj (e-mail: ouafaeelaeraj@gmail.com)

these challenges, we propose our own innovative multilayer model, the Snort Support Vector Machine (SSVM), which incorporates a hierarchical approach for more effective intrusion detection.

The NIST Cybersecurity Framework is a guide developed by the National Institute of Standards and Technology (NIST) to help organizations improve the management and reduction of cybersecurity risks. It is structured around five key functions: Identify, Protect, Detect, Respond, and Recover; the framework provides a strategic model for managing cybersecurity at all levels of the organization. This strategic model is crucial to understanding how the SSVM model, which we have developed, fits into a holistic and dynamic approach to cybersecurity.

The SSVM model is structured into three strategic layers, and each designed to address specific functions of the NIST framework. The first layer, which captures and preprocesses network traffic, filters potential threats to protect resources, aligning its functions primarily with "Identify" and "Protect." This step is essential to establish a solid security foundation that prevents intrusions before they compromise the network. The second layer of the model uses ML techniques to analyze the data collected, enabling effective detection of abnormal or malicious behavior. This in-depth analysis capability reinforces the framework's "Detect" function, essential for rapid reaction to emerging threats. Finally, the third layer evaluates the overall effectiveness of the detection system by measuring indicators such as detection rate and false positive and negative rates. In doing so, it plays a crucial role in the "Respond" and "Recover" functions, analyzing incident response performance and facilitating the restoration of operations after an attack.

What's more, the SSVM model is designed to be adaptable to a variety of network environments, from traditional infrastructures to cloud-based and hybrid configurations. This flexibility ensures that the model can be easily integrated into different technological contexts, ensuring that the system operates effectively whatever the organization's network architecture. This broad compatibility reinforces the SSVM model's ability to meet diverse cybersecurity needs, making it not only robust but also extremely versatile.

This paper is structured as follows: initially, Section II provides an overview of related work. Section III then provides an in-depth explanation of the automated model. Section IV discloses the results obtained and their analysis. Section V provides a discussion and comparison with other studies. Section VI concludes with a summary of the main points of the paper.

II. RELATED WORK

Building on the fundamental aspects discussed in the introduction, in-depth studies have focused on improving network intrusion detection using advanced ML techniques. These studies looked at various algorithms and their applications to meet the challenges of false positives and false negatives, as mentioned earlier. ML processes raw data and develops diagnostic models [12, 13]. Over the past 20 years, this field has developed rapidly, offering powerful methods and techniques for a wide variety of domains [14,15].

One important piece of research, described in paper [16], explores the use of ML techniques such as SVM, decision tree, Long Short-Term Memory, and RF to detect Distributed Denial of Service (DDoS) attacks directed at The Internet of Things (IoT) devices. The RF model stood out for its superior performance, achieving an accuracy rate of 99.321% and the highest F1 scores compared with other models. The study reveals the increased vulnerability of IoT devices to DDoS attacks and examines the potential of ML, data mining, and Big Data technologies for effective detection. It also proposes a specific detection model that integrates ML and data mining methods, highlighting the effectiveness of the AdaBoost and XGBoost algorithms in identifying DDoS traffic. Finally, the paper highlights the value of developing ML models specifically designed to combat DDoS attacks in IoT networks, proposing an innovative approach based on software-defined networks (SDNs).

The authors of paper [17] demonstrate that decision trees offer better results in intrusion detection. They presented a hybrid intrusion detection method using decision trees for feature selection. This approach improves the efficiency and accuracy of IDSs by combining signature-based and anomaly-based techniques. Results show a detection rate of 97.95% and a significant reduction in the false alarm rate.

Another significant work [18] presents a novel intrusion detection method that relies on ensemble ML to outperform individual detection systems. This research was tested on various recognized datasets, such as KDD99, UNSW-NB15, and CIC-IDS2017, and demonstrated a notable improvement in reducing the false positive rate and increasing accuracy. The study reveals the effectiveness of ensemble classifiers in detecting intrusions and highlights the importance of continuing to develop methods for detecting various types of attack. It proposes a detection system based on ensemble models such as RF, AdaBoost, and Light Gradient Boosting Machine, using a soft voting mechanism to improve accuracy and reduce false positives, achieving up to 99.9% accuracy and a low false positive rate on Network Security Laboratory-Knowledge Discovery in Databases (NSL-KDD) and UNSW-NB15 datasets.

The paper [19] presents M-Multi SVM, an ML-based IDS with an optimized feature selection method to improve detection. The paper details the technical approach underlying M-Multi SVM, highlighting its effective use of SVMs and a refined feature selection process. Experimental results illustrate the system's ability to accurately identify intrusions while minimizing the number of features required. Paper [20] explores the creation of an advanced network IDS, based on a two-stage architecture that integrates KNN and SVM to refine attack classification. The first stage uses KNN to sort incidents into normal, severe, and minor attacks. Then, SVM takes over to further refine, identifying whether severe attacks fall under DoS or Probe and minor ones under U2R or R2L. A distinctive feature of this method is the adoption of Common Correlated Feature Selection (CCFS), aimed at improving the selection of relevant attributes. This approach proceeds by dividing the training data into three distinct segments, with an emphasis on preprocessing to refine the representativeness and statistical normality of network traffic-related features. Tests carried out on the NSL-KDD dataset validate the effectiveness of this method.

In addition, the study [21] examined various ML classifiers using the KDD intrusion dataset, with particular emphasis on performance indicators such as false positive and false negative rates to improve the detection capabilities of IDSs. Although recall and F1 scores were not reported, the RF demonstrated the best performance with a precision rate of 93.77%, as well as the lowest Root Mean Square Error values and false positive rates. The "IDS-ML" open-source initiative [22] provides tools for the development of ML-based IDS, using publicly available network traffic datasets, thus providing a valuable resource for researchers and practitioners wishing to experiment with and deploy innovative IT security solutions.

Liu and Lang [23] examine the revolutionary influence of ML and deep learning techniques on IDS, highlighting their indispensable role for cybersecurity in today's digital age. Faced with the shortcomings of traditional IDSs, particularly in terms of detection

accuracy and reduction of false alarms, ML is emerging as a promising solution, capable of accurately differentiating between normal activities and malicious actions, and recognizing new threats. Deep learning, with its ability to deliver outstanding performance in intrusion detection, is particularly appreciated. This article proposes an innovative classification of IDS studies exploiting ML and deep learning, discussing key algorithms, evaluation criteria, and future challenges, paving the way for significant advances in network defense against cyberattacks. Finally, the SVM model is to perform classification, achieving accuracy, precision, recall, and F-measure scores of 98.6%, 97.4%, 99.7%, and 98.5% respectively.

In their study [24], the researchers address the challenges of intrusion detection in IoT environments, characterized by the variety of attacks and the need for real-time identification. They observe that most studies focus on a narrow spectrum of IoT attacks or fail to provide real-time detection, often due to the use of simple binary classifiers unable to distinguish specific types of IoT attacks. To overcome these obstacles, the paper introduces an advanced IDS that uses an ensemble method based on Lambda architecture, designed to improve detection accuracy and efficiency through Big Data analysis. This approach achieves a remarkable accuracy of over 99.93%, highlighting the importance of data selection and preparation, feature selection, as well as training and detection processes for effective, real-time classification using deep learning techniques, particularly suited to attacks on IoT devices.

Approach [25] introduces a new technique aimed at enhancing intrusion detection in the vast and heterogeneous IoT ecosystem. The method is based on entropy- and set-theoretic-driven feature selection and extraction, exploited on the IoTID20 and NSL-KDD datasets with the application of ML algorithms such as Bagging, Multilayer Perceptron, J48, and IBk. This strategy isolated a set of essential and highly relevant features, resulting in a remarkable classification accuracy of 99.98%, underlining the potential of IDS to secure IoT environments against cyberattacks through optimized feature selection for accurate anomaly identification.

A comparative analysis [26] explores the effectiveness of IDSs using ML approaches in SDNs, highlighting the performance evaluation of different learning algorithms, such as deep learning and SVMs, for reliable intrusion detection. This study highlights the strategic integration of IDS into the adaptable architecture of SDNs, assessing their flexibility in the face of network evolutions, their ability to identify various types of attack, and the related challenges, including the need for large training datasets and minimizing false positives. It also shows how these systems can be optimized to enhance network security, presenting real-life use cases and recommended practices for implementing ML-based IDS in SDNs, to effectively counter a wide range of cyber threats.

Another comparative study [27] evaluates the performance of four ML algorithms for network intrusion detection, namely RF, linear support vector machine (LSVM), Gaussian Naive Bayes (GNB), and LG, using CICIDS-2017 datasets. The confusion matrix of the RF model demonstrates its high accuracy compared to the other algorithms, indicating that the RF model correctly predicted the majority of attacks.

In the context of the significant advances in network intrusion detection using ML algorithms presented in previous work, our model stands out by addressing and resolving many of the weaknesses identified in these studies, while achieving superior precision. While previous research has established a solid foundation for effective intrusion detection, highlighting the crucial role of ensemble methods, deep learning, and judicious feature selection, our model introduces a notable advance. It enriches the field with a novel approach, enhancing detection capabilities in a significantly improved way.

III. METHODS

A. ARCHITECTURE OF THE PROPOSED INTRUSION DETECTION MODEL

Attacks on these platforms are increasing in number and intensity every day, because there is no such thing as an impenetrable computer system.

Implementing a multilayered protection model is essential for defense-in-depth, creating a more secure network environment and limiting opportunities for attackers to exploit it. Our own SSVM model, illustrated in Figure 1, embodies this approach with its three-layer structure dedicated to robustly reinforcing security. The first layer, operating an IDS, is responsible for collecting and analyzing network traffic. It acts as a first line of defense against threats, filtering out suspicious activity as soon as it appears. The second layer uses advanced ML techniques to identify and classify anomalies in the collected data, thus compensating for potential deficiencies in the first detection layer. The third layer evaluates overall system performance by measuring critical parameters such as execution time, energy consumption, and physical resource usage. This model ensures that SSVM operates efficiently while remaining economically viable. Collectively, these layers form a unified system not only capable of countering today's threats but also of adapting to evolving attack strategies, offering dynamic, future-proof protection for targeted networks.

The proposed model, SSVM, is designed to operate in a fully automated fashion, using a script to orchestrate all the tasks associated with the various layers of the IDS. This process begins by periodically launching Snort every five minutes to capture

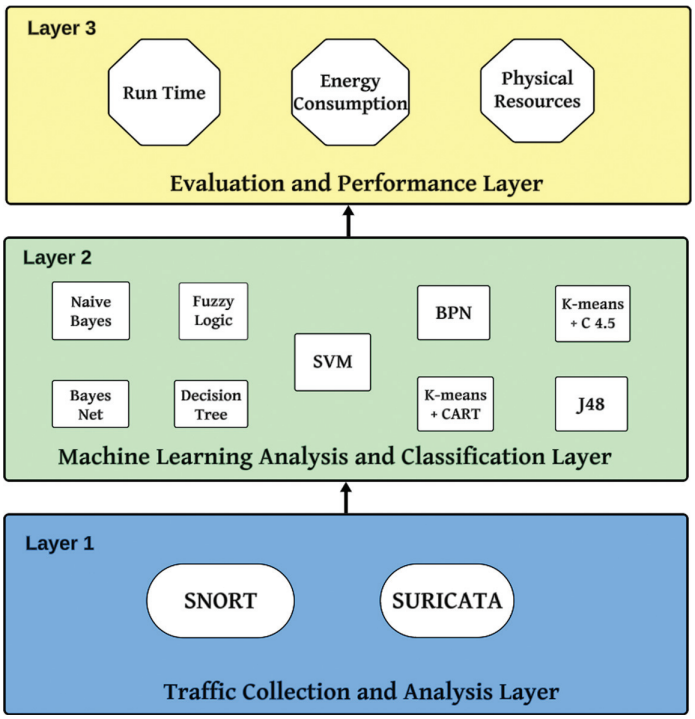

Fig. 1. SSVM multilayer intrusion detection model.

network traffic. The log files generated by Snort are then converted into CSV files, which are particularly suitable for use as input by the selected ML algorithm. This conversion is crucial as it prepares the data for the next analysis, which is the final step in our process and consists of evaluating the model's performance in terms of precision and intrusion detection efficiency.

To train our model, we exploit a dataset extracted directly from the traffic activity captured by Snort, ensuring that the scenarios used for model testing and validation accurately reflect real-life conditions. Our dataset consists of 1,042,455 packets collected over a six-day period, providing a rich source of information for detecting anomalies and malicious behavior.

The dataset includes a variety of network protocols, such as FTP, SSH, Telnet, SMTP, DNS, HTTP, HTTPS, PostgreSQL, and alternative HTTP, ensuring that the model is tested against a wide range of environments and traffic types. What's more, the packets come from different segments of a large organization's network, including academic and research segments. This diversity ensures that the model is exposed to a wide range of traffic scenarios, increasing its effectiveness under various operational conditions.

The first layer of our model handles the collection and preprocessing of traffic, using advanced techniques to capture and filter data. It acts as a first line of defense, isolating relevant information and paving the way for more detailed analysis. The second layer then takes over, applying ML techniques to the preprocessed data. Prior to analysis, data are normalized, essential features are selected, and dimension reduction is performed to optimize the performance of ML models. These models are trained with large datasets that incorporate attack scenarios as well as normal traffic, enabling us to effectively distinguish between normal and malicious behavior.

B. LAYER 1: TRAFFIC COLLECTION AND ANALYSIS

The layer 1 gives several key features:

1) PACKET CAPTURE. Packet capture is the first and one of the most crucial aspects of this layer. Tools, like Snort, operate in promiscuous mode, capturing all traffic passing through the network interface where they are installed. This capture capability is fundamental to identifying suspicious or malicious activity.

2) TRAFFIC FILTERING. Following the capture phase, it is crucial to perform selective traffic filtering to target the most relevant information. In our IDS configuration, Snort is configured with customized filtering rules. These rules are designed to isolate packets requiring scrutiny, based on criteria such as IP addresses, port numbers, or specific patterns contained within packets. For our IDS, filtering rules are focused on specific port numbers, including 21 (FTP), 22 (SSH), 23 (Telnet), 25 (SMTP), 53 (DNS), 80 (HTTP), 443 (HTTPS), 5432 (PostgreSQL), and 8080 (HTTP alternative). The aim of this filtering is to minimize the volume of data to be analyzed, thus optimizing the performance and effectiveness of the IDS in detecting suspicious or malicious activity.

3) PRETREATMENT. Pretreatment captured packets is a vital step. It includes packet defragmentation, which is crucial for detecting attacks that exploit fragmentation to evade detection. Protocol standardization is also carried out to simplify subsequent analysis, by providing a consistent structure for data processing.

The decision to adopt Snort as the core of this layer was based on a detailed comparative analysis with Suricata, two of the most efficient and widely used tools on the market. According to the study [28], although Suricata is able to process network traffic faster than Snort and has a lower packet loss rate, it requires more computing resources. In comparison, Snort is less resource-intensive, making it more suitable for resource-constrained environments. Snort also demonstrated an ability to accurately identify six of the seven types of malicious traffic examined, achieving a true positive rate (TPR) of 99% for each. Suricata, on the other hand, accurately detected only certain types of malicious traffic such as FTP, SSH, HTTP, DoS/DDoS, and ICMP, also with a TPR of 99%. However, Suricata failed to correctly detect ARP and Scan malicious traffic, also recording a TPR of 0% for these categories.

When using Snort, managing false positives and negatives is a considerable challenge. False positives can overload cybersecurity teams with irrelevant alerts, while false negatives risk leaving malicious activity undetected, threatening network security. Furthermore, adding ML to Snort can improve threat detection but requires careful attention to balance precision and reduce these errors.

C. LAYER 2: ML ANALYSIS AND CLASSIFICATION

This layer plays an essential role in an IDS system that incorporates ML algorithms. The function of this layer is to analyze the traffic data collected by the NIDS SNORT tool and then categorize this data into normal or suspicious activity. It employs various ML models to detect and identify abnormal or malicious behavior.

The layer 2 is featured as follow:

1) DATA PROCESSING. Prior to analysis, the data collected by the Network Intrusion Detection System (NIDS) are transformed from a logging format to a.csv file. This process is followed by a series of preprocessing steps such as normalization, selection of essential features, and dimension reduction, in order to simplify and optimize the data for analysis by ML models, making the dataset easier to manage and more relevant for model training.

2) MODEL TRAINING. Historical datasets are used to train ML models to identify patterns associated with malicious or anomalous behavior, improving detection precision.

3) CLASSIFICATION AND PREDICTION. The trained ML models are then applied to classify network traffic in real time, enabling normal and potentially malicious activity to be distinguished and acted upon accordingly.

The integration of ML into Snort enhancement, in particular through the adoption of the SVM algorithm, has been instrumental in resolving the problems of false positives and false negatives, while significantly improving the precision of the system. The selection of the SVM algorithm is the result of a careful evaluation of various ML algorithms, an approach inspired by research works such as article [29], which highlights Snort's shortcomings and explores different methods to remedy them. These comparisons revealed the superior efficiency of SVM, eloquently demonstrated by Table I. These comparisons revealed the superior efficiency of SVM, highlighting an exceptional 99% precision in threat detection, a low false positive rate of 0.6%, and a recall rate of 100%. This rigorous, evidence-based approach ensures a significant improvement in Snort's reliability and effectiveness in detecting threats to network security. Examining Table I in detail, we observe that the improved precision and reduced errors are attributable to the optimization of SVM parameters and its ability to model complex decision frontiers, which is essential for effectively

Table I. Snort improvement results through machine learning [29].

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
[13]	J48	CICIDS	NIDS	98	N/A	98
[14]	Support Vector Machines	NSA SNORT IDS	NIDS	95.6	0.7	96.8
	Decision Trees			82	2.9	79.2
	Fuzzy Logic	DARPA IDS		92.3	0.2	94.5
	Bayes Net			73	3.5	65
	Naïve Bayes	NSL-KDD IDS		70	3	62
[15]	Support Vector Machines	Snort IDS Alert	NIDS	99	0.6	100

discriminating between legitimate and malicious activity in network traffic.

D. LAYER 3: EVALUATION AND PERFORMANCE LAYER

Layer 3, an essential part of the proposed intrusion detection model, is responsible for in-depth evaluation of the system's performance. Its main objective is to measure the effectiveness of intrusion detection, evaluate the precision of the classifications made by the ML algorithms, and judge the overall performance of the system. To achieve this, it uses a range of metrics and parameters such as detection rate, false positive and negative rates, recall rate, precision, and F1 score. In addition, this layer takes into account crucial aspects such as execution time, energy consumption, and resource utilization. These complementary measures are essential to ensure that the model is not only accurate in its detection but also efficient and viable from an operational point of view. This multidimensional assessment ensures that the system not only performs well in all respects but also adapts proactively to meet future challenges.

When evaluating the execution of a computer program, there are several key parameters we can consider to measure its performance and effectiveness. Parameters for evaluating the execution of a computer program can vary according to the nature of the program, its objectives, and specific needs. Here are some common evaluation parameters for the execution of a computer program:

1) PERFORMANCE MEASUREMENT. To evaluate system effectiveness, we use several specific metrics that reflect the unique challenges associated with threat detection. These measures include:

- *Detection rate*: a high detection rate is essential to quickly and effectively identify malicious or suspicious activity. Greater sensitivity is required to minimize the risk of major disruptions caused by undetected attacks, such as intrusions or malware.
- *False positive rate*: This rate is of paramount importance in network security. A high volume of false positives can quickly saturate systems, reducing the effectiveness with which operators can identify and respond to real threats. Consequently, accurate and reactive alert management is essential to preserve network functionality and operability, while minimizing unnecessary downtime.
- *False negative rate*: Particularly critical in network security, a low false-negative rate is vital to ensure that no real threat goes undetected. The consequences of undetected attacks can be disastrous, including major data loss or service interruptions.
- *Precision*: This measures the precision with which the system correctly identifies legitimate versus malicious activity. High precision is essential to avoid unnecessary interruptions to normal network operations, which could result from incorrect identification.
- *F1 score*: This score is particularly suited to environments where false positives and false negatives have serious consequences. It provides a balance between recall and precision, offering a holistic view of security system performance.

2) EXECUTION TIME. It measures the time required for the program to complete a specific task. It can be expressed in milliseconds, seconds, or any other appropriate unit of time. This time may vary according to factors such as algorithm complexity, available hardware resources, system workload, and code quality. Execution time measurement is commonly used in computer science to optimize performance, evaluate the efficiency of algorithms, detect performance problems, and make decisions to improve the responsiveness and efficiency of computer systems.

The $T_{execution}$ time is calculated using the formula:

$$T_{execution} = T_{end} - T_{start}$$

where:

$T_{execution}$ is the program execution time.
T_{end} is the time at which the program starts.
T_{start} is the time at which the program completes the task.

3) POWER CONSUMPTION. For mobile, embedded or power-sensitive applications, it measures the amount of power consumed by the program during execution. We need to know the power of the device (in watts) and the duration for which it is running (in hours). We can then use the following formula to calculate energy consumption in kilowatt-hours (kWh):

$$\text{Energy (kWh)} = \text{Power (kW)} \times \text{Time (h)}$$

4) RESOURCE UTILIZATION. It evaluates the use of hardware resources, such as CPU and memory:

$$\text{CPU utilization (\%)} = (\text{CPU time used} \div \text{Total time}) \times 100$$

This formula measures the percentage of time the processor is actively used by the program in relation to the total time:

$$\text{Memory usage}\,(\%) = (\text{Memory used by program} \div \text{Total available memory}) \times 100$$

This formula indicates the percentage of total memory used by the program.

Ultimately, this intelligent SSVM model seeks to strengthen the security environment and minimize vulnerabilities that attackers could exploit, ensuring constant monitoring and seamless integration with existing security infrastructures to increase network resilience.

IV. RESULTS

The heart of our SSVM system is based on an Ubuntu machine, which serves as a platform for Snort, an advanced IDS, as well as for our Python implementation of the SVM algorithm. This configuration captures and processes the log data produced by Snort, which is then transformed into a CSV file by a specially designed Shell script. This CSV file becomes the input for analysis and intrusion detection by our SSVM model. In parallel, we use a Kali machine, equipped to automatically launch attacks via a script, thus creating an authentic test environment to evaluate the effectiveness of our model against various types of cyber threats such as denial-of-service attacks, SQL injections, and network malware.

Our SSVM model demonstrated significantly better detection and classification capabilities than other basic and advanced models, offering robust protection against a variety of attack scenarios. In particular, SSVM effectively identified and mitigated denial-of-service attacks by handling large volumes of malicious network traffic, accurately detected SQL injections exploiting database vulnerabilities, and showed exceptional results in identifying behaviors associated with malware propagation via the network. Figure 2 illustrates this performance in detail, presenting a comprehensive classification report that not only affirms our system's precision in detecting intrusions but also exposes other key performance indicators such as false positive, false negative, and recall rates. Each metric is analyzed to show how SSVM stands out from other models in terms of its ability to detect specific threats in a realistic, dynamic attack environment. This exposure detailed in Figure 2 not only validates the effectiveness of our model but also provides a solid basis for evaluating our infrastructure in the realistic simulation of various threats, thus ensuring a reliable platform for testing the effectiveness of our solution under varied network conditions.

The classification report reveals the model's exceptionally high performance for each predicted class, with indicators such as precision, recall, F1 score, and support showing excellent results. For both classes evaluated, precision reached 99%, meaning that almost all the model's predictions were correct, and no false positives were detected. False positives occur when a model wrongly predicts an instance as belonging to the positive class, while false negatives occur when the model fails to identify an actual instance of the positive class. The values for both metrics are zero for our model, suggesting that all positive and negative predictions were correct. Recall is 100%, indicating that the model has succeeded in correctly identifying all true positive cases for each class. As for the F1 score, which represents the harmonic mean between precision and recall, it also reaches 100% for both classes, underlining the model's optimal performance, remarkable precision, and robustness on the dataset tested.

System performance, illustrated in Figure 3, shows CPU utilization at 44.3% and memory utilization at 32.1%, with a total available memory of 6237462528 bytes. These indicators reveal the efficiency with which the SSVM model manages resources during threat detection and classification operations. Energy consumption for this operation is measured at 0.72 kWh, underlining the system's energy efficiency despite a computationally intensive task. In addition, the task's execution time of 20.651 seconds confirms the model's agility in test situations, capable of delivering fast and reliable results. Classification by the SSVM model is not only flawless but also achieved with significant but efficient use of system resources, illustrating an optimal balance between performance and resource consumption.

These results confirm that the SSVM model offers exceptional predictive precision while optimizing resource consumption. It therefore represents a highly effective and efficient cybersecurity solution, perfectly suited to the complex challenges of modern network environments.

```
[[  22    0]
 [   0 2378]]
              precision    recall  f1-score   support

           0       0.99      1.00      1.00        22
           1       0.99      1.00      1.00      2378

    accuracy                           1.00      2400
   macro avg       0.99      1.00      1.00      2400
weighted avg       0.99      1.00      1.00      2400

CPU usage : 44.3%

Memory usage : 32.1%

Total memory : 6237462528 bytes

Energy consumption is 0.72 kWh


The execution time is 20.651 seconds.
```

Fig. 2. Results of the SSVM automated model.

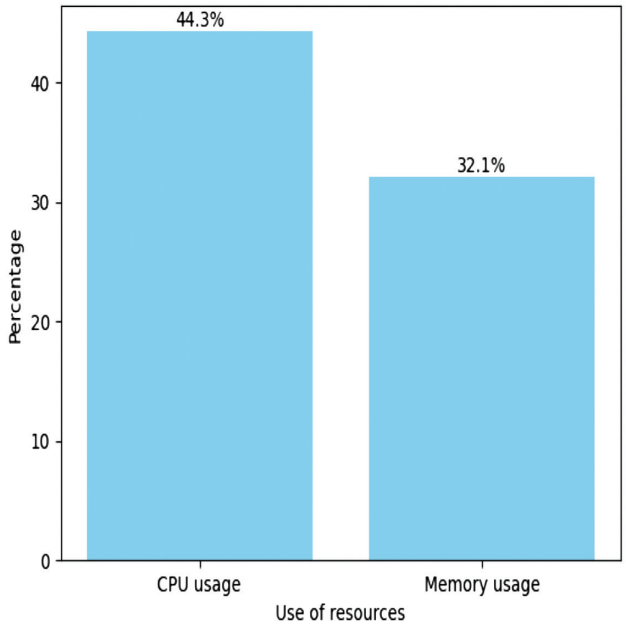

Fig. 3. Resource utilization.

V. DISCUSSION AND COMPARISON

The field of network intrusion detection has attracted considerable interest within the cybersecurity community, particularly with the integration of ML algorithms to improve detection capabilities. A considerable amount of research has been devoted to exploring the potential of various ML approaches to identify and mitigate malicious activity in network traffic. Researchers have investigated a myriad of models, from traditional classifiers such as decision trees and SVMs to more complex architectures such as neural networks and ensemble methods, seeking to capitalize on their ability to learn and adapt to evolving threats. These studies underline the essential role of ML in the development of robust, dynamic, and predictive security mechanisms that keep pace with the sophisticated tactics employed by modern cyber adversaries.

A. Ugale and A. Potgantwar [30] have developed a sophisticated network IDS using advanced ML techniques. This project is designed to improve cyber defense capabilities by proactively detecting anomalous behavior, thus playing a crucial role in preventing cyberattacks and protecting networks. At the heart of this study is an anomaly detection approach that identifies deviations from typical network traffic behavior. Feature selection and extraction methods are implemented to distinguish the most relevant traffic indicators, thereby increasing the precision of the detection system. The solution is designed for uninterrupted monitoring and the generation of immediate alerts when suspicious activity is detected, integrating and reinforcing existing security measures to solidify network security. The study offers a detailed comparative evaluation of the performance of several classification algorithms, such as SVM, RF, NB, and convolutional neural network (CNN), on various datasets. These results are neatly summarized in Table II, which highlights the superiority of CNN in terms of precision, recall, F1 score, and overall accuracy. Each metric is discussed to illustrate the CNN's ability to handle data complexity and variability, outperforming other models despite higher computational demands. In addition, Table III offers an overview of the runtimes of the different algorithms, highlighting a longer runtime for the CNN compared to other models. This information is crucial, as it indicates that although the CNN offers better performance in terms of classification, this comes with a cost in terms of processing time. This analysis enables users to weigh up the benefits of increased precision against extended runtime, providing a comprehensive framework for assessing the operational efficiency of models according to their specific needs.

The project [31] explores the refinement of a ML ensemble model enriched with genetic algorithms for network intrusion

Table II. IDS dataset [30]

Sr. No	Dataset files	Type	File size	No of records
1	friday_working_hours_morning_pcap_iscx [33]	CSV	56 MB	1.99 L
2	friday_working_hours_afternoon_ portscan_pcap_iscx [33]	CSV	75 MB	2.86 L
3	thursday_working_hours_afternoon_infilteration_pcap_iscx [33]	CSV	81 MB	2.88 L
4	tuesday-working_hours.pcap_iscx [33]	CSV Multi class	131 MB	4.45

Table III. Model evaluation of paper [30]

			Performance measure (in %)				
S. No.	Dataset files	Method	Precision	Recall	F1-Score	Accuracy	Time (sec)
1	[1]	SVM	68.58	80.43	73.01	98.48	70.93
2	[2]		98.86	98.56	98.70	98.72	59.90
3	[3]		72.72	77.77	74.99	99.28	130.22
4	[4]*		63.15	82.36	68.86	96.09	380.17
Avg			**75.8275**	**84.78**	**78.89**	**98.14**	
1	[1]	RF	52.72	90.72	49.48	81.42	21.16
2	[2]		99.39	99.50	99.44	99.45	25.55
3	[3]		50.28	98.89	49.99	97.78	44.48
4	[4]*		88.13	99.37	92.66	99.25	83.76
Avg			**72.63**	**97.12**	**72.89**	**94.47**	
1	[1]	NB	98.92	80.80	88.76	81.42	35.4
2	[2]		98.50	97.44	98.95	98.55	48.6
3	[3]		81.87	87.67	84.45	87.18	62.02
4	[4]*		94.24	92.46	88.76	97.79	129.51
Avg			**93.38**	**89.59**	**90.23**	**91.23**	
1	[1]	CNN	99.14	99.25	99.11	99.25	19.73
2	[2]		99.61	99.61	99.61	99.61	28.62
3	[3]		99.98	99.99	99.98	99.98	32.34
4	[4]*		99.27	99.27	99.16	99.27	47.88
Avg			**99.5**	**99.53**	**99.46**	**99.52**	

Table IV. Model evaluation of the paper [31]

Model	Cross validation score	Accuracy	Root square score/variance
Naive Baye Classifier	0.909238235	0.909896798	0.637990159
Decision Tree Classifier	0.993252917	0.995766076	0.982989259
K Neighbors Classifier	0.988753285	0.993649114	0.974483888
Logistic Regression	0.958192298	0.956866896	0.826703072
Basic Bagging	0.996427695	0.998279968	0.993089386
Genetically Optimized Random Forest Bagging	0.99642752	0.998544589	0.994152558

detection. This innovative method uses genetic programming to optimize the selection and tuning of the underlying models, thereby enhancing the accurate detection of anomalous activity. The overall approach capitalizes on the combination of various algorithms to form a unified system that offers increased robustness and better accuracy in different contexts. The introduction of genetic components enables fluid adaptation to new threats, underlining the project's aim of providing a scalable and responsive cybersecurity solution. As demonstrated in Table IV, the tool developed excels in identifying suspicious actions within changing network environments, exploiting a diverse dataset simulating intrusions in a military network context.

Our SSVM model demonstrates outstanding performance, as shown in Table V. This table illustrates precision and recall, both evaluated at 99%, as well as a perfect F1 score of 100%. These results highlight the model's effectiveness in accurately classifying and reliably detecting correct instances among the data tested, reflecting an excellent ability to correctly identify threats while minimizing classification errors. The weighted average F1 score of 100% underlines a consistency in performance across different classes, indicating that the model operates with uniform reliability, regardless of the variability of the input data. Table V not only summarizes these high rates but also serves as evidence of the overall robustness of the SSVM model, confirming its ability to maintain high levels of precision and efficiency under a variety of conditions.

In three recent studies, the SSVM proved its exceptional efficiency, achieving 99% precision and 100% recall. It thus demonstrates a remarkable ability to classify instances impeccably, without generating false positives or false negatives. Meanwhile, articles [30,31] present a comparative analysis of various classification techniques applied to several datasets. These studies show that, although the CNN and genetically optimized RF aggregation models shine on several performance indicators, they lack robustness in the face of evolving threats, making them less effective in fluctuating network environments or in the face of innovative attack strategies. However, SSVM excels in its extreme precision in classifying individual classes, as shown by the data in Figure 4, and is specifically designed to adapt and operate effectively under a variety of network conditions. Using advanced techniques for

Table V. Model evaluation of our SSVM model.

Class	Precision (%)	Recall (%)	F1-Score (%)
0	99	100	100
1	99	100	100
Macro Avg	99	100	100
Weighted Avg	99	100	100

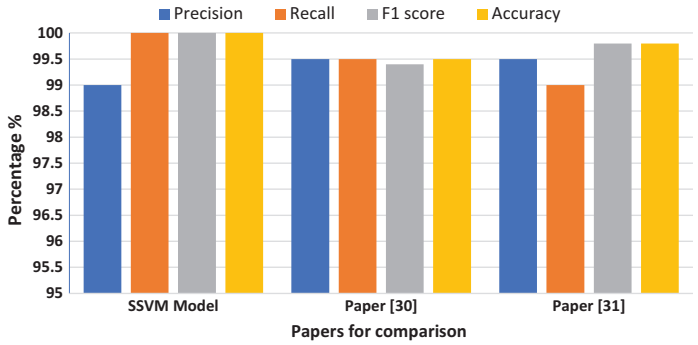

Fig. 4. Comparison of precision, recall, F1 score, and accuracy across model.

proactive detection of abnormal behavior, and incorporating sophisticated feature selection and extraction methods, the model significantly improves the precision of anomaly detection.

With constant monitoring and immediate alerts when suspicious behavior is detected, our model reinforces existing security measures and proves extremely effective against a wide range of threats, including emerging and evolving ones.

The classification model we have developed is designed to integrate effortlessly into a wide range of environments, from devices with limited capabilities to the most extensive and sophisticated systems. It features optimized memory and processor consumption, the result of extensive optimization, making it particularly well suited to contexts where resources are a scarce and precious commodity. Its flexibility of use and ease of integration into different software ecosystems attest to its universal compatibility and invariable performance. This model transcends standards thanks to its versatility and adaptability, affirming its status not only as a high-performance tool but also as a resilient and scalable solution.

The SSVM model embodies the perfect balance between cybersecurity sophistication and ease of use, eliminating the complications common to advanced IT security deployments. It is designed for simplified commissioning and maintenance, without the need for specialized technical skills on the part of the user. Its architecture, designed for full automation, guarantees smooth operations management and rapid integration with multiple levels of protection. A script specifically designed for this model orchestrates the entire process, from Snort initialization to performance evaluation, including data conversion for ML. In this way, the SSVM model proves to be a reliable, non-binding security solution, offering multilayered defense while setting itself apart from other models through its ease of administration and operational efficiency.

VI. CONCLUSIONS

The development and implementation of the SSVM model marked a significant milestone in the evolution of network IDSs. The advanced architecture of this model ensured accurate detection of current threats, while remaining flexible and scalable to meet emerging cybersecurity challenges. Our results, after a thorough evaluation against other referenced systems [30,31], demonstrated that SSVM was superior, setting a new standard for a more resilient and adaptable defense. This breakthrough reflected our determination to constantly innovate and adapt to the changing technological landscape, with the aim of maintaining first-rate network protection.

Looking to the future, we plan to focus our efforts on refining SSVM. This approach aims to strike an optimal balance between responsiveness and reliability, contributing to a network security infrastructure that not only meets today's requirements but also prepared for future advances. By pursuing this path, we aspire to offer security that never goes out of fashion, security that evolves in concert with technologies and threats, for truly proactive, forward-thinking cybersecurity.

CONFLICT OF INTEREST STATEMENT

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] S. Waskle, L. Parashar, and U. Singh, "Intrusion detection system using PCA with random forest approach," in *International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp.803–808, 2020. doi: 10.1109/ICESC48915.2020.9155656.

[2] E. Spafford, "James P. Anderson: an information security pioneer," *IEEE Security & Privacy*, vol. 6, no. 01, pp. 9, 2008. doi: 10.1109/MSP.2008.15.

[3] H. Liao, C. Lin, Y. Lin, and K. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, pp. 16–24, 2013.

[4] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, pp. 1–22, 2019.

[5] B. Shubham, "Intrusion detection and prevention systems (IDS/IPS) for OS protection," *International Journal of Scientific Research in Engineering and Management*, vol. 08, pp. 1–5, 2024.

[6] D. I. Edeh, "Network intrusion detection system using deep learning technique," Master of Science, Department of Computing, University of Turku, 2021.

[7] A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2017.

[8] A. Prieto, B. Prieto, E. Ortigosa, E. Ros, F. Pelayo, J. Ortega, and I. Rojas, " Neural networks: an overview of early research, current frameworks and new challenges," *Neurocomputing*, vol. 214, pp. 242–268, 2016.

[9] H. Dathar and M. Abdulazeez, "A modified convolutional neural networks model for medical image segmentation," *Test Engineering and Management*, vol. 83, pp. 16798–16808, 2020.

[10] K. Shashank and M. Balachandra, "Review on network intrusion detection techniques using machine learning," *IEEE Distributed Computing*, pp. 104–109, 2018.

[11] A. Ashwitha, M. Banu, and K. Puneet, "Fusing nature inspired fuzzy neural networks for hypervisor intrusion detection," *International Journal of Information Technology*, vol. 16, pp. 2915–2924, 2024.

[12] P. Amala, G. Gayathri, and S. Dinesh, "Effective intrusion detection system using support vector machine learning," *International Journal of Advanced Science and Engineering Research*, vol. 3, pp. 302–305, 2018.

[13] E. Matel, A. Sison, and R. Medina, "Optimization of network intrusion detection system using genetic algorithm with improved feature selection technique," in *IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pp. 1–6, 2019.

[14] L. Hakim, R. Fatma and Novriandi, "Influence analysis of feature selection to network intrusion detection system performance using NSL-KDD Dataset," in *International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, pp. 217–220, 2019.

[15] A. Bhumgara and A. Pitale, "Detection of Network Intrusions using Hybrid Intelligent Systems," in *1st International Conference on Advances in Information Technology (ICAIT)*, pp. 500–506, 2019.

[16] Y. Xie, "Machine learning-based DDoS detection for IoT networks," *Applied and Computational Engineering*, vol. 29, pp. 99–107, 2023.

[17] U. Mubarak Albarka, C. Zhanfang, and L. Yan, "A hybrid intrusion detection with decision tree for feature selection," *Information & Security An International Journal*, vol. 49, 2021.

[18] K. Gandhi, S. Balaji, S. Srikanth, and V. Suba, "Ensemble machine learning-based network intrusion detection system," *Evolution in Computational Intelligence*, pp. 135–144, 2023.

[19] A. Turukmane and R. Devendiran, "M-MultiSVM: an efficient feature selection assisted network intrusion detection system using machine learning," *Computers & Security*, vol. 137, pp. 103587, 2024.

[20] S. Patthi, S. Singh, and ICK P, "2-layer classification model with correlated common feature selection for intrusion detection system in networks," *Multimedia Tools and Applications*, vol. 83, pp. 1–26, 2024.

[21] M. Almseidin, M. Alzubi, K. Szilveszter, and M. Alkasassbeh, "evaluation of machine learning algorithms for intrusion detection system," in *IEEE 15th International Symposium on Intelligent Systems and Informatics*, pp. 000277–000282, 2017.

[22] L. Yang and A. Shami, "IDS-ML: An open source code for intrusion detection system development using machine learning," *Software Impacts*, vol. 14, pp. 100446, 2022.

[23] H. Liu and Bo. Lang, "Machine learning and deep learning methods for intrusion detection systems: a survey," *Applied Sciences*, vol. 9, pp.4396, 2019.

[24] R. Alghamdi and M. Bellaiche, "An ensemble deep learning based IDS for IoT using Lambda architecture," *Cybersecurity*, vol. 6, 2023.

[25] K. Albulayhi, Q. Abu Al-Haija, S. Alsuhibany, A. Jillepalli, M. Ashrafuzzaman, and F.T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, pp. 5015, 2022.

[26] K. Sudar and P. Deepalakshmi, "Comparative study on IDS using machine learning approaches for software defined networks," *International Journal of Intelligent Enterprise*, vol. 7, pp. 15, 2020.

[27] M. Al lail, A. Garcia, and S. Olivo, "Machine learning for network intrusion detection—a comparative study," *Future Internet*, vol. 15, pp. 243, 2023.

[28] S. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.

[29] O. El Aeraj and C. Leghris, "Study of the SNORT intrusion detection system based on machine learning," in *7th IEEE Congress on Information Science and Technology (CiSt)*, pp. 33–37, 2023.

[30] A. Ugale and A. Potgantwar, "Anomaly Based Intrusion Detection through Efficient Machine Learning Model," *International Journal of Electrical and Electronics Research*, vol. 11, pp. 616–622, 2023.

[31] M. Akhtar, S. Qadri, M. Siddiqui, N. Mustafa, S. Javaid, and S. Ali, "Robust genetic machine learning ensemble model for intrusion detection in network traffic," *Scientific Reports*, vol. 13, pp. 17227, 2023.