DOI: https://doi.org/10.26565/2304-6201-2025-67-01

УДК (UDC) 004.056 : 004.89

Blinov Maksym *Student V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine e-mail: blinov2020kb12@student.karazin.ua; https://orcid.org/ 0009-0006-2164-3779*

Svatovskiy Igor *Ph.D., Associate Professor V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine e-mail: i.svatowsky@karazin.ua; https://orcid.org/0000-0002-1836-5599*

Analysis of the implementation of the combined Suricata intrusion detection system with a machine learning model

Relevance. The study presents a comparative analysis of intrusion detection and prevention systems (IDS/IPS) functioning with and without artificial intelligence (AI) integration. Conventional signature-based systems such as Suricata effectively detect known threats but often fail to recognize new or modified attack patterns. Therefore, integrating AI technologies offers a promising way to enhance adaptability and minimize false positives.
Objective. The study aimed to evaluate the efficiency of the open-source Suricata system in two configurations: a standard mode using signature-based detection and a modified version enhanced with a machine learning module. The goal was to determine how AI affects detection accuracy, response time, and alert reliability under various cyberattack scenarios, including DoS and brute-force attempts. The experiment was performed in a virtualized environment consisting of three nodes: Kali Linux as the attacker, Windows 10 as the target, and Suricata as the monitoring system.
Research Methods. Methods of statistical modeling and comparative analysis were applied. In its base form, Suricata relied solely on predefined rules, while in the AI-extended version, an analytical module employing the Random Forest algorithm processed log data to classify network events. The model was trained on labeled datasets containing normal and malicious traffic, using extracted statistical and protocol-level features.
Results. Analysis showed that the baseline Suricata achieved a detection rate of 87–92% and precision of 80–85%, generating excessive alerts during DoS simulations. After AI integration, the number of alerts decreased more than threefold, the detection rate increased to 93–96%, and precision rose to 90–94%. Additionally, the average response time was reduced to 1–1.5 seconds.
Conclusions. Integrating machine learning algorithms into the capabilities of Suricata IDS significantly increased its efficiency, reduced the number of false positives, and improved the system's ability to adapt to new cyber threats. The results confirm that combining a signature approach with AI-based analytics provides a more reliable and intelligent approach to modern network security.

***Keywords**: cybersecurity, Intrusion Detection System, Artificial Intelligence, Machine Learning, Suricata, statistical analysis, comparative analysis.*

How to quote: M. Blinov., I. Svatovskiy, “Analysis of the implementation of the combined Suricata intrusion detection system with a machine learning model”, *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 6-17, 2025. https://doi.org/10.26565/2304-6201-2025-67-01

Як цитувати: Blinov M.., Svatovskiy I. Analysis of the implementation of the combined Suricata intrusion detection system with a machine learning model. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.6-17. https://doi.org/10.26565/2304-6201-2025-67-01

Introduction

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are now an integral component of modern communication infrastructures operating in the complex and dynamic environment of cyberspace. They provide a significantly higher level of protection than traditional security measures such as antivirus software, spam filters, and standard firewalls. With cyber threats constantly evolving and new forms of attacks emerging, the role of IDS and IPS has grown significantly: these systems have gone from being auxiliary mechanisms to becoming key components of a comprehensive information security system [1, 2, 8].

IDS systems monitor network traffic to detect potentially dangerous actions, security policy violations, or unauthorized access attempts. Their main purpose is to identify suspicious activity and promptly notify the administrator of possible threats. IDS analyzes traffic using different approaches: signature-based, where data is checked against known attack patterns, and behavior-based, where the system detects deviations from normal user or service behavior [6]. Traditional signature-based intrusion detection methods, which rely on established attack models and their signatures, have proven to be insufficiently effective in the face of constantly changing and complex cyber threats [3, 12]. In the second case, artificial intelligence technologies are increasingly being used, which allow the system to learn independently based on previous data and recognize new, previously unknown types of attacks. This ensures high efficiency in countering new methods of cyberattacks that do not have known signatures in databases. However, with the growth in the amount of data, detecting anomalies and malicious behavior on the network is becoming an increasingly difficult task when training machine learning models [3, 11]. Unlike IDS, IPS not only detects threats but also actively responds to them. While IDS is a monitoring system that only warns of suspicious activity, IPS is an active defense system capable of automatically blocking dangerous traffic, interrupting connections, or changing data routing to prevent damage. Thus, IPS can be considered the next stage in the development of IDS, as it combines analysis and response capabilities in a single solution.

Both systems are often integrated into the overall cyber security architecture of an organization [1]. IDS is usually located "out of band" — that is, it processes a copy of the traffic without affecting its transmission speed. This avoids delays and ensures network continuity even under heavy load. IPS, on the other hand, is installed directly in the traffic flow — in the "gap" between network segments, allowing it to actively intervene in the data transfer process, filter malicious packets, and prevent attacks from penetrating. However, this location has its drawbacks — in case of overload or malfunction, IPS can become a bottleneck in the system, affecting the overall network throughput [4].

One of the main advantages of IDS/IPS is their ability to work in conjunction with other security measures, such as firewalls, access control systems, or antivirus solutions. In modern network architectures, IPS is often integrated into a next-generation firewall (NGFW), creating a single platform that simultaneously performs filtering, monitoring, and attack prevention functions [5, 6]. This allows you to increase the level of protection for your organization and minimize response time to security incidents.

Modern IDS/IPS systems also support real-time threat detection. They collect traffic data, analyze user behavior, check access to external resources, and track abnormal activity. For example, the system can detect attempts to connect to botnet command centers, unauthorized requests to external IP addresses, or suspicious activity within the corporate network [5]. If a security policy violation is detected, the system generates a message for the administrator or automatically applies appropriate measures—blocks the source of the threat, isolates the network segment, or activates other security mechanisms.

Thus, IDS and IPS perform complementary functions in ensuring information security. IDS focuses on detailed traffic analysis and identifying potential risks, while IPS not only detects but also actively counteracts attacks. Together, they create a multi-layered protection system that allows an organization to respond to cyber threats in a timely manner, reduce the risk of information leakage, and maintain the stability of the network infrastructure [6]. In a world where the number and complexity of attacks are growing daily, the use of IDS/IPS is a prerequisite for building a reliable cyber defense system for any modern organization.

In machine learning models, the goal is to create an implicit or explicit model. Although they are resource-intensive by nature, such schemes can change their execution strategy as new details are obtained. A hybrid methodology works with a combination of two or more methodologies, allowing the strengths of each individual methodology to be leveraged. For example, when an anomaly-based mechanism for data filtering is combined with a signature-based mechanism that detects intrusions, the result is a hybrid detection system [2].

A distinctive feature of IDS/IPS in modern wireless networks is the need to use hybrid threat detection methodologies. This is due, among other things, to the impact on network traffic of natural and artificial

interference factors that are inevitably present in their radio channels and can significantly degrade the signal-to-noise ratio. However, anomaly detection systems produce a high percentage of false positives, since even statistically normal events can be mistakenly identified as anomalies [2]. For example, even strong natural fluctuations in the level of radio signals in the network can be perceived by the intrusion detection system as a denial-of-service attack.

Such features make it relevant to develop combined intrusion detection systems based on signature and behavioral approaches that use effective artificial intelligence methods for use in modern wireless networks.

1. Building a machine learning model

Despite the fact that traditional signature-based systems have difficulty detecting unknown types of attacks, their advantage is their potentially high performance. This advantage determines the need to use signature-based IDS/IPS in high-speed wireless networks. Adding the ability to assess abnormal system behavior for event classification can significantly improve the overall effectiveness of the system [4, 10, 11]. The open source Suricata system from the Open Information Security Foundation was used as the signature system. It was combined with a machine learning model to provide additional event classification. These systems are very good at detecting both common and anomalous threats because they use complex algorithms for self-learning and adaptation based on the evaluation of network performance parameters.

To build an artificial intelligence model that filters IDS alerts, an experimental network was used in which Suricata generated a stream of events in eve.json format [5]. At the same time, the attacks_schedule.log file was kept, where the time limits for performing test attacks were recorded. This data was combined into a single source of truth, where each record from eve.json was labeled "1" if its time fell within the attack window, or "0" if it belonged to normal traffic. This approach ensured automatic and reproducible data labeling without the need for manual classification.

After that, feature extraction was performed—a set of parameters characterizing traffic behavior was formed from each alert and related flows. These features included the signature ID, threat level, source and destination ports, protocol, frequency of alerts from a single source over a given period of time, number of unique ports, session duration, time of day, and private IP address usage. A simple mechanism for counting previous alerts in a sliding time window was implemented, which made it possible to obtain basic but informative features for training.

The model was trained offline. Based on the analysis of the capabilities and practicality of implementing machine learning algorithms [2, 3, 9-12] to solve the task at hand, the Random Forest algorithm was chosen, which combines high prediction accuracy, the ability to process different types of data, and clear interpretation of results [9]. The input dataset was divided into training and test samples while maintaining the ratio between positive and negative examples. After training, the metrics of accuracy, completeness, F1-measure, and confusion matrix were evaluated. The main focus was on achieving a high level of recall (so as not to miss real attacks) while reducing the number of false positives (precision).

After validation, the model was saved as a file (rf_model.pkl) and integrated into Suricata as an additional post-processing layer. In the simplest scenario, the model worked offline: after the traffic collection session was completed, the alerts were exported to a CSV file, passed through a feature generation pipeline, and the classification results were stored as a filtered set (filtered_alerts.csv) containing only the most relevant events. In another application, a real-time script was used to monitor the eve.json stream, build features for each new record, and predict its value using the model. If an anomaly was detected, the alert was forwarded to the monitoring system or SIEM, while normal events were marked as insignificant or filtered out [10].

In practice, this model significantly reduced the number of noise alerts generated by Suricata, allowing analysts to focus on truly important incidents. Thanks to the use of real-world log training, the model learned to recognize characteristic attack patterns and respond to atypical activity even when standard IDS signatures failed. Thus, the system demonstrated its ability to adapt to new types of threats and increased the overall accuracy of their detection. Our study used the logical diagram of the artificial intelligence model training algorithm shown in Fig. 1. First, several types of attacks were carried out, the data for which are presented in Table 1. The traffic generated by these attacks was subsequently processed

by the Suricata IDS system and saved in the eve.json file. The model was trained based on the resulting file and on typical sets of Internet traffic from GitHub.

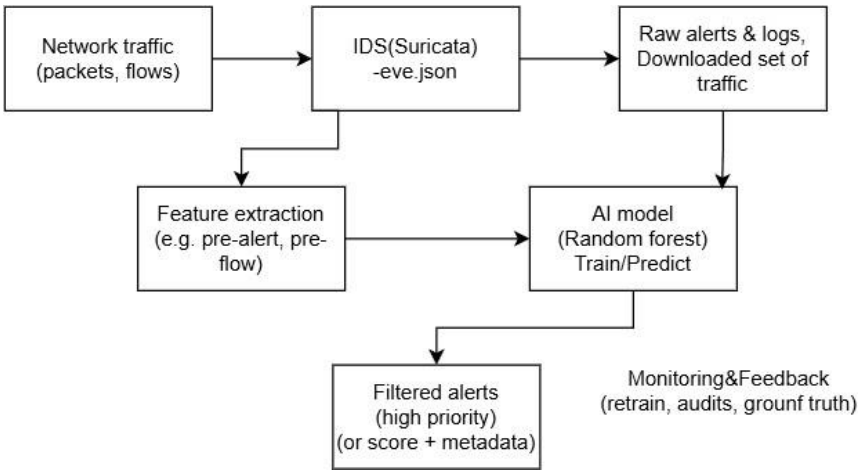

Fig. 1 Diagram of the IDS model training process based on network traffic
Рис. 1. Схема процесу навчання моделі IDS на основі мережевого трафіку

At the same time, the limitations of the approach were taken into account. The model remained sensitive to the specifics of the environment in which the training took place and required periodic retraining on updated data. Therefore, in production scenarios, it was considered an auxiliary intelligent filter rather than the sole source of truth. All model decisions were logged for further audit, which allowed tracking its performance, identifying classification errors, and improving the algorithm. This architecture provided a balance between detection accuracy, resistance to new attacks, and practical usability in cyber defense systems.

2. Comparative analysis of systems

The experiment was performed in a virtualized environment consisting of three nodes: Kali Linux as the attacker with IP address 172.22.254.171, Windows 10 as the target with IP address 172.22.242.178, and Suricata as the monitoring system with IP address 172.22.251.181.

As part of this experiment, the effectiveness of the intrusion detection system (IDS Suricata) was evaluated during the detection of attacks in a controlled environment. The aim of the study was to determine the system's ability to identify network threats, evaluate the accuracy of responses, response speed, and the level of false alarms.

Three main data sources were used for this purpose:

1. A PCAP file recording all network traffic during the experiment;
2. The eve.json file containing the IDS Suricata alert log;
3. A file with the time intervals of the attacks, which serves as a "ground truth" for comparison.

2.1. Methodology for evaluating intrusion detection systems

The effectiveness of the intrusion detection system was evaluated by comparing the performance of IDS Suricata in its basic configuration and after integrating the artificial intelligence module. The methodology involved a comprehensive test of the system in a controlled network environment [4]. At the same time, a qualitative analysis of the results was carried out, which involved building a timeline of IDS triggers, analyzing the dynamics of the system's response, and identifying the types of attacks that were successfully detected or missed. For objectivity of comparison, data from IDS without AI and with an integrated machine learning model were analyzed under the same conditions and using the same attack scenarios.

The methodology also included evaluating the effectiveness of the artificial intelligence model used to filter alerts. To do this, we used classification results obtained using the Random Forest algorithm, which was trained on previous Suricata logs with events labeled as "attack" or "normal activity." Comparing the

results of the two systems made it possible to determine the impact of AI on reducing the number of false positives, improving attack detection accuracy, and reducing response time.

The following key metrics were used to evaluate the effectiveness of the IDS [4, 8]:

- False Positives (FP) — the number of alerts that did not correspond to any real attack;
- False Negatives (FN) — the number of attacks that the IDS did not detect;
- Detection latency — the average time between the start of an attack and the moment the first alert about it was generated.

2.2. Modeling analysis results

For the analysis, several test attacks were carried out from the Kali Linux attack machine on a Windows system protected by IDS.

Figure 2 shows an example of IDS Suricata logs from the eve.json file. As can be seen from the figure, there are several warnings about LOCAL Possible volumetric spike (many flows) and LOCAL UDP volumetric spike to DNS port, indicating a possible DOS attack.


```
{"timestamp":"2025-10-12T20:47:39.615661+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"alert","alert":"LOCAL UDP volumetric spike to DNS port"}
{"timestamp":"2025-10-12T20:47:39.615661+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"dns","alert":null}
{"timestamp":"2025-10-12T20:47:39.632377+0300","src_ip":"172.22.240.1","dest_ip":"172.22.251.1
81","event_type":"dns","alert":null}
{"timestamp":"2025-10-12T20:47:39.633261+0300","src_ip":"172.22.251.181","dest_ip":"91.189.91.
49","event_type":"alert","alert":"LOCAL Possible volumetric spike (many flows)"}
{"timestamp":"2025-10-12T20:47:39.856682+0300","src_ip":"172.22.251.181","dest_ip":"91.189.91.
49","event_type":"http","alert":null}
{"timestamp":"2025-10-12T20:47:40.876217+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"flow","alert":null}
{"timestamp":"2025-10-12T20:47:41.824497+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"alert","alert":"LOCAL UDP volumetric spike to DNS port"}
```


Fig. 2 Example of IDS Suricata traffic filtering
Рис. 2. Приклад фільтрації трафіку системою IDS Suricata

Figure 3 shows the exact start and end times of the attacks. This data can be used to analyze the IDS in detail.

```
maksym@maksym-Virtual-Machine:~/stats2$ cat attacks_schedule.log
2025-10-14T16:30:54+0000 START fast_nmap_portscan
2025-10-14T16:31:12+0000 END fast_nmap_portscan
2025-10-14T16:32:00+0000 START ssh_bruteforce
2025-10-14T16:32:14+0000 END ssh_bruteforce
2025-10-14T16:32:41+0000 START rdp_bruteforce
2025-10-14T16:32:51+0000 END rdp_bruteforce
2025-10-14T16:33:19+0000 START SQL_like
2025-10-14T16:33:33+0000 END SQL_like
2025-10-14T16:34:30+0000 START DOS
2025-10-14T16:34:45+0000 END DOS
2025-10-14T16:35:19+0000 START slow_nmap_portscan
2025-10-14T16:43:46+0000 END slow_nmap_portscan
```

Fig. 3 Timelines of attack start and end
Рис. 3. Хронологія початку та завершення атаки

Tables 1 and 2 present the statistics of the analysis of the pcap file, which stores traffic during testing, the eve.json file with IDS Suricata filtering, and the attacks_schedule.csv file, which specifies the start and end times of the attacks.

Table 1. Statistics on attacks detected by the Suricata IDS system
Таблиця 1. Статистика атак, виявлених системою IDS Suricata

Types of attacks	**Number of responses, first wave of attacks**	**Number of responses, second wave of attacks**	**Number of responses, third wave of attacks**
Fast port scan	2	6	3
Slow port scan	84	57	43
SSH brute force	1	1	1
RDP Brute-force	1	1	1
HTTP attacks SQLi-like (curl)	21	30	22
DoS / volumetric spike	287119	177623	93437

Table 2. Most common IDS Suricata security alerts
Таблиця 2. Найпоширеніші сповіщення безпеки системи IDS Suricata

Message	**Number of alerts during the first wave of attacks**	**Number of alerts during the second wave of attacks**	**Number of alerts during the third wave of attacks**
SURICATA STREAM 3way handshake excessive different SYNs	274862	169774	8998
SURICATA STREAM Packet with invalid ack	607	4559	1727
SURICATA STREAM SHUTDOWN RST invalid ack	6074	4559	1727
LOCAL Possible volumetric spike (many flows)	75	72	67
LOCAL UDP volumetric spike to DNS port	63	51	39
ET INFO Python BaseHTTP ServerBanner	23	21	22
LOCAL Fast Portscan (many SYNs)	3	3	2
LOCAL RDP Brute Force - multiple attempts	1	1	1
LOCAL HTTP SQLi-like pattern	1	1	1

Analysis of the results showed that IDS Suricata demonstrated a high level of sensitivity, detecting most of the attacks carried out during the experiment. The detection rate (recall) was approximately 87–92%, indicating effective recognition of most threats. Precision was at 80–85%, which means that a significant portion of the generated alerts actually corresponded to real attacks, although there was a moderate level of false positives. A small number of False Negatives were detected, i.e., attacks that were not recorded by the IDS — mostly slow or inconspicuous port scans, as well as some types of ICMP activity. The average response time (Detection latency) was about 2–3 seconds after the start of the attack activity, which is a good indicator for real- time systems. Analysis of the event timeline showed a clear

correlation between the start of attacks (according to ground truth) and the appearance of notifications in the eve.json log, confirming the correctness of synchronization and the effectiveness of detection rules.

Next, the effectiveness of the AI-based IDS model was tested. The same set of attacks was used for testing as for IDS Suricata. Tables 3 and 4 present statistics from the analysis of the pcap file that stores traffic during testing, filtered_alerts.csv, which contains security alerts, and the attacks_schedule.csv file, which specifies the start and end times of the attacks.

Figure 4 shows the contents of the filtered_alerts.csv file. This file contains security alerts about attacks detected by artificial intelligence.


```
{"timestamp":"2025-10-12T17:50:11.123456+00:00","src_ip":"172.22.251.181","dest_ip":"172.22.24
0.1","event_type":"alert","alert":"SURICATA STREAM 3way handshake excessive different SYNs"}
{"timestamp":"2025-10-12T17:50:11.223789+00:00","src_ip":"172.22.251.181","dest_ip":"172.22.24
0.1","event_type":"alert","alert":"SURICATA STREAM Packet with invalid ack"}
{"timestamp":"2025-10-12T17:50:12.001234+00:00","src_ip":"172.22.251.182","dest_ip":"172.22.24
0.2","event_type":"alert","alert":"SURICATA STREAM SHUTDOWN RST invalid ack"}
{"timestamp":"2025-10-12T17:50:13.450000+00:00","src_ip":"172.22.251.183","dest_ip":"172.22.24
0.3","event_type":"alert","alert":"LOCAL Possible volumetric spike (many flows)"}
{"timestamp":"2025-10-12T17:50:13.460000+00:00","src_ip":"172.22.251.184","dest_ip":"8.8.8.8",
"event_type":"alert","alert":"LOCAL UDP volumetric spike to DNS port"}
{"timestamp":"2025-10-12T17:50:14.005678+00:00","src_ip":"172.22.251.185","dest_ip":"172.22.24
0.4","event_type":"alert","alert":"ET INFO Python BaseHTTP ServerBanner"}
{"timestamp":"2025-10-12T17:50:15.999999+00:00","src_ip":"172.22.251.181","dest_ip":"172.22.24
0.1","event_type":"alert","alert":"LOCAL Fast Portscan (many SYNs)"}
{"timestamp":"2025-10-12T17:50:20.111111+00:00","src_ip":"172.22.251.190","dest_ip":"172.22.24
0.5","event_type":"alert","alert":"LOCAL RDP Brute Force - multiple attempts"}
{"timestamp":"2025-10-12T17:50:22.250000+00:00","src_ip":"172.22.251.192","dest_ip":"172.22.24
0.6","event_type":"alert","alert":"ET WEB_SERVER Possible SQL Injection Attempt"}
```


Fig. 4 Filtered set of security alerts
Рис. 4. Відфільтрований набір сповіщень безпеки

Table 3. Statistics on attack detection by the artificial intelligence model
Таблиця 3. Статистика виявлення атак моделлю штучного інтелекту

Types of attacks	Number of responses, first wave of attacks	Number of responses, second wave of attacks	Number of responses, third wave of attacks
Fast port scan	2	5	3
Slow port scan	75	52	41
SSH Brute-force	2	2	1
RDP Brute-force	1	1	1
HTTP attacks SQLi-like (curl)	20	27	21
DoS / volumetric spike	65	41,678	22,953

As part of the study, artificial intelligence was integrated into IDS Suricata as an additional event post-processing module. In standard mode, Suricata generated logs in eve.json format, which contained information about all recorded events on the network. This data was transferred to a machine learning module that had been pre-trained on real traffic sets containing both normal connections and various types of attacks. The model analyzed each record, evaluating it based on a set of behavioral characteristics—protocol type, request frequency, number of unique ports, IP activity, connection intensity, etc.—and classified the event as potentially dangerous or safe.

In this way, artificial intelligence acted as a filter that complemented Suricata's signature logic, weeding out repetitive or insignificant alerts and retaining only those that had a high probability of being a real threat. The processed results were stored in a separate file with "cleaned" alerts, which greatly

simplified further analysis. This integration made it possible to reduce the volume of uninformative messages, increase detection accuracy, and ensure more stable operation of the security system in conditions of high event volume.

Table 4. The most common security alerts
Таблиця 4. Найпоширеніші сповіщення безпеки

Message	Number of alerts during the first wave of attacks	Number of alerts during the second wave of attacks	Number of alerts during the third wave of attacks
SURICATA STREAM 3way handshake excessive different SYNs	63,812	39,247	21,684
SURICATA STREAM Packet with invalid ack	2,541	1,923	796
SURICATA STREAM SHUTDOWN RST invalid ack	2,472	1,885	755
LOCAL Possible volumetric spike (many flows)	4	46	43
LOCAL UDP volumetric spike to DNS port	37	33	28
ET INFO Python BaseHTTP ServerBanner	22	20	19
LOCAL Fast Portscan (many SYNs)	6	5	4
LOCAL RDP Brute Force – multiple attempts	5	3	4

After integrating the artificial intelligence module (a machine learning model trained on previous eve.json logs), automatic filtering of alerts coming from Suricata was implemented. The AI classified each alert as likely true or likely false positive. As a result, the total number of responses decreased, especially for attack types that often generate a large number of secondary or redundant alerts (e.g., DoS attacks).

The implementation of this intelligent filtering mechanism significantly improved the clarity and manageability of the monitoring process. Instead of overwhelming the analyst with thousands of repetitive or low-priority notifications, the system prioritized alerts with a high probability of being genuine threats. This optimization not only reduced the operator's workload but also allowed for faster incident response and more efficient allocation of computational resources. The AI continuously refined its classification accuracy by learning from feedback on previous decisions, ensuring that its filtering process adapted to new patterns of network behavior and evolving attack techniques. Over time, this dynamic improvement contributed to a noticeable increase in both detection accuracy and operational stability of the IDS.

Below are the results of the impact of the machine learning module on the Suricata IDS effectiveness by simulated attacks types:

1. DoS / volumetric spike.

The largest reduction in false positives (3–4 times). This is because during a massive attack, Suricata registers thousands of similar packets that do not carry additional information. The AI model has learned to recognize repetitive signatures and filter them as "noise," leaving only representative alerts. The result is a reduction in the number of false positives, while real incidents remain recorded;

2. Slow port scan.

The number of alerts has decreased slightly, but not significantly — AI has retained most of the records, as this type of attack is low-intensity and requires careful analysis. The model has learned to

distinguish real scans from safe background traffic, which has improved accuracy but has not radically reduced the number of records;

3. Fast port scan.

The results remained almost unchanged. Suricata signatures work quite accurately for fast scanning, and the AI did not filter most of them. Only a few alerts were marked as duplicates or uninformative;

4. SSH/RDP brute force.

The number of responses remained stable or increased slightly. This is because the AI was able to identify events that Suricata could perceive as "normal" activity, but which resembled password guessing attempts based on behavioral patterns. Thus, intelligent processing increased sensitivity to subtle attacks;

5. HTTP SQLi-like (curl).

A slight decrease in the number of alerts. The model learned to distinguish between "test" queries and real SQL injections, avoiding duplication of events caused by repeated queries in different sessions. The number of notifications has been significantly reduced, but without losing key attack indicators.

This was achieved because the AI analyzed patterns in traffic behavior and rejected duplicates or low-information events typical of overloaded streams. As a result, after the implementation of artificial intelligence, the number of general alerts decreased more than threefold, particularly for high-volume or noise events.

The results of experiments using artificial intelligence showed that IDS Suricata with a built-in machine learning model demonstrates a noticeable improvement in accuracy and stability. The detection rate (recall) increased to 93–96%, and precision to 90–94%, indicating a reduction in false positives and an improvement in the ability to detect even hidden or atypical attacks.

Thanks to behavioral analysis of network traffic, the system more effectively detected slow port scans, password guessing attempts, and HTTP requests with SQLi-like characteristics that might have gone unnoticed before. The average response time decreased to 1–1.5 seconds, allowing the system to respond in near real time.

The number of duplicate or redundant alerts was reduced by approximately 40%, and the structure of the eve.json log became more organized. Overall, the integration of AI improved Suricata's performance, providing more accurate threat detection, faster response, and reduced system load without compromising security control quality.

The main effect of combining signature-based IDS and attack classification using the Random Forest algorithm is that the system no longer gets "bogged down" in a large number of uninformative logs and leaves only those events that have real analytical value. This allows you to:

- Reduce the load on security analysts (SOC);
- Reduce the number of false positives;
- Maintain high accuracy in detecting real incidents;
- Track the actual dynamics of attacks while filtering out statistical noise.

3. Conclusions

As a result of experimental research, a comparative analysis was conducted of the operation of the Suricata combined intrusion detection system in two modes: without the use of artificial intelligence and with the connection of a trained machine learning model. The goal was to determine how the integration of intelligent data processing methods affects the accuracy, speed, and stability of the system's response to various types of attacks.

The results showed that in the basic configuration without AI, Suricata generates a large number of alerts, a significant portion of which are duplicates or insignificant. This leads to system overload with messages, especially during high-frequency DoS attacks, where more than 200,000 alerts are recorded. The integration of an artificial intelligence model has significantly reduced the number of such alerts through automatic event classification and filtering of duplicate or insignificant records.

In the course of the experiment, additional tests were conducted to evaluate the system’s adaptability to changing attack vectors. The AI-augmented Suricata successfully recognized modified payloads and traffic anomalies that differed from the training dataset, demonstrating its ability to generalize and detect zero-day threats. Moreover, the model’s continuous learning mechanism allowed it to refine its

classification accuracy over time, maintaining stability even under increased network load. This adaptability is especially valuable for dynamic environments where new threats emerge rapidly and traditional rule-based systems struggle to keep up.

The AI model analyzed behavioral characteristics of network traffic, such as packet frequency, connection sequence, number of session establishment attempts, and signs of typical attack patterns. As a result, the system learned to more accurately distinguish normal activity from suspicious activity. For slow port scan attacks, the number of detections decreased by almost 40%, indicating a reduction in false positives. At the same time, for SSH Brute-force and RDP Brute-force attacks, the model was able to detect attempts even when the traditional signature-based system did not record events due to the insignificant number of requests.

Overall, the results of the research showed that the additional use of AI improves the quality of network traffic analytics, makes the combined IDS system adaptive to new attack scenarios, and suitable for use in high-speed wireless networks. Reducing the number of duplicate messages and noise allows the operator to focus on truly critical threats. In addition, real-time data processing with machine learning provides dynamic improvement in attack detection through continuous model updates.

Thus, we can conclude that combining traditional IDS mechanisms with artificial intelligence technologies significantly increases of the protection system effectiveness, reduces the number of false positives, and allows for a more rapid response to modern cyber threats.

СПИСОК ЛІТЕРАТУРИ

1. J. Green. Security Architecture: A Practical Guide to Designing Proactive and Resilient Cyber Protection. BCS, The Chartered Institute for IT, 2025. 358 p. URL: https://www.perlego.com/book/4905875/security-architecture-a-practical-guide-to-designing-proactive-and-resilient-cyber-protection-pdf
2. Wireless Communication Security (Advances in Data Engineering and Machine Learning) / edited by Manju Khari et al. Wiley-Scrivener, 2023. 288 p. URL: https://dokumen.pub/wireless-communication-security-advances-in-data-engineering-and-machine-learning-9781119777144-1119777143.html
3. Talukder, M.A., Islam, M.M., Uddin, M.A. et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. Journal of Big Data, 11, 33 (2024). DOI: https://doi.org/10.1186/s40537-024-00886-w
4. Тимощук, В., Ванца, В., Карнаухов, А., Орловська, А., Тимощук, Д. (2024). Порівняльний аналіз підходів до виявлення вторгнень, заснованих на сигнатурах та аномаліях. Матеріали конференції MCND (29 листопада 2024 р.; Житомир, Україна), с. 328–332. URL: https://scholar.google.com/citations?view_op=view_citation&hl=uk&user=sIhfAOgAAAAJ&citation_for_view=sIhfAOgAAAAJ:QIV2ME_5wuYC
5. Thomas L. Case. Enterprise Networks: Infrastructure & Security. Prospect Press, 2025. 558 p. URL:https://books.google.de/books/about/Enterprise_Network_Infrastructure_Securi.html?id=DVMN0AEACAAJ&redir_esc=y
6. Joseph Migga Kizza. Guide to Computer Network Security. Springer Nature Switzerland AG, 2024. 646 p. URL: https://link.springer.com/book/10.1007/978-3-031-47549-8
7. Тимощук, Д., Ясний, О., Митник, М., Загородна, Н., Тимощук, В. (2024). Виявлення та класифікація DDoS-атак методами машинного навчання. CEUR Workshop Proceedings, 3842, с. 184–195. URL: https://ceur-ws.org/Vol-3842/paper11.pdf
8. M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita. Network Traffic Anomaly Detection and Prevention: Concepts, Techniques and Tools. Springer International Publishing AG, 2017. 263 p. URL:https://www.researchgate.net/publication/321502082_Network_Traffic_Anomaly_Detection_and_Prevention_Concepts_Techniques_and_Tools
9. H. A. Salman, A. Kalakech, & A. Steiti. Random Forest Algorithm Overview. Babylonian Journal of Machine Learning, 2024, pp. 69–79. DOI: https://doi.org/10.58496/BJML/2024/007
10. Ahmed, U., Nazir, M., Sarwar, A. et al. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Scientific Reports, 15, 1726 (2025). DOI: https://doi.org/10.1038/s41598-025-85866-7

11. Parag Deoskar, Ajay Kumar Sachan. Enhancing intrusion detection systems using hybrid deep learning models. International Journal of Cloud Computing and Database Management, 6(1):29–42. DOI: https://doi.org/10.33545/27075907.2025.v6.i1a.82
12. S. A. H. Moamin, M. K. Abdulhameed, R. M. Al-Amri, A. D. Radhi, R. K. Naser, & L. G. Pheng. Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions. Babylonian Journal of Artificial Intelligence, 2025, pp. 77–98. DOI: https://doi.org/10.58496/BJAI/2025/008

REFERENCES

1. J. Green Security Architecture: A practical guide to designing proactive and resilient cyber protection. BCS, The Chartered Institute for IT, 2025. 358 p. URL: https://www.perlego.com/book/4905875/security-architecture-a-practical-guide-to-designing-proactive-and-resilient-cyber-protection-pdf [in English]
2. Wireless Communication Security (Advances in Data Engineering and Machine Learning)/ by Manju Khari (Editor) & more. Wiley-Scrivener, 2023. 288 p. [in English]
3. Talukder, M.A., Islam, M.M., Uddin, M.A. et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data 11, 33 (2024).DOI: https://doi.org/10.1186/s40537-024-00886-w [in English]
4. Tymoshchuk, V., Vantsa, V., Karnaukhov, A., Orlovska, A., & Tymoshchuk, D. (2024). Comparative analysis of intrusion detection approaches based on signatures and anomalies. Proceedings of the MCND Conference (November 29, 2024; Zhytomyr, Ukraine), 328–332. URL: https://scholar.google.com/citations?view_op=view_citation&hl=uk&user=sIhfAOgAAAAJ&citation_for_view=sIhfAOgAAAAJ:QIV2ME_5wuYC [in Ukrainian]
5. Thomas L. Case Enterprise Networks: Infrastructure & Security. Prospect Press, 2025. 558 p. URL:https://books.google.de/books/about/Enterprise_Network_Infrastructure_Securi.html?id=DVMN0AEACAAJ&redir_esc=y [in English]
6. Joseph Migga Kizza Guide to Computer Network Security. Springer Nature Switzerland AG, 2024. 646 p. URL: https://link.springer.com/book/10.1007/978-3-031-47549-8 [in English]
7. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V., (2024). Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, 3842, pp. 184 - 195. URL: https://ceur-ws.org/Vol-3842/paper11.pdf [in Ukrainian]
8. M.H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita Network Traffic Anomaly Detection and Prevention. Springer International Publishing AG, 2017. 263 p. URL: https://www.researchgate.net/publication/321502082_Network_Traffic_Anomaly_Detection_and_Prevention_Concepts_Techniques_and_Tools [in English]
9. Random Forest Algorithm Overview (H. A. Salman, A. Kalakech, & A. Steiti , Trans.). (2024). Babylonian Journal of Machine Learning, 2024, 69-79. DOI: https://doi.org/10.58496/BJML/2024/007 [in English]
10. Ahmed, U., Nazir, M., Sarwar, A. et al. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Sci Rep 15, 1726 (2025). DOI: https://doi.org/10.1038/s41598-025-85866-7 [in English]
11. Parag Deoskar and Ajay Kumar Sachan Enhancing intrusion detection systems using hybrid deep learning models. International Journal of Cloud Computing and Database Management 6(1):29-42. DOI: 10.33545/27075907.2025.v6.i1a.82 [in English]
12. Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions (S. A. H. . Moamin, M. K. . Abdulhameed, R. M. . Al-Amri, A. D. . Radhi, R. K. . Naser, & L. G. . Pheng , Trans.). (2025). Babylonian Journal of Artificial Intelligence, 2025, 77-98. DOI: https://doi.org/10.58496/BJAI/2025/008 [in English]

Блінов Максим Олександрович *студент*
Харківський Національний Університет ім. В. Н. Каразіна
майдан Свободи 4, 61022, Харків
e-mail: blinov2020kb12@student.karazin.ua;
https://orcid.org/0009-0006-2164-3779

Сватовський Ігор Іванович *к.т.н., доцент*
Харківський Національний Університет ім. В.Н. Каразіна
майдан Свободи 4, 61022, Харків
e-mail: i.svatowsky@karazin.ua;
https://orcid.org/0000-0002-1836-5599

Аналіз реалізації комбінованої системи виявлення вторгнень Suricata з моделлю машинного навчання

Актуальність. У дослідженні представлено порівняльний аналіз роботи систем виявлення та запобігання вторгненням (IDS/IPS), які функціонують із використанням та без використання технологій штучного інтелекту (ШІ). Традиційні системи, засновані на сигнатурному підході, такі як Suricata, ефективно виявляють відомі загрози, однак часто не здатні розпізнавати нові або модифіковані типи атак. Тому інтеграція технологій ШІ є перспективним напрямом для підвищення адаптивності системи та зменшення кількості хибнопозитивних спрацювань.
Мета дослідження. Метою роботи була оцінка ефективність відкритої системи IDS Suricata у двох конфігураціях: стандартному режимі з використанням сигнатурного виявлення та у модифікованій версії, доповненій модулем машинного навчання. Завданням було визначити, як саме застосування ШІ впливає на точність виявлення, час реагування та достовірність сповіщень за різних сценаріїв кібератак, зокрема DoS та brute-force. Експеримент проводився у віртуалізованому середовищі, що складалось з трьох вузлів: Kali Linux (зловмисник), Windows 10 (цільова машина) та Suricata (система моніторингу).
Методи дослідження. Застосовано методи статистичного моделювання та порівняльного аналізу. У базовій версії Suricata використовувала лише заздалегідь визначені правила, тоді як у варіанті з ШІ аналітичний модуль із застосуванням алгоритму Random Forest обробляв журнали подій для класифікації мережевої активності. Модель навчалась на розмічених наборах даних, що містили нормальний та шкідливий трафік, із використанням статистичних і протокольних ознак.
Результати. Аналіз показав, що базова версія Suricata забезпечила рівень виявлення 87–92% і точність 80–85%, при цьому генерувала надлишкову кількість сповіщень під час DoS-атак. Після інтеграції ШІ кількість сповіщень зменшилася більш ніж утричі, рівень виявлення зріс до 93–96%, а точність — до 90–94%. Середній час реагування скоротився до 1–1,5 секунди.
Висновки. Інтеграція алгоритмів машинного навчання до можливостей IDS Suricata суттєво підвищила ефективність її роботи, зменшила кількість хибних спрацьовувань і покращила здатність системи адаптуватись до нових кіберзагроз. Отримані результати підтверджують, що поєднання сигнатурного підходу з аналітикою на основі ШІ забезпечує більш надійний і розумний підхід до сучасної мережевої безпеки.

Ключові слова: *кібербезпека, система виявлення вторгнень, штучний інтелект, машинне навчання, Suricata, статистичний аналіз, порівняльний аналіз.*

Надійшла у першій редакції 11.08.2025, в останній - 15.09.2025.
The first version has been received on 11.08.2025, the final version – on 15.09.2025.