IBIMA Publishing
Communications of the IBIMA
https://ibimapublishing.com/articles/CIBIMA/2026/467519/
Vol. 2026 (2026), Article ID 467519, 10 pages, ISSN: 1943-7765
https://doi.org/10.5171/2026.467519

Research Article

Performance Evaluation of AI-Driven IDS: From Offline Forensic Accuracy to Real-Time Implementation Challenges

Maurycy DZIEDZIC, Piotr KOZIKOWSKI and Maciej SOBIERAJ

Poznan University of Technology, Poznan, Poland

Correspondence should be adressed to: Maciej SOBIERAJ; maciej.sobieraj@put.poznan.pl

Received date:5 November 2025; Accepted date:5 February 2026; Published date: 18 March 2026

Academic Editor: Joanna Wisniewska

Abstract

This study presents a comparative performance analysis between a modern Artificial Intelligence-based Intrusion Detection System (AI-IDS) and the traditional Snort 3 platform. While traditional signature-based systems like Snort are highly efficient for real-time traffic processing, they often struggle to detect previously unknown "zero-day" threats. To address this, a proprietary AI-IDS utilizing the RandomForestClassifier algorithm was developed and tested within a virtualized environment against UDP flood attacks. The proposed AI model achieved 99% accuracy in detecting UDP flood attacks, demonstrating superior adaptability and predictive capabilities. However, testing revealed that the AI system's current Python-based implementation is better suited for offline forensic analysis due to real-time performance bottlenecks. The findings suggest that a hybrid architecture, combining the efficiency of signature-based methods with the precision of machine learning, provides the optimal defense against evolving cyber threats.

Keywords: intrusion detection system, artificial intelligence, machine learning, Snort, RandomForestClassifier, cybersecurity.

Introduction

The contemporary cybersecurity landscape is characterized by a dynamic increase in both the number and sophistication of network attacks, which necessitates the continuous evolution of defense systems (Diana et al., 2025). Traditional Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), primarily based on signature-based methods, are facing growing

Cite this Article as: Maurycy DZIEDZIC, Piotr KOZIKOWSKI and Maciej SOBIERAJ (2026)," Performance Evaluation of AI-Driven IDS: From Offline Forensic Accuracy to Real-Time Implementation Challenges", Communications of the IBIMA, Vol. 2026 (2026), Article ID 467519, https://doi.org/10.5171/2026.467519 challenges in detecting previously unknown threats, particularly zero-day attacks (Cloudflare Learning Center, 2025). The integration of artificial intelligence (AI) and machine learning (ML) techniques into network security systems has emerged as a promising response to these challenges (Chavan and Alone, 2025)

.

Intrusion Detection Systems represent a crucial component of modern security architectures, enabling the identification of and response to unauthorized access attempts targeting network resources (Wrzesień et al., 2012). While traditional solutions such as Snort rely on predefined attack signatures, modern AI-driven heuristic systems are distinguished by their ability to learn from data and adapt to new threat patterns (Otoum et al., 2021). This adaptability allows them to enhance detection accuracy, improve response time, and provide a proactive layer of protection against evolving cyber threats.

This study aligns with the ongoing research efforts in the field of cybersecurity by undertaking a practical implementation and performance evaluation of selected IDS/IPS systems in diverse testing environments. The analysis included solutions based on widely used open-source tools such as Snort (version 3). As a complementary component, an original AI-based approach was developed and tested, utilizing machine learning models for real-time network attack detection.

The main objective of this article is to implement and evaluate Intrusion Detection and Prevention Systems (IDS/IPS) across various test environments, including virtual machines running Linux-based operating systems, using Snort 3 and selected artificial intelligence components. The study focuses on a comprehensive analysis of the effectiveness, efficiency, and performance of these systems, as well as the identification of potential advantages and limitations arising from the integration of AI mechanisms with traditional IDS/IPS architectures.

By comparing classical signature-based approaches with AI-enhanced detection models, the work aims to contribute to the ongoing discussion on hybrid security solutions that combine the precision of traditional systems with the adaptability and predictive capabilities of machine learning techniques.

The article is organized as follows. Section 2 describes IDS systems and platforms. In Section 3, the implementation of the AI-based system and testing environment were presented. Section 4 presents research results and comparative analysis. In Section 5, future work was described. Section 6 concludes the article.

Intrusion Detection Systems

Classification and Methods

Intrusion detection systems can be classified according to various criteria, with the most important distinction being between host-based systems (HIDS) and network-based systems (NIDS) (Jabez and Muthukumar, 2015). In terms of functionality, we can distinguish between intrusion detection systems (IDS) and intrusion prevention systems (IPS), with the latter characterized by the ability to actively block detected threats (Lepide Blog, 2025).

Detection methods in IDS can be divided into three main categories:

1. Signature-based detection – relies on predefined patterns of known attacks; it is highly effective against recognized threats but limited in detecting new types of attacks (Farnaaz and Jabbar, 2016).

2. Heuristic detection – uses the analysis of behavioral and contextual anomalies, enabling the identification of deviations from normal system behavior (NetScout Systems, 2021).
3. Machine learning-based detection – utilizes ML algorithms to classify network traffic and identify both known and unknown threats (Achuthan et al., 2024).

The Snort Platform as a Representative of Traditional IDS Systems

Snort is one of the most recognizable intrusion detection systems based on signature-based methods (Snort Official Website, 2025a). It employs a mechanism for verifying signatures and capturing network packets according to a predefined set of rules that analyze packet contents (Snort Manual, 2025). The architecture of Snort consists of four main modules: the packet sniffer, the preprocessor, the intrusion detection engine, and the output module (Snort Users Manual, 2025a).

The evolution of the Snort platform can be divided into three main generations. Snort 1 (1998–2001) introduced the classic operational modes in a single-threaded C-based application (Snort Users Manual, 2025a). Snort 2 (2001–2020) expanded the system by introducing the Data Acquisition interface and the first fully functional inline IPS mode (Snort Users Manual, 2025b). Snort 3 (2021–present) represents a complete rewrite in C++, featuring a modern, modular, multithreaded "inspector" model along with a publisher-subscriber event-handling mechanism (Snort Official Website, 2025b).

Compared to its predecessors, Snort 3 offers improved scalability, more efficient rule processing, and integration capabilities with containerized environments and modern orchestration systems such as Kubernetes. Furthermore, it introduces a Lua-based configuration language that enhances flexibility and ease of customization for security analysts. Despite its traditional signature-based foundation, Snort continues to evolve toward hybrid approaches that incorporate heuristic and behavioral analysis.

Artificial Intelligence in Intrusion Detection Systems

The integration of artificial intelligence (AI) methods into cybersecurity systems enables comprehensive anomaly identification and automation of defense mechanisms against sophisticated attacks (Dilhara, 2025). Machine learning algorithms such as Random Forest, Support Vector Machines (SVM), and neural networks demonstrate the capability to detect both known and unknown threats (Roshanaei et al., 2024).

Random Forest, an ensemble learning algorithm, is particularly effective in network traffic classification tasks. Its advantages include robustness to overfitting, the ability to handle large and imbalanced datasets, and the capability to assess feature importance. Research indicates that Random Forest-based models can achieve detection accuracies exceeding 97% across various categories of cyberattacks (Reddy, 2024).

Moreover, the application of AI in IDS systems extends beyond detection. Modern approaches involve deep learning architectures—such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs)—to analyze traffic patterns and temporal dependencies in real time. The integration of AI with traditional IDS, exemplified by platforms like Snort or Suricata, has led to the emergence of hybrid IDS/IPS solutions, combining rule-based logic with adaptive, data-driven intelligence.

These advancements mark a paradigm shift from reactive detection toward proactive and predictive cybersecurity, enabling faster incident response, reduced false positives, and enhanced resilience against zero-day exploits.

Implementation of the AI-Based System and Testing Platform

Implementation of the AI-IDS System

The proprietary solution to mitigate UDP flood attacks using an artificial intelligence-based model was designed. The primary goal of the project was to develop a system capable of automatically detecting and blocking anomalies in network traffic, thereby enhancing the security and resilience of network infrastructures.

To obtain training data, the network analysis tool Wireshark was utilized, allowing for packet capture and detailed inspection of network traffic. The dataset was composed of two categories of traffic samples:

Malicious traffic, representing UDP flood attack scenarios, was generated within a virtualized environment running Kali Linux, a popular penetration testing distribution equipped with specialized tools for controlled attack simulation.

Benign traffic, representing normal network behavior, was collected under conditions of high load within a local area network (LAN) operating in an open and heterogeneous environment that included typical user interactions, file transfers, and web traffic.

This dual-source approach ensured the collection of a diverse and representative dataset, covering a broad spectrum of both anomalous (attack-related) and legitimate network activities. The inclusion of realistic traffic variability improved the generalization capability of the AI model and reduced the risk of false positives during detection.

Following the data acquisition phase, the dataset underwent preprocessing, which included packet filtering, feature extraction (such as packet size, flow duration, and inter-arrival time), and normalization. These features were then used to train a supervised machine learning model designed for real-time anomaly classification.

The trained AI-IDS model was integrated into a lightweight monitoring framework capable of capturing live traffic and performing on-the-fly inference. Upon detecting suspicious activity indicative of a UDP flood, the system triggers an automated mitigation response, temporarily blocking the source IP address or throttling the corresponding traffic flow.

This implementation not only demonstrates the practical potential of AI in intrusion detection but also establishes a scalable and extensible foundation for future enhancements, such as multi-attack detection, adaptive learning, and integration with existing IDS/IPS platforms like Snort or Suricata.

Data Collection and Processing Using Wireshark

To prepare data for training and validating the IDS/IPS model, a controlled network traffic acquisition process was conducted in the PCAP format. A clear distinction was maintained between training and validation datasets, as well as between the two traffic classes: normal traffic and attack traffic.

The acquisition process was carried out using Wireshark and command-line tools compatible with the PCAP format, such as tcpdump. Captured data were saved directly to files with the .pcap extension, ensuring full compatibility with standard analysis tools and the subsequent feature extraction pipeline. Packet captures were performed on specific network interfaces, and, in the case of large datasets, the data were consolidated into single, continuous files to preserve the chronological order of packets.

Example of command used for data capture:

```
tcpdump -i enp0s3 -w normal_traffic.pcap
```

The validation data were collected independently from the training dataset, following the same acquisition methodology and using isolated environments to prevent data contamination. The validation set consisted of a 100 MB file containing normal traffic and a 162 MB file containing attack traffic. This approach allowed for testing the detection accuracy on previously unseen data while maintaining consistency in format and recording conditions.

Data processing was performed in a streaming mode, enabling real-time reading of packets from PCAP files and the extraction of features characteristic of volumetric UDP attacks. Among the analyzed attributes were:

- IP and transport layer header parameters,
- payload length and entropy,
- temporal and flow-based metrics, such as inter-packet intervals and packet counts per flow.

The use of streaming processing significantly reduced memory consumption and preserved the temporal order of events, which is crucial for the analysis of packet sequences and short transmission intervals typical of UDP flood attacks.

Data Preparation and Model Training

The dataset for UDP flood attack detection was constructed from two primary sources: legitimate traffic (recorded within a heavily loaded LAN) and malicious traffic (generated on a virtual machine running Kali Linux). Traffic samples in PCAP format were captured using Wireshark and then processed in a streaming manner through the Scapy library.

From each captured packet, a comprehensive set of features was extracted, grouped into three main categories:

- IP and transport layer header parameters – such as packet length, TTL, source and destination ports, and protocol type.
- Payload characteristics – including payload length, entropy, and the

proportion of empty or zero-payload packets.

- Temporal and flow-based features – such as the number of packets within a flow, the time interval since the previous packet, and the average number of bytes per packet.

In UDP flood attacks, packets are typically characterized by high frequency, short inter-packet intervals, and low payload entropy, making these attributes useful indicators for anomaly detection.

Before training, continuous features were normalized, and the parameter class_weight='balanced' was applied to compensate for class imbalance between normal and attack traffic samples.

For classification, the RandomForestClassifier from scikit-learn was employed with the following configuration:

- n_estimators = 100
- max_depth = 15
- min_samples_split = 5
- random_state = 42

The model's performance was evaluated using cross-validation and independent PCAP-based test sets, reporting standard quality metrics such as accuracy, precision, recall, and F1-score. After final training, the model and the scaler were saved together to ensure consistent real-time inference during online deployment.
This setup enabled the development of an AI-driven IDS module capable of continuously analyzing live network traffic and dynamically identifying UDP flood patterns with high reliability and low computational overhead.

Test Environment

The experiments were conducted within a dedicated virtualization environment built using Oracle VirtualBox, which provided full control over network topology, resource allocation, and isolation between components. The testbed was designed to replicate a realistic network infrastructure while ensuring reproducibility and safety of attack simulations.

The testing topology consisted of three virtual machines (VMs) interconnected through a virtual LAN configured in bridged mode to allow direct communication and traffic monitoring:

- Detection Machine (Ubuntu 22.04) – equipped with Snort 3, and the custom AI-IDS module.

- Hardware configuration: 6 CPU cores, 8 GB RAM, 25 GB virtual disk.

- This machine served as the primary monitoring and detection node, responsible for capturing traffic, performing signature-based detection (via Snort), and executing the AI-based anomaly detection model in real time.

- Server Environment Machine – simulated a legitimate service endpoint such as a web or application server.

  - Hardware configuration: 6 CPU cores, 8 GB RAM.

  - This node generated normal user-like network traffic (e.g., HTTP requests, DNS queries, SSH connections) to provide a realistic background for testing the accuracy of intrusion detection under load.

- Attacking Machine (Kali Linux) – used to generate and control various attack scenarios, primarily UDP flood attacks, but also configurable for TCP SYN floods and ICMP floods.

  - Hardware configuration: 10 CPU cores, 8 GB RAM.

  - Equipped with penetration testing tools such as hping3, LOIC, and Metasploit, this machine provided controlled attack vectors aimed at evaluating both the reactive and proactive capabilities of the detection systems.

The network configuration ensured bidirectional visibility of all traffic flows, allowing Snort and the AI-IDS system to capture packets in real time for analysis and response. VirtualBox's internal network mode was selected to isolate traffic from the host operating system, thereby preventing unintentional interference or external exposure during stress testing.

Research Results and Comparative Analysis

Effectiveness of the Traditional Snort 3 System (IDS Mode)

The Snort 3-based Intrusion Detection System (IDS) demonstrated a detection rate of 99.37% for UDP flood attacks during controlled testing. This high accuracy confirms the reliability of signature-based detection methods when attack patterns closely match predefined rule sets.

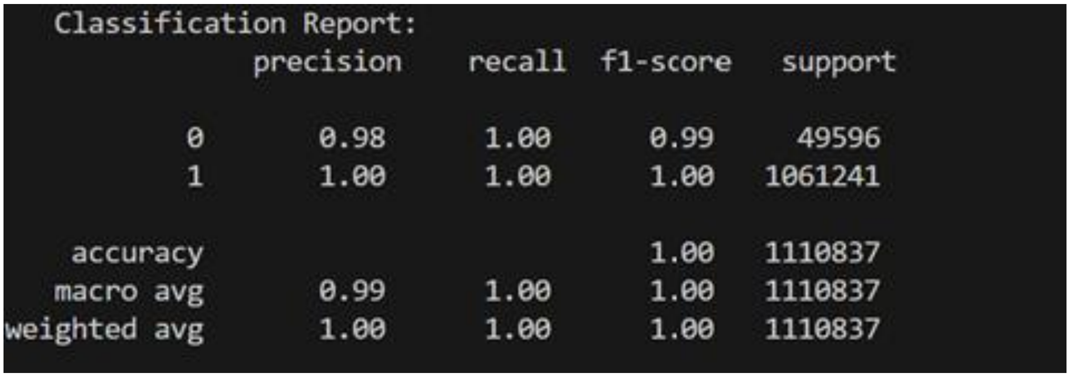
During the experiments, Snort 3 operated in passive IDS mode, where network packets were captured and analyzed in real time without active interference in traffic flow. The system used a customized rule set adapted to UDP flood detection, including signatures targeting high packet frequency, abnormal payload characteristics, and repetitive source/destination combinations.

Test Results for the AI-Based Soluion

The artificial intelligence model demonstrated very high precision, achieving a value of 0.99, as illustrated in Figure 1. This outstanding accuracy also translated into excellent results during testing with previously collected network datasets containing real attack traffic, as shown in Figure 2.

The evaluation confirmed that the AI-based system was capable of accurately distinguishing between malicious and legitimate network activity, even when exposed to traffic patterns not encountered during training. The model maintained high sensitivity to volumetric anomalies while minimizing false positives, which is critical in operational environments with high data throughput.

The results confirm that the integration of machine learning techniques with intrusion detection workflows significantly improves both accuracy and adaptability compared to traditional, signature-based systems such as Snort.



```
  Classification Report:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99     49596
           1       1.00      1.00      1.00   1061241

    accuracy                           1.00   1110837
   macro avg       0.99      1.00      1.00   1110837
weighted avg       1.00      1.00      1.00   1110837
```

Fig 1. Detection accuracy values

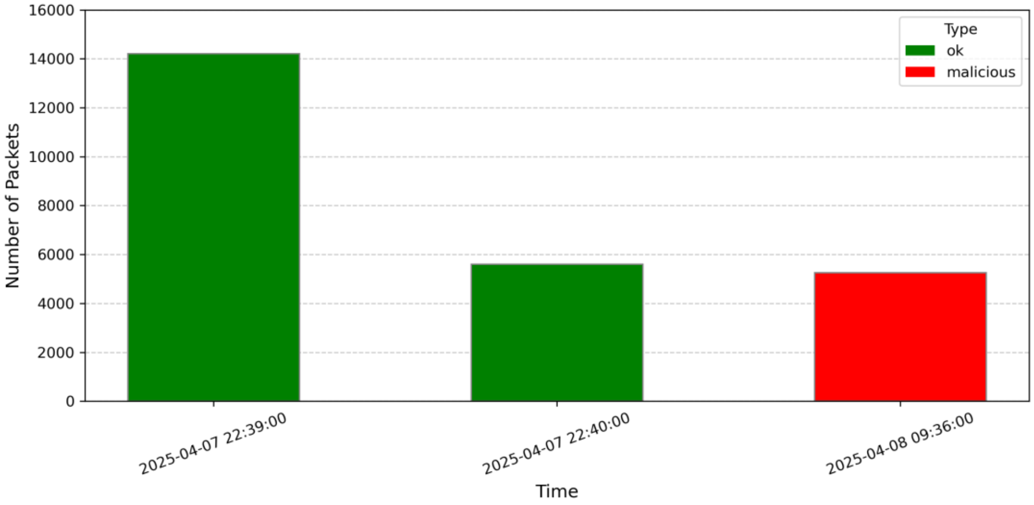


Fig 2. Traffic over time (per minute)

In the real-time network traffic test, the model proved to be less effective, which can be attributed to its relatively slow performance. The most probable cause of this limitation lies in its implementation using the Python scripting language, which does not provide sufficient execution speed for scenarios requiring real-time packet analysis.

In contrast, when tested on previously captured traffic loaded into the model in offline mode, detection performance remained highly accurate and stable. Under these conditions, the system was able to process large datasets efficiently, achieving results consistent with those obtained during validation.
The findings from these experiments indicate that the proposed tool can be effectively employed for post-incident analysis and offline forensic processing, where time constraints are less critical. However, in the context of live network monitoring and real-time intrusion detection, the current implementation is not suitable for practical deployment without further optimization or reimplementation in a high-performance language such as C++.

The conducted research reveals the complementary nature of AI-based and traditional signature-based systems. While Snort-type systems demonstrate high effectiveness in detecting and blocking known attack types in real-time, AI-based systems are characterized by their ability to identify previously unknown threats. Detection effectiveness comparison is presented in Tab. 1.

Table 1: Detection Effectiveness Comparison

System	***Type***	***Accuracy***	***Zero-day Detection***	***Response Time***
Snort 3	*IDS*	*97–99%*	*Limited*	*Real-time*
AI-IDS (RandomForest)	*IDS*	*99%*	*High*	*Offline: excellent, Real-time: limited*

Implementation Analysis

Traditional IDS solutions, such as Snort, primarily rely on signature-based mechanisms, making them proven and widely used in practice. Their advantage lies in the ability to process data in real-time with relatively low computational resource requirements, which makes them efficient in environments with limited hardware infrastructure. However, the drawback of this approach is the need for regular signature updates and a limited ability to detect new, previously unknown attacks that have not yet been included in the signature database.

In contrast, AI-based IDS systems demonstrate a significantly higher convergence capability, allowing them to more effectively identify previously unknown types of attacks and adaptively adjust to changing network conditions. Their strength comes from the ability to automatically learn from large volumes of data, resulting in increased flexibility and detection efficiency. It should be noted, however, that these systems require substantial computational resources, limiting their full deployment in real-time mode. Additionally, their effectiveness heavily depends on the quality and size of the training datasets, which form the foundation of the model learning process.

Practical Implementation Aspects

For Snort 3, the implementation process can be considered relatively straightforward, primarily due to the ease of installation and configuration. A particular advantage is the support provided by the community and the Cisco Talos team. Additionally, Snort being open-source allows for obvious business cost savings.

The challenges associated with deploying AI-based IDS systems are different. Their implementation requires advanced knowledge in machine learning, which limits the availability of specialists capable of effective configuration and maintenance. Another barrier is the high cost of the computational infrastructure needed to train and operate AI models. It is also necessary to regularly retrain the models to maintain their effectiveness and address issues related to the interpretability of decisions, which hinders full transparency of these systems' operations.

Practical Implications

In environments characterized by a high risk of zero-day attacks, it is recommended to use hybrid solutions that integrate traditional signature-based mechanisms with artificial intelligence algorithms. This approach allows for the simultaneous use of mature technologies with proven effectiveness and methods capable of detecting previously unknown threats.

In environments with limited infrastructure resources, signature-based systems, such as Snort, remain the optimal choice. They feature low hardware requirements and long-established effectiveness in network threat detection, making them a stable and economically justified solution.

Meanwhile, in the process of analyzing security incidents, AI-IDS systems operating in offline mode are particularly useful. Their ability to process large volumes of data and identify complex correlations enables in-depth post-incident analysis and the detection of broader attack patterns.

Future Work

Optimizing the performance of IDS systems is a significant direction for further research, particularly in the implementation of artificial intelligence algorithms using high-performance programming languages such as C++. These efforts aim to increase data processing efficiency and reduce computational resource load, thereby enabling broader deployment of AI solutions in production environments that require real-time operation.

Another important area of development is hybrid systems, whose architecture integrates traditional signature-based methods with artificial intelligence techniques. This approach allows the combination of proven effectiveness in detecting known threats with the ability to identify new, previously unrecognized attacks. As a result, hybrid systems can effectively merge the advantages of both approaches while minimizing their individual limitations.

Conclusions

The research showed that AI-based intrusion detection systems, represented by the RandomForestClassifier model, achieve high accuracy (99%) in detecting UDP flood attacks. However, performance limitations in real-time mode make them more suitable for post-incident analysis than active protection.

Traditional systems, such as Snort 3, maintain an advantage in applications requiring real-time processing and demonstrate proven reliability in production environments.

Optimal deployment of intrusion detection systems should consider the specifics of the environment, available resources, and threat profile. In the context of increasingly sophisticated cyberattacks, the future of IDS is likely to belong to hybrid solutions that effectively combine proven signature-based methods with the innovative capabilities of artificial intelligence.

Acknowledgment

This research was funded by the Polish Ministry of Science and Higher Education under Grant 0313/SBAD/1315.

References

- Achuthan, K., Ramanathan, S., Srinivas, S., and Raman, R. (2024) 'Advancing Cybersecurity and Privacy with Artificial Intelligence: Current Trends and Future Research Directions,' Frontiers in Big Data, 7.
- Chavan, P. V., and Alone, N. V. (2025) 'Optimizing Intrusion Detection with Random Forest: A High-Accuracy Approach using CIC-IDS 2017,' International Journal of Computer Applications, 187(3), 17-21.
- Cloudflare Learning Center. (2025) UDP flood DDoS attack. [Online], [Retrieved October 27, 2025]. Available: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/
- Diana, L., Dini, P., and Paolini, D. (2025) 'Overview on Intrusion Detection Systems for Computers Networking Security,' Computers, 14(3), 87.
- Dilhara, A. (2025) 'Impact of Machine Learning and AI on Cybersecurity Risks and Opportunities,' KeAi Cyber Security Applications.
- Farnaaz, N., and Jabbar, M. A. (2016) 'Random Forest Modeling for Network Intrusion Detection System,' Procedia Computer Science, 89, 213-217.
- Jabez, J., Muthukumar, B. (2015) 'Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach,' Procedia Computer Science, 48, 338-346
- Lepide Blog. (2025) Top 10 Most Common Types of Network Attacks. [Online], [Retrieved October 27, 2025]. Available: https://www.lepide.com/blog/common-types-of-network-attacks/
- NetScout Systems. (2021) What is an ICMP Flood Attack? [Online], [Retrieved October 27, 2025]. Available: https://www.netscout.com/what-is-ddos/icmp-flood
- Otoum, S., Kantarci, B., and Mouftah, H. (2021) 'A Comparative Study of AI-based Intrusion Detection Techniques in Critical Infrastructures,' ACM Transactions on Internet Technology (TOIT), 21(4), 81, 1 - 22.
- Reddy, A. K. (2024) 'AI Vs. Traditional IDS: Comparative Analysis of Real-World Performance Metrics,' International Journal of Research in Information Technology and Computer Communications, 12(2).
- Roshanaei, M., Khan, M. R., and Sylvester, N. N. (2024). 'Enhancing Cybersecurity through AI and ML: Strategies, Challenges, and Future Directions,' Journal of Information Security, 15, 320-339.
- Snort Manual. [Online], [Retrieved October 27, 2025]. Available: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node11.html
- Snort Official Website (2025) What is Snort? [Online], [Retrieved October 27, 2025]. Available: https://www.snort.org/
- Snort Official Website (2025) Why Snort 3? [Online], [Retrieved October 27, 2025]. Available: https://www.snort.org/snort3/
- Snort Users Manual Snort Release: 1.8.4. [Online], [Retrieved October 27, 2025]. Available: https://www.inf.fu-berlin.de/lehre/WS07/ITS/aufgaben/SnortManual184.pdf
- Snort Users Manual 2.9.16. [Online], [Retrieved October 27, 2025]. Available: https://snort-org-

site.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf

- Wrzesień, M., Olejnik, Ł., and Ryszawa, P. (2012) 'IDS/IPS: systemy wykrywania i zapobiegania włamaniom do sieci komputerowych,' Pomiary Automatyka Kontrola, 58(2), 114-121.