An Impactful Change In Packet Sniffing Using Machine Learning

Lithiga Jayaprakash
Chandigarh University
lithigasindhu@gmail.com

Aryan Thakur
Chandigarh University
thakurrraryan90@gmail.com

Atishaya Jain
Chandigarh University
Atishayajain10@gmail.com

Prince Kumar Roy
Chandigarh University
Pproychd@gmail.com

Aksat
Chandigarh University
Akshatdaddy@gmail.com

Prof.Swapnil Raj
Chandigarh University
swapnil.e13017@cumail.in

Abstract*—With the escalating sophistication of cyber threats in today's digital landscape, the efficacy of traditional packet sniffing methods in network security analysis is increasingly challenged. In response, this paper investigates the paradigm shift towards leveraging advanced machine learning techniques for packet sniffing applications. Traditional approaches, reliant on signature-based detection and manual analysis, are found to be inadequate in addressing the complexities of modern cyber threats, especially within encrypted data streams. Conversely, machine learning, a subset of artificial intelligence, offers a promising alternative by enabling autonomous learning and decision-making from vast datasets. Through a comprehensive review of literature, this study delineates the advancements in machine learning methodologies applied to packet sniffing, highlighting their superior adaptability, scalability, and efficacy in threat detection. By synthesizing insights from empirical studies, this paper elucidates the advantages of machine learning over traditional methods, including enhanced accuracy, real-time analysis, and the ability to detect evolving threats. Furthermore, this research underscores the transformative potential of machine learning in augmenting network security efforts, heralding a new era of proactive threat mitigation and resilience.

Keywords*—cybersecurity, machine learning, packet sniffing, models, traditional methods.

I. Introduction

In today's digital age, cybersecurity has become a critical concern for individuals, businesses, and governments alike. The continuously evolving communication network architecture and integration of a diverse range of devices have resulted in sophisticated challenges for network security [1].

The recent developments in 5G networks and beyond have provided higher data rates and speeds, leading to a significant increase in data traffic and connected devices. This increase in data traffic and connected devices also brings about vulnerabilities, threats, and potential attacks, which can have catastrophic financial, social, and humanitarian consequences. To tackle these challenges, traditional methods of scrutinizing and analyzing Big Data for suspicious activities may no longer be sufficient.

One area of cybersecurity that has gained significant attention is packet sniffing. Packet sniffing refers to the process of intercepting and analyzing network traffic in order to gain information about the data being transmitted. Packet sniffing has been used for various purposes, including network troubleshooting, performance monitoring, and security analysis. However, the traditional approach to packet sniffing relies heavily on signature-based security software, which can be prone to false positives and false negatives. This limitation has prompted researchers to explore alternative methods, such as machine learning, to improve the accuracy and effectiveness of packet sniffing for cybersecurity purposes.

Within artificial intelligence, machine learning is a subfield that focuses on algorithms and statistical models, allowing computers to learn and make decisions without being explicitly programmed. Machine learning algorithms have shown promise in various fields, including cybersecurity. One area where machine learning can be applied to improve packet sniffing is in distinguishing normal from malicious TCP sessions, even when encryption is in place. This research paper aims to explore the application of machine learning techniques in packet sniffing for cybersecurity purposes. The purpose of this paper is to provide an overview of machine learning methods applied to packet sniffing for cybersecurity [2].

Moreover, it seeks to highlight the limitations of traditional packet sniffing approaches and demonstrate how machine learning can overcome these limitations.

Machine learning algorithms have the potential to improve the accuracy and effectiveness of packet sniffing for cybersecurity purposes. By utilizing machine learning algorithms, packet sniffing can analyze large amounts of network traffic and detect malicious activities with greater precision. Furthermore, machine learning algorithms have the ability to adapt and learn from new data, enabling them to detect evolving cyber threats that may not be captured by traditional methods.

This research paper also aims to identify the frequently used machine learning methods within supervised, unsupervised, and semi-supervised machine learning for packet sniffing for cybersecurity.

In order to conduct this research, a systematic review of existing literature on machine learning methods applied to packet sniffing for cybersecurity will be conducted. This review will provide insights into the different approaches, objectives, and results of previous studies in this area. Additionally, the research paper will highlight the benefits of using machine learning over traditional methods.

II. Overview Of Packet Sniffing

A. Outline Of Packet Sniffing

The process of intercepting and examining data packets as they move over a network is known as packet sniffing. This process allows security professionals to inspect the contents of these packets, identify potential security threats or anomalies, and take appropriate action to mitigate risks. However, despite its benefits in monitoring network traffic,

packet sniffing also poses certain risks such as unauthorized access to sensitive information, privacy violations, and potential misuse by malicious actors.

Packet sniffing is a technique used in network security to monitor and analyse the data that is being transmitted over a network. It plays a crucial role in detecting malicious activities, identifying vulnerabilities, and ensuring overall network integrity. With the increasing complexity of cyber threats, the need for accurate packet sniffing methods has become more pressing than ever.

B. Traditional Packet Sniffing Methods

Port Mirroring/Spanning:

Description: Network switches are configured to duplicate traffic from one or more ports to another port to which a packet sniffer is connected in a process known as port mirroring or spanning. This method allows the packet sniffer to capture all packets passing through the mirrored ports.

Complexity: Port mirroring is less feasible in some contexts due to its complexity and the need for network switch access.

Visibility: Port mirroring may not capture all network traffic, especially if traffic doesn't pass through the mirrored ports.

Promiscuous Mode:

Description: A network interface card (NIC) records all packets on the network segment, even those that are not addressed to it, while it is in promiscuous mode. This mode lets the NIC to intercept packets that aren't meant for the host, which makes packet sniffing possible.

Detection: Promiscuous mode can be detected by network intrusion detection systems (IDS) or other security measures, leading to potential alerting or blocking of the sniffing host.

Encrypted Traffic: Promiscuous mode is ineffective for encrypted traffic since it captures encrypted packets without being able to decipher their content.

ARP Spoofing:

Description: By sending fictitious ARP packets, one can link the attacker's MAC address to the victim's machine's IP address through Address Resolution Protocol (ARP) spoofing. This allows the attacker to intercept traffic meant for the victim's machine.

Visibility: ARP spoofing may not capture all traffic on the network, especially if the attacker's machine is not in the communication path between the victim and other network nodes.

Detection: ARP spoofing can be detected by monitoring ARP traffic and detecting inconsistencies between ARP requests and responses.

Physical Taps:

Description: Physical taps are hardware devices inserted into the network cabling that passively copy all data flowing through the network. These taps can be difficult to detect but require physical access to the network infrastructure.

Physical Access: Physical taps require access to network cabling, making them less practical for remote or distributed network environments.

Maintenance: Physical taps may require regular maintenance and monitoring to ensure proper functioning.

Packet Capture Software:

Description: Various software tools like Wireshark, tcpdump, and Snort can capture and analyze network packets.

Manual Analysis: Packet capture software often requires manual analysis of captured data, which can be time-consuming and may not scale well for large networks.

Limited Context: Captured packets may lack context, making it challenging to differentiate between normal and malicious activity without additional analysis.

III. Impact Of Machine Learning

Machine learning (ML) techniques can be effectively applied to packet sniffing for various purposes, including intrusion detection, network traffic classification, anomaly detection, and network traffic analysis.

A real positive occurs when the model correctly predicts the positive class. A real negative, on the other hand, is a result in which the model accurately predicts the negative class.

When the model forecasts the positive class inaccurately, the result is called a false positive. Furthermore, a false negative occurs when the model forecasts the negative class inaccurately.

The benefits of using machine learning for threat intelligence and security analytics are numerous. Machine learning can improve malware detection by analyzing attack vectors and intrusions, providing better analysis of suspicious traffic and unauthorized access. It also automates daily security activities, saving time and effort for more important work. Additionally, machine learning can predict forthcoming events and trends, and it can be used for intrusion detection, behavior-based analysis, and security policy validation. Furthermore, machine learning models such as Neural Networks, Support Vector Machines, Logistic Regression, and Linear Regression can be applied to calculate the probability of suspicious packets, leading to more accurate threat detection. Incremental learning, a methodology for machine learning that relies on new examples as they appear and model adaptation, is particularly suitable for real-time industrial applications, and it can efficiently use memory, CPU, and storage. Overall, machine learning offers the potential for proactive threat intelligence, improved security analytics, and automated intrusion detection [3].

The input data file used for training and testing in the ML-based approach contains the following components:

FlagFlood (Boolean): Indicates whether the macof tool was in the 'idle' period (0) or 'flooding' the suspected host with artificial packets (1).

Mean (real number): Represents the ping Round-Trip Time (RTT) value or the curl download data rate for each period.

Median (real number): Represents the ping RTT or curl download data rate median value for each period.

Standard Deviation (real number): Represents the ping-based RTT result or curl download data rate standard deviation value for each period.

FlagSniff (Boolean): Indicates whether the sniffer was running on the suspected host (1) or not (0).

These statistical measures for each period are used as input for the ML-based experiments, and they help train a model that can reliably predict the presence of a sniffer on the network. The input file is used to determine whether the host is malicious or not, and it is split into training and testing datasets for cross-validation. This approach reduces the input file size from millions of entries to 1600 entries, making it more manageable for training and testing the ML model.

The trained ML model for sniffing detection can be executed using the Driveless AI (dai) software, which is an AI framework that provides features and supports GPU processing. The software allows for the auto-tuning of ML algorithms, which enables a focus on the main problem of sniffing detection without the need for tuning ML parameters. The model needs to be periodically re-executed, and the training phase depends on the performance of the hardware. It is not required to have enterprise-class hardware for the training phase, as commercial cloud providers like IBM or Google offer special platforms and products for AI applications. Once a trained model provides good results, it can be used without concerns. Therefore, the hardware requirements for executing the trained ML model for sniffing detection are not fixed and can vary based on the performance needs and the availability of specialized platforms or products for AI applications [4].

Machine learning offers the advantage of accurately determining the congestion status of router ports, enabling efficient detection of attack scenarios in packet sniffing. By using machine learning algorithms, such as perceptrons, the system can differentiate between attack traffic and regular traffic, improving the accuracy of identifying malicious nodes. This approach reduces false predictions, enhances localization efficiency, and adapts to dynamic network traffic scenarios in real-time, making it a powerful tool for packet sniffing and attack detection [5].

Here are some ML techniques commonly used for packet sniffing:

A. *Supervised Learning*

- *Classification:* Using algorithms such as Support Vector Machines (SVM), Random Forests, Decision Trees, or Neural Networks to classify network traffic into predefined categories such as normal traffic or different types of attacks (e.g., DoS, DDoS).
- *Deep Learning*: Techniques like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can learn complex patterns in network traffic data for classification tasks.

Advantages:

a) Provides accurate classification when trained on labeled data.

b) Well-established algorithms like SVM, Random Forests, and Neural Networks offer good performance.

c) Suitable for scenarios where various types of network traffic need to be classified (e.g., normal traffic vs. different types of attacks).

Disadvantages:

d) Requires a large amount of accurately labeled data for training, which can be difficult and time-consuming to obtain.

e) May suffer from performance degradation if the training data does not adequately represent real-world scenarios.

f) Limited to detecting known attack patterns and may struggle with detecting novel or sophisticated attacks.

B. *Unsupervised Learning*

- *Clustering:* Algorithms like K-Means or DBSCAN can group network traffic data into clusters based on similarity, which can help identify anomalies or unusual patterns.
- *Anomaly Detection:* Techniques such as Isolation Forests, One-Class SVM, or Autoencoders can detect unusual behaviour or anomalies in network traffic that may indicate potential attacks.

Advantages:

a) Does not require labeled data, making it suitable for detecting unknown or emerging attack patterns.

b) Can identify anomalies and unusual patterns in network traffic that may indicate potential security threats.

c) Allows for exploratory analysis of network data without predefined categories.

d) Allows for exploratory analysis of network data without predefined categories.

Disadvantages:

e) Prone to false positives due to the lack of labeled data for validation.

f) Difficulty in distinguishing between benign anomalies and actual security threats.

g) May suffer from performance degradation if the training data does not adequately represent real-world scenarios.

h) May require expert domain knowledge for interpreting detected anomalies and taking appropriate actions.

C. *Semi-supervised Learning*

- *Self-training:* Using a small amount of labelled data combined with a larger amount of unlabelled data to improve classification accuracy.

Advantages:

a) Combines the benefits of supervised and unsupervised learning by leveraging both labeled and unlabeled data.

b) More scalable than traditional supervised learning approaches as it can utilize large amounts of unlabeled data.

c) Can improve model generalization by incorporating diverse examples from unlabeled data.

Disadvantages:

d) Still requires a certain amount of labeled data for initial model training, which may be challenging to obtain

e) The representativeness and quality of the labeled data have a major impact on performance.

f) May not fully exploit the benefits of unsupervised learning if labeled data is scarce or of poor quality.

D. Reinforcement Learning

- Reinforcement learning techniques can be used for dynamic packet sniffing strategies, where the model learns to make decisions on which packets to capture based on feedback from the environment (e.g., network performance metrics, security alerts).

Advantages:

a) Can adapt to dynamic environments and evolving attack strategies by learning from interaction with the environment

b) Enables autonomous decision-making in real-time based on feedback from the network.

c) Suitable for packet sniffing tasks where the optimal action depends on the context and network conditions.

Disadvantages:

d) Requires extensive training and exploration in complex environments, which may be computationally expensive.

e) Limited interpretability, making it challenging to understand the rationale behind the model's decisions.

f) Vulnerable to adversarial attacks that exploit weaknesses in the learning process.

E. Feature Engineering

- Extracting relevant features from packet headers or payloads to represent network traffic data effectively for ML algorithms. Features could include packet size, protocol type, destination IP, port numbers, payload content, etc.

Advantages:

a) Allows for the extraction of relevant features from network traffic data, improving the performance of ML algorithms.

b) Enables customization of feature representations based on domain knowledge and specific detection requirements.

c) Helps reduce dimensionality and computational complexity by focusing on informative features.

Disadvantages:

d) Manual feature engineering can be labor-intensive and subjective, requiring expertise in network protocols and security.

e) May overlook important but non-obvious features that could improve detection accuracy

f) Difficult to adapt feature representations to changing network environments and attack patterns.

F. Ensemble Learning

- Combining multiple ML models to improve overall performance and robustness. Techniques like bagging (e.g., Random Forests) or boosting (e.g., AdaBoost) can be applied.

Advantages:

a) Combines multiple base learners to improve classification accuracy and robustness.

b) Helps mitigate overfitting and reduces the risk of model bias by aggregating predictions from diverse models.

c) Enables the integration of different learning algorithms and feature representations for enhanced performance.

Disadvantages:

d) Increased computational complexity and resource requirements due to training and maintaining multiple models

e) Higher implementation and tuning complexity compared to individual base learners.

f) Limited interpretability when combining predictions from multiple models, especially in complex ensemble architectures.

G. Online Learning

- Techniques that can adapt to changes in network traffic patterns over time. Online learning algorithms continuously update the model based on incoming data streams, making them suitable for real-time packet sniffing applications.

Advantages:

a) Adapts to changing network conditions and attack patterns in real-time without requiring retraining from scratch.

b) Suitable for environments with continuous data streams where batch processing is not feasible.

c) Enables rapid response to emerging threats and dynamic network events.

Disadvantages:

d) Vulnerable to concept drift and noisy data streams, which may degrade model performance over time.

e) Limited by the availability of online learning algorithms that can effectively handle streaming data.

f) Requires careful parameter tuning and monitoring to maintain model stability and performance in dynamic environments

It's important to note that the choice of ML technique depends on the specific objectives of the packet sniffing task, the characteristics of the network traffic data, and the computational resources available. Additionally, preprocessing steps such as data cleaning, normalization, and feature selection are often crucial for effective application of ML techniques to packet sniffing.

IV. LIMITATIONS OF TRANDITIONAL METHODS COMPARED TO MACHINE LEARNING TECHNIQUES

Traditional methods for packet sniffing typically involve manual analysis and rule-based detection, which can be time-consuming and less accurate compared to machine learning methods. Machine learning algorithms, on the other hand, can automatically learn patterns from network traffic data, enabling more efficient and accurate detection of anomalies or intrusions in real-time. Machine learning methods can

adapt to new threats and data patterns, making them more effective for packet sniffing in dynamic and complex network environments [6].

Machine learning replaces traditional methods for packet sniffing by utilizing algorithms like Support Vector Machine (SVM) and Random Forest (RF) to categorize intrusion detection system (IDS) data. These algorithms outperform traditional methods during training and testing phases, adapting to different datasets and attributes. Machine learning enhances accuracy and efficiency in detecting and classifying network intrusions compared to traditional signature-based detection methods [11].

Traditional methods for packet sniffing often rely on predefined rules and signatures to analyze network traffic, which can be limited in handling complex and evolving network behaviors. In contrast, machine learning techniques, especially deep learning algorithms, can adapt and learn from data without the need for explicit programming of rules. Machine learning models can automatically extract features and patterns from large volumes of traffic data, enabling more accurate and efficient packet sniffing compared to traditional methods [9].

Features	***Traditional method***	***Machine Learning Techniques***
Encryption Handiling	Traditional packet sniffing methods struggle to analyze encrypted traffic since they cannot decipher the content of encrypted packets.	Machine learning models can still extract valuable insights from encrypted data by analyzing metadata, traffic patterns, and other features without decrypting the content.
Adaptability	Traditional methods rely on predefined rules or signatures, making them less effective at detecting emerging threats or unknown attack patterns.	Machine learning algorithms can adapt and learn from new data, enabling them to detect novel threats and evolving attack techniques.
Automated Analysis	Manual analysis of captured packets is often required, leading to slower detection and response times.	Machine learning models can automate the analysis of network traffic, enabling real-time threat detection and response without human intervention.
Contextual Understanding	Traditional methods may capture raw data without providing much context or insight into the nature of the traffic.	Machine learning models can incorporate contextual information to better understand the nature of network traffic and distinguish between normal and malicious behavior.
Pattern Recognization	Traditional methods may rely on simple	Machine learning algorithms are quite good at recognizing

Features	***Traditional method***	***Machine Learning Techniques***
	pattern matching or heuristics, making them less effective at identifying subtle or complex attack patterns.	patterns; they can recognize intricate patterns that point to malevolent or suspicious activity.

Fig. 1.

Fig. 1. Difference between Traditional method and Machine Learning Techniques

V. APPLICATIONS OF MACHINE LEARNING FOR PACKET SNIFFING

Efficiency: Machine learning algorithms can quickly analyze large volumes of network traffic data, identifying patterns and anomalies more efficiently than traditional methods.
Adaptability: ML models can adapt and learn from new data, improving over time and staying up-to-date with evolving network threats.
Detection Accuracy: ML can detect complex and subtle patterns in network traffic that may go unnoticed by traditional methods, enhancing the accuracy of packet sniffing for identifying attacks in real-time [16].
Automating Detection: ML algorithms can automatically detect patterns and anomalies in network traffic, improving efficiency.
Adapting to New Threats: ML can adapt to new threats and patterns in real-time, providing proactive security measures [17].
Enhanced Detection: Machine learning algorithms can improve the accuracy and efficiency of detecting malicious packets in network traffic.
Anomaly Detection: ML models can identify unusual patterns in packet data, signaling potential security threats or abnormal behavior.
Predictive Analysis: By analyzing historical packet data, machine learning can predict future network vulnerabilities or attacks based on patterns and trends [8].
Real-Time IoT Traffic Analysis: Machine learning algorithms are used to identify malicious patterns in real-time IoT traffic.
SPADE Algorithm for Network Security: SPADE algorithm minimizes computational and I/O costs by reducing database scans in network security applications.
Unsupervised ML Approaches for Smart Grid Security: Utilizing unsupervised ML approaches for smart grid security to detect cyber-attacks without performance reduction [12].

Machine learning techniques have been applied to packet sniffing for anomaly detection in industrial control systems (ICS) and network security. Methods like autoencoders, convolutional neural networks (CNNs), and variational autoencoders (VAEs) have been used to analyze packet payloads and detect anomalies. These models aim to distinguish between normal and attack input data by encoding and decoding packet information to identify potential threats in network traffic [13].

Machine learning is applied in packet sniffing for network anomaly detection. It helps in identifying unusual patterns in network traffic that could indicate potential security threats

or attacks. By analyzing network data using machine learning algorithms, it can detect anomalies in real-time and improve the efficiency of intrusion detection systems [14].

VI. CONCLUSION

The shift from traditional security solutions to data-driven models is justified due to the limitations of conventional methods in addressing the evolving cyber threat landscape. Traditional solutions like firewalls and access control may not effectively mitigate modern cyber risks. Data-driven models leverage machine learning algorithms to extract insights from security data, enabling more proactive and intelligent cybersecurity solutions. This shift allows for dynamic updating based on new security patterns, enhancing the efficiency and effectiveness of cybersecurity measures in response to emerging threats [7].

Ensuring the accuracy and reliability of the data used to train ML models for packet sniffing is crucial to avoid false positives or negatives. ML models used for packet sniffing, especially deep learning models, can be complex and lack transparency, making it challenging to interpret their decisions. Networks are dynamic, and ML models need to adapt quickly to new threats and changing network conditions to remain effective in packet sniffing tasks. Future limitations in applying machine learning for packet sniffing may include the need for continuous updates to keep up with evolving cyber threats. Ensuring the scalability of ML models to handle the increasing volume of network traffic and data generated in modern networks could be a challenge. Addressing privacy concerns and regulatory requirements related to analyzing network data using ML algorithms may pose limitations on the application of machine learning for packet sniffing [10].

Imbalance in cyber-security datasets can affect the performance of machine learning models. The sensitivity of evaluation metrics like F1-score to imbalanced datasets can be a challenge in the cyber-security domain. Only 2.7% of selected papers reported the time complexity (training and inference) of machine learning models, indicating a lack of comprehensive understanding in this area [15].

REFERENCES

[1] Aiyanyo, I. D., Samuel, H., & Lim, H. (2019). A Systematic Review of Defensive and Offensive Cybersecurity with Machine Learning. *Applied Sciences, 10*(17), 5811.

[2] Ali, M. L., Thakur, K., & Atobatele, B. (2019, July). Challenges of cyber security and the emerging trends. In *Proceedings of the 2019 ACM international symposium on blockchain and secure critical infrastructure* (pp. 107-112).

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] M. Gregorczyk, P. Żórawski, P. Nowakowski, K. Cabaj and W. Mazurczyk, "Sniffing Detection Based on Network Traffic Probing and Machine Learning," in IEEE Access, vol. 8, pp. 149255-149269, 2020, doi: 10.1109/ACCESS.2020.3016076

[5] M. Sinha, S. Gupta, S. S. Rout and S. Deb, "Sniffer: A Machine Learning Approach for DoS Attack Localization in NoC-Based SoCs," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 11, no. 2, pp. 278-291, June 2021, doi: 10.1109/JETCAS.2021.3083289

[6] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[7] Sarker, I.H., Kayes, A.S.M., Badsha, S. *et al*. Cybersecurity data science: an overview from machine learning perspective. *J Big Data* **7**, 41 (2020). https://doi.org/10.1186/s40537-020-00318-5

[8] Shaukat K, Luo S, Varadharajan V, Hameed IA, Chen S, Liu D, Li J. Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity. *Energies*. 2020; 13(10):2509.https://doi.org/10.3390/en13102509

[9] Mahmoud Abbasi, Amin Shahraki, Amir Taherkordi,Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey,Computer Communications,Volume 170,2021,Pages 19-41,ISSN 0140-3664,https://doi.org/10.1016/j.comcom.2021.01.021.

[10] Amir Afaq, Noman Haider, Muhammad Zeeshan Baig, Komal S. Khan, Muhammad Imran, Imran Razzak,Machine learning for 5G security:Architecture,recentadvances,andchallenges,AdHocNetwork Volume123,2021,102667,ISSN15708705,https://doi.org/10.1016/j.adhoc.2021.102667.

[11] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi and R. Ahmad, "Machine Learning and Deep Learning Approaches for CyberSecurity: A Review," in IEEE Access, vol. 10, pp. 19572-19585, 2022, doi: 10.1109/ACCESS.2022.3151248.

[12] Pinto, S. J., Siano, P., & Parente, M. (2022). Review of Cybersecurity Analysis in Smart Distribution Systems and Future Directions for Using Unsupervised Learning Methods for Cyber Detection. *Energies*, *16*(4), 1651. https://doi.org/10.3390/en16041651.

[13] Kwon, H., Kim, T., & Lee, M. (2021). Advanced Intrusion Detection Combining Signature-Based and Behavior-Based Detection Methods. *Electronics*, *11*(6), 867. https://doi.org/10.3390/electronics11060867.

[14] A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," in IEEE Access, vol. 9, pp. 78658-78700, 2021, doi: 10.1109/ACCESS.2021.3083060.

[15] Mvula, P.K., Branco, P., Jourdan, GV. *et al.* A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning. *Discov Data* **1**, 4 (2023). https://doi.org/10.1007/s44248-023-00003-x

[16] Keserwani, H. ., Rastogi, H. ., Kurniullah, A. Z. ., Janardan, S. K. ., Raman, R. ., Rathod, V. M. ., & Gupta, A. . (2022). Security Enhancement by Identifying Attacks Using Machine Learning for 5G Network. *International Journal of Communication Networks and Information Security (IJCNIS)*, *14*(2), 124–141. https://doi.org/10.17762/ijcnis.v14i2.5494

[17] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran and Q. Emad Ul Haq, "Machine Learning Techniques for 5G and Beyond," in IEEE Access, vol. 9, pp. 23472-23488, 2021, doi: 10.1109/ACCESS.2021.3051557.

[18] J. Vykopal, P. Seda, V. Švábenský and P. Čeleda, "Smart Environment for Adaptive Learning of Cybersecurity Skills," in IEEE Transactions on Learning Technologies, vol. 16, no. 3, pp. 443-456, 1 June 2023, doi: 10.1109/TLT.2022.3216345.

[19] H. Karimipour, A. Dehghantanha, R. M. Parizi, K. -K. R. Choo and H. Leung, "A Deep and Scalable Unsupervised Machine Learning System for Cyber-Attack Detection in Large-Scale Smart Grids," in IEEE Access, vol. 7, pp. 80778-80788, 2019, doi: 10.1109/ACCESS.2019.2920326.

[20] Bharadiya, J. . (2023). Machine Learning in Cybersecurity: Techniques and Challenges. *European Journal of Technology*, *7*(2), 1 - 14. https://doi.org/10.47672/ejt.1486

[21] Dasgupta, D., Akhtar, Z., & Sen, S. (2020). Machine learning in cybersecurity: A comprehensive survey. *The Journal of Defense Modeling and Simulation*. https://doi.org/10.1177/1548512920951275

[22] Varun Shah. (2022). Machine Learning Algorithms for Cybersecurity: Detecting and Preventing Threats. *Revista Espanola De Documentacion Cientifica*, *15*(4), 42–66. Retrieved from https://redc.revistas-csic.com/index.php/Jorunal/article/view/156

[23] Sarker, Iqbal H., Yoosef B. Abushark, Fawaz Alsolami, and Asif Irshad Khan. 2020. "IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model" *Symmetry* 12, no. 5: 754. https://doi.org/10.3390/sym12050754