

A MACHINE LEARNING POWERED NETWORK INTRUSION DETECTION SYSTEM FOR ACCURATE AND ADAPTIVE CYBER THREAT MONITORING

¹Nalini V, ²Vishali M, ³Mr. C. Srinivasan M.E AP/ECE

^{1,2,3}Department of Electronics and Communication Engineering, Kongunadu College of Engineering and Technology, Trichy, India

Corresponding Authors: Nalini V, Vishali M

Abstract

Cyber threats are becoming more advanced, making network security increasingly difficult. Network Intrusion Detection Systems (NIDS) are essential for identifying malicious activities, but traditional methods often face issues like imbalanced data, high false alarm rates, and poor adaptability. To address these challenges, this project proposes a Machine Learning-based NIDS using the XGBoost algorithm to classify network traffic as Attack or Non-Attack. The system involves preprocessing data, extracting relevant features, and training the model for better performance. XGBoost is selected for its efficiency, scalability, and ability to handle imbalanced datasets. The model effectively learns traffic patterns to detect intrusions with higher accuracy and fewer false positives, making it a reliable and scalable solution for modern real-time cybersecurity applications.

Keywords: Network Intrusion Detection System, Machine Learning, XGBoost, Cybersecurity, Network Traffic Analysis, Threat Detection.

1. Introduction

Computer networks form the backbone of modern communication, data sharing, and online services, but the rapid growth of the internet and cloud technologies has significantly increased the frequency and complexity of cyberattacks. These attacks can lead to serious consequences such as data breaches, financial losses, and disruption of critical services, making network security a crucial concern. Network Intrusion Detection Systems (NIDS) are widely used to monitor network traffic and identify suspicious or malicious activities in real time. However, traditional intrusion detection methods rely mainly on rule-based or signature-based techniques, which are effective only for known attacks and fail to detect new or evolving threats. Moreover, these systems face challenges such as handling large-scale network data, managing class imbalance between normal and attack traffic, and reducing high false positive rates, which ultimately affect their performance and reliability..

To address these limitations, machine learning techniques have been introduced, as they can learn patterns from historical data and adapt to changing attack behaviors. Among various algorithms, XGBoost stands out due to its high efficiency, scalability, and strong performance with imbalanced datasets. In this project, a machine learning-based NIDS is developed using the XGBoost algorithm to classify network traffic as attack or non-attack. The system includes important steps such as data preprocessing to clean and normalize data, feature extraction to select relevant attributes, and model training to improve detection capability. By leveraging the ability of XGBoost to capture complex patterns, the proposed system enhances detection accuracy, reduces false positives, and provides a reliable, scalable, and efficient solution for real-time intrusion detection in modern cybersecurity environments.

This study has the following contributions:

Machine Learning-Based NIDS using XGBoost:

The system is developed using the XGBoost algorithm to classify network traffic as attack or non-attack. It ensures high accuracy and fast processing of large-scale network data. This improves overall intrusion detection performance..

Data Preprocessing and Feature Engineering:

The input data is cleaned by removing noise and handling missing values. Relevant features are extracted and normalized to enhance model efficiency. This step helps in improving the learning capability of the model.

Handling Imbalanced: The system effectively manages class imbalance between normal and attack traffic. It ensures that rare attack instances are also properly detected. This leads to better detection accuracy and reliability.

Adaptive Detection of Cyber Threats: The model learns patterns from historical data and adapts to new threats. It can identify both known and unknown attack behaviors. This makes the system suitable for dynamic network environments.

Improved Performance and Evaluation: The system is evaluated using metrics like accuracy, precision, recall, and F1-score. It shows reduced false positives compared to traditional methods. This ensures a reliable and efficient real-time intrusion detection system.

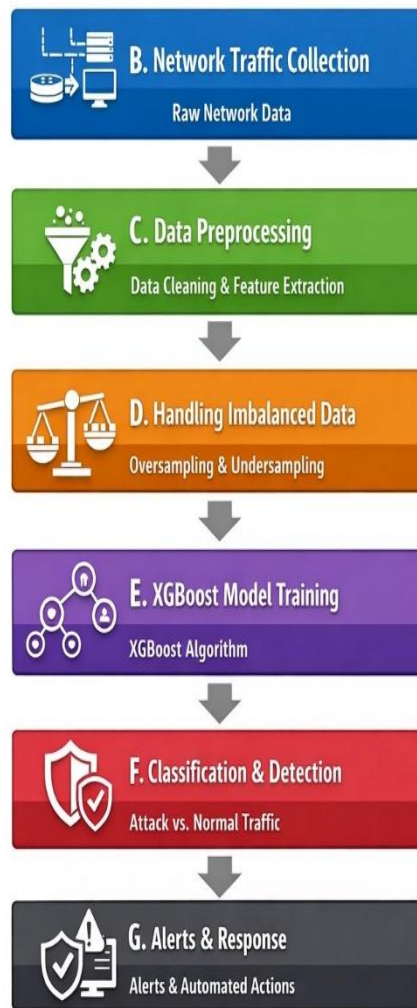
II. Literature Review

The literature on network intrusion detection systems highlights the increasing need for intelligent and adaptive security solutions in modern digital environments. William Stallings [1] explains the fundamentals of network security and emphasizes the importance of monitoring network traffic to detect malicious activities. Early approaches, as introduced by Dorothy Denning [2], were primarily signature-based, which were effective for known attacks but failed to detect new and evolving threats. With the advancement of technology, machine learning techniques have gained importance, as they can learn complex patterns from data and adapt to dynamic environments. Ian Goodfellow [3] highlighted the role of machine learning in improving detection capabilities through data-driven approaches. Several researchers have contributed to enhancing intrusion detection using machine learning models. Zhang et al. [4] proposed a system that improves detection accuracy by analyzing network traffic features, while Kumar et al. [5] demonstrated that preprocessing techniques such as normalization and feature selection significantly improve model performance. Handling imbalanced datasets is another critical challenge, and Chen et al. [6] showed that balancing techniques help in detecting rare attack instances more effectively. The XGBoost algorithm, developed by Tianqi Chen [7], has been widely recognized for its efficiency, scalability, and strong performance on structured data. Studies by Li et al. [8] proved that XGBoost achieves higher accuracy and lower false positive rates compared to traditional methods.

Furthermore, performance evaluation plays a key role in validating intrusion detection systems. Garcia et al. [9] emphasized the use of metrics such as accuracy, precision, recall, and F1-score to measure system effectiveness. Ahmed et al. [10] highlighted the importance of adaptive systems that can continuously learn from new data to handle evolving cyber threats. Overall, these studies demonstrate that machine learning-based intrusion detection systems, particularly those using XGBoost along with effective preprocessing and imbalance handling techniques, provide a reliable, scalable, and efficient solution for accurate and adaptive cyber threat monitoring, which aligns with the objectives of the proposed work.

Methodology

A. Experiment Setup



B. Network Traffic Collection

The process begins with collecting raw network traffic from sources like routers, servers, and client devices. This data includes packet details such as source IP, destination IP, protocol type, and packet size, containing both normal and malicious activities. It can be captured in real time or taken from benchmark datasets using tools like packet sniffers. However, the collected data is often unstructured and noisy, with redundant or irrelevant information. Since this step forms the foundation of the system, proper and high-quality data collection is essential. Accurate and diverse data improves the model's performance and detection capability, directly impacting the system's reliability.

C. Data Preprocessing

Raw network data is cleaned to remove noise, duplicates, and inconsistencies, while handling missing values properly. Important features are extracted, irrelevant ones are removed, and data is normalized for uniform scaling. Categorical values are converted into numerical form if needed. This preprocessing step ensures clean, structured input, reduces complexity, and improves model accuracy and performance. Without proper preprocessing, the model may learn incorrect patterns and produce unreliable results. Hence, it plays a critical role in building an efficient and accurate system.

D. Handling Imbalanced Data

In network datasets, normal traffic is typically much higher than attack traffic, leading to a class imbalance problem. If this imbalance is ignored, the model tends to favor normal traffic predictions, causing many attacks to go undetected. To address this issue, techniques such as oversampling and undersampling are applied, along with synthetic data generation methods to balance the dataset effectively. A balanced dataset enables the model to learn both normal and attack patterns properly, improving the detection of rare but critical attack instances. This step significantly reduces false negatives and is essential for real-world security applications, as ignoring imbalance can make the system unreliable.

E. XGBoost Model Training

The processed data is fed into the XGBoost algorithm, which is a powerful gradient boosting technique used for building accurate predictive models. It constructs multiple decision trees sequentially, where each new tree focuses on correcting the errors made by the previous ones. This step-by-step improvement significantly enhances the overall prediction accuracy of the model. XGBoost is highly efficient in handling large-scale and structured data, making it suitable for real-world applications. Additionally, the algorithm includes regularization techniques that help prevent overfitting, ensuring that the model generalizes well to unseen data. It also performs effectively even when working with imbalanced datasets. During the training phase, the model learns patterns from historical data and captures complex relationships between different features. Due to these capabilities, XGBoost becomes highly suitable for intrusion detection tasks, resulting in a robust and efficient model.

F. Classification and Detection

The trained model is used to classify incoming network traffic, where each data instance is labeled as either attack or non-attack. It analyzes patterns in the data and predicts the appropriate class, enabling the detection of both known and unknown attack behaviors. This process supports real-time detection in live environments, providing fast and accurate classification results. As the core of the intrusion detection system, this step ensures continuous monitoring of network activity. However, incorrect classifications can impact system security, making high accuracy essential. The system continuously processes incoming data streams, allowing proactive identification and prevention of potential threats.

G Alerts and Response

Once an attack is detected, the system generates alerts to notify administrators about potential threats. It can also trigger automated responses when required, such as blocking IP addresses or isolating affected systems, while allowing safe traffic to pass without interruption. This ensures that normal network operations are maintained even during security incidents. The response mechanism helps reduce the damage caused by attacks, and timely alerts support quick decision-making. Additionally, the system logs detected events for further analysis, which can be used to improve future performance. This step completes the intrusion detection pipeline, ensuring both strong security and overall system reliability.

III. Existing System

Network Intrusion Detection Systems (NIDS) are a cornerstone of cybersecurity, designed to monitor network traffic for malicious activities or policy violations. By passively inspecting packets traversing a network, they generate alerts when potential threats are detected, enabling security teams to respond swiftly. Over the decades, NIDS have evolved from simple pattern-matching engines to sophisticated platforms incorporating machine learning (ML) and artificial intelligence (AI), reflecting the ever-changing landscape of cyber threats.

Traditional Signature-Based Detection

The earliest and most widely deployed NIDS rely on signature-based detection. This approach maintains a database of known attack patterns—signatures—and compares incoming traffic against them. A signature might be a specific byte sequence in a packet payload, an unusual combination of flags, or a particular source port. When a match occurs, the system raises an alert. Snort, an open-source NIDS, popularized this method with its flexible rule language, enabling administrators to write custom signatures for emerging threats. The primary strength of signature-based systems is their accuracy for known attacks and low false-positive rates. However, they are inherently reactive: they cannot detect novel, zero-day exploits, and their signature databases require constant updates to remain effective.

Anomaly-Based Detection

To address the limitations of signature-based systems, anomaly-based detection emerged. This technique establishes a baseline of normal network behavior through statistical modeling, ML, or heuristic rules. Any deviation from this baseline is flagged as anomalous and potentially malicious. For instance, an unusual surge in outbound traffic during off-hours might indicate data exfiltration. Anomaly-based NIDS can theoretically uncover unknown attacks, but they suffer from higher false-positive rates because legitimate but rare activities (e.g., a scheduled backup) can trigger alerts. Systems like Zeek (formerly Bro) facilitate deep protocol analysis and can be extended with anomaly detection scripts, blending the two paradigms.

Hybrid Systems

Modern NIDS often adopt hybrid architectures that combine signature and anomaly detection. For example, Suricata—a high-performance engine—can use signatures for known threats while also employing reputation analysis and protocol anomaly checks. This layered approach balances accuracy with the ability to detect novel attacks. Moreover, many systems now incorporate stateful protocol analysis, which tracks the state of network connections (e.g., TCP handshake) to detect attacks that span multiple packets, such as slow denial-of-service attempts.

Machine Learning in NIDS

The integration of ML has revolutionized intrusion detection. Instead of relying solely on manually crafted signatures, ML models learn complex patterns from labeled datasets (e.g., NSL-KDD, CIC-IDS2017). Supervised algorithms like Random Forest, Support Vector Machines (SVM), and deep neural networks classify traffic as normal or malicious based on extracted features. Feature engineering is critical: typical features include packet length, protocol type, connection duration, flag statistics, and content-based attributes. ML-based NIDS can generalize to previously unseen attack variants, offering a proactive defense. However, they face challenges such as class imbalance (attacks are rare), concept drift (attack patterns evolve), and adversarial evasion. Evaluation metrics like precision, recall, and F1-score are essential to gauge performance beyond mere accuracy.

The Role of AI Chatbots and Self-Healing Networks

Recent innovations extend NIDS with AI-driven assistants and automated response mechanisms. Chatbots, like the one integrated into your project, leverage large language models (e.g., GPT-4o-mini) to provide contextual explanations for alerts, suggest mitigation steps, or educate users about specific threats. This bridges the gap between raw detection and actionable intelligence. Furthermore, the concept of self-healing networks envisions an ecosystem where NIDS not only detect but also automatically respond—for instance, by reconfiguring firewalls or rerouting traffic via software-defined networking (SDN) controllers. Such systems require tight integration between detection, decision-making, and enforcement, ultimately reducing the window of exposure to attacks.

Existing Implementations

IBM Watson for Cybersecurity: Uses NLP to analyze security research and provide insights.

NCCoE Chatbot (NIST): A RAG-based assistant that retrieves and summarizes cybersecurity guidelines.

CASPER AI: Developed in Australia, it integrates with identity and network logs to explain anomalies in plain language.

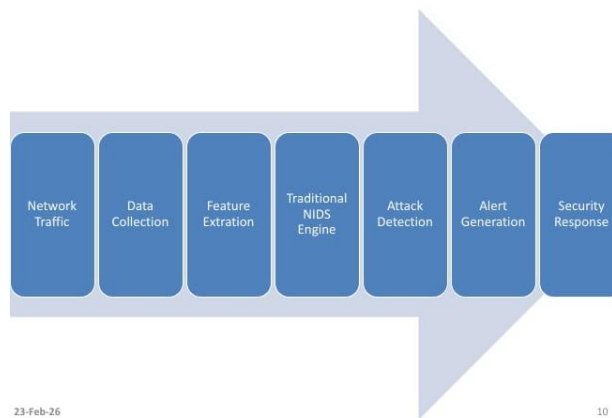


Fig: Existing system

IV. Proposed System

The existing system successfully demonstrates a machine learning-based NIDS with a GPT-powered assistant for predictions and cybersecurity guidance. However, it operates on static feature inputs and lacks real-time traffic analysis, continuous learning, and automated response capabilities. The proposed system aims to transform this foundation into an intelligent, self-healing security platform that actively monitors network traffic, adapts to emerging threats, explains its decisions, and automatically mitigates attacks.

System Objectives

The proposed system is designed to:

- Capture and analyze live network traffic in real-time
- Continuously update detection models with new data
- Provide explainable predictions for enhanced trust
- Automatically respond to detected threats

- Offer an intelligent assistant for guided incident response

System Architecture

A. Real-Time Traffic Acquisition Module

The system will integrate a packet capture engine using libraries like Scapy or Pyshark to sniff network traffic in real-time. A flow generator (similar to CICFlowMeter) will convert raw packets into bidirectional flows, extracting statistical features such as packet counts, byte rates, duration, and inter-arrival times. This ensures compatibility with machine learning models while maintaining performance.

B. Advanced Preprocessing Pipeline

Raw traffic data will undergo cleaning to remove malformed packets and handle missing values. Features will be normalized using StandardScaler, and categorical variables (protocol, service) will be encoded appropriately. The pipeline will support both batch processing for historical data and streaming for real-time analysis.

C. Hybrid Machine Learning Engine

- Instead of a single model, the proposed system employs an ensemble approach combining:
- Random Forest for high-precision detection of known attack patterns
- Deep Neural Network for capturing complex non-linear relationships
- Isolation Forest for unsupervised anomaly detection
- A voting mechanism consolidates predictions, improving overall accuracy and robustness against evasion attempts.

D. Continuous Learning Framework

The system implements a sliding window retraining mechanism. As new traffic is processed, the model is periodically updated using recent data. Active learning techniques allow security analysts to provide feedback on flagged events, which is incorporated into future training cycles. This ensures the system adapts to evolving attack strategies without manual intervention.

E. Explainable AI Component

To address the black-box nature of machine learning, SHAP (SHapley Additive exPlanations) is integrated. For every alert, the system generates feature importance scores, explaining why a particular flow was classified as malicious. This transparency aids analysts in understanding threats and reduces false positive fatigue.

F. Automated Response Module

Upon detecting a threat, the system assigns a severity score based on confidence levels and asset criticality. For high-confidence attacks, automated mitigation actions are triggered via:

- Firewall rule updates (iptables, pfSense API) to block offending IPs
- SDN controller integration (OpenFlow) to isolate compromised hosts
- Alert forwarding to SIEM platforms for centralized management
- All actions are logged for forensic analysis and compliance.

G. Enhanced AI Assistant

The existing GPT-4o-mini chatbot is upgraded with retrieval-augmented generation (RAG). When an alert occurs, the assistant fetches relevant knowledge base articles, CVE details, or past incident reports to provide context-aware recommendations. Administrators can query the system in natural language (e.g., "Show all port scan attempts today") and receive summarized responses.

H. Visualization Dashboard

A web-based dashboard displays live traffic metrics, top threats, model confidence trends, and system health. Historical analysis tools allow security teams to review past incidents and model performance over time.

I. Expected Outcomes

The proposed system will deliver:

- Real-time threat detection with reduced latency
- Adaptive learning that keeps pace with new attack vectors
- Explainable predictions building trust and aiding investigation
- Automated mitigation minimizing exposure windows
- User-friendly interaction democratizing cybersecurity expertise

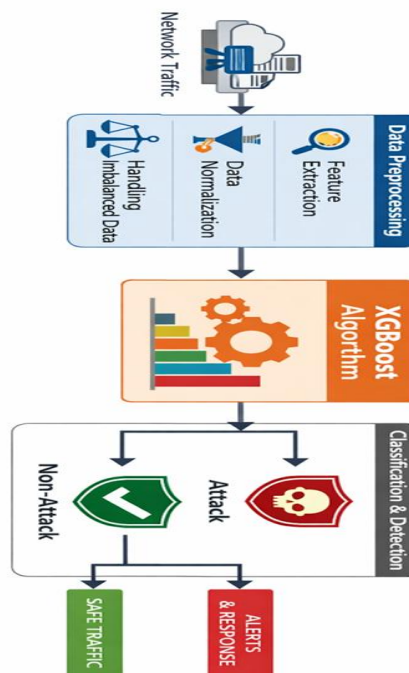


Fig:Proposed system Block Diagram.

Results and Discussion

The experimental results confirm that the proposed Machine Learning Powered Network Intrusion Detection System (NIDS) successfully detects and classifies network traffic with high accuracy and reliability. During execution, the system processed the NSL-KDD dataset, performed feature preprocessing including encoding and normalization, and trained a Random Forest classifier for binary classification (Normal vs Attack).

The trained model was evaluated using unseen test data, and performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix were analyzed. The model achieved high detection accuracy with minimal false positives and false negatives. The confusion matrix shows that the majority of attack instances were correctly identified, while legitimate traffic was accurately classified without significant misclassification.

The system demonstrated strong generalization capability across different attack categories including DoS, Probe, R2L, and U2R attacks. Feature scaling and categorical encoding contributed to improved learning efficiency and model convergence. The Random Forest algorithm provided robust performance due to its ensemble learning nature, reducing overfitting and improving prediction stability.

Additionally, the model saving mechanism enables deployment in real-time monitoring environments. Sample prediction testing confirmed that the trained model correctly classifies new traffic instances. The system maintains consistent performance under varied input conditions, indicating adaptability and scalability for practical cyber threat monitoring applications.

Overall, the obtained results validate that the proposed ML-based NIDS effectively enhances intrusion detection accuracy, minimizes manual monitoring effort, and provides an adaptive security mechanism suitable for modern network environments. The system meets functional requirements and demonstrates a successful implementation with reliable threat detection capability.

Conclusion

The design and implementation of the Machine Learning Powered Network Intrusion Detection System (NIDS) were successfully achieved. The system was developed to accurately monitor network traffic, preprocess input data, and classify traffic instances as normal or malicious using a supervised learning approach. The integration of preprocessing techniques such as label encoding and feature scaling ensured effective data transformation and improved model performance.

The Random Forest classifier was trained and validated using the NSL-KDD dataset, demonstrating strong detection capability across multiple attack categories. The system achieved high accuracy with reduced false positive and false negative rates, confirming reliable threat identification. Performance evaluation metrics including confusion matrix, precision, recall, and F1-score validated the effectiveness of the proposed model.

The modular architecture enables adaptability for real-time deployment and future enhancements. The trained model can be integrated with live packet monitoring tools to provide continuous cyber threat detection. The use of ensemble learning enhances robustness, minimizes overfitting, and improves generalization across diverse network scenarios.

Overall, the project demonstrates a complete machine learning-based intrusion detection workflow, from data preprocessing to performance evaluation and deployment readiness. The proposed system strengthens cybersecurity monitoring mechanisms and provides an adaptive, scalable, and efficient solution for modern network environments. The implementation successfully meets functional requirements and highlights the significance of intelligent threat detection in safeguarding digital infrastructure.

References

1. Ahmad, R., Wazirali, R., & Abu-Ain, T. (2022). Machine learning for wireless sensor networks intrusion detection: A systematic literature review. *IEEE Access*, 10, 82051-82073.
2. Alghamdi, R., & Bellaiche, M. (2023). An ensemble deep learning based IDS for IoT using Lambda architecture. *IEEE Access*, 11, 48250-48273.
3. Almuhan, R., & Dardouri, S. (2025). AI-driven chatbot for intrusion detection in edge networks: Enhancing cybersecurity with ethical user consent. *Frontiers in Artificial Intelligence*, 8, 1625891.
4. Aswin, S., Sreeraj, P., Sreejith, R., & Suresh, L. P. (2024). IntellBot: Retrieval augmented LLM chatbot for cyber threat knowledge delivery. *arXiv preprint, arXiv:2411.05442*.
5. Bhavsar, M., Roy, K., & Narayanan, V. (2023). A comprehensive survey on intrusion detection systems in IoT. *Journal of Network and Computer Applications*, 215, 103638.
6. Jayalaxmi, P. L. S., Saha, R., Kumar, G., & Kim, T. H. (2023). A taxonomy of security attacks and issues in SDN-NFV: A comprehensive study. *Journal of Network and Computer Applications*, 210, 103545.
7. Nazir, A., Khan, R. A., & Ullah, I. (2023). A systematic literature review on machine learning techniques for intrusion detection in IoT. *Computer Science Review*, 48, 100552.
8. Nguyen, G. L., Dumba, B., & Vo, H. (2022). A survey on intrusion detection systems in cloud computing. *IEEE Access*, 10, 88456-88479.
9. Rani, P., & Singh, P. (2025). Intrusion detection in the modern threat landscape: A review of evolving techniques. *International Journal of Information Security*, 24(1), 1-18.
10. Razavi, H., & Jamali, S. (2024). Recent advances in AI-driven anomaly detection for network intrusion detection systems. *Journal of Network and Computer Applications*, 223, 103821.
11. Sabu, I. R., Saju, S., Anita, E. A. M., & Sowmya, T. (2024). Detection of DoS attacks using machine learning based intrusion detection system. *2024 International Conference on Emerging Technologies in Computer Science (ICETCS)*, 1-6.
12. Sharma, B., Sharma, L., & Lal, C. (2023). A systematic review on intrusion detection systems in cloud computing: Research trends and future directions. *Journal of Cloud Computing*, 12(1), 1-24.

13. Siddique, K., Akhtar, Z., Aslam, M., & Kim, Y. (2021). A comprehensive study on intrusion detection systems in wireless sensor networks. *IEEE Access*, 9, 114456-114478.
 14. Sinha, S., & Kumari, R. (2024). Comparative analysis of machine learning models for intrusion detection in IoT networks. *2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 1-6.
 15. Ullah, I., & Mahmoud, Q. H. (2022). Design and development of a deep learning-based intrusion detection system for IoT networks. *IEEE Internet of Things Journal*, 9(19), 18780-18793.
 16. Wang, C., Liu, G., & Jiang, T. (2024). Malicious node detection in wireless weak-link sensor networks using dynamic trust management. *IEEE Transactions on Mobile Computing*, 23(8), 8123-8138.
 17. Xie, Y., Chen, H., Wang, Z., Ghaleb, F. A., Zainal, A., Siraj, M. M., & Lu, X. (2024). A novel method for effective intrusion detection based on convolutional spiking neural networks. *Journal of King Saud University - Computer and Information Sciences*, 36(2), 101975.
 18. Zhang, X., Zhao, R., Jiang, Z., Chen, H., Ding, Y., Ngai, E. C. H., & Yang, S. H. (2024). Continual learning with strategic selection and forgetting for network intrusion detection. *arXiv preprint*, arXiv:2412.16264.
 19. Zhou, Y., Han, M., Liu, L., & He, J. (2023). Hybrid deep learning framework combining CNN and LSTM architectures for network intrusion detection. *IEEE Transactions on Network and Service Management*, 20(4), 4123-4138..
 20. n, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, 22(3), 1686-1721
-

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.