WAVEPULSE DEFENDER: MULTI-SCALE AI FRAMEWORK FOR PERSISTENT NETWORK THREAT ANALYSIS

Goutham Kunamalla[1], Kokkula Sandeep[2], Emmadi Angel[2], Merugu Raghavi[2], Palabindela Akhil[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2]Vaagdevi Engineering College, Bollikunta, Warangal, 506005, Telangana, India.

ABSTRACT

Modern digital infrastructures including enterprise networks, cloud computing platforms, IoT ecosystems, and online service environments continuously generate vast volumes of real-time data such as network packets, authentication logs, and system performance metrics. This rapid data generation necessitates efficient, intelligent, and real-time security monitoring mechanisms. Conventional security approaches rely on manual log analysis, rule-based intrusion detection systems, and static threshold techniques, which are time-consuming, labor-intensive, and unable to adapt to evolving cyber threats. As a result, they often lead to delayed responses, high false positive rates, limited scalability, and increased vulnerability to sophisticated attacks. To address these challenges, this work proposes an automated and scalable AI-driven security framework for real-time anomaly detection and authentication threat analysis. Initially, machine learning models such as K-Nearest Neighbor (KNN) and Support Vector Classifier (SVC) are employed to learn network behavior patterns and distinguish between normal and malicious activities. However, these models exhibit limitations including high computational complexity, sensitivity to feature scaling, and inefficiency with large-scale or probabilistic data. To overcome these issues, a Naive Bayes Classifier (NBC) is adopted as the primary probabilistic inference model. By leveraging Bayesian decision theory and modeling conditional feature dependencies, NBC enables efficient threat probability estimation with reduced computational overhead and improved scalability for high-dimensional datasets. The system integrates data preprocessing, class balancing, multi-model training, and deployment through a Flask-based web interface. Performance evaluation using accuracy, precision, recall, and F1-score demonstrates reliable anomaly detection, validating the framework as a robust and efficient solution for real-time security intelligence.

Keywords: Anomaly Detection, Cybersecurity, Machine Learning, Naive Bayes, Intrusion Detection, IoT, Cloud Security.

1. INTRODUCTION

In today's rapidly advancing technological environment, sensors play a crucial role in supplying essential data for decision-making across a wide range of applications. Maintaining the reliability and accuracy of this sensor data is vital, as even minor inconsistencies can result in significant errors and potential system failures [1]. This concern becomes even more critical within embedded systems and Internet of Things (IoT) environments, where sensors function under diverse and often harsh conditions. Ensuring data integrity in such settings is particularly challenging due to continuous exposure to dynamic and unpredictable environments [2]. As shown as Figure 1 Conventional anomaly detection techniques, including signature-based Intrusion Detection Systems (IDS), have demonstrated effectiveness in identifying known threats.

However, they face limitations when dealing with new and sophisticated attack patterns. With the continuous evolution of cyber threats and system vulnerabilities, there is an increasing demand for advanced

detection mechanisms that can adapt to emerging risks while maintaining system reliability and security [3]. This research addresses the identified challenges by proposing a novel approach that incorporates discrete wavelet transforms (DWT) within microcontroller-based systems for real-time anomaly detection and fault isolation [4]. Wavelet transforms, especially DWT, are widely recognized as effective tools in signal processing, enabling the decomposition of signals into multiple frequency components for localized analysis.

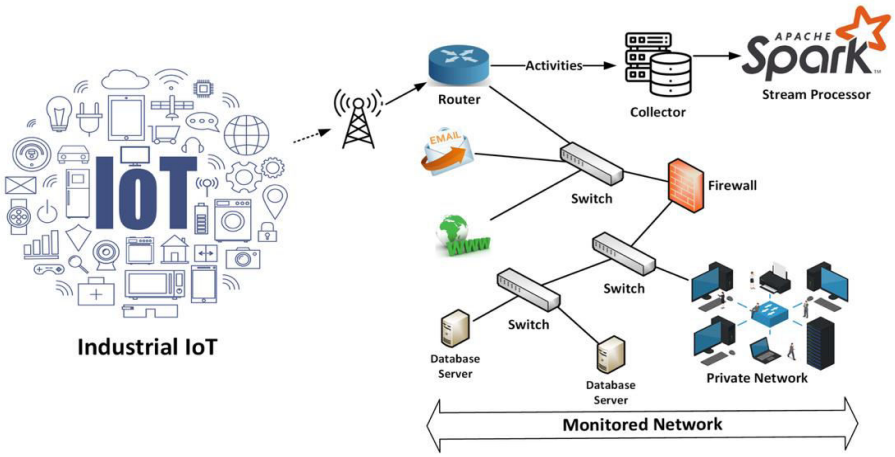

Figure 1: Monitoring of IoT-based network traffic flows.

The Haar wavelet, a member of the DWT family, is selected for this study due to its simplicity and efficiency in handling non-stationary signals. It facilitates the detection of both transient and persistent anomalies in sensor data by providing precise time and frequency localization. Furthermore, the computational simplicity of Haar wavelets makes them highly suitable for real-time implementation in resource-constrained environments such as microcontrollers [5]. To enhance anomaly detection, Euclidean distance is employed alongside DWT to measure deviations between transformed sensor data and a reference model, offering an efficient and straightforward mechanism for fault identification.

2. LITERATURE SURVEY

Paracha, et al. [6] proposed an investigation model based on AI/ML techniques that can analyse network traffic and behavioural patterns to identify any prior or potential cyberattacks. The proposed AI-based network forensics model speeds up the investigation process, boosting network monitoring without human intervention. This also aimed to provide timely and accurate information to network administrators for quick and effective decisions, enabling them to avoid and circumvent future cyberattacks. Chiriac, et al. [7]. presented an approach for implementing a generic model of a network-based intrusion detection system for Industry 4.0 by integrating the computational advantages of the Nvidia Morpheus open-source AI framework. The solution is modularly built with two pipelines for data analysis. The pipelines used a pre-trained XGBoost (eXtreme Gradient Boosting) model that achieved an accuracy score of up to 90%. The proposed IDS has a fast rate of analysis, managing more than 500,000 inputs in almost 10 s, due to the application of the federated learning methodology.

Abuali, et al. [8] aimed to propose a SVM based deep learning system that will classify the data extracted from servers to determine the intrusion incidents on social media. To implement deep learning-based IDSs for multiclass classification, the CSE-CIC-IDS 2018 dataset has been used for system evaluation. The CSE-CIC-IDS 2018 dataset was subjected to several preprocessing techniques to prepare it for the training phase.

The proposed model has been implemented in 100,000 instances of a sample dataset. Toldinas, et al. [9]. proposed a novel approach for network intrusion detection using multistage deep learning image recognition. The network features are transformed into four-channel (Red, Green, Blue, and Alpha) images. The images then are used for classification to train and test the pre-trained deep learning model ResNet50. The proposed approach is evaluated using two publicly available benchmark datasets, UNSW-NB15 and BOUN Ddos. On the UNSW-NB15 dataset, the proposed approach achieves 99.8% accuracy in the detection of the generic attack. On the BOUN DDos dataset, the suggested approach achieves 99.7% accuracy in the detection of the DDos attack and 99.7% accuracy in the detection of the normal traffic.

Zhang, et al. [10] proposed a novel algorithm based on both wavelet leader multifractal analysis (WLM) and machine learning (ML) principles. In earlier research on unmanned aerial systems (UAS), intrusion detection systems (IDS) based on multifractal (MF) spectral analysis have been used to provide accurate MF spectrum estimations of network traffic. Such an estimation is then used to detect and characterize flooding anomalies that can be observed in an unmanned aerial vehicle (UAV) network. However, the previous contributions have lacked the consideration of other types of network intrusions commonly observed in UAS networks, such as the man in the middle attack (MITM). In this work, this promising methodology has been accommodated to detect a spoofing attack within a UAS. Ali, et, al. [11] implemented various deep learning models, including multilayer perceptron (MLP), convolutional neural network (CNN), and long short-term memory (LSTM), alongside traditional machine learning algorithms such as logistic regression, naive Bayes, random forest, K-nearest neighbors, and decision trees.

Mao, et, al. [12] employed a dynamic routing mechanism to map sample feature vectors into robust class vector representations, achieving superior generalization when detecting unseen attack types. Compared to existing FCN–Transformer models, MFEI-IDS incorporates inductive learning to handle data imbalance and small-sample scenarios. Experiments on ISCX 2012 and CIC-IDS 2017 datasets show that MFEI-IDS outperforms mainstream IDS methods in accuracy, precision, recall, and F1-score, excelling in cross-dataset validation and demonstrating strong generalization capabilities. Mari, et al. [13] demonstrated a way to create adversarial instances of network traffic that can be used to evade detection by a machine learning-based IDS. Moreover, this traffic can be used for training in order to improve performance in the case of new attacks. Thus, a generative adversarial network (GAN) i.e., an architecture based on a deep-learning algorithm capable of creating generative models was implemented.

Thapa, et al. [14] proposed an IDS using different ML and DL models. Both ML and DL models achieved an accuracy of 99% on the CIDDS dataset with a high detection rate, low false alarm rate, and relatively low training costs. Feature importance was also studied using the Classification and regression tree (CART) model. Mohammad et, al. [15] highlighted how the proposed methods of deep learning-based intrusion detection can be seamlessly integrated into cybersecurity frameworks, enhancing the ability to detect and mitigate sophisticated network attacks. The outcomes of this study have shown that the intrusion detection models have achieved high accuracy (up to 91% for the augmented CIC-IDS-2017 dataset) and are strongly influenced by the quality and quantity of the dataset used.

3. PROPOSED SYSTEM

The proposed system architecture converts raw network traffic into meaningful and discriminative features through structured preprocessing, followed by machine learning–based classification for detecting anomalies and authentication failures with high accuracy. The workflow is designed to capture variations in network behaviour, identify deviations in authentication patterns, and analyse packet-level characteristics

that are often difficult to detect using traditional approaches. By integrating preprocessing techniques with multiple machine learning models, the system improves detection capability, minimizes false alarms, and supports efficient real-time intrusion detection, as illustrated in Figure 2.

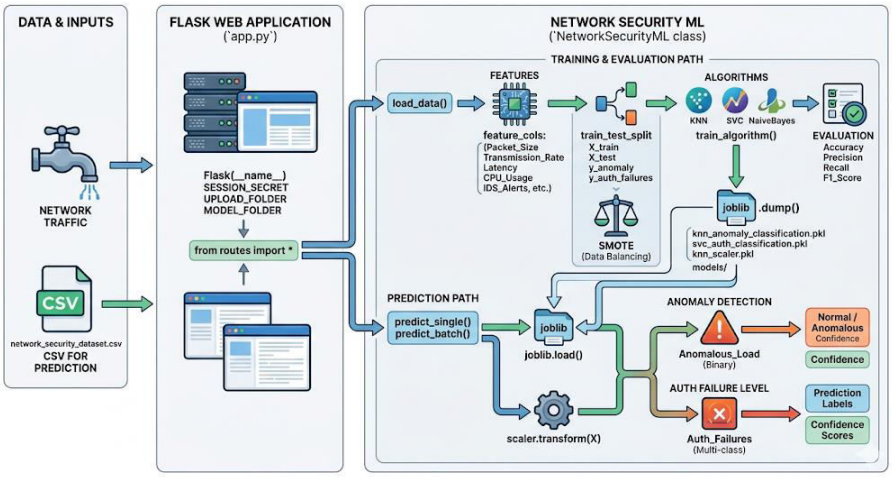

Figure 2: Proposed system architecture

Raw Network Data Acquisition: The process begins with the collection of raw network traffic and input datasets. Data is obtained from network traffic streams and CSV files used for prediction. These inputs include packet-level information and security-related attributes, forming the base for further analysis. The collected data represents various network conditions, including normal operations and potential anomalous activities.

Data Preprocessing and Cleaning: After data acquisition, preprocessing is performed to ensure consistency and quality. This stage removes incomplete or redundant entries and structures the dataset into a usable format. Feature columns such as packet size, transmission rate, latency, CPU usage, and IDS alerts are prepared for analysis. The data is also normalized and organized to maintain uniformity across all attributes, enabling effective model training and prediction.

Feature Engineering and Dataset Preparation: Relevant features are identified and structured to represent network behavior effectively. The system splits the dataset into training and testing subsets using train-test split techniques. Additionally, class imbalance is addressed using SMOTE (Synthetic Minority Over-sampling Technique), ensuring balanced representation of normal and anomalous classes. This step enhances the model's ability to learn patterns without bias toward dominant classes.

Model Training Using Machine Learning Algorithms: The prepared dataset is used to train multiple machine learning models, including KNN, SVC, and NBC. Each model learns to distinguish between normal and abnormal network activities based on the extracted features. The training process includes algorithm execution and performance evaluation using metrics such as accuracy, precision, recall, and F1-score. Trained models and scalers are then stored using joblib for later use in prediction.

Real-Time Prediction and Intrusion Detection: In the prediction phase, the system utilizes the Flask-based web application to process incoming data. Functions such as single and batch prediction are applied to evaluate new inputs. The stored models are loaded, and feature scaling is performed before classification. The system identifies anomalies through binary classification and detects authentication failures using

multi-class classification. Outputs include prediction labels and confidence scores, indicating whether the activity is normal or anomalous.

System Deployment and Integration: The entire pipeline is integrated within a Flask web application, enabling user interaction and real-time processing. The application manages data input, model loading, and prediction handling through defined routes and configurations. This integration ensures seamless communication between data processing, model execution, and result visualization.

Continuous Monitoring and Feedback: The system continuously processes incoming data and updates predictions based on model outputs. Detected anomalies and classification results contribute to ongoing monitoring, allowing consistent evaluation of network behavior. This ensures that the system maintains reliable detection performance during real-time operation.

4. RESULTS ANALYSIS

The results description section provides a clear summary of the key findings obtained from the study or experiment. It highlights the main outcomes, patterns, or trends observed in the collected data. This section focuses on presenting facts without interpretation, often supported by graphs, or figures. It helps readers understand what the data reveals in a structured and concise manner. Additionally, it ensures that the results are organized logically for easy comprehension. It forms the foundation for further analysis and discussion in the following sections.

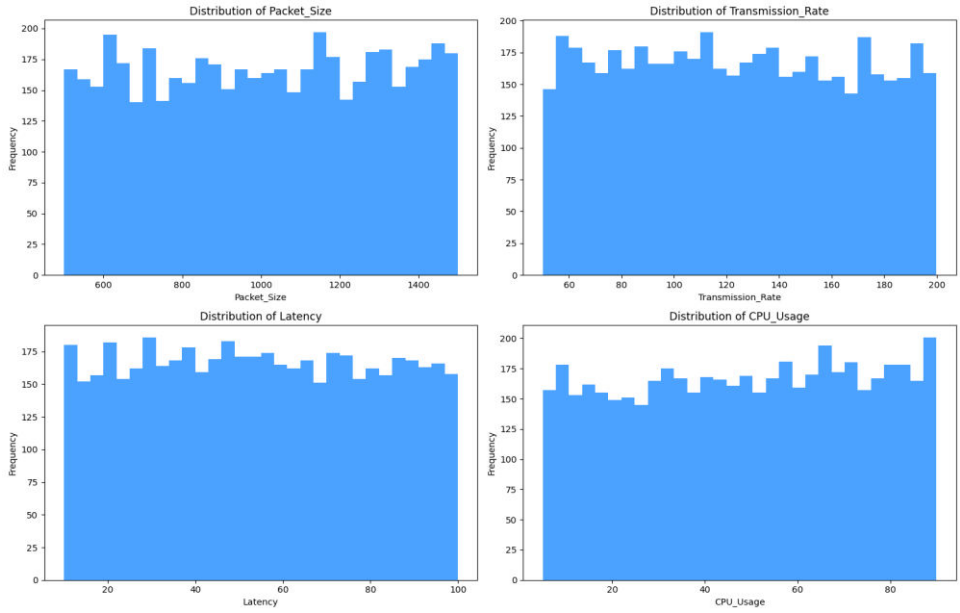

Figure 3: Visualizing Key Feature Distributions using Histogram.

Figure 3 depicts the distribution of important network traffic features using histogram-based visualizations. The figure includes histograms for packet size, transmission rate, latency, and CPU usage. Packet size values range approximately between 500 and 1500 units, while transmission rate values range between about 50 and 200 units. Latency values are distributed between roughly 10 and 100 units, and CPU usage ranges between approximately 5 and 90 percent. These distributions help researchers understand how network characteristics vary across the dataset. Such analysis is useful for identifying feature patterns and detecting potential outliers before model training.

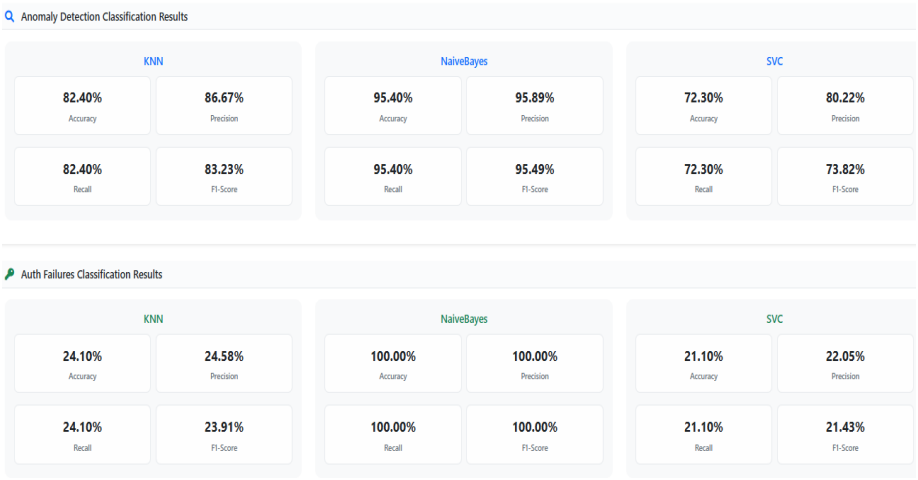

Figure 4: Model Performance Evaluation Interface after Training the models.

Figure 4 presents the evaluation results obtained after training the selected machine learning models. For anomaly detection classification, Naive Bayes achieved an accuracy of 95.40 percent, precision of 95.89 percent, recall of 95.40 percent, and F1 score of 95.49 percent. The KNN model achieved an accuracy of 82.40 percent with an F1 score of 83.23 percent, while the SVC model achieved an accuracy of 72.30 percent with an F1 score of 73.82 percent. In authentication failure classification, Naive Bayes achieved 100 percent accuracy, precision, recall, and F1 score.

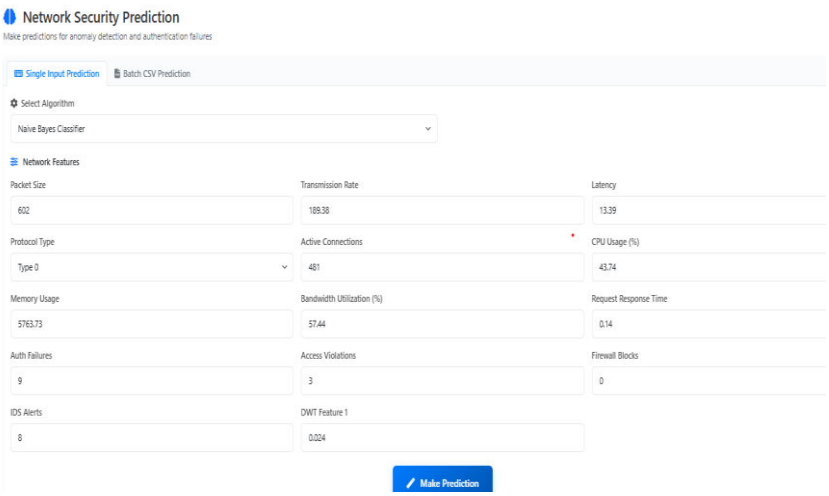

Figure 5: Network Security Prediction Interface for Single Input Analysis.

Figure 5 illustrates the prediction interface used for performing single input analysis in the network security application. The interface allows users to provide network traffic parameters including packet size, transmission rate, latency, protocol type, active connections, CPU usage, memory usage, bandwidth utilization, request response time, authentication failures, access violations, firewall blocks, IDS alerts, and DWT feature values. These input parameters are processed by the trained machine learning model to determine the security status of the network activity. The interface supports real time prediction of anomalous network behavior and authentication failure levels.

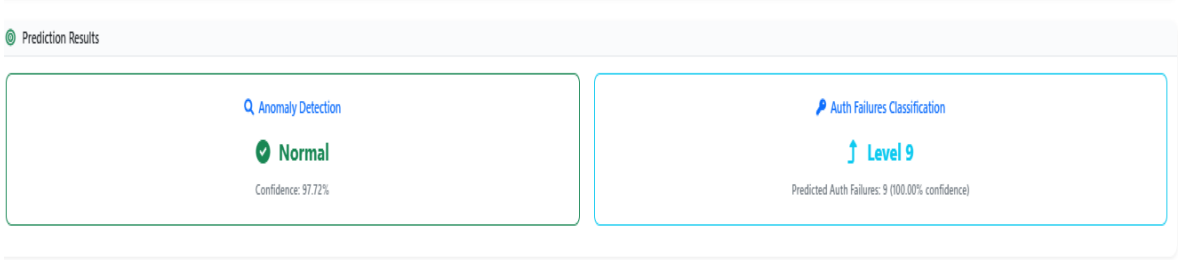

Figure 6: Output of the Test Data.

Figure 6 shows the prediction results generated by the system after processing the provided network traffic data. The anomaly detection module predicts the network activity as normal with a confidence level of approximately 97.72 percent. In addition, the authentication failure classification module predicts level 9 with 100.00 percent confidence. These prediction results demonstrate the ability of the trained machine learning model to analyse network features and provide security related insights.

5. CONCLUSION

The research presents a robust Flask-based web application that seamlessly integrates machine learning for intelligent network security analysis. The system leverages a network dataset enriched with wavelet-derived attributes, particularly DWT_Feature_1, to strengthen anomaly detection and authentication failure classification. Multiple classifiers, including KNN, SVC, and NBC, are evaluated, with NBC achieving superior performance such as 95.40% accuracy, precision, recall, and F1-score for anomaly detection, and a perfect 100% across all metrics for authentication failure classification. In contrast, KNN and SVC demonstrate comparatively lower performance, achieving 82.40% and 72.30% anomaly detection accuracy, and only 24.10% and 21.10% accuracy for authentication failure prediction. Performance optimization strategies include SMOTE-based class balancing to address dataset skew (1,354 anomalous vs. 3,646 normal samples), StandardScaler-driven feature normalization for stable learning, and joblib-based model persistence for efficient deployment. These enhancements collectively improve predictive reliability, computational efficiency, and real-time operational readiness of the intrusion detection framework.

REFERENCES

[1]. Li, D.; Wang, Y.; Wang, J.; Wang, C.; Duan, Y. Recent Advances in Sensor Fault Diagnosis: A Review. Sens. Actuators A Phys. 2020, 309, 111990.

[2]. Fatima, N.; Riaz, S.; Ali, S.; Khan, R.; Ullah, M.; Kwak, D. Sensors Faults Classification and Faulty Signals Reconstruction Using Deep Learning. IEEE Access 2024, 12, 100544–100558.

[3]. Yang, J.W.; Lee, Y.D.; Koo, I.S. Convolutional Autoencoder-Based Sensor Fault Classification. Int. Conf. Ubiquitous Future Netw. 2018, 2018, 865–867

[4]. Jiang, X.; Zhang, X.; Zhang, Y. Establishment and Optimization of Sensor Fault Identification Model Based on Classification and Regression Tree and Particle Swarm Optimization. Mater. Res. Express 2021, 8, 085703.

[5]. Jiang, C.Y.; Li, L.C.; Ye, C.L.; Yu, S.Y. Research on Sensor Fault Identification Based on Improved 1-v-r SVM Classification Method. Int. J. Adv. Media Commun. 2016, 6, 235–245.

[6]. Paracha, M.A.; Jamil, S.U.; Shahzad, K.; Khan, M.A.; Rasheed, A. Leveraging AI for Network Threat Detection—A Conceptual Overview. Electronics 2024, 13, 4611. https://doi.org/10.3390/electronics13234611

[7]. Chiriac, B.-N.; Anton, F.-D.; Ioniță, A.-D.; Vasilică, B.-V. A Modular AI-Driven Intrusion Detection System for Network Traffic Monitoring in Industry 4.0, Using Nvidia Morpheus and Generative Adversarial Networks. Sensors 2025, 25, 130. https://doi.org/10.3390/s25010130

[8]. Abuali, K.M.; Nissirat, L.; Al-Samawi, A. Advancing Network Security with AI: SVM-Based Deep Learning for Intrusion Detection. Sensors 2023, 23, 8959. https://doi.org/10.3390/s23218959

[9]. Toldinas, J.; Venčkauskas, A.; Damaševičius, R.; Grigaliūnas, Š.; Morkevičius, N.; Baranauskas, E. A Novel Approach for Network Intrusion Detection Using Multistage Deep Learning Image Recognition. Electronics 2021, 10, 1854. https://doi.org/10.3390/electronics10151854

[10]. Zhang, R.; Condomines, J.-P.; Lochin, E. A Multifractal Analysis and Machine Learning Based Intrusion Detection System with an Application in a UAS/RADAR System. Drones 2022, 6, 21. https://doi.org/10.3390/drones6010021

[11]. Ali, M.L.; Thakur, K.; Schmeelk, S.; Debello, J.; Dragos, D. Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study. Appl. Sci. 2025, 15, 1903. https://doi.org/10.3390/app15041903

[12]. Mao, J.; Yang, X.; Hu, B.; Lu, Y.; Yin, G. Intrusion Detection System Based on Multi-Level Feature Extraction and Inductive Network. Electronics 2025, 14, 189. https://doi.org/10.3390/electronics14010189

[13]. Mari, A.-G.; Zinca, D.; Dobrota, V. Development of a Machine-Learning Intrusion Detection System and Testing of Its Performance Using a Generative Adversarial Network. Sensors 2023, 23, 1315. https://doi.org/10.3390/s23031315

[14]. Thapa, N.; Liu, Z.; KC, D.B.; Gokaraju, B.; Roy, K. Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems. Future Internet 2020, 12, 167. https://doi.org/10.3390/fi12100167

[15]. Mohammad, R.; Saeed, F.; Almazroi, A.A.; Alsubaei, F.S.; Almazroi, A.A. Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach. Systems 2024, 12, 79. https://doi.org/10.3390/systems12030079