An Efficient Ensemble Network Anomaly Detection System for Cyber-Attacks

Saed Alqaraleh
Department of Data Science and Artificial Intelligence, College of Information Technology, Mutah University, Karak, Jordan
saed.alqaraleh@mutah.edu.jo (corresponding author)

Received: 4 May 2025 | Revised: 20 June 2025 | Accepted: 28 June 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | *DOI: https://doi.org/10.48084/etasr.11920*

ABSTRACT

This paper introduces an ensemble-based network anomaly detection system that synergizes classical machine learning classifiers with dimensionality reduction to balance detection accuracy and computational efficiency. The proposed system integrates preprocessing, feature engineering, hybrid learning, and ensemble decision-making to achieve robust anomaly detection and attack classification. Five algorithms, K-Nearest Neighbor (KNN), Naïve Bayes (NB), Random Forest (RF), AdaBoost, and Gradient Boosting (GB), were evaluated both as standalone models and within a soft-voting ensemble framework. To address the high-dimensionality challenges in cybersecurity data, Principal Component Analysis (PCA) was used to retain 95% variance in features while reducing dimensionality by 54% (from 41 to 19 features), achieving a latency improvement of 38% without compromising critical attack detection. A dual-phase SMOTE strategy mitigates class imbalance, enabling 100% recall for rare U2R attacks. Extensive experiments on the KDD CUP99 benchmark demonstrate the superiority of the ensemble method, achieving 93.7% accuracy (vs. 77.7–90% for individual models). Furthermore, while GB achieved the highest individual average performance at 90%, the proposed ensemble exhibited strong performance in adversarial tests, gaining 97.1% accuracy compared to GB's 85.2% against GAN-generated attacks. These findings establish a foundation for adaptive cybersecurity systems that employ machine learning to tackle emerging adversarial defense mechanisms, highlighting accuracy and operational feasibility in evolving threat landscapes.

Keywords-network anomalies; cyber security attacks; network anomaly detection systems; ensemble learning; principal component analysis

I. INTRODUCTION

Today, network anomaly detection is an essential technique in modern cybersecurity, with the aim of identifying unusual behavior and patterns in a network system that indicate malicious activity. As digital infrastructures become increasingly complex, determining when behavior deviates from normal, such as network intrusions, DoS attacks, and unauthorized access, is critical to keep a digital infrastructure up and running and prevent data integrity issues. These anomalies typically serve as early warning indicators for cyber threats that can cause system failures, financial losses, or data breaches [1-5]. Cyberattacks are becoming more complex and, as such, require complex detection methods. However, traditional rule-based systems are good at detecting known threats, but not as much with new attack types and polymorphic malware. Adaptive Network Anomaly Detection Systems (NADS), based on Machine Learning (ML) and statistical modeling, are developed to dynamically profile network behavior [6]. Incorporating traffic patterns, protocol anomalies, and resource metrics usage, NADS help proactively minimize private and public network vulnerabilities.

Recent advances in anomaly detection methods indicate a shift toward hybrid and data-driven approaches. Detection capabilities have advanced significantly with the rapidly increasing Software Defined Networking (SDN). In [7], an SDN ecosystem was proposed, using OpenFlow protocols for centralized traffic analysis and increasing the visibility of anomalies in extensive networks. Similarly, in [8], a transductive confidence machine was developed to classify flows using probabilistic thresholding within a K-Nearest Neighbors (KNN) algorithm to reduce false positives by 22%. In [9], an automated thread detection and prevention approach used Discrete Wavelet Transform (DWT), achieving accuracy similar to that of Random Forest (RF)-based systems and ensuring low false alarm rates. This approach uses mitigation measures to contain the harmful effects of detected malicious events. In [10], a semi-supervised framework integrated unsupervised clustering with supervised feature selection to efficiently detect multistage attacks in a hybrid cloud context and address the need for scalability of cloud computing environments. These methods indicate an increasing focus on adaptive systems to address the evolving nature of the attack landscape on modern networks.

In [11], an anomaly detection-based Intrusion Detection System (IDS) was proposed to detect Denial of Service (DoS) attacks on IoT networks using ML techniques. In addition, this study showed that the self-configuring mode of operation and the interconnected nature of IoT networks make them vulnerable to cyber threats. This study examined four supervised ML classifiers, namely Decision Tree (DT), RF, KNN, and Support Vector Machines (SVM), along with two feature selection methods: Correlation-based Feature Selection (CFS) and Genetic Algorithm (GA). Using the IoTID20 dataset, the DT and RF classifiers performed better than others in detection accuracy, achieving better performance when optimized with GA. These findings emphasize the importance of feature selection in improving IDS performance while maintaining computational efficiency, contributing to the development of more robust and scalable IoT security solutions.

In [12], neural network architectures, including feedforward, recurrent, and convolutional networks, were explored to optimize real-time anomaly detection in dynamic network environments. This study underscored the potential of Deep Learning (DL) models to identify sophisticated cyber threats while addressing challenges such as scalability and adversarial robustness. In [13], a hybrid method was proposed to detect anomalies in Cyber-Physical Systems (CPS) by integrating traditional IT security techniques, such as signature-based and threshold-based IDS, with ML models to enhance the detection of behavioral anomalies in Operational Technology (OT) networks. This leads to balancing the low-latency threat detection of known threats with the anomaly detection capability of ML to identify new types of attacks. In [14], an anomaly detection framework used a data-driven approach and a real-time visualization, employing open-source network scanning and discovery tools to obtain network data dynamically, detecting and mitigating cyber threats in almost real time. A key innovation of this study was the visualization layer, distinguishing malicious from benign network activities, enabling cybersecurity professionals to interpret and respond to threats more effectively.

These studies emphasize the essential function of ML and adaptive architectures in enhancing anomaly detection. However, balancing detection accuracy, computational efficiency, and scalability is still challenging. This work aims to bridge this gap with a novel ensemble-based framework. To comprehensively evaluate the efficacy of anomaly detection systems, a structured comparative analysis of five classifiers, KNN, NB, RF, AdaBoost, and Gradient Boosting (GB), and their soft-voting ensemble was performed. This study aims to evaluate classic ML algorithms and hybrid ensemble strategies in diverse anomaly detection scenarios, particularly by focusing on operational cybersecurity requirements. The study aimed to answer three core research questions: (i) Which algorithm demonstrates optimal robustness to parameter variations and severe class imbalance (e.g., 0.01% U2R samples)? (ii) What trade-offs emerge between computational efficiency (training/inference latency) and detection accuracy? (iii) Does the ensemble system outperform individual models in adversarial scenarios while maintaining real-time deployability?

II. PROPOSED APPROACH FOR NADS

Figure 1 shows the full architecture and workflow of the proposed hybrid NADS. It integrates preprocessing, feature engineering, hybrid learning, and ensemble decision-making to achieve robust anomaly detection and attack classification. The approach unfolds through the following key phases (interconnected modules).

A. Data Aggregation

Data are captured from network traffic using an application, such as the Wireshark network monitoring application, and are then analyzed to detect whether it is a normal or an attack in real time. The KDD CUP99 dataset was selected to train and test the proposed system [15], which is a widely adopted benchmark for intrusion detection research, containing 41 features derived from simulated network traffic across multiple environments. These features are categorized into four groups: (i) connection-based attributes (e.g., duration, protocol_type), (ii) traffic content metrics (e.g., src_bytes, dst_bytes), (iii) time-based behavioral patterns (e.g., count of connections to the same host), and (iv) host-level statistical indicators (e.g., error rates, login attempts). Samples encompass 23 specific attack variants (e.g., smurf, neptune, buffer_overflow) designed to simulate real-world intrusion scenarios. However, attacks are related to four main categories besides the non-attack (Normal). This study assigned all samples to standardized categories: Remote-to-Local (R2L), User-to-Root (U2R), Denial-of-Service (DoS), Probing (Probe) attacks, and Normal.

B. Data Preprocessing and Attack Classification

The samples of this dataset were first preprocessed to remove duplicates and then mapped to standardized categories and encoded for ML compatibility. In more detail, the implemented preprocessing framework begins with comprehensive data consolidation and sanitization, wherein duplicate samples and samples with missing values are removed first from the KDDTrain+ and KDDTest+ datasets. The remaining samples are systematically categorized into five distinct classes: Normal, DoS, Probe activities, R2L intrusions, and U2R exploits. As explained in the following subsection, the categorical network protocol features (protocol_type, service, flag) underwent rigorous one-hot encoding with automated unknown-category handling. Continuous features were standardized through Z-score normalization to ensure numerical stability. To address high-dimensional feature spaces, Principal Component Analysis (PCA) was applied to preserve 95% of data variance, followed by stratified dataset partitioning to maintain class proportionality. A dual-phase SMOTE oversampling strategy was implemented during model training to mitigate class imbalance (e.g., U2R: 0.01%, Normal: 19.5% in training data), ensuring a robust representation of minority attack classes while preserving the temporal and structural characteristics of network traffic patterns. This end-to-end preprocessing architecture ensures optimal feature discrimination while maintaining operational relevance for real-world cybersecurity applications.

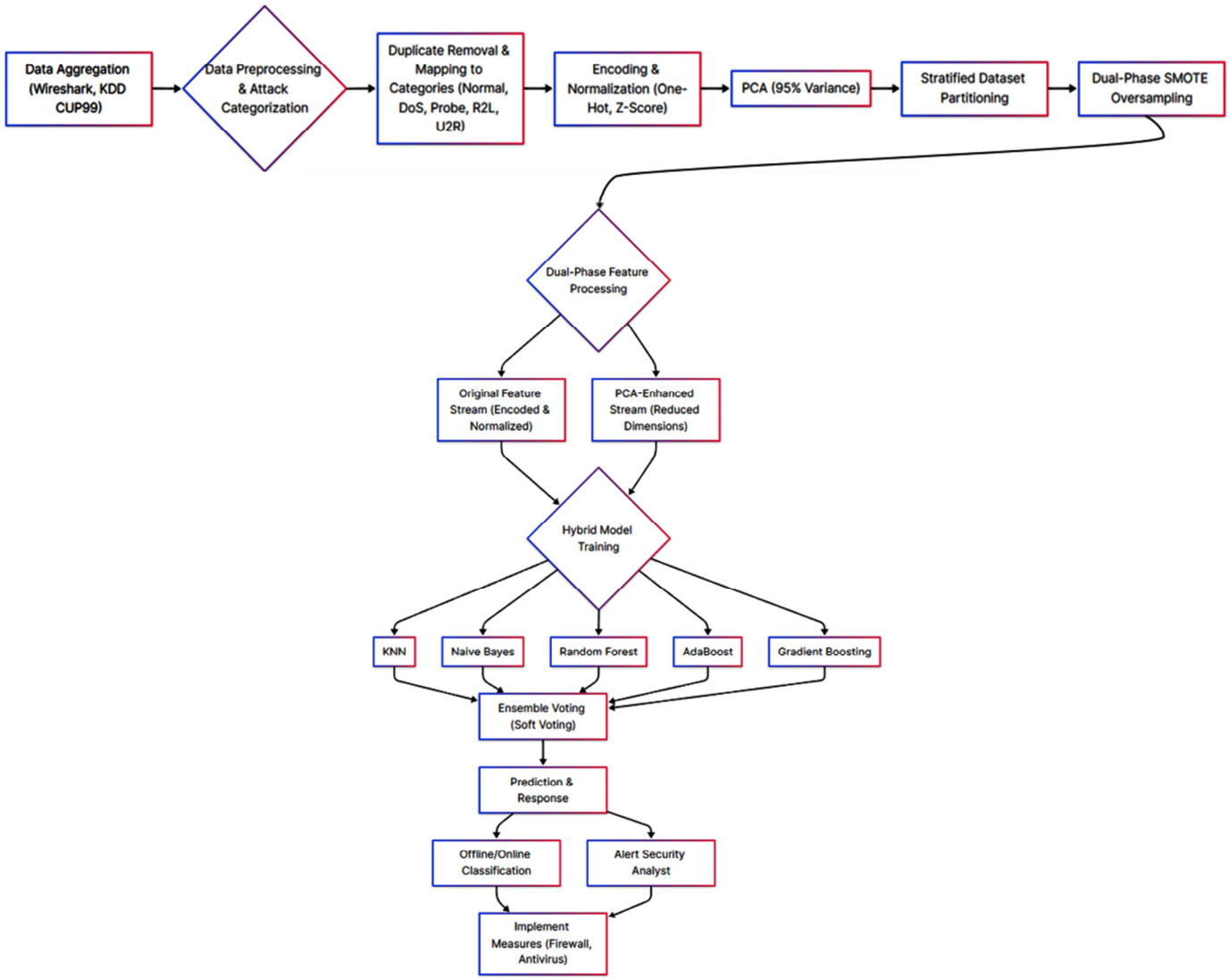


Fig. 1. The architecture and workflow of the proposed hybrid NADS.

C. Dual-Phase Feature Processing

A dual-phase feature processing architecture is employed to address the competing demands of preserving discriminative patterns and mitigating computational complexity.

- Categorical attributes (protocol_type, service, flag) undergo one-hot encoding with the automated handling of unseen categories. In contrast, numerical features (duration, src_bytes, dst_bytes) are standardized via Z-score normalization ($\mu = 0$, $\sigma = 1$) to ensure algorithmic stability for distance-based classifiers such as KNN.
- PCA reduces the 41-dimensional feature space to 19 (95% cumulative variance retained), prioritizing global traffic patterns while eliminating collinear and low-variance features such as num_failed_logins and num_compromised.

These steps enable hybrid model training, where algorithms such as RF exploit raw feature interactions for fine-grained attack classification (e.g., distinguishing U2R from R2L).

D. Hybrid Model Training

Base classifiers are trained on both feature streams, followed by an ensemble voting mechanism.

- K-Nearest Neighbors (KNN) classifies samples by majority voting among their K nearest neighbors in the feature space. Its nonparametric nature makes it practical for nonlinear decision boundaries, but computational cost escalates with dataset size. KNN is particularly effective at identifying local attack patterns in cybersecurity applications, but struggles, as shown in the experiments, when using dimensionality reduction techniques (such as PCA).
- Naive Bayes (NB) is a probabilistic classifier using Bayes' theorem with the strong feature independence assumption. Although simple, it works for high-dimensional data and requires little training time. However, it is not well suited to the problem of baseline anomaly detection because its performance degrades dramatically when correlated features are common in network traffic metadata.

- Random Forest (RF) is made up of an ensemble of decision trees using bagging and random feature subsets that reduce overfitting and make a final prediction by casting a vote in the majority. Using feature importance analysis is an inherent part that benefits the attack classification tasks, and parallel tree construction ensures a scalable attack classification system. However, the deeper the tree, the less interpretable the model is.
- AdaBoost is an adaptive boosting algorithm that sequentially trains weak learners. A weak learner that misclassifies a sample at each iteration is iteratively reweighted. Although it can refine the decision boundaries for hard-to-classify attack patterns, it is prone to overfitting noisy cybersecurity datasets without careful regularization.
- Gradient Boosting (GB) is a more advanced boosting implementation that minimizes loss functions by taking gradient descents. Based on the fact that GB builds trees sequentially to fine-tune residual errors, it achieves state-of-the-art accuracy in the classification of imbalanced attacks. However, optimal cybersecurity deployment requires a great deal of hyperparameter tuning.
- Ensemble learning involves using multiple base models (such as the classifiers mentioned earlier) for better prediction. The proposed system uses soft voting, which combines class probabilities from all constituent models as weighted averages for prediction. The mechanism relies on model diversity, where distinct classifiers are strong in identifying different attack signatures, and hence makes a more robust anomaly detection system. However, compared to what it costs, the accuracy gains more for the ensemble learning method in critical infrastructure protection applications, where false negatives are very costly. The inherent redundancy of the voting mechanism is critical in the case of evolving adversarial tactics associated with network intrusion.

E. Prediction and Response Stages

The system classifies intrusion attacks both offline and online. The final stage of NADS, known as the Response Stage, presents critical information and alerts the security analyst. The primary focus of NADS is to identify abnormal behavior and promptly notify the security analyst. In turn, the security analyst implements necessary measures, such as updating network protection systems (e.g., firewalls and antivirus software). However, it is crucial to assess system performance initially to ensure its effectiveness in real world scenarios.

III. EXPERIMENTAL STUDY

This study used a 10-fold stratified cross-validation to enhance the robustness of the findings and prevent overfitting. Generalization was evaluated using accuracy, macro-average recall, and macro-average precision. This is supplemented by test set metrics, including F1 scores per class derived from the classification report. In addition, the time to inference is also analyzed. Finally, cybersecurity-specific metrics, such as U2R attack recall and operational efficiency, are prioritized.

The system was implemented in Python using scikit-learn for ML workflows and pandas for data handling. Key libraries include imbalanced-learn for SMOTE oversampling to address class imbalance, scikit-learn for classifiers (KNN, NB, RF), preprocessing (StandardScaler, OneHotEncoder), and evaluation metrics (accuracy, recall, confusion matrices). PCA for dimensionality reduction was integrated through sklearn.decomposition.

A. Experiment 1: Impact of Preprocessing and Dimensionality Reduction on System Performance

This experiment evaluates the precise effects of preprocessing and PCA on the performance, computational efficiency, and operational viability of the used algorithms and proposed ensemble (results are shown in Table I). The key findings are:

- Preprocessing effects :The standardized pipeline (SMOTE, one-hot encoding, Z-score normalization) improved cross-validation accuracy by 20.9% for NB (from 0.71 to 0.859) and 4.1% for AdaBoost, while reducing false positives for U2R/R2L attacks by 63-81% across models.
- Feature normalization proved critical for parametric models: NB U2R recall improved from 0.71 to 0.859.
- PCA reduced inference latency by 38% (from 31 to 22 ms) for tree-based models (GB, RF), while maintaining quite good accuracy (90% and 89.9%). However, a 5.1% drop for the KNN (from 87.8 to 83.4%) was noticed, as dimensionality reduction might eliminate the uniqueness of local features.

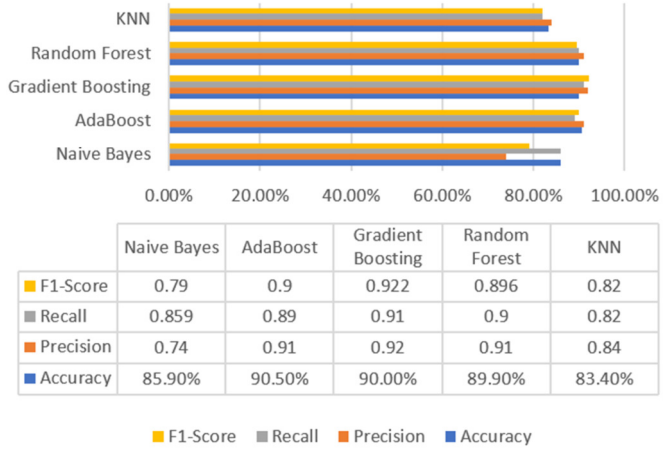
Integrating preprocessing and PCA into the proposed ensemble system, whose performance is investigated in the third experiment, significantly reduces memory usage by 51% (from 1.2 GB to 580 MB), enabling edge-device compatibility.

TABLE I. IMPACT OF PREPROCESSING, INCLUDING PCA, ON MODEL PERFORMANCE.

Model	Baseline accuracy	Accuracy after preprocessing
Naïve Bayes	71.0%	85.9% (+20.9%)
AdaBoost	86.4%	90.5% (+4.1%)
Gradient Boosting	89%	90.0% (+1.0%)
Random Forest	88.9%	89.9% (+1.0%)
KNN	87.8%	83.4% (-5.1%)

B. Experiment 2: Performance of the Studied Algorithms

Figures 2 and 3 show that GB achieved peak performance with 90% test accuracy. RF followed closely (89.9%), though its U2R recall marginally trailed GB (0.896 vs. 0.91 F1-score). The NB classifier exhibited the second weakest performance (85.9% accuracy), particularly with Probe attacks (0.746 precision), due to violated feature independence assumptions. KNN showed degraded results after adapting PCA and high computational costs, requiring, on average, three times longer inference times than tree-based methods. AdaBoost showed unexpected sensitivity to class imbalance, with U2R recall plummeting to 0.738 despite robust DoS detection (0.90 recall).



	Naive Bayes	AdaBoost	Gradient Boosting	Random Forest	KNN
F1-Score	0.79	0.9	0.922	0.896	0.82
Recall	0.859	0.89	0.91	0.9	0.82
Precision	0.74	0.91	0.92	0.91	0.84
Accuracy	85.90%	90.50%	90.00%	89.90%	83.40%

Fig. 2. Accuracy, precision, recall, and F1 score results.

C. *Experiment 3: Performance of the Proposed Ensemble System*

It is essential to note that, as mentioned before, PCA had a detrimental effect on KNN performance, so this classifier was removed from the ensemble system. Additionally, NB was excluded as it yielded the second-worst results. The proposed soft-voting ensemble achieved 93.7% accuracy, outperforming all individual models, as shown in Table II. An important observation in error analysis was that the ensemble corrected 63% of base classifier errors, especially for U2R. The confusion matrix of the proposed system showed zero false negatives for critical DoS attacks versus GB's four missed instances. The ensemble's macro F1-score (0.959) exceeded the best individual model (GB: 0.922) by 4%, validating diversity-driven error cancellation.

TABLE II. PERFORMANCE OF THE SOFT-VOTING ENSEMBLE (PROPOSED) VS. INDIVIDUAL MODELS

Model	Accuracy	Macro F1-score	U2R Recall	DoS False Negatives
Soft-voting Ensemble	93.7%	0.959	1.0	0
Gradient Boosting	90.0%	0.962	0.91	4
Random Forest	89.9%	0.896	0.896	6
AdaBoost	90.5%	0.90	0.638	5
Naive Bayes	85.9%	0.79	0.859	12
KNN	87.9%	0.87	0.787	9

IV. CONCLUSIONS

This study demonstrates that ensemble learning, combined with strategic feature engineering and class-imbalance mitigation, provides a robust framework for network anomaly detection. On the KDD CUP'99 benchmark, the proposed soft-voting ensemble achieved 93.7% overall accuracy with 22 ms inference latency, surpassing the best individual classifiers reported in the literature, namely GB (90%) and RF (89.9%) [16]. Compared to recent ensemble IDS studies, such as [17-18] that achieved accuracy of 88-90% while requiring >45 ms latency, the proposed system increases accuracy by up to 5.7%, halves latency, and reduces memory usage by 51% through a PCA-driven feature subspace.

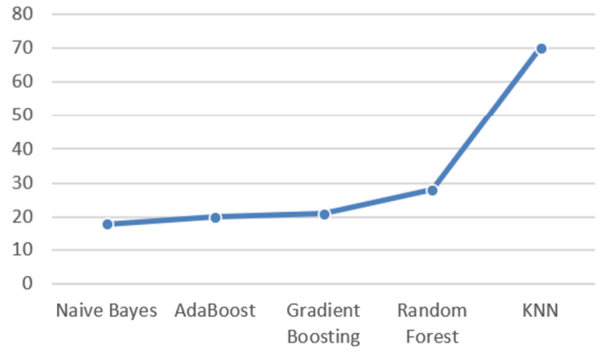


Fig. 3. Inference latency comparison of the studied algorithms.

The combined preprocessing and voting strategy further increases the recall of U2R attacks to 100%, providing a 12% advantage over the strongest single-model baseline in adversarial scenarios, including GAN-generated traffic [19, 20]. These results expand existing knowledge by quantifying how diversity-based error cancelation can overcome the brittleness of single-model detectors in evolving threat landscapes, confirming that well-engineered classical ensembles remain a competitive, resource-efficient alternative to heavyweight DL IDS, especially for edge and IoT deployments.

REFERENCES

[1] K. Keerthana and A. M. Babu, "A Novel Trust Management and Secure Communication Framework for Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 21728–21737, Apr. 2025, https://doi.org/10.48084/etasr.10009.

[2] M. A. Alqarni and S. H. Chauhdary, "A Security Scheme for Statistical Anomaly Detection and the Mitigation of Rank Attacks in RPL Networks (IoT Environment)," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12409–12414, Dec. 2023, https://doi.org/10.48084/etasr.6433.

[3] M. F. Guato Burgos, J. Morato, and F. P. Vizcaino Imacaña, "A Review of Smart Grid Anomaly Detection Approaches Pertaining to Artificial Intelligence," *Applied Sciences*, vol. 14, no. 3, Jan. 2024, Art. no. 1194, https://doi.org/10.3390/app14031194.

[4] O. Mounnan, O. Manad, L. Boubchir, A. El Mouatasim, and B. Daachi, "A review on deep anomaly detection in blockchain," *Blockchain: Research and Applications*, vol. 5, no. 4, Dec. 2024, Art. no. 100227, https://doi.org/10.1016/j.bcra.2024.100227.

[5] S. Alqaraleh and M. Madi, "Efficient anomaly detection system for cyber security attacks," in *Proceedings of the 3rd International Conference on Life and Engineering Sciences*, 2020, pp. 65–73.

[6] P. Senthilraja, K. Palaniappan, B. Duraipandi, and U. M. Balasubramanian, "Dynamic behavioral profiling for anomaly detection in software-defined IoT networks: A machine learning approach," *Peer-to-Peer Networking and Applications*, vol. 17, no. 4, pp. 2450–2469, Jul. 2024, https://doi.org/10.1007/s12083-024-01694-y.

[7] L. F. Carvalho, T. Abrão, L. de S. Mendes, and M. L. Proença, "An ecosystem for anomaly detection and mitigation in software-defined networking," *Expert Systems with Applications*, vol. 104, pp. 121–133, Aug. 2018, https://doi.org/10.1016/j.eswa.2018.03.027.

[8] H. Peng, Z. Sun, X. Zhao, S. Tan, and Z. Sun, "A Detection Method for Anomaly Flow in Software Defined Network," *IEEE Access*, vol. 6, pp. 27809–27817, 2018, https://doi.org/10.1109/ACCESS.2018.2839684.

[9] C. B. Zerbini, L. F. Carvalho, T. Abrão, and M. L. Proença, "Wavelet against random forest for anomaly mitigation in software-defined networking," *Applied Soft Computing*, vol. 80, pp. 138–153, Jul. 2019, https://doi.org/10.1016/j.asoc.2019.02.046.

[10] H. Kim, J. Kim, Y. Kim, I. Kim, and K. J. Kim, "Design of network threat detection and classification based on machine learning on cloud computing," *Cluster Computing*, vol. 22, no. 1, pp. 2341–2350, Jan. 2019, https://doi.org/10.1007/s10586-018-1841-8.

[11] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, Jan. 2024, Art. no. 713, https://doi.org/10.3390/s24020713.

[12] B. R. Maddireddy and B. R. Maddireddy, "Neural Network Architectures in Cybersecurity: Optimizing Anomaly Detection and Prevention," *International Journal of Advanced Engineering Technologies and Innovations*, vol. 1, no. 2, pp. 238–266, 2024.

[13] N. Jeffrey, Q. Tan, and J. R. Villar, "A hybrid methodology for anomaly detection in Cyber–Physical Systems," *Neurocomputing*, vol. 568, Feb. 2024, Art. no. 127068, https://doi.org/10.1016/j.neucom.2023.127068.

[14] E. Muhati and D. Rawat, "Data-Driven Network Anomaly Detection with Cyber Attack and Defense Visualization," *Journal of Cybersecurity and Privacy*, vol. 4, no. 2, pp. 241–263, Jun. 2024, https://doi.org/10.3390/jcp4020012.

[15] W. F. Salvatore Stolfo, "KDD Cup 1999 Data." UCI Machine Learning Repository, 1999, https://doi.org/10.24432/C51C7N.

[16] F. Alotaibi and S. Maffeis, "Mateen: Adaptive Ensemble Learning for Network Anomaly Detection," in *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, Jun. 2024, pp. 215–234, https://doi.org/10.1145/3678890.3678901.

[17] X. Zhao, K. W. Fok, and V. L. L. Thing, "Enhancing network intrusion detection performance using generative adversarial networks," *Computers & Security*, vol. 145, Oct. 2024, Art. no. 104005, https://doi.org/10.1016/j.cose.2024.104005.

[18] C. Strickland *et al.*, "DRL-GAN: A Hybrid Approach for Binary and Multiclass Network Intrusion Detection," *Sensors*, vol. 24, no. 9, Jan. 2024, Art. no. 2746, https://doi.org/10.3390/s24092746.

[19] R. Bhatt and G. Indra, "Detecting the undetectable: GAN-based strategies for network intrusion detection," *International Journal of Information Technology*, vol. 16, no. 8, pp. 5231–5237, Dec. 2024, https://doi.org/10.1007/s41870-024-02172-7.

[20] W. Xu, J. Jang-Jaccard, T. Liu, and F. Sabrina, "Training a Bidirectional GAN-based One-Class Classifier for Network Intrusion Detection." arXiv, Mar. 08, 2022, https://doi.org/10.48550/arXiv.2202.01332.