EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

A Computationally Efficient Ensemble Boosting Framework for Optimized DoS Attack Detection in Wireless Sensor Networks

Mohamed Loughmari[1,*] , Anass El Affar[1]

[1]Engineering Sciences Laboratory (LSI), Polydisciplinary Faculty of Taza, Sidi Mohamed Ben Abdellah University, Taza, Morocco

Abstract

As Wireless Sensor Networks (WSNs) become more prevalent in modern IoT deployments, Denial of Service (DoS) attacks, which are capable of exhausting node resources and disrupting entire network segments, have emerged as their most significant security threat. In this research, we propose a computationally efficient intrusion detection framework based on the eXtreme Gradient Boosting (XGBoost) model, specifically tailored for energy-constrained environments, and designed for deployment at the cluster-head or gateway levels of WSN architectures. The framework integrates a systematic feature selection process that reduces the initial 19-column curated dataset to an optimal 16-feature subset, addressing correlation and redundancy while maintaining high-fidelity detection. Unlike many existing studies that optimize for accuracy alone, this work uniquely combines four complementary evaluation dimensions: classification accuracy, per-fold stability (standard deviation), ultra-low prediction latency, and adaptive threshold analysis for deployment flexibility. We validated our results through five-fold cross-validation on the WSN-DS dataset. The proposed framework achieved a mean accuracy of 99.74% with an exceptionally low standard deviation of ±0.000174, a prediction latency of 0.0017 ms per sample, and demonstrated a 20× computational efficiency advantage over state-of-the-art ensemble learners like CatBoost, with a compressed model size of 235 KB and an estimated inference cost of 2,550 FLOPs per prediction. Crucially, we empirically validate these theoretical metrics by physical deployment on a Raspberry Pi 3 Model B+, achieving an operational hardware latency of 0.033 ms per packet, and demonstrate the model's exceptional resilience under a 10:1 high-intensity flooding stress test. The overall outcomes of the study support that the proposed framework provides a resilient, adaptable solution for real-world WSN deployment.

Keywords: IoT/WSN Security, Wireless Sensor Networks, Intrusion Detection System (IDS), Denial of Service (DoS) Attacks, Cybersecurity, Machine Learning, Computationally Efficient IDS

Received on 23 May 2026, accepted on 02 August 2026, published on 11 August 2026

doi: 10.4108/eetiot.13137

*Corresponding author. Email: mohamed.loughmari@usmba.ac.ma

1. Introduction

Wireless Sensor Networks (WSNs) form a vital component of the modern Internet of Things (IoT) infrastructure and have widespread applications in environmental monitoring, e-health, smart buildings, and military surveillance [1]. These networks are often deployed in large numbers and consist of numerous tiny, low-power sensor nodes that cooperatively transmit the collected physical or environmental data to a central base station [2,3]. However, their inherent characteristics, such as their reliance on wireless

communication, distributed nature, deployment in unattended environments, and significant constraints on computational power and energy, render them highly vulnerable to cyber threats, particularly Denial of Service (DoS) attacks, which aim to disrupt network availability by exhausting node resources, leading to service interruptions and data loss [4]. While cryptography provides a first line of defense, it is insufficient against insider threats, which necessitates the implementation of Network-based Intrusion Detection Systems (NIDS) as a critical additional layer of security [5].

Traditional NIDS, particularly those utilizing conventional signature-based systems, are often unsuitable for WSNs because they fail to accommodate the stringent resource limitations of sensor nodes, struggle to detect unknown or zero-day attacks, and are often overwhelmed by the dynamic traffic of IoT and WSN environments [6]. To overcome these issues, Artificial Intelligence (AI) and Machine Learning (ML) have become essential technologies for modern IDS, including anomaly-based, misuse-based, and hybrid detection approaches. As a result, an effective security solution for WSNs must be highly accurate, fast, and computationally lightweight to ensure minimal overhead. This has created an urgent need for lightweight, specialized intrusion detection techniques [7].

Even with the significant amount of research on AI and ML-based intrusion detection, the development of a practical, deployable solution capable of reliably identifying threats in WSN continues to pose a major hurdle, where network traffic is naturally dynamic and imbalanced, and where the Normal class can constitute over 90% of records, which often leads current solutions to report high overall accuracy but struggle with the detection of minority attack classes such as TDMA and Flooding. Furthermore, many recent studies do not measure the computational cost, such as prediction latency and training time, needed to run the model on devices with stringent resource limitations; this selective reporting creates a misleading picture of deployment readiness, since a computationally intractable model, regardless of its accuracy, cannot be practically deployed on a resource-constrained cluster-head or gateway node.

This research addresses these critical gaps by proposing a framework that treats computational feasibility as an equally important evaluation criterion alongside detection accuracy. Crucially, we distinguish our approach in both design philosophy and evaluation scope from recent studies that also leverage gradient boosting frameworks for WSN security. While many contemporary methodologies focus primarily on maximizing accuracy by integrating computationally expensive meta-heuristic optimization layers, these architectural additions inadvertently multiply the required CPU cycles, rendering them impractical for true Edge-AI deployment. By contrast, we hypothesize that a carefully optimized ensemble boosting model can deliver robust, real-time security without overloading the network. This claim is substantiated through the simultaneous measurement of classification accuracy, prediction latency, serialized model size, and estimated inference FLOPs, ultimately presenting not merely a classification benchmark, but a reproducible, efficiency-aware, highly effective, and structurally compressed framework for safeguarding WSNs against DoS attacks and potential threats.

The main contributions of this paper are as follows:

- We propose a lightweight XGBoost-based framework specifically designed to detect DoS and potential attacks in WSNs. The algorithm selection was motivated by its built-in architectural efficiency properties (such as regularization, parallel tree construction, and memory-efficient learning mechanisms), which make it inherently more suitable for computationally constrained deployment scenarios, such as cluster-head or gateway nodes, compared to architecturally heavier alternatives.
- We demonstrate the effectiveness of the model on a feature-optimized subset derived from the WSN-DS dataset, obtained through correlation analysis; we conduct a comprehensive and multi-dimensional performance evaluation covering both classification metrics and computational efficiency benchmarks, critical indicators for WSN deployment suitability.
- We rigorously validate the framework through 5-fold cross-validation combined with standard deviation analysis (σ), guaranteeing robustness, generalizability, and performance consistency across all folds. Furthermore, we prove the operational stability under extreme conditions via a 10:1 scenario-based flooding stress test.
- The proposed evaluation further adopts a uniquely multi-dimensional design that simultaneously reports classification accuracy, prediction latency benchmarks, and physical edge-deployment performance on a Raspberry Pi 3 Model B+, a combination rarely addressed simultaneously in existing WSN-IDS benchmark studies on the WSN-DS dataset.
- Furthermore, we provide a threshold analysis to evaluate the model's operational flexibility and decision-making trade-offs in different WSN deployment scenarios.

Remaining sections of this work are organized as follows: Section 2 reviews the related work. Section 3 details the proposed research methodology. Section 4 presents the experimental results, while Section 5 provides a comprehensive examination of the outcomes. Finally, Section 6 concludes the paper and suggests possible future research directions.

2. Related work

The field of WSN security has seen extensive research focused on Machine Learning (ML) and Deep Learning (DL) to combat the increasing threat of cyberattacks. ML has broadly demonstrated significant potential across various cybersecurity areas [8], and AI/ML-based intrusion detection systems provide an integrated alternative for securing WSN environments, by facilitating adaptive, data-driven detection with adjustable parameter dimensions as well as, for distributed models, diminished dependence on centralized data aggregation. A comprehensive examination of

computational intelligence methodologies for IDS in WSN [9] confirms that the field includes traditional supervised models, deep learning frameworks, generative anomaly detectors, and emerging reinforcement learning approaches.

Misuse-based or signature-based approaches frame intrusion detection as a supervised classification problem utilizing labeled traffic data, predominating recent literature due to the accessibility of labeled attack samples and consistently strong performance metrics. Wazirali & Ahmed [10] evaluate seven models, including kNN, LR, SVM, Gradient Boosting, DT, LSTM, and MLP, on the basis of a WSN DoS dataset and notably examine the effect of detected attacks on WSN network lifetime. The research reports an accuracy of 99.60%, with LSTM and Gradient Boosting emerging as the top performers, making this one of the few works to consider energy lifetime as an operational metric alongside detection performance. In [11], Jiang et al. present SLGBM, a stochastic LightGBM-based IDS for WSN smart environments; the method provides high accuracy yet indicates reduced precision in low-attack-rate scenarios. DL variants further exploit temporal and spatial feature dependencies: The studies in [12] leverage VGG-19 combined with a hybrid deep neural network to reduce training data requirements via transfer learning while improving detection rates; in [13] the authors propose an edge-deployable improved kNN model that is enhanced via Parallel Lévy-flight Aquila Optimizer (PL-AOA), reporting 99.00% accuracy on WSN-DS with approximately 10% improvement over standard kNN while explicitly targeting WSN edge deployment to reduce centralized processing overhead; and in [14] Kumar et al. proposed an adaptive spatio-temporal deep learning framework for robust cyber threat detection in IoT environments, emphasizing the need to capture complex traffic patterns across distributed nodes. Optimization-assisted pipelines further automate feature selection and hyperparameter tuning: [15] benchmarks seven classifiers on the WSN-DS and reports that AdaBoost achieves up to 100.00% accuracy with zero false-positive rate on certain attack classes while Random Forest reaches 99.72%; yet, such perfect detection rates on specialized datasets should be interpreted with caution, as they often indicate a high risk of overfitting or pattern memorization rather than robust generalization. Similarly, CatBoost has emerged as a powerful symmetric tree-based ensemble learner for cybersecurity tasks; for instance, Nguyen et al. [16] report an accuracy of 99.65% when combining it with Genetic Sacrificial Whale Optimization (GSWO). However, while such meta-heuristic optimizations achieve high precision, they often introduce significant computational complexity and energy consumption, which remains a primary challenge for real-time WSN deployment. Furthermore, Panday et al. [17] propose a Tabu Search-optimized Random Forest, claiming superior multi-metric performance over standard RF. More recently, Jiang et al. [18] propose ST-IAOA-XGBoost, combining spatial-temporal feature extraction with an improved Aquila optimization algorithm on the WSN-DS dataset, demonstrating that gradient-boosted models can be further enhanced through topology-aware preprocessing, though at the cost of additional architectural complexity. Meanwhile [19] applies the Whale Optimization Algorithm to tune CNN architectural parameters with reported improvements over fixed-architecture baselines. An imbalance-aware pipeline combining ML classifiers with SMOTE-Tomek resampling is evaluated in [20] on WSN-DS dataset, achieving 99.78% binary accuracy and 99.92% multiclass accuracy, demonstrating that explicit class-balance handling reduces misclassification on minority attack classes. Nevertheless, misuse-based methods are fundamentally limited by their reliance on previously observed attack patterns; while they may generalize to detect certain unseen variants, they do not consistently identify genuinely novel or zero-day threats, which represents a critical limitation in adversarial environments where attack behaviors evolve beyond the closed-world assumptions of fixed labeled datasets.

Anomaly-based approaches learn the pattern of the normal behavior of WSN and flag any deviations detected; theoretically this gives these approaches the capabilities to detect zero-day attacks without the need for labeled attack samples. This paradigm is less prevalent than supervised classification, primarily due to the difficulty in obtaining clean normal-traffic traces for calibration. However, Generative Adversarial Networks have emerged as a prominent direction: The work presented in [21] frames detection utilizing a Conditional GAN (CGAN) to differentiate between normal and attack patterns without explicit labels during training. Building on this, Meenakshi & Karunkuzhali [22] propose a more sophisticated optimized self-attention-based Variational Autoencoder GAN (VAE-GAN), which combines the representation learning capabilities of VAEs with adversarial discriminative training. The Graph Neural Networks offer a uniquely structured anomaly-aware approach by modeling WSN topology as a first-class detection signal. Instead of treating each node independently, GNN-based systems consolidate information from neighboring nodes to detect coordinated multi-hop attack behaviors that are invisible to node-local classifiers [23], and [24] enhance a GNN with Osprey Optimization-driven feature selection on WSN-DS, reporting 99.68% accuracy with improved false-positive rate and sensitivity compared to baseline models. Although anomaly-based methods are theoretically superior in detecting unknown vulnerabilities, they consistently exhibit high false positive rates compared to supervised methods. Furthermore, their calibration sensitivity, particularly in generative models prone to training instability and pattern collapse, complicates reliable tuning in the context of limited and lacking stable data traffic across dispersed wireless sensor network nodes.

Hybrid approaches integrate the principles of both misuse and anomaly detection, aiming to capture both known and behavioral anomalies simultaneously. Ramana et al. [25] introduce WOGRU-IDS, which combines Whale Optimization with Gated Recurrent Units in an IoT-WSN context. Their method reports high precision but reduced specificity, hence revealing a persistent false-alarm trade-off in the anomaly component. Authors in [26] integrate a two-

tier node/cluster-head trust evaluation model into a deep learning pipeline optimized via Sea Lion Optimization. Their method claims improvements over standalone deep learning approaches. In 2023, Subramani & Selvi [27] integrate fuzzy inference with a deep CNN to model uncertainty in WSN routing behavior. Their method reports improved accuracy, a lower false-positive rate, and reduced mean detection delay relative to existing methods. Despite the benefits of hybrid designs, they introduce additional architectural complexity and often computational overhead and depend on the consistent availability of trust or behavioral signals, such as cluster-head stability and node honesty scores, which may not be consistently observable in dynamic or heterogeneous WSN deployments where topology changes frequently.

Reinforcement learning (RL) represents an emerging approach to do adaptive intrusion detection in WSNs, instead of using a fixed labeled dataset, an IDS agent learns the best way to detect intrusions by interacting with the network environment. Rameshkumar et al. in [28] apply a progressive transfer learning Deep Q-Network to DDoS defense in WSN, leveraging transfer learning to accelerate policy convergence across deployment contexts. Shakya et al. [29] propose IRADA, which forms an integrated RL and deep learning framework formulated as a Markov Decision Process with Bayesian hyperparameter optimization. Their method reports a 99.50% accuracy, 98.26% F1-score, 99.94% True Negative Rate, and 99.38% Area Under the ROC Curve. Although RL-based approaches offer runtime adaptation to unknown attack patterns, these methods acknowledge elevated detection time and computational overhead as constraints for resource-limited node deployment.

Table 1 summarizes all reviewed studies across detection approach, model/technique, dataset, and key reported results.

Table 1. Summary of state-of-the-art literature

Author(s)	Year	Approach	Model / Technique	Key Result
Wazirali & Ahmad [10]	2022	Misuse-based	kNN, LR, SVM, GBoost, DT, LSTM, MLP	99.6% acc ;98.8% F1-Score (GBoost)
Jiang et al. [11]	2020	Misuse-based	SLGBM	99.73% acc
Manjula & Priya [12]	2022	Misuse-based (deep)	VGG-19 + Hybrid DNN	98.86% acc
Liu et al. [13]	2022	Misuse-based (edge)	PL-AOA improved kNN	99.00% (+10% vs std kNN)
Arabiat & Eljaafreh [15]	2025	Misuse-based	AdaBoost, CN2, RF, NB, J48, SVM, MLP	100.00% (AdaBoost); 99.00% (CN2, RF)
Nguyen et al. [16]	2024	Misuse-based	GSWO-CatBoost	99.65%
Pandey et al. [17]	2025	Misuse-based + Optimization	Tabu Search–optimized RF	99.67%
Vembu and D. Ramasamy [19]	2022	Misuse-based + Optimization	WOA-tuned CNN	97.01% acc, 94.09% recall, 95.50% F1-score
Talukder et al. [20]	2024	Misuse-based (imbalance-aware)	ML + SMOTE-Tomek	99.78% acc(binary); 99.92% acc (multi-class)
Jiang et al. [18]	2025	Misuse-based + Optimization	ST-IAOA-XGBoost	99.71% acc
Sood et al. [21]	2022	Anomaly-based (generative)	Conditional GAN (CGAN)	90.43% acc
Meenakshi & Karunkuzhali [22]	2024	Anomaly-based (generative)	Optimized self-attention VAE-GAN	99.2% acc
Gharavian et al. [23]	2023	Anomaly-based (topology-aware)	Lightweight GNN	99.1% acc
Mohan et al. [24]	2025	Anomaly + Graph-based	OOA feature selection + GNN	99.68% acc
Ramana et al. [25]	2022	Hybrid (anomaly + misuse)	WOGRU-IDS (WO + GRU)	99.85% acc
Kagade and Jayagopalan [26]	2022	Hybrid (trust + DL)	Two-tier trust eval + SLnO-NN	95.00%
Subramani and Selvi [27]	2023	Hybrid (fuzzy + DL)	Deep Fuzzy CNN	98.9% acc
Rameshkumar et al. [28]	2023	Anomaly-based (RL)	Progressive TL + Deep Q-Network	89.00% acc
Shakya et al. [29]	2024	Hybrid (RL + DL)	Bayesian-opt DL + RL as MDP	Acc: 99.50%, F1: 98.26%, AUC: 99.38%

Despite the advancements reviewed above, a critical research gap remains regarding the alignment between algorithmic selection and the unique operational requirements of WSNs. While existing models achieve high

detection rates, a critical examination reveals that many studies rely on generalized network datasets (KDD-99, NSL-KDD, CICIDS2017...) that may not accurately reflect the specific traffic patterns and resource constraints of wireless sensor environments. Furthermore, the predominant focus on overall accuracy often obscures the computational efficiency and performance stability required for real-time deployment. This frequent disregard for the synergy between dataset specifications and internal algorithm functionality often leads to results that represent pattern memorization rather than robust and generalized detection. Consequently, there is an urgent need for research that justifies high performance through the lens of mathematical optimization and resource-awareness. This work addresses these limitations by proposing a computationally efficient framework that prioritizes the accuracy-efficiency trade-off, using 5-fold cross-validation and standard deviation analysis to ensure robust, real-time security.

3. Research methodology

This section details the systematic methodology employed in this study, as illustrated in the schematic diagram (Figure 1). The process begins with data collection, followed by preprocessing and feature selection, model training, and a comprehensive performance evaluation.

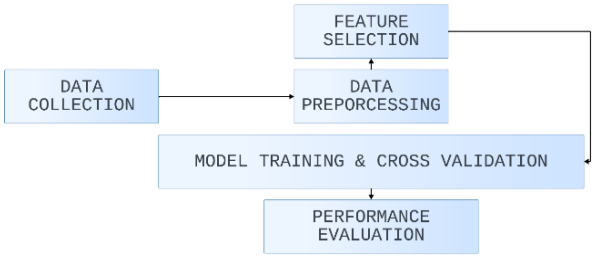

Figure 1. Representational diagram of the proposed system

3.1. Dataset collection and description

The experiments were conducted on the WSN-DS dataset, a specialized, labeled compilation introduced by Almomani et al. [3] specifically for evaluating DoS attack detection in WSNs. Data collection for this dataset was performed using Network Simulator 2 (NS-2) based on the LEACH (Low Energy Adaptive Cluster Hierarchy) routing protocol. The dataset was constructed to characterize four specific types of DoS attacks: Blackhole, Grayhole, Flooding, and TDMA (Scheduling) attacks, in addition to normal network traffic. While the original dataset comprised 374,661 records and 23 extracted attributes, this study employed a widely adopted, curated release by the dataset maintainers that provides a standardized CSV format with 19 columns. The robustness and correctness of the collected data were validated through analytical verification of LEACH phases and comparison against NS-2 simulation outputs under normal conditions, showing a 100% match for key messaging parameters. The main characteristics of this dataset are summarized in Table 2, while the full technical parameters and network configuration used for its generation are detailed in Table 3.

Table 2. Dataset Summary

Characteristic	Detail	Notes
Total Instances	374,661	The full size of the dataset.
Feature Count	Original: 23 attributes; Curated: 19 columns	The curated 19-parameter CSV used in this study.
Number of Classes	5	4 Attack types + 1 Normal class.
Attack Types	4	Blackhole, Grayhole, Flooding, TDMA (Scheduling)
Normal Traffic	Included	Baseline data
Simulation Platform	NS-2	Network Simulator 2 used for data generation
Network Protocol	Leach	Low Energy Adaptive Cluster Hierarchy
Key feature groups	IDs, Distances, Counters (ADV/JOIN/SCH/ DATA), Energy Metrics	Categories of parameters used for detection

Table 3. WSN Simulation and Network Configuration Parameters used for WSN-DS Dataset Generation [3]

No.	Parameter Name	Description	Data Type	Units/Range (ex. values)	Category
1	id	Unique identifier for the node.	Integer	N/A	Node/Topology

2	Time	Simulation timestamp of the record.	Float	Seconds	Temporal
3	Is_CH	Binary indicator: 1 if the node is a Cluster Head, 0 otherwise.	Integer	{0, 1}	Node/Role
4	who CH	ID of the node acting as the CH for this specific node.	Integer	Node ID	Node/Topology
5	Dist_To_CH	Euclidean distance between node and its CH.	Float	Meters	Spatial
6	ADV_S	Number of advertisement packets sent by node.	Integer	Count	Communication
7	ADV_R	Number of advertisement packets received by node.	Integer	Count	Communication
8	JOIN_S	Number of join-request packets sent by node.	Integer	Count	Communication
9	JOIN_R	Number of join-request packets received by node.	Integer	Count	Communication
10	SCH_S	Number of schedule packets sent by the Cluster Head.	Integer	Count	Communication
11	SCH_R	Number of schedule packets received by the node.	Integer	Count	Communication
12	Rank	Rank of the node within its cluster or network.	Integer	N/A	Node/Hierarchy
13	DATA_S	Number of data packets sent by the node.	Integer	Count	Communication
14	DATA_R	Number of data packets received by the node.	Integer	Count	Communication
15	Data_Sent_To_BS	Total data packets sent by the node towards the Base Station.	Integer	Count	Communication
16	dist_CH_To_BS	Distance from the node's Cluster Head to the Base Station.	Float	Meters	Spatial
17	send_code	An internal code related to packet transmission/reception status.	Integer	E.g., {0, 1}	Communication
18	Expaned Energy	Total energy consumed by the node since the start of simulation.	Float	Joules (J)	Energy
19	Attack type	The class label indicates normal traffic or a specific DoS attack type.	Categorical	{'Normal', 'Grayhole', 'Blackhole', 'TDMA', 'Flooding'}	Target Class

3.2. Data preprocessing and feature selection

Data preprocessing

To prepare the data for ML modeling, a multi-step preprocessing phase was conducted to ensure data quality and suitability for ML model training. Initially, the textual labels of the target feature (Attack type) were encoded as numerical values. The encoding scheme mapped 0 for ‘Normal’ traffic, 1 for ‘Blackhole’, 2 for ‘Grayhole’, 3 for ‘Flooding’, and 4 for ‘TDMA’. Subsequent preprocessing steps entailed the elimination of duplicate records and the management of infinities and missing or null values to uphold data integrity.

Feature Selection

Selecting the optimal features is a critical phase in the ML workflow. This phase aims to select the optimal features in a dataset, which directly impacts the performance of classifiers, the understanding of data, and computational efficiency. This phase is especially crucial in WSN settings, where devices must avoid high-dimensional issues while operating under strict energy and processing constraints.

In this study, despite the early exploration of complex strategies like Recursive Feature Elimination (RFE) and Sequential Backward Selection (SBS), a more straightforward approach proved most effective for optimizing the feature set. The process began with excluding the index column, as it was deemed statistically insignificant for the detection task. Subsequently, we examined the Seaborn (SNS)-generated heatmap of feature correlations

(Figure 2) to mathematically identify collinearity. The analysis revealed that the 'SCH_R' and 'JOIN_S' features are highly correlated.

Because this correlation is a known structural artifact of the LEACH cluster formation handshake, the 'SCH_R' feature was manually excluded prior to model training to eliminate redundancy. Furthermore, the dataset was meticulously verified to confirm the absence of zero-variance columns. This protocol-driven preprocessing phase resulted in a final, optimized set of 16 features, which, alongside the target variable, were utilized for all subsequent training and evaluation stages.

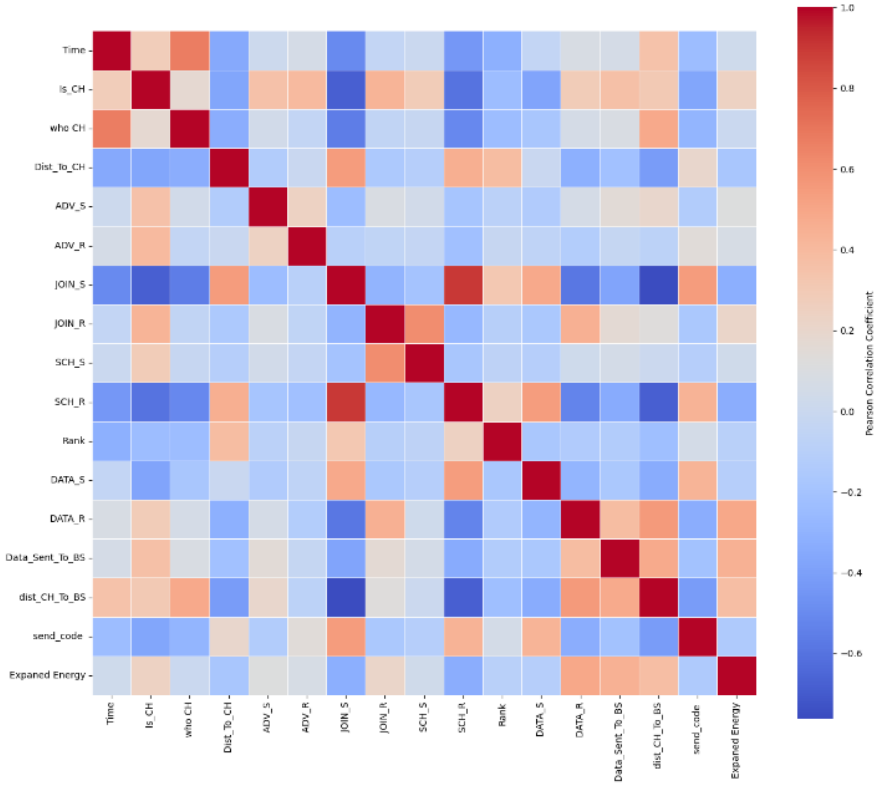

Figure 2. Heatmap of Feature Correlations for the WSN-DS Dataset

3.3. Model training and validation

To evaluate the hypothesis that our proposed model offers a computationally-efficient solution, we implemented five machine learning algorithms chosen for their complementary strengths in intrusion detection: XGBoost and CatBoost as state-of-the-art gradient boosting methods, Decision Tree (DT) as a simple yet interpretable baseline, Support Vector Machine (SVM) for its ability to capture complex, high-dimensional boundaries, and Naïve Bayes (NB) as a lightweight probabilistic model often used in security-related tasks. This selection provides a balanced evaluation across ensemble, kernel-based, tree-based, and probabilistic learning paradigms.

Default hyperparameters were initially employed for all models, as they provided competitive baseline results. XGBoost, as the best-performing algorithm, was then further optimized through systematic hyperparameter tuning using the scikit-learn's GridSearchCV module to enhance and maximize its performance prior to its proposal for WSN DoS detection. In order to ensure reproducibility and model optimality, this tuning process utilized a 5-fold GridSearchCV strategy. The search space covered the learning rate [0.01, 0.4], maximum tree depth [2, 10], and the number of estimators [50, 125], this search space was deliberately designed to be targeted rather than exhaustive, balancing tuning thoroughness against computational overhead consistent with the resource-efficiency focus of this work, as a broader random search risked overfitting the tuning process to the training data distribution. The optimization identified the ideal configuration as: learning_rate=0.4, max_depth=5, and n_estimators=102. These parameters were selected as they provided the optimal balance between high-fidelity classification and the performance stability required for consistent WSN monitoring.

The implementation of XGBoost is governed by an objective function designed to optimize predictive performance while controlling model complexity. At each iteration t, the algorithm minimizes the next objective function.

$$Obj^{(t)} = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (1)$$

Where, L represents a differentiable convex loss function that measures the training error between the prediction $\hat{y}_i$ and the target y_i. To ensure the model remains lightweight and avoids overfitting, as this is a critical requirement for the dynamic traffic of WSN environments, the framework utilizes the internal XGBoost regularization term, Ω, formulated as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2 \quad (2)$$

Where : T is the number of leaves in the tree and w represent the leaf weights. The parameters γ and λ act as penalties that constrain the model's complexity, ensuring stable and consistent performance across the dataset's partitions.

To rigorously validate our findings and ensure generalizability, we employed a 5-fold cross-validation strategy. In this approach, the dataset was partitioned into five subsets; in each iteration, the model was trained on four folds and evaluated on the remaining one. This process was repeated until every subset had served once as the test set. Crucially, to prevent data leakage and provide strictly controlled baseline experiments, all models were evaluated under an identical validation architecture. For scale-sensitive baseline algorithms (SVM and Naïve Bayes), preprocessing and Z-score normalization were securely encapsulated within a scikit-learn Pipeline. This architectural choice ensured that the scaler was fit exclusively on the training fold and then applied blindly to the test fold, thereby guaranteeing that the comparative metrics are free from preprocessing leakage and represent a fair, one-to-one operational benchmark. The final score was calculated as the average across the five folds, providing a reliable estimate of the model's ability to generalize to unseen data.

3.4. Performance metrics

The models were evaluated using a comprehensive set of metrics covering both classification performance and computational efficiency. Classification metrics were derived from the confusion matrix, including True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Processing Time: The computational cost of each model was assessed using processing-time metrics, including the total execution time of the complete 5-fold cross-validation procedure, the average training time per fold, and the prediction latency per sample. Prediction latency is a critical indicator in WSN environments, where real-time detection must operate under limited computational and energy resources.

Accuracy: The proportion of total correct classifications.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

Precision: The proportion of predicted attacks that were actual attacks.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Recall: The proportion of actual attacks that were correctly identified.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

F1-score: The harmonic mean of precision and recall.

$$F1 - score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (6)$$

Standard Deviation: The model stability and performance consistency across all folds.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (7)$$

Where N is the number of folds (N=5), x_i is the accuracy of the i-th fold, and μ represents the mean accuracy across all folds.

Binary threshold analysis: A binary evaluation (Attack vs. Normal) was conducted to analyze operating-point trade-offs by varying the decision threshold. For each threshold value, the True Positive Rate (TPR), corresponding to the recall of the Attack class, and the False Positive Rate (FPR), representing the proportion of Normal samples incorrectly classified as attacks, were computed to assess the balance between detection performance and false-alarm generation.

3.5. Experimental environment

All experiments were conducted on a personal computer equipped with an Intel Core i5-10210U CPU (1.60 GHz, 4 cores / 8 threads) and 12 GB of RAM, running Windows 10 Professional 64-bit; this local hardware setup guarantees high-quality benchmarking of computational overhead. The implementation was carried out using Python 3.10.9, along with main libraries, including scikit-learn 1.7.1 (for GaussianNB, DecisionTree, and Support Vector Classification), XGBoost 3.0.2, and CatBoost 1.2.8. In order to ensure consistency and reproducibility, all experiments were performed on the same hardware and software configuration.

4. Results

This section presents detailed performances of the five ML models on the final, feature-optimized dataset, which was

evaluated using a 5-fold cross-validation strategy for the multi-class classification task.

4.1. Overall model performance

The feature selection methods explored in the study failed to produce a notable improvement in model performance; in some cases, a slight decrease was observed. This outcome suggests that the curated attributes provided in the publicly available dataset were already the result of a careful selection process from the initially reported 23 features, ensuring their relevance to WSN security applications [3]. Despite this, we further investigated feature reduction to test the hypothesis that a smaller, well-chosen subset could enhance computational efficiency without compromising classification performance.

Table 4 presents the mean accuracies and standard deviations (σ) for both configurations: using all features (No FS) and after feature selection (FS). The total cross-validation (CV) time in seconds is also included to illustrate the computational impact of feature reduction.

Table 4. Comparative Analysis of Detection Accuracy and Computational Efficiency across Full (No FS) and Optimized (FS) Feature Configurations

Model	Mean Acc μ (No FS) Eq.(3)	σ (No FS) Eq.(7)	CV Time(s) (No FS)	Mean Acc μ (FS) Eq.(3)	σ (FS) Eq.(7)	CV Time(s) (FS)
XGBoost	0.997347	0.000149	32.75	0.997398	0.000174	30.6
CatBoost	0.996939	0.000117	737.28	0.996955	0.000135	603.94
Decision Tree	0.995156	0.000319	12.2	0.995129	0.000293	9.67
SVM	0.96839	0.000561	57.7	0.969271	0.000503	35.81
Naive Bayes	0.953339	0.00108	4.73	0.953435	0.001092	3.2

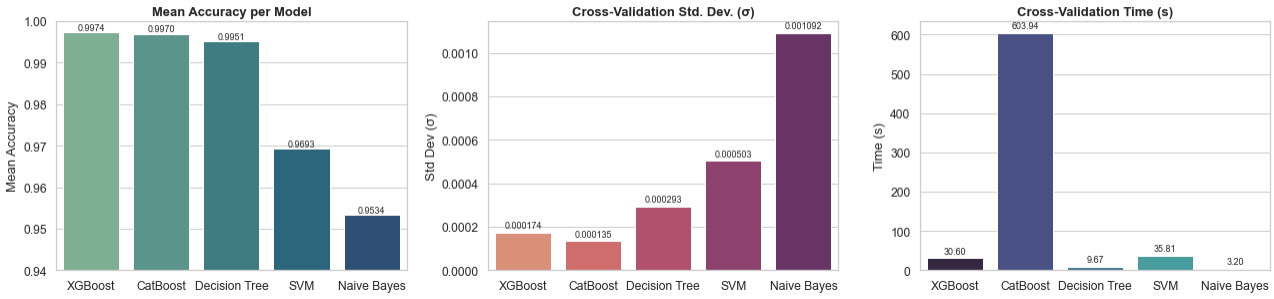

Figure 3. Performance Comparison Across Models

The performance metrics presented in Table 4 and illustrated in Figure 3 show that the tree-based ensemble models (XGBoost, CatBoost, and Decision Tree) consistently outperform the SVM and Naive Bayes classifiers. Notably, the XGBoost model recorded the highest mean accuracy (μ = 0.9974) along with an extremely low standard deviation (σ = ± 0.000174) as calculated by Eq.(7). This indicates highly stable and reliable performance across all cross-validation folds. However, although CatBoost achieved comparable accuracy, its cross-validation runtime was approximately 20 times longer than that of XGBoost (603.94 s vs. 30.60 s), and this acts as a limitation for its deployment feasibility in resource-constrained WSN nodes. The Decision Tree had the lowest complexity among tree-based learners but with noticeably less accuracy, while SVM and Naive Bayes remained significantly below 97% and 96% accuracy, respectively, with higher latency. Furthermore, it is essential to note that while classification accuracy remained stable, the FS method resulted in significant improvements in computational efficiency. For instance, the total CV time for SVM was reduced by approximately 37% (from 57.7s to 35.81s).

4.2. Detailed analysis of the optimal model

Following the performance evaluation, XGBoost provides the best overall balance. To evaluate the model's performance for each specific attack class, the average recall, precision, F1-score, average FP, and average FN were calculated across the five folds to ensure statistical reliability. The aggregated results are presented in Table 5.

Table 5. Per-Class Performance Metrics for the Optimized XGBoost Model

Attack Type	Recall	Precision	F1-Score	Support	Avg FP	Avg FN
Normal	0.99932	0.99822	0.99876	340066	122.2	46.8
Grayhole	0.98954	0.99226	0.99088	14596	22.6	30.6
Blackhole	0.99606	0.99148	0.99376	10049	17.2	8.0
TDMA	0.92728	0.99536	0.96008	6638	5.8	96.6
Flooding	0.9801	0.95982	0.96982	3312	27.2	13.0

As shown in the table above, the model maintained strong performance across most attack classes, with particularly high recall and precision values for Normal, Blackhole, and Flooding traffic. A slight drop is observed in the TDMA class, suggesting a relative challenge in identifying this attack type compared to others.

Error distribution analysis

The full examination of security effectiveness and per-class error analysis indicates that the percentages of false positives and false negatives reveal particular strengths and limitations in the model's detection abilities across several attack classes.

- Strengths: The model demonstrates high efficacy in detecting 'Blackhole' and 'Flooding' attacks, as evidenced by their low average false negative counts (8.00 and 13.00, respectively). This suggests that when these critical attacks occur, the system has a high probability of correctly identifying them.
- Identified Limitation: The model's primary weakness lies in the detection of 'TDMA' attacks, for which it recorded a high average false negative count of 96.60. This implies that the model frequently misclassifies TDMA attack traffic as another class. The correspondingly high average false positive count for 'Normal' traffic (122.20) further suggests that TDMA attacks are often misclassified as benign network activity. This limitation is consistent with the findings of the original WSN-DS dataset study [3], in which the TDMA attack class also recorded the lowest per-class detection rate, further confirming that this is an intrinsic challenge of the dataset's feature representation rather than a model-specific failure.

For time and complexity metrics, the average training time per fold was 5.53 seconds, with a prediction latency of 0.0017 milliseconds per sample. To further substantiate the computational efficiency of the proposed framework, the compressed and serialized XGBoost model occupies only 235 KB of storage and requires an estimated 2,550 floating-point operations (FLOPs) per prediction. This compact footprint is well within the storage capacity of modern cluster-head and gateway devices, and the ultra-low FLOPs count confirms that inference remains computationally trivial, producing predictions in a single pass without iterative or recursive computation overhead. This rapid prediction capability ensures minimal processing overhead on constrained deployment nodes, which is essential for preserving battery life and enabling immediate response to potential threats.

Scenario-Based Stress Testing

Although k-fold cross-validation establishes a robust performance baseline, real-world DoS attacks in WSNs rarely occur in balanced distributions. To validate the model's stability under extreme conditions, we designed a scenario-based stress test that simulated a high-intensity Flooding replay attack. We isolated a strict 30% hold-out test set to ensure zero data leakage. Using bootstrap sampling to simulate rapid packet duplication, we generated a heavily skewed attack scenario comprising a 10:1 ratio of malicious-to-normal traffic (5,000 Flooding packets and 500 Normal packets) from this unseen data.

Despite the heavy traffic saturation, the XGBoost model maintained exceptional classification accuracy, achieving 98.49%. The model correctly identified 100% of normal traffic and 98.34% of flooding packets, showing that its detection capability does not degrade with heavy class imbalance. Furthermore, the inference latency during this stress test was recorded at 0.0032 milliseconds per sample. This microscopic latency profile confirms that the proposed framework can process sudden, massive traffic surges successfully without inducing a CPU bottleneck or computational failure at the resource-constrained cluster head node.

Physical Edge Hardware Validation

To conclusively validate the feasibility of deploying the proposed framework, the trained XGBoost model was deployed on a Raspberry Pi 3 Model B+, a widely adopted, representative, resource-constrained edge device. Following standard Edge-AI paradigms, our previously trained XGBoost model was serialized and transferred to the Raspberry Pi for an inference-only evaluation. When tested on a live traffic partition of 74,933 network packets, the edge-deployed model maintained an operational accuracy of 99.72%. Most importantly, the physical inference latency was recorded at 0.033 milliseconds per packet. This physical hardware validation confirms that the proposed framework is fully capable of executing real-time, high-speed DoS detection at the WSN cluster-head level without exceeding edge-computing capabilities.

Threshold analysis and operational flexibility

To further assess the robustness and adaptability of the proposed XGBoost-based intrusion detection model, we analyzed its performance at different decision thresholds ranging from 0.10 to 0.99. Adjusting the decision threshold allows the system to be fine-tuned for high sensitivity or

extreme precision depending on the specific security requirements of the WSN deployment.

As shown in Table 6, at a low threshold of 0.10, the model achieved a high recall of 98.74%, effectively capturing almost all intrusion attempts, though at the cost of a slightly higher false positive rate (0.28%). Increasing the threshold to 0.50, which represents the standard operating point, resulted in the best overall balance, with 99.76% accuracy, 98.25% recall, and a false positive rate below 0.09%. At higher thresholds, such as 0.95, the model maintained high accuracy (99.53%) and significantly reduced false positives (0.0097%), but with a modest drop in recall (95.05%). These results highlight the model's stability and versatility, allowing it to be fine-tuned for different deployment scenarios.

Table 6. Impact of Decision Thresholds on XGBoost Detection Performance

Threshold	Accuracy	Recall (TPR)	FPR
0.10	99.6309%	98.7426%	0.2788%
0.30	99.7336%	98.4940%	0.1403%
0.50	99.7568%	98.2483%	0.0897%
0.70	99.7395%	97.6875%	0.0518%
0.90	99.6335%	96.2307%	0.0203%
0.95	99.5345%	95.0542%	0.0097%
0.99	99.0866%	90.1315%	0.0024%

5. Discussion

The experimental results offer some significant insights into the design of efficient intrusion detection systems for wireless sensor networks (WSNs). Overall, the experiments demonstrate that XGBoost achieves the optimal balance between detection accuracy and computational cost and thereby fulfilling the objective of developing a computationally-efficient denial-of-service (DoS) attack detection model for WSN environments. Of all the evaluated classifiers, XGBoost achieved the highest accuracy (99.74%) with extremely low variability across cross-validation folds, indicating strong robustness and generalization capability.

In the case of WSN deployment, where nodes operate under strict energy and processing constraints, computational efficiency becomes a decisive factor beyond predictive performance. Although CatBoost had comparable accuracy, it took significantly longer runtime (≈604s for 5-fold cross-validation) compared to XGBoost and is therefore impractical for energy-constrained environments. This performance gap reflects the efficiency of XGBoost's optimized gradient-boosting framework, which combines regularization, parallel tree construction, and memory-efficient learning mechanisms [30]. These characteristics make XGBoost a more suitable candidate for deployment at the gateway or cluster-head levels.

The comparative analysis also highlights the limitations of simpler models. Even though the decision tree and Naive Bayes classifiers offered faster execution, their detection capabilities were weak, especially for minority attack categories. Similarly, the SVM struggled to capture the nonlinear relationships inherent to WSN traffic patterns [31], and thus the accuracy was significantly lower. These findings support the idea that boosted ensemble learners are better suited to modeling complex network behaviors. These observations are consistent with previous WSN intrusion-detection studies that emphasize gradient-boosted approaches [11,16,18,32,33]. An additional contribution of this work is the explicit evaluation of model stability through standard deviation σ calculated in Eq. (7). XGBoost produced the second and nearly the lowest standard deviation (±0.00017), confirming its consistent performance across different data partitions. Evaluating accuracy and deviation separately provides a clearer representation by describing both central tendency (performance level) and dispersion (stability). This consistency is especially important for WSN environments, where traffic characteristics can change dynamically over time.

The feature-selection analysis further reinforces the objective of a lightweight design. Reducing the feature space slightly improved accuracy while simultaneously decreasing total cross-validation time. This conclusion demonstrates that removing redundant attributes can enhance efficiency without compromising predictive capability. Accordingly, this observation supports the hypothesis that compact feature representations reduce noise and unnecessary computation, which is essential for real-time intrusion detection in constrained environments.

From a practical deployment perspective, the proposed model displays strong operational feasibility. Its short training duration (approximately 5.5 seconds per fold) coupled with ultra-low prediction latency (0.0017 ms per sample) enables near real-time detection. Moreover, false alarm analysis is found to be balanced in terms of traffic categories with high recall for the Normal and Flooding classes while keeping the misclassification low for more subtle attack patterns (such as TDMA). This balanced detection capability helps reduce unnecessary alerts and preserve network resources.

Operational configuration parameters, alongside model architecture, also affect deployment behavior. In particular, the decision threshold is crucial in the adaptation of the proposed IDS to various constraints of WSN deployment. As shown in Table 6, varying the classification threshold produces a controlled trade-off between detection capability and false-alarm rate. Lower thresholds enhance recall via increased attack sensitivity, while higher thresholds reduce false positives at the expense of marginally diminished detection performance. These results highlight the model's stability and versatility, allowing it to be fine-tuned for different deployment scenarios. For instance, security-sensitive WSN applications may favor lower thresholds to maximize detection rates, while resource-constrained or low-alert

environments may adopt higher thresholds to minimize false alarms without severely impacting detection performance.

When compared to previous approaches, our proposed XGBoost model remains highly competitive. While certain deep learning architectures may report marginally higher accuracy, they typically require substantially greater computational resources and are more complex to train. This makes them poorly suited for resource-constrained edge nodes. Furthermore, even within gradient-boosting implementations, recent studies often use complex feature extractors or meta-heuristic layers [15, 17] to improve performance on the WSN-DS dataset. However, this multiplies the required CPU cycles. In contrast, our approach demonstrates that a structurally compressed model (235 KB) with a strictly bounded inference path (510 trees requiring a maximum of 2,550 logical comparisons) can achieve high-fidelity detection comparable to that of other models (99.74%) without the prohibitive energy costs. This establishes our framework not as a competitor in raw algorithmic complexity, but as a superior, realistic solution for immediate cluster-head deployment.

Overall, the findings demonstrate that the proposed XGBoost-based IDS successfully combines high detection accuracy, low computational cost, and strong stability. These characteristics confirm its suitability as an optimal lightweight solution for detecting DoS attacks in WSNs.

Table 7. Comparative analysis of the proposed framework with state-of-the-art studies

Reference	Methodology	Accuracy
Rameshkumar et al. [28]	Progressive TL + Deep Q-Network	89.00%
Subramani and Selvi [27]	Deep Fuzzy CNN	98.90%
Meenakshi & Karunkuzhali [22]	Optimized self-attention VAE-GAN	99.20%
Shakya et al. [29]	Bayesian-opt DL + RL	99.50%
Nguyen et al. [16]	GSWO-CatBoost	99.65%
Pandey et al. [17]	Tabu Search–optimized RF	99.67%
Jiang et al. [18]	ST-IAOA-XGBoost	99,71%
Proposed Work	Optimized XGBoost	99.74%

6. Conclusion

This paper presented a computationally efficient intrusion detection framework designed to defend against the growing threat of DoS attacks in wireless sensor networks. By combining the XGBoost algorithm with a meticulously optimized 16-feature vector, we developed a solution that successfully balances high detection accuracy with minimal computational overhead. Our experimental evaluation confirmed that tree-based ensemble learners, particularly XGBoost, provide superior stability and precision compared to traditional kernel-based or probabilistic models when operating on imbalanced network traffic.

The findings indicated that resolving feature redundancies not only preserves predictive performance but also significantly reduces the total cross-validation runtime, a critical factor for extending the operational longevity of sensor nodes. The proposed model exhibited nearly perfect recall for the majority of attack classes. However, the identified limitation in detecting TDMA attacks highlights a specific area for further refinement.

Future research will concentrate on enhancing the detection sensitivity for low-frequency attack patterns through advanced sampling techniques or hybrid feature engineering. Furthermore, building upon our initial hardware validation, we plan to deploy the suggested architecture across a fully realistic operational platform to evaluate its network-wide energy efficiency and real-time responsiveness in heterogeneous WSN deployments.

References

[1] D. Kandris and E. Anastasiadis, Advanced Wireless Sensor Networks: Applications, Challenges and Research Trends, Electronics **13**, 2268 (2024).

[2] A. Förster, J. Dede, A. Könsgen, K. Kuladinithi, V. Kuppusamy, A. Timm-Giel, A. Udugama, and A. Willig, A beginner's guide to infrastructure-less networking concepts, IET Networks **13**, 66 (2024).

[3] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks, Journal of Sensors **2016**, (2016).

[4] M. Faris, M. N. Mahmud, M. F. M. Salleh, and A. Alnoor, Wireless sensor network security: A recent review based on state-of-the-art works, International Journal of Engineering Business Management **15**, 18479790231157220 (2023).

[5] Fawaz, A Survey of Intrusion Detection Schemes in Wireless Sensor Networks, American Journal of Applied Sciences **9**, 1636 (2012).

[6] M. Hosseini Shirvani and A. Akbarifar, Anomaly-based Detection of Blackhole Attacks in WSN and MANET Utilizing Quantum-metaheuristic algorithms, Journal of Communication Engineering (2020).

[7] I. Batra, S. Verma, Kavita, and M. Alazab, A lightweight IoT-based security framework for inventory automation using wireless sensor network, International Journal of Communication Systems **33**, e4228 (2020).

[8] K. Shaukat, S. Luo, V. Varadharajan, I. Hameed, S. Chen, D. Liu, and J. Li, Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity, Energies **13**, 2509 (2020).

[9] V. Sivagaminathan, M. Sharma, and S. K. Henge, Intrusion detection systems for wireless sensor networks using computational intelligence techniques, Cybersecurity **6**, (2023).
[10] R. Wazirali and R. Ahmad, Machine Learning Approaches to Detect DoS and Their Effect on WSNs Lifetime, Computers, Materials & Continua **70**, 4922 (2022).
[11] S. Jiang, J. Zhao, and X. Xu, SLGBM: An Intrusion Detection Mechanism for Wireless Sensor Networks in Smart Environments, IEEE Access **8**, 169548 (2020).
[12] P. Manjula and S. B. Priya, An effective network intrusion detection and classification system for securing WSN using VGG-19 and hybrid deep neural network techniques, IFS **43**, 6419 (2022).
[13] G. Liu, H. Zhao, F. Fan, G. Liu, Q. Xu, and S. Nazir, An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs, Sensors **22**, 1407 (2022).
[14] Narendra Kumar, Kalai Vani Ys, Muneshwara M S, Chittibabulu Sape, V. Subba Reddy, and Idimadakala Madhavilatha, Adaptive Spatio-Temporal Deep Learning for Robust Cyber Threat Detection Across IoT Environments, EAI Endorsed Trans IoT **11**, (2026).
[15] A. M. Arabiat and Y. G. Eljaafreh, Intrusion Detection in Wireless Sensor Networks Using ML Based Classification of Denial of Service (DoS) Attacks, JCM 501 (2025).
[16] T. M. Nguyen, H. H.-P. Vo, and M. Yoo, Enhancing Intrusion Detection in Wireless Sensor Networks Using a GSWO-CatBoost Approach, Sensors **24**, 3339 (2024).
[17] V. K. Pandey, S. Prakash, T. K. Gupta, P. Sinha, T. Yang, R. S. Rathore, L. Wang, S. Tahir, and S. T. Bakhsh, Enhancing intrusion detection in wireless sensor networks using a Tabu search based optimized random forest, Sci Rep **15**, 18634 (2025).
[18] L. Jiang, H. Gu, L. Xie, H. Yang, and Z. Na, ST-IAOA-XGBoost: An Efficient Data-Balanced Intrusion Detection Method for WSN, IEEE Sensors J. **25**, 1768 (2025).
[19] G. Vembu and D. Ramasamy, Optimized deep learning-based intrusion detection for wireless sensor networks, Int J Communication **36**, e5254 (2023).
[20] Md. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs, Int. J. Inf. Secur. **23**, 2139 (2024).
[21] T. Sood, S. Prakash, S. Sharma, A. Singh, and H. Choubey, Intrusion Detection System in Wireless Sensor Network Using Conditional Generative Adversarial Network, Wireless Pers Commun **126**, 911 (2022).
[22] B. Meenakshi and D. Karunkuzhali, Enhancing cyber security in WSN using optimized self-attention-based provisional variational auto-encoder generative adversarial network, Computer Standards & Interfaces **88**, 103802 (2024).
[23] V. Gharavian, R. Khosrowshahli, Q. H. Mahmoud, M. Makrehchi, and S. Rahnamayan, *Intrusion Detection for Wireless Sensor Network Using Graph Neural Networks*, in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)* (IEEE, Mexico City, Mexico, 2023), pp. 807–813.
[24] D. B. Mohan, P. Arumugam, and A. R., Osprey optimization algorithm integrated with graph neural networks for intrusion detection in wireless sensor networks, Sci Rep **15**, 44849 (2025).
[25] K. Ramana, A. Revathi, A. Gayathri, R. H. Jhaveri, C. V. L. Narayana, and B. N. Kumar, WOGRU-IDS — An intelligent intrusion detection system for IoT assisted Wireless Sensor Networks, Computer Communications **196**, 195 (2022).
[26] R. B. Kagade and S. Jayagopalan, Optimization assisted deep learning based intrusion detection system in wireless sensor network with two-tier trust evaluation, Int J Network Mgmt **32**, e2196 (2022).
[27] S. Subramani and M. Selvi, Intelligent IDS in wireless sensor networks using deep fuzzy convolutional neural network, Neural Comput & Applic **35**, 15201 (2023).
[28] S. Rameshkumar, R. Ganesan, and A. Merline, Progressive Transfer Learning-based Deep Q Network for DDOS Defence in WSN, Computer Systems Science and Engineering **44**, 2379 (2023).
[29] V. Shakya, J. Choudhary, and D. P. Singh, IRADA: integrated reinforcement learning and deep learning algorithm for attack detection in wireless sensor networks, Multimed Tools Appl **83**, 71559 (2024).
[30] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, San Francisco California USA, 2016), pp. 785–794.
[31] M. A. Hamzah and S. H. Othman, A Review of Support Vector Machine-based Intrusion Detection System for Wireless Sensor Network with Different Kernel Functions, Int J Innov Comp **11**, 59 (2021).
[32] D. Meng, H. Dai, Q. Sun, Y. Xu, and T. Shi, Novel Wireless Sensor Network Intrusion Detection Method Based on LightGBM Model, IAENG International Journal of Applied Mathematics **52**, (2022).
[33] K. Vikas, W. Charu, B. S. Bharat, and M. Manisha, Ensemble Learning Based Intrusion Detection for Wireless Sensor Network Environment, Int J Performability Eng **20**, 541 (2024).