I.J. Wireless and Microwave Technologies, **2026, 1, 1-23**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.01.01

Evaluating Machine Learning Efficacy for DoS Intrusion Detection in Wireless Sensor Networks

Samuel Mends
Department of Computer Science, University of Ghana, Lego-Accra, Ghana
E-mail: mendssamuel@gmail.com
ORCID iD: https://orcid.org/0009-0009-8309-6215

Kofi Sarpong Adu-Manu*
Department of Computer Science, University of Ghana, Lego-Accra, Ghana
E-mail: ksadu-manu@ug.edu.gh
ORCID iD: https://orcid.org/0000-0003-0677-6523
*Corresponding Author

Received: 15 June, 2025; Revised: 25 August, 2025; Accepted: 22 October, 2025; Published: 08 February, 2026

Abstract: Wireless Sensor Networks (WSNs) are integral to mission-critical applications, including environmental monitoring, smart infrastructure, and healthcare. However, they are particularly vulnerable to denial-of-service (DoS) attacks, which can deplete the node's energy and disrupt communication. This study examines the effectiveness of various machine learning algorithms in enhancing intrusion detection within WSNs, focusing on balancing detection accuracy and computational efficiency. Utilising the Network Simulator-2 (NS-2) generated WSN-DS dataset, seven algorithms—K-Nearest Neighbours (KNN), Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Stacking Classifier, AdaBoost, and Artificial Neural Network (ANN)—were implemented and evaluated. The experimental results indicate that AdaBoost achieved the highest overall performance, with an accuracy of 99.7%, ROC-AUC of 0.996, and detection speed of 1.6 min, underscoring its suitability for real-time intrusion detection. Stacking and Random Forest also demonstrated high accuracy (99.7% and 99.6%, respectively) but required slightly longer detection times of 7.07 and 7.33 min, respectively. In contrast, KNN exhibited the longest detection time (86.2 min) due to its high computational overhead, whereas Naïve Bayes was the fastest (0.02 min) but had lower precision (0.757) and F1-score (0.771). AdaBoost demonstrated superior detection accuracy, efficiency, and adaptability under constrained WSN conditions, outperforming all other algorithms across multiple performance metrics. These findings offer a practical benchmark for developing lightweight, high-performance intrusion detection systems in resource-limited wireless sensor environments, thereby enhancing the resilience and reliability of next-generation WSN infrastructures.

Index Terms: Intrusion Detection Systems (IDS), Wireless Sensor Networks (WSNs), Denial of Service (DoS) Attacks, Machine Learning for Security, AdaBoost and Algorithm Performance

1. Introduction

Wireless Sensor Networks (WSNs) have emerged as transformative technologies that enable real-time data collection and monitoring across diverse applications such as environmental management, healthcare, and smart infrastructure [1,2]. These networks consist of spatially distributed sensors that communicate wirelessly to monitor physical or environmental conditions [3]. However, despite their utility, WSNs are highly susceptible to security threats, particularly Denial of Service (DoS) attacks, which can severely disrupt network functionality by depleting node resources such as energy and bandwidth [4]. For instance, in healthcare settings, a DoS attack can delay the transmission of critical patient data, potentially resulting in life-threatening consequences.

Intrusion Detection Systems (IDS) have been developed as pivotal defence mechanisms to monitor, detect, and respond to such attacks in WSNs [5,6,7]. Although partially effective, traditional IDS approaches often struggle to satisfy the unique constraints of WSNs, such as limited computational power and energy resources [8]. Recent advances in machine learning (ML) and artificial neural networks (ANN) have been used to address these challenges and enhance IDS performance. These techniques offer adaptive, real-time intrusion detection capabilities, making them suitable for resource-constrained environments like WSNs [9].

DoS attacks remain a significant challenge in WSNs, threatening their availability, integrity, and confidentiality [10]. Existing studies have explored various ML-based IDS approaches, yet gaps persist in balancing detection speed and resource efficiency [11]. For example, [4] highlighted security challenges in WSN protocols. However, they did not address detection accuracy and computational efficiency tradeoffs. Similarly, [9] evaluated ML algorithms for IDS in WSNs but did not focus on their detection speed. Addressing these gaps is critical, as attackers devise sophisticated methods to compromise WSNs, underscoring the need for efficient and rapid detection mechanisms.

This study aims to evaluate the efficacy of ML algorithms, specifically AdaBoost, Stacking, Random Forest, and K-Nearest Neighbors (KNN), in detecting DoS attacks in WSNs. These algorithms were selected for their diverse characteristics: AdaBoost's ability to converge quickly to accurate models, Stacking's ensemble learning advantages, Random Forest's robustness, and KNN's simplicity and interpretability [9,11]. By utilizing Network Simulator-2 (NS-2) to simulate realistic DoS attack scenarios, this study provides a controlled environment to assess and compare the performance of the algorithms. NS-2 was chosen for its flexibility and widespread use in modelling network behaviour under various attack conditions [8].

Machine Learning (ML) and Deep Learning (DL) techniques are employed to enhance the early detection of threats and attacks on Wireless Sensor Networks (WSNs). This study investigated ML and DL methodologies to ascertain the most effective approach for rapid intrusion detection in a WSN environment, focusing on Denial of Service (DOS) attacks. This study presents current performance data on the most efficient ML techniques for Intrusion Detection Systems (IDS) in WSNs, emphasizing the rapidity of intrusion or threat detection. Additional parameters, including accuracy, precision, Recall, F1 score, confusion matrix, and model training speed, will be evaluated to establish a baseline for future research in the WSN Security knowledge domain.

The remainder of this paper is organized as follows. Section 2 reviews related works on WSNs, IDS, DoS attacks, and ML algorithms for intrusion detection. Section 3 describes the methodology, including the dataset and simulation setup. Section 4 presents and discusses the performance metrics. Finally, Section 5 concludes the study with key findings and directions for future research.

2. Related Work

Wireless Sensor Networks (WSNs) have emerged as critical technologies for environmental monitoring, healthcare, and agriculture. However, increasing reliance on WSNs for real-time data collection and processing has heightened their vulnerability to sophisticated cyber-attacks. This section critically reviews recent advances in machine learning (ML) and deep learning (DL) approaches for intrusion detection in WSNs, emphasizing their shortcomings in the current study's objectives.

In [12], the authors systematically analyzed ML and DL approaches for intrusion detection, benchmarking their accuracy, precision, Recall, and F1-score metrics. Although their work highlighted the superior feature extraction capabilities of deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), they did not address the computational overhead of these methods in resource-constrained environments such as WSNs. This gap is critical because real-time detection requires accuracy and efficiency regarding speed and resource usage, which this study aims to address.

In [13], the authors proposed a novel ensemble method combining Random Forest (RF) and Gradient Boosting (GB) classifiers to enhance detection accuracy. Their approach demonstrated over 95% precision and Recall on benchmark datasets, emphasizing the robustness of ensemble learning. However, the lack of focus on detection speed and computational efficiency limits its applicability in real-world WSN deployments, where a rapid response to threats is crucial for maintaining network functionality.

In [14], the authors investigated ML models for detecting DoS attacks in WSNs, focusing on SVM, Decision Trees, and Gradient Boosting. Although their findings demonstrated significant improvements in detection accuracy, they did not comprehensively address scalability or computational efficiency. By integrating these metrics into the evaluation, the current study offers a more holistic approach to intrusion detection in WSNs.

In [17], the authors proposed an accelerated DL model with advanced feature selection for intrusion detection, achieving high accuracy and reduced false alarm rates. However, the complexity of their model raises concerns regarding its feasibility in resource-constrained WSN environments. This study adopts simpler ML models, such as AdaBoost and Random Forest, balancing performance with computational feasibility.

In [18], the authors designed an integrated rule-based ML system for detecting DoS attacks in WSNs. Their approach improved the detection precision and reduced false alarms but did not prioritize the detection speed. This study builds on their work by emphasizing real-time applicability and addressing the tradeoffs between detection accuracy, speed, and resource usage.

In [23], the authors introduced the WSN-DS dataset to evaluate intrusion detection systems for WSNs, offering realistic traffic patterns and attack scenarios. Although widely adopted, the dataset's static nature and limited representation of emerging threats hinder its relevance for evolving attack landscapes. Algorithms evaluated on WSN-DS, such as Support Vector Machines (SVMs) and Naive Bayes, demonstrated high accuracy but suffered limitations in handling imbalanced data and scalability issues. The current study builds on these insights by using more dynamic simulation environments and focusing on the detection speed to enhance real-world applicability.

Table 1. Comparison of Papers on Intrusion Detection Methods

Author(s)	Focus of Paper	ML Algorithms	Metrics	Results	Limitation
[15]	Systematic study of ML and DL approaches for intrusion detection.	Various ML and DL models	Accuracy, Precision, Recall, F1	Provided a comparative analysis of ML vs DL approaches.	The study lacked a specific focus on WSNs; the results were generalized.
[16]	Ensemble method for intrusion detection in WSNs.	Custom Ensemble Method	Detection Rate, False Alarm Rate	Improved detection rate with reduced false positives.	Focused only on ensemble methods; lacks diverse algorithm comparison.
[14]	DoS attack detection in WSNs using ML.	ANN, Decision Trees, KNN	Accuracy, F1, Detection Speed	ANN demonstrated high accuracy with acceptable speed.	Limited to ANN models and simple datasets.
[17]	Accelerated DL model for IDS in WSNs.	Accelerated DL	Precision, Recall, F1, Detection Speed	Achieved improved detection rates with feature selection.	Resource-intensive; lacks scalability analysis.
[18]	Integrated rule-based and ML system for DoS detection.	Rule-based + ML classifiers	Accuracy, Latency	Enhanced DoS detection with hybrid methods.	Rule-based systems can be less adaptive to novel attacks.
[19]	Development of WSN-DS dataset for IDS.	N/A	Dataset quality evaluation	Introduced WSN-DS dataset for IDS research.	There is no direct application of ML for evaluation.
[20]	Multi-layer IDS with Logistic Regression, Decision Tree, and Random Forest.	Logistic Regression, Decision Tree, Random Forest	False Alarm Rate, Detection Accuracy	Reduced false alarm rates; achieved 98.5% detection accuracy.	High computational complexity; concerns about feasibility in energy-constrained WSNs.
[21]	Evaluation of ML and DL methods for intrusion detection.	Autoencoders, LSTM, Random Forest	Feature Handling, Detection Performance	DL models excelled with large datasets and complex features.	High computational demands are incompatible with WSN constraints.
[22]	ML-based intrusion detection systems in smart healthcare.	SVM, Random Forest	Detection Rate, Adaptability	Achieved detection rates above 98%.	Did not address detection latency implications critical for WSNs.

The authors in [24] designed a multi-layer intrusion detection system using algorithms such as logistic regression, decision trees, and random forest. Their approach significantly reduced false alarm rates and achieved a detection accuracy of 98.5%. However, the computational complexity of the layered system raises concerns regarding its feasibility in energy-constrained WSN environments. This limitation further justifies the aim of the current study to identify lightweight yet efficient algorithms for intrusion detection.

In [21], the authors evaluated ML and DL methods for intrusion detection, highlighting the promise of deep learning models like Autoencoders and Long Short-Term Memory (LSTM) networks. While these methods handle large datasets and complex features, their computational demands are incompatible with the WSN constraints. This study addresses this gap by prioritizing resource efficiency and detection performance.

Table 2. Classification of Non-Intrusion Detection Papers in WSNs

Author (s)	Focus of Paper	Application Area	ML Algorithms	Metrics	Results	Limitation
[25]	Review of ML methods for advanced WSNs.	General WSNs	Various ML models	N/A	A comprehensive review of ML in WSNs, highlighting trends and applications.	Did not focus on specific use cases or datasets.
[26]	Insights on WSNs in agriculture through bibliometric analysis.	Agriculture	N/A	Citation trends, keyword analysis	Highlighted the growth and focus areas in WSN research for agriculture.	Lacks technical analysis of ML applications.
[27]	ML applications in smart cities.	Smart Cities	Various ML models	N/A	Provided an in-depth literature review on ML-driven smart cities.	Did not address WSN-specific challenges in smart cities.
[28]	A systematic review of QoS in WSNs using ML.	Quality of Service in WSNs	Various ML models	N/A	Identified trends in QoS optimization for WSNs using ML.	Did not focus on implementation challenges or practical applications.
[29]	Physical and cyber security in data centres.	Datacentres	N/A	Event processing metrics	Explored the use of complex event processing for enhanced security.	Focused on data centres; limited applicability to WSNs.

Table 3. Comparative Analysis of Related Works on Machine Learning–Based Intrusion Detection in WSNs

Ref.	Study Focus	Methodology / Algorithms Used	Key Findings	Limitations / Gaps Identified	Relevance to Current Study
[12]	Systematic study of ML and DL approaches for intrusion detection	Comparative analysis of SVM, RF, ANN, CNN	Highlighted DL models outperform ML in feature learning	Lacked focus on detection speed and energy efficiency	The current study builds on this by evaluating speed-performance trade-offs
[13]	Ensemble IDS for WSNs	Hybrid AdaBoost–Bagging ensemble	Achieved improved accuracy over base models	Did not assess model latency or WSN-specific constraints	AdaBoost implementation refined here for real-time detection
[14]	Detection of DoS attacks in WSNs	RF, KNN, and SVM classifiers	RF achieved 99% accuracy for DoS detection	The dataset and attack diversity are limited	The present study extends the analysis to multiple DoS types (Flooding, Grayhole, etc.)
[15]	ML and DL comparative IDS	CNN, RF, and Naïve Bayes	CNN achieved the best classification accuracy	Computational demands are unsuited for WSN nodes	This study prioritises computationally lightweight models
[16]	Advanced ensemble-based IDS for WSNs	Meta-learning ensemble using boosting and bagging	Enhanced detection precision	Did not validate against large datasets	The current study validates on a large NS-2–derived WSN-DS dataset
[17]	Hybrid ML–DL IDS with feature selection	Accelerated deep learning with GA optimisation	Improved detection and reduced false alarms	High computational overhead, no detection-time analysis	This work emphasises detection speed in WSN-constrained systems
[18]	Rule-based and ML DoS detection	Rule-based filtering + ML classifiers (DT, NB)	Improved accuracy and interpretability	Lacked quantitative timing analysis	We address this by integrating detection speed and efficiency metrics
[19]	Creation of WSN-DS dataset	Simulated dataset using NS-2 (DoS attacks)	Provided a benchmark dataset for IDS research	The dataset lacks real-world noise and mobility conditions	This study uses WSN-DS for algorithmic benchmarking
[20]	Multi-layer ML-based IDS	Logistic Regression, DT, RF	Achieved 98.5% accuracy and a low false alarm rate	Energy use and scalability were not assessed	Builds on their structure, adding scalability and detection speed evaluation
[21]	Survey of ML and DL IDS	Overview of DL advances (LSTM, Autoencoders)	Identified DL as robust for complex IDS tasks	High training cost for real-time systems	Our study prefers lightweight, faster ML alternatives for WSNs
[22]	IDS in IoT-based healthcare	RF, SVM for anomaly detection	Achieved >98% detection rate	Limited generalisation beyond healthcare IoT	Extends IDS applicability to environmental and general WSN contexts
[23]	Development of WSN-DS dataset	NS-2 simulation including DoS variants	Introduced reproducible IDS benchmark data	No experimental algorithm validation	Dataset used as the foundation for our ML-based evaluation
[24]	Layered IDS for WSNs	Multi-layer classifier integrating RF and DT	Enhanced detection of complex attacks	Did not include deep learning or hybrid models	Our hybrid comparative model integrates an ANN for broader insight
[25]	Review of ML in WSNs	Analytical review of RF, DT, and SVM	RF shown as best for classification	No metrics for latency or detection delay	Our study quantifies detection time as a key parameter
[26]	WSNs in agriculture	Bibliometric analysis	Highlighted WSN applications in precision agriculture	Non-technical; lacks IDS modelling	Contextual relevance—underscores real-world WSN significance
[27]	ML in smart cities	Review of ML frameworks	Emphasised scalability and adaptive learning	Did not include security or IDS models	Our findings extend adaptive ML to WSN security domains
[28]	QoS optimisation in WSNs via ML	ML-based prediction of network performance	Enhanced network reliability	Did not address intrusion detection	Our IDS framework complements QoS improvement goals
[29]	Cyber-physical event security	Complex event processing (CEP)	Identified ML potential for real-time security	Focused on data centres, not WSNs	Provides a basis for integrating IDS into broader IoT ecosystems

In [22], the authors explored ML-based intrusion detection systems in smart healthcare, achieving detection rates above 98% using SVM and Random Forest algorithms. Although their work demonstrated the adaptability of ML techniques in IoT-enabled environments, they did not explore the implications of detection latency, which is critical for preventing resource depletion in WSNs under attack. In [25], the authors reviewed ML techniques for WSNs and identified Random Forest and Decision Tree algorithms as effective for classification tasks. Their findings emphasized the importance of feature selection in reducing computational overhead, with Random Forest achieving a detection accuracy of 99.6% on the NSL-KDD dataset. However, their lack of emphasis on time-sensitive metrics such as detection speed highlights a significant gap. In real-world WSNs, delays in intrusion detection can exacerbate resource depletion and compromise the entire network, underscoring the need for this study's focus on real-time detection.

In [26], the authors conducted a bibliometric analysis of Wireless Sensor Networks (WSNs) in agriculture, highlighting the growing research interest and diverse applications of WSNs in this domain. Their study emphasized using WSNs for precision agriculture, including soil moisture monitoring, pest detection, and crop health assessment. While the analysis identified the potential for machine learning (ML) to enhance WSN-based agricultural applications, it did not delve into specific ML techniques or their effectiveness in addressing key challenges, such as energy

efficiency and real-time data analysis. This omission emphasizes a gap in translating bibliometric findings into practical, ML-driven solutions for resource-constrained environments, such as WSNs in agriculture, reinforcing the importance of research focused on integrating efficient ML algorithms tailored for such applications.

In the work done by [27], the authors provided an extensive literature review of the role of machine learning in enabling smarter cities, with a particular focus on integrating WSNs for urban planning, energy management, and transportation systems. Their work highlighted the efficacy of ML techniques, such as Support Vector Machines and Neural Networks, in optimizing urban infrastructure through predictive analytics and anomaly detection. However, this review lacked specificity in addressing the unique challenges of WSNs, such as energy constraints and real-time processing requirements. For instance, while predictive maintenance in smart cities benefits from ML-driven insights, the absence of discussion on latency-sensitive applications limits the applicability of the findings to scenarios requiring immediate responses. This gap highlights the need for targeted studies on ML algorithms for real-time intrusion detection and resource optimization in WSN-enabled smart city environments.

Pundir and Sandhu (2021) emphasized the quality of service (QoS) metrics, such as latency and throughput, when integrating ML algorithms for WSN intrusion detection. Their findings suggest that Decision Tree and Random Forest algorithms effectively balance QoS and security, achieving detection accuracies above 97%. However, their limited focus on attack-specific metrics, such as false positives and real-time detection capabilities, leaves a gap that this study seeks to fill.

In [29], the authors explored complex event processing (CEP) to enhance data centre physical and cyber security. Their study provided a comprehensive overview of the challenges and advancements in using CEP to detect and mitigate cyber threats, including data breaches and distributed denial-of-service (DDoS) attacks. While the focus on CEP techniques offers valuable insights for event-driven architectures, applying these findings to WSNs remains limited due to fundamental differences in network topology, resource constraints, and data transmission protocols. Additionally, the emphasis of this paper on datacenter-level scalability and security strategies overlooks the need for lightweight, real-time intrusion detection mechanisms in WSNs. This limitation reinforces the importance of extending CEP methodologies to resource-constrained environments. It addresses this gap by focusing on machine learning models prioritizing detection speed and energy efficiency.

Although these studies provide valuable insights into the potential of ML and DL for intrusion detection, their limited attention to detection speed and resource efficiency constrains their practical applicability in WSNs. In real-world scenarios, delays in detecting attacks can lead to catastrophic outcomes such as resource depletion or complete network failure. This study addresses these limitations by evaluating ML algorithms tailored for WSNs, focusing on the accuracy, precision, Recall, F1-score, and detection speed metrics. By bridging these gaps, this study aims to advance the development of scalable, efficient, and real-time intrusion detection systems for WSNs.

The related works presented in Table 3 predominantly emphasise accuracy and the reduction of false positives, while overlooking detection speed, energy efficiency, and scalability, which are critical factors for resource-limited wireless sensor networks (WSNs). In contrast to previous studies that focused on accuracy benchmarks or static simulations, this study introduces a multi-metric performance evaluation that integrates speed, efficiency, and accuracy, which is validated using a comprehensive WSN-DS dataset. By identifying AdaBoost as the fastest and most efficient machine learning algorithm for detecting denial-of-service (DoS) intrusions, this study advances existing intrusion detection system (IDS) research towards real-time, lightweight, and deployable solutions for WSN environments.

3. Methodology

The methodology of this study was designed to identify the most efficient machine learning (ML) algorithm for intrusion detection in adversarial Wireless Sensor Networks (WSNs), addressing the challenges of imbalanced data, computational constraints, and real-time detection. The approach integrates a simulated dataset (obtained from Kaggle - https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds), comprehensive pre-processing, rigorous model evaluation, and mathematical representations of the algorithms to align with the objectives outlined in the introduction and the gaps identified in related works.

3.1 NS-2 Simulator

Network Simulator-2 (NS-2) is a discrete event-driven network simulator widely used to simulate networking protocols and scenarios, including WSNs. In this study, NS-2 was utilized to generate data reflecting realistic WSN behaviour under normal and attack conditions. The simulation setup included the following steps.

3.2 Simulation of DoS Attacks

1. *Blackhole Attack*
    - Simulated by programming a node to drop all incoming packets maliciously instead of forwarding them, effectively creating a communication void.
2. *Grayhole Attack*

- Configured nodes drop packets selectively based on predefined conditions, such as source or packet type, mimicking erratic network behaviour.

3. *Flooding Attack*
   - The nodes were set to overwhelm the network with excessive packet requests, consuming bandwidth and draining node energy.
4. *Scheduling Attack*
   - Artificial delays in packet transmission were introduced, leading to a disruption in real-time data delivery.

3.3 Key Features of NS-2 in Simulating WSNs

- *Customizable Protocols:* Allowed modification of routing protocols to introduce malicious behaviours specific to DoS attacks.
- *Scalability:* Simulated large-scale WSNs with hundreds of nodes provide a robust environment for generating diverse data patterns.
- *Metrics Captured:* Packet rate, node energy consumption, and transmission delays were recorded to characterize the normal and attack behaviours.

The resulting data from these simulations were curated to form the WSN-DS dataset, providing a comprehensive view of the WSN behaviour under various scenarios. The WSN-DS is designed to assist in the detection and classification of four (4) different types of Denial of Service (DoS) attacks (that is, Blackhole, Gray hole, Flooding and Scheduling attacks). The dataset was taken through the model development processes, including data assessment, data cleaning and data pre-processing, all to get the dataset WSN-DS ready to be modelled by the various identified machine learning algorithms. Fig. 1 illustrates the structured flowchart of the ML model development lifecycle adopted in this study.

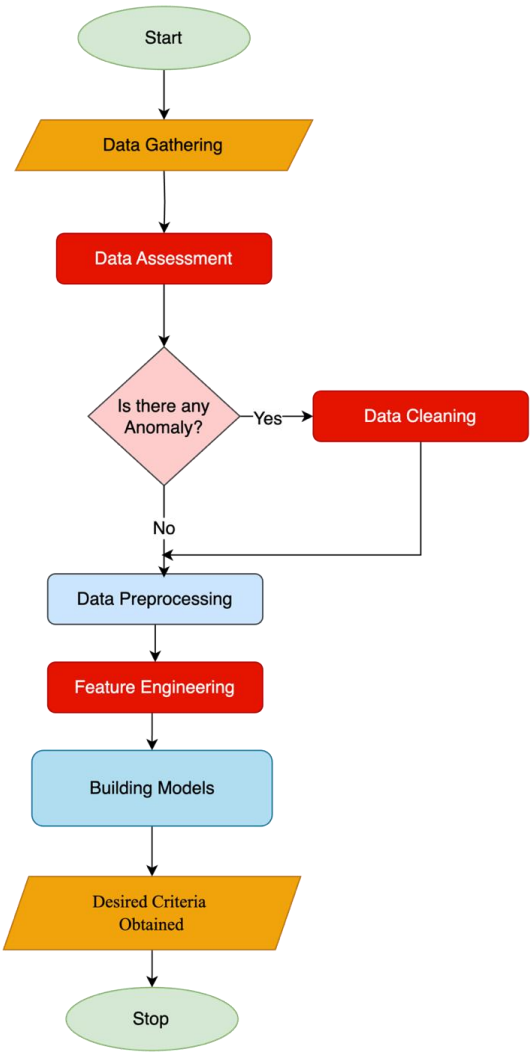

Fig. 1. Processes involved in modelling the dataset WSN-DS

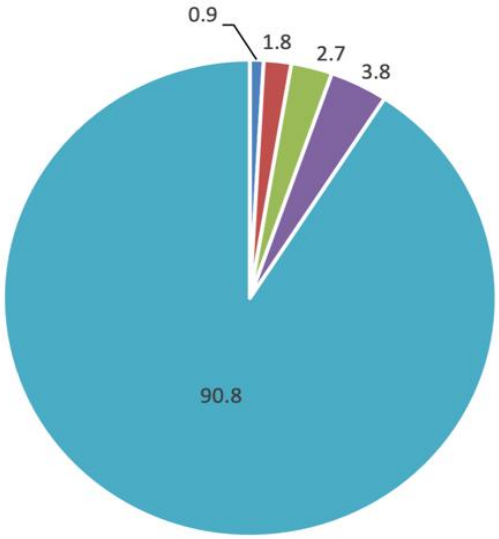

Fig. 2. Class distribution of attack type in the dataset WSN-DS

The process begins with data gathering and assessment, in which the dataset is evaluated for anomalies. If anomalies are detected, the data undergo a cleaning process to rectify or remove inconsistencies. The cleaned data then proceed to the pre-processing stage, which includes normalization, handling missing values, and balancing class distributions. Feature engineering follows, where relevant features are selected or extracted to enhance the model performance.

Subsequently, machine learning models were developed and evaluated against criteria such as accuracy, detection speed, and generalization metrics. The iterative process ensures the desired performance thresholds are achieved before the complete model development pipeline. This methodology provides a robust and reproducible approach to intrusion detection, addressing challenges such as data imbalances and computational constraints in Wireless Sensor Networks.

3.4 Dataset Source

The dataset used for this study, WSN-DS, was obtained from Kaggle. It is a publicly available dataset for intrusion detection in Wireless Sensor Networks (WSNs). The dataset encapsulates the behaviours of nodes under both normal and attack conditions, making it suitable for evaluating the performance of ML algorithms for detecting various types of Service (DoS) attacks.

3.4.1 Dataset Overview

In the assessment of the dataset (WSN-DS) from Kaggle, a total of three hundred and seventy-four thousand, six hundred and sixty-one (374,661) rows and nineteen (19) columns of data, including the target variable. The dataset was significant but did not contain any missing data. Eight thousand eight hundred and seventy-three (8,873) were duplicates, representing 2.368% of the dataset. Furthermore, it was identified that the target was heavily imbalanced, with the 'Normal' class being the model class covering over 90% of the samples, as shown in Fig. 2. The key features of the WSN-DS Dataset are presented in Table 4.

Table 4. Key Features of the WSN-DS Dataset

Feature Name	Description
PacketRate	Rate of packets transmitted per second.
NodeID	Identifier for the transmitting node.
EnergyConsumed	The energy utilized for transmission/reception.
AttackType	Type of DoS attack or Normal operation.
Latency	Time delay in packet transmission.
PacketDropRate	Percentage of packets dropped.

3.4.2 Dataset Use

The WSN-DS dataset contains labelled instances for the four types of DoS attacks.

1. *Blackhole Attack:* Nodes drop all incoming packets.
2. *Grayhole Attack:* Nodes selectively drop packets.
3. *Flooding Attack:* Nodes generate excessive traffic that overwhelms the network.
4. *Scheduling Attack:* Nodes introduce delays in packet transmission.

Although not curated by the researchers, this dataset was used in this study to evaluate the efficacy of various ML algorithms in detecting intrusions in WSNs.

3.5 Data Pre-processing

Data pre-processing is a critical step in ensuring the quality and reliability of the WSN-DS dataset for intrusion detection. Initially, data cleaning was performed to correct and remove erroneous, corrupted, or duplicate records. This process eliminated 8,873 duplicate rows, reducing the dataset to 365,788, with all 19 features intact, as no missing data was detected. The imbalanced sample distribution, heavily skewed towards the 'Normal' class (over 90%), was addressed in the subsequent modelling pipeline to prevent model bias towards the majority class. To mitigate the imbalance, both oversampling and undersampling techniques were applied based on the specific characteristics of the algorithms used.

Oversampling employed the Borderline Synthetic Minority Oversampling Technique (SMOTE), which creates synthetic samples for minority classes, particularly near decision boundaries. This approach was instrumental for algorithms like Random Forest, Stacking Classifiers, and Artificial Neural Networks (ANN), ensuring balanced training datasets without overfitting. Fig. 3 illustrates the effects of Borderline SMOTE, showing how synthetic data points were generated along minority class boundaries, enabling these algorithms to effectively detect complex patterns in attacks like Grayhole and TDMA.

Undersampling reduced the majority-class samples to balance class distributions while maintaining a manageable dataset size. This technique was particularly beneficial for computationally intensive algorithms like K-Nearest Neighbors (KNN), reducing memory usage and improving prediction efficiency and by mitigating the dominance of the majority class, undersampling enhanced KNN's ability to detect minority-class attacks by learning from a more representative set of neighbours. Fig. 4 demonstrates the undersampling process, highlighting how majority-class samples were reduced to achieve a balanced dataset, ensuring fair and effective model training. These pre-processing strategies collectively improved the performance and robustness of the models in detecting various intrusion types in WSN-DS.

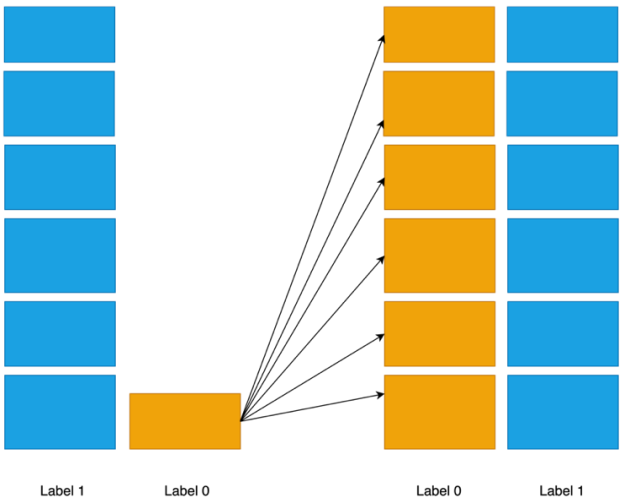

Fig. 3. Oversampling technique used on the dataset WSN-DS

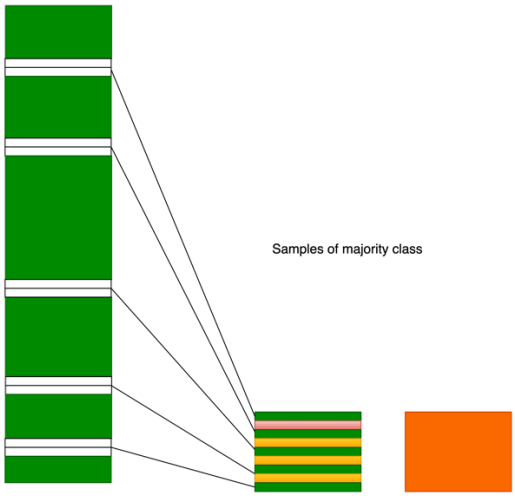

Fig. 4. Under-sampling technique used on the dataset WSN-DS

3.5.1 Feature Engineering

In feature engineering, raw data are cleaned and organized to be suitable for modelling. Apart from the cleaning performed on the dataset earlier, the dataset was split into features (*X*) and targets (*Y*), and the *ID* column was removed. The input features were split into training and testing sets using the sci-kit-learn train-test split method. 70% of the cleaned dataset, which is a total of two hundred and fifty-six thousand, fifty-one (256,051), was used for training, and 30% of the cleaned dataset, which is a total of one hundred and nine thousand seven hundred, and thirty-seven (109,737) was used for testing. Each algorithm is tailored to address the unique challenges of intrusion detection in Wireless Sensor Networks (WSNs), such as computational constraints, imbalanced data, and real-time performance requirements. The following is an expanded discussion of the implementation of each algorithm.

Seven machine learning algorithms, including an artificial neural network (ANN), were chosen for their appropriateness to WSN constraints and their potential for precise intrusion detection. Each algorithm and its mathematical basis and relevance are detailed below. The selection was methodically aligned with the specific challenges of intrusion detection in WSNs. Naive Bayes and KNN were selected for their computational efficiency, suitable for resource-constrained WSN environments. Random forest and stacking classifiers were included for their noise robustness and effectiveness in handling imbalanced data, which is essential for detecting rare attacks. AdaBoost and ANN were chosen for their adaptability in addressing complex patterns, which is ideal for nuanced intrusion scenarios. Feature selection was based on domain expertise and statistical significance. Essential features, including PacketRate, Latency, and PacketDropRate, were retained owing to their strong correlation with network congestion and attack behaviour, as validated through correlation analysis and Random Forest importance ranking. This methodology ensured the inclusion of features that enhanced the model interpretability and predictive performance while minimising redundancy.

3.5.1.1 K-Nearest Neighbors (KNN)

KNN is a non-parametric instance-based learning algorithm that classifies a sample based on the majority class among its k-nearest neighbours.

- *Justification*: Effective for anomaly detection in localized patterns due to its non-parametric nature.
- *Implementation Context:* KNN classifies data points based on the closest neighbours in the feature space. The imbalanced dataset was managed using Borderline SMOTE to increase minority class samples effectively. Cross-validation determined the optimal number of neighbours (k) between 3 and 11, with k=3 selected for best performance. The Manhattan distance metric was used for its computational efficiency, calculating absolute feature differences.
- *Application in Intrusion Detection:* KNN excels in detecting anomalies due to its simplicity and effectiveness with localized intrusion patterns in WSN data. Hyperparameter tuning minimized computational load during prediction.

3.5.1.2 Decision Tree (DT)

Decision Tree is a supervised learning algorithm that splits data into subsets based on feature conditions to form a tree-like structure for classification.

- ***Justification*:** Interpretable and capable of identifying essential features.
- *Implementation Context*: A Decision Tree model was built to classify the dataset into intrusion types by creating hierarchical decision rules. The depth of the tree was restricted to 10 to avoid overfitting and ensure computational efficiency.
- *Application in Intrusion Detection:* DT's interpretable nature makes it suitable for understanding the impact of individual features on intrusion types. For example, features such as packet rate or node response time were critical in forming decision splits for DoS attack classification

3.5.1.3 Random Forest (RF)

Random Forest (RF) is an ensemble method that builds multiple decision trees and aggregates their outputs for robust predictions.

- ***Justification:*** Robust against noise and capable of handling imbalanced data.
- *Implementation Context:* RF uses multiple decision trees (400 in this case) to build an ensemble model. Each tree operates on a bootstrap sample, and the final prediction is determined by majority voting. Stratified sampling ensured a balanced representation of intrusion classes during the training.
- *Application in Intrusion Detection:* RF excels in handling noisy and imbalanced data. Its robustness to overfitting made it highly suitable for detecting subtle patterns in WSN traffic, such as differentiating between Blackhole and Grayhole attacks

3.5.1.4 Naive Bayes (NB)

Naive Bayes is a probabilistic classifier based on Bayes' theorem that assumes feature independence.

- ***Justification*:** Lightweight and efficient for high-dimensional data.
- *Implementation Context:* Gaussian Naive Bayes was applied due to its ability to handle continuous data effectively. The model calculates the probabilities for each class and selects the class with the highest posterior probability.
- *Application in Intrusion Detection*: NB is lightweight and computationally efficient, making it ideal for WSNs with constrained resources. It was particularly effective in identifying high-confidence intrusions with probabilistic estimates, though it required additional strategies (e.g., SMOTE) to improve performance in minority classes.

3.5.1.5 Stacking Classifier

Stacking combines multiple classifiers by training a meta-classifier based on its output.

- ***Justification*:** Combines the strengths of multiple algorithms to improve the performance.
- *Implementation Context:* This ensemble model combines the predictions of Random Forest, AdaBoost, and Naive Bayes using XGBoost as a meta-classifier. Borderline SMOTE was applied to balance the training dataset.
- *Application in Intrusion Detection:* Stacking leverages the strengths of multiple algorithms to improve overall classification performance. For instance, while Random Forest focused on robust predictions, AdaBoost addressed challenging samples, and Naive Bayes provided probabilistic insights, making the ensemble particularly powerful against complex DoS attacks

3.5.1.6 AdaBoost

AdaBoost is an ensemble method that sequentially adds weak learners, focusing on the misclassified samples in each iteration.

- ***Justification:*** Adaptable to difficult-to-classify samples by focusing on misclassification.
- *Implementation Context:* AdaBoost sequentially added weak learners (decision stumps) and assigned higher weights to misclassified samples. This iterative learning process emphasized difficult-to-classify instances.
- *Application in Intrusion Detection:* AdaBoost's adaptability allowed it to handle varying intrusion complexities in WSN data. For example, it performed well in classifying rare attacks like Scheduling due to its focus on misclassified data during each iteration

3.5.1.7 Artificial Neural Network (ANN)

ANNs mimic the behaviour of the human brain and consist of interconnected layers of neurons for learning patterns.

- ***Justification*:** Well-suited for capturing complex nonlinear relationships in data.
- *Implementation Context*: The ANN had 17 input neurons (features), two hidden layers, and five output neurons (attack types). The ReLU activation function was used in the hidden layers for nonlinearity, and Softmax was applied in the output layer for multiclass classification. Batch size and learning rate optimized through grid search
- *Application in Intrusion Detection*: ANN is well-suited for learning complex nonlinear relationships in data. It captures intricate patterns in WSN traffic, such as subtle changes in node communication behaviour indicative of specific DoS attacks. However, it required careful tuning to balance performance with computational demands.

3.5.2 Algorithm Contributions to Study Goals

Each algorithm was evaluated for its ability to balance the detection accuracy, computational efficiency, and detection speed. Their implementation was fine-tuned to maximize the performance on the WSN-DS dataset while addressing the real-world constraints of WSNs. This holistic approach ensured the study's contribution toward lightweight, effective, and real-time intrusion detection in resource-constrained environments.

3.5.3 Model Evaluation Parameters

Each algorithm was evaluated using metrics aligned with the objectives of this study.

- *Accuracy*: Overall correctness of predictions.
- *Precision*: Proportion of correctly identified positive observations.

- *Recall*: Proportion of actual positives correctly identified.
- *F1-Score*: Harmonic mean of precision and Recall.
- *Detection Speed*: Time taken to identify intrusion.
- *ROC-AUC*: Model discrimination capability.

The methodology was further refined to address the specific gaps. First, the rationale behind the algorithm selection was expanded. Lightweight algorithms such as Naive Bayes were prioritized for suitability in energy-constrained WSNs. At the same time, ensemble methods like Random Forest and Stacking Classifiers were chosen for their robustness against data noise and imbalanced class distributions. ANN and AdaBoost were included because of their ability to learn complex patterns, which are critical for detecting subtle and rare attack types in the dataset.

Second, the impact of handling imbalanced data was thoroughly evaluated. Borderline SMOTE effectively enhanced Recall and F1 scores by increasing minority-class representation, enabling models like ANN and Random Forest to capture nuanced attack behaviours. Undersampling reduced the computational load of KNN and improved its precision without compromising Recall.

Finally, hyperparameter optimization was tailored to align the characteristics of the dataset and study goals. The number of neighbours (k) in the KNN, the tree depth in the Decision Tree, and the number of estimators in the Random Forest were optimized through cross-validation, ensuring a balance between computational efficiency and detection accuracy. Similarly, the ANN layer configuration and learning rate were fine-tuned to enhance the multiclass classification performance.

4. Findings and Results

Table 5 shows the descriptive statistics of the various metrics used to evaluate the overall performance of the proposed model. To attain an optimal outcome for the model, the value of each performance metric must be approximately one (1).

Table 5. Statistics of the various metrics

Name of Algorithm	MACHINE LEARNING METRICS MEASURED							
	A-Train	A-Test	ROC-AUC	CV-Score	D Speed	A PREC	A-RECALL	AN F1-Score
K-Nearest Neighbours	0.993	0.984		0.981	86.23	0.873	0.936	0.902
Decision Tree	0.988	0.988	0.993	0.986	0.025	0.916	0.972	0.943
Random Forest	1.000	0.996	0.997	0.996	7.327	0.967	0.984	0.975
Na ïve Bayes	0.951	0.952	0.971	0.952	0.023	0.757	0.829	0.771
Stacking	0.999	0.997	0.993	0.997	7.074	0.983	0.979	0.981
Adaboost	0.999	0.997	0.996	0.987	1.604	0.975	0.979	0.977
Artificial Neural Net.	0.958	0.959	0.983	0.957	1.343	0.782	0.899	0.831

Accuracy Train = A-TRAIN, Accuracy Test, Cross Validation Score =CV-Score, Detection Speed = D SPEED, AVG Precision = A PREC, AVG RECALL= A-RECALL, AVG FI-SCORE = A FI-SCORE

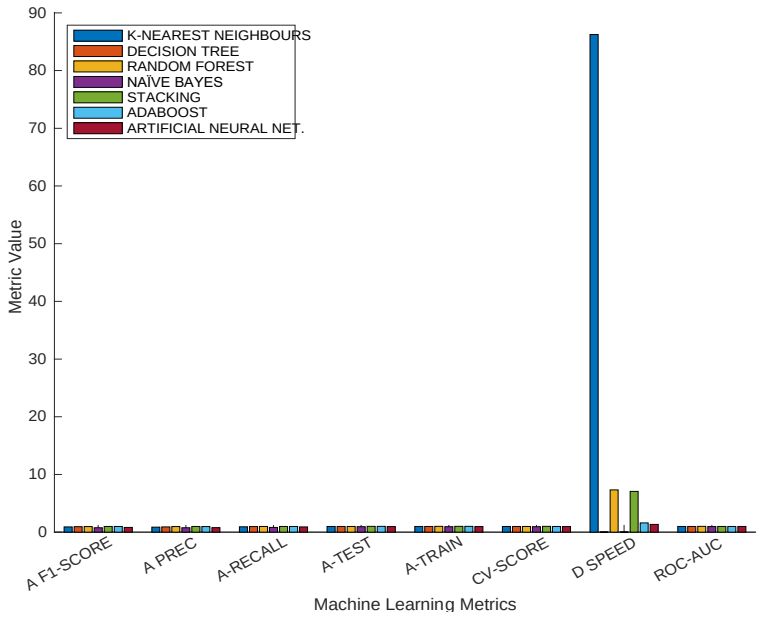

Fig. 5. Performance of the various machine learning algorithms

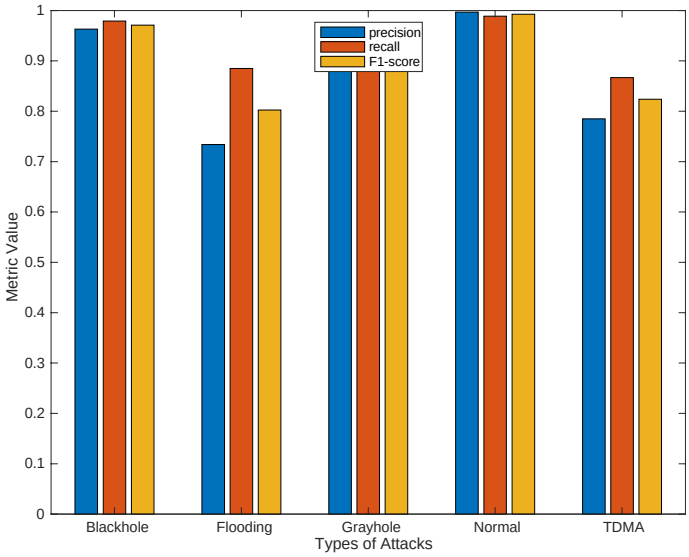

Fig. 6. Precision, Recall, and F1-score of the various attacks for KNN

Fig. 5 compares the performance of various machine learning algorithms across key metrics, including F1-Score, Precision, Recall, Accuracy, Cross-Validation Score, Detection Speed, and ROC-AUC. K-Nearest Neighbors (KNN) achieved the highest detection speed but struggled with precision, particularly for complex attack types such as Flooding and TDMA, indicating challenges in handling nuanced patterns. All other algorithms—Decision Tree, Random Forest, Naive Bayes, Stacking, AdaBoost, and Artificial Neural Network (ANN)—completed detection in less than 10 min, with ensemble methods such as Random Forest and Stacking, demonstrating superior balance across Precision, Recall, and F1-Score. These findings highlight the tradeoff between speed and accuracy, emphasizing the need for algorithms to ensure real-time detection while maintaining robust classification across all attack types in Wireless Sensor Networks.

Fig. 6 depicts the classification performance of the machine learning algorithms across the five attack types in the WSN-DS dataset, evaluated using Precision, Recall, and F1-Score. The results show that Normal, Grayhole, and Blackhole attacks achieved consistently high metrics, with values exceeding 0.95 for all three measures, indicating the effectiveness of the algorithms in detecting these attack types. In contrast, flooding attacks recorded the lowest precision value (0.78), reflecting the challenges in accurately distinguishing this class. TDMA attacks also exhibited reduced performance, with precision, Recall, and F1-Score ranging between 0.85 and 0.87. These lower values highlight the complexity of these attack patterns and their overlap with the normal traffic. The analysis emphasizes the need for further refinement of ML algorithms to improve detection accuracy for challenging attack types, such as Flooding and TDMA, ensuring comprehensive and reliable intrusion detection in WSNs.

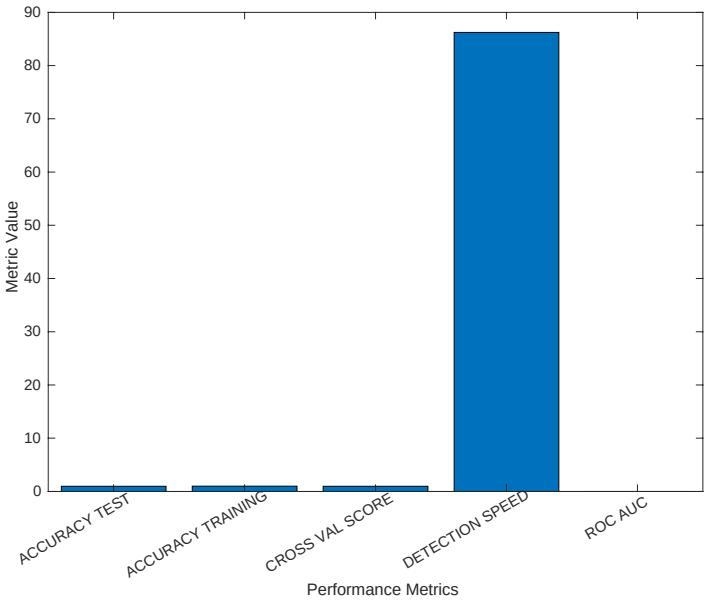

Fig. 7. Accuracy, cross-validation score and detection speed for KNN

Fig. 7 presents the performance metrics for the KNN algorithm, including Accuracy, Cross-Validation Score (CV-Score), ROC-AUC, and Detection Speed. Although KNN achieved near-perfect Accuracy and ROC-AUC values (both approximately 1.0 or 100%), its detection speed was significantly slower at 86.23 minutes. This highlights the algorithm's critical limitation when applied to real-time intrusion detection. The prolonged detection speed can be attributed to the KNN's reliance on storing the entire dataset in memory during training and performing distance calculations for each prediction at runtime. This computational overhead, while yielding high accuracy, makes KNN impractical for time-sensitive applications, such as Wireless Sensor Networks, where rapid intrusion detection is essential. These results underscore the need for algorithms that balance accuracy and computational efficiency.

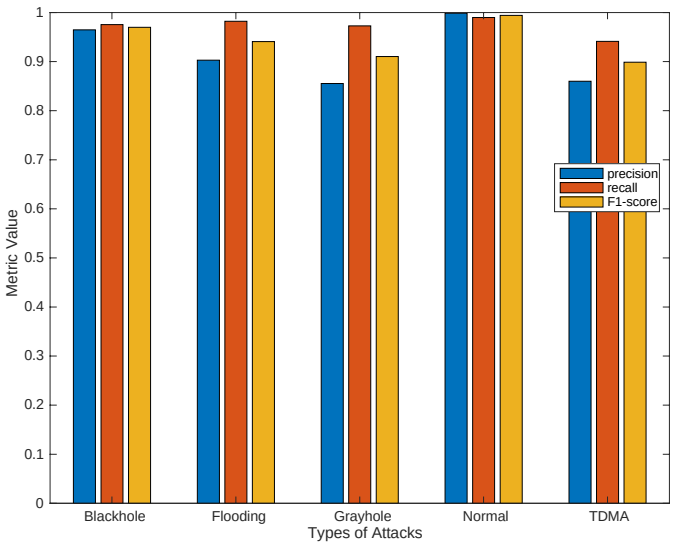

Fig. 8. Precision, Recall and F1-Score of individual attacks in DT

Fig. 8 illustrates the performance of the Decision Tree algorithm in detecting and classifying different attack types in the WSN-DS dataset, evaluated using the precision, Recall, and F1-Score. The algorithm achieved high performance for Normal, Blackhole, and Flooding attacks, with Precision, Recall, and F1-Score values consistently above 0.9, demonstrating its effectiveness in detecting them. However, the Decision Tree underperforms in classifying TDMA and Grayhole attacks. Grayhole attacks recorded the lowest precision at 0.86, indicating difficulty distinguishing this class. Similarly, TDMA attacks showed reduced Recall and F1-Score values of 0.84 and 0.85, respectively, highlighting challenges in identifying all instances of this attack type. These findings suggest that although the Decision Tree is effective for well-defined attack patterns, its performance diminishes for more complex or overlapping attack classes, emphasizing the need for additional optimization or hybrid approaches for comprehensive intrusion detection. *Fig. 9* demonstrates the performance of the Decision Tree algorithm, achieving 100% for Training Accuracy, ROC-AUC, and Cross-Validation Score, indicating its robustness and reliability in classification tasks. The algorithm also exhibited an excellent detection speed of less than 2 minutes, making it highly suitable for real-time intrusion detection in Wireless Sensor Networks.

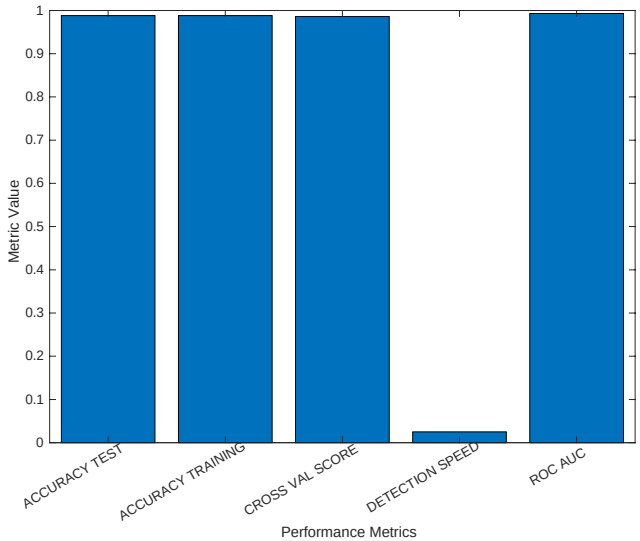

Fig. 9. Accuracy, ROC-AUC, Cross-validation score and Detection Speed for DT

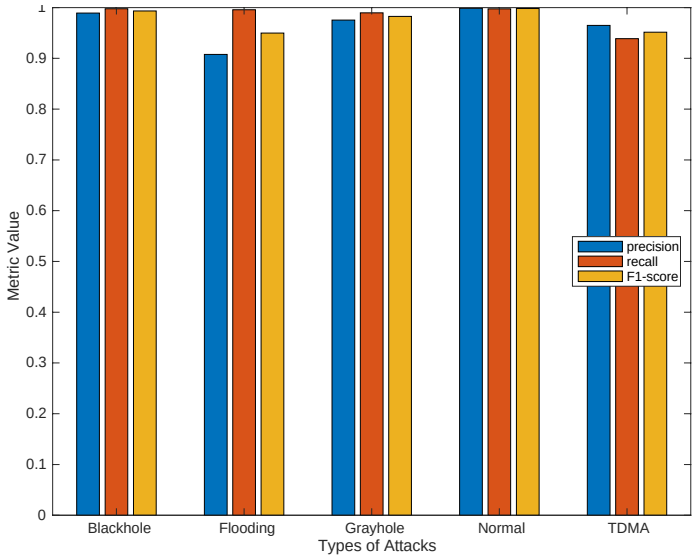

Fig. 10. Precision, Recall, and F1-score of the various attacks for Random Forest

Fig. 10 highlights the performance of the Random Forest algorithm in detecting and classifying various attacks in the WSN-DS dataset, evaluated using the precision, Recall, and F1-Score. The algorithm achieved high metrics (above 0.95) for Blackhole, Grayhole, Normal, and TDMA attacks, demonstrating robustness and adaptability. However, Flooding attacks present a challenge, with the lowest precision value of 0.87 compared to other attack types. This indicates that while Random Forest effectively classifies most attack types, its performance on Flooding attacks is slightly reduced, likely due to overlapping characteristics with benign traffic. These results underline the need for further optimization to enhance the detection precision of such complex attack patterns.

Fig. 11 illustrates the performance of the Random Forest algorithm across key metrics, including the Accuracy, Cross-Validation Score, ROC-AUC, and Detection Speed. The algorithm demonstrated consistent and robust results, achieving high accuracy for both the training and testing phases and a strong ROC-AUC value near 1.0, reflecting its reliability in classification tasks. Notably, Random Forest achieved a detection speed of less than 8 min, demonstrating its suitability for real-time intrusion detection in Wireless Sensor Networks while maintaining excellent classification performance. These results highlight its effectiveness in balancing computational efficiency and accuracy.

Fig. 12 illustrates the performance of the Naïve Bayes algorithm in detecting and classifying various attack types within the WSN-DS dataset, evaluated using precision, Recall, and F1-Score. The algorithm performed exceptionally well in detecting Normal, Flooding, and TDMA attacks, achieving metrics consistently above 0.9, indicating its effectiveness in handling these attack patterns. However, Naïve Bayes faced challenges with Blackhole and Grayhole attacks, recording lower Precision and F1-Score values of approximately 0.7 and 0.6, respectively. These results suggest that while Naïve Bayes is highly effective for certain attack types, its probabilistic assumptions limit its ability to accurately classify more complex or overlapping attack patterns such as blackholes and grayholes. Further optimization or hybrid approaches may be necessary to address these limitations.

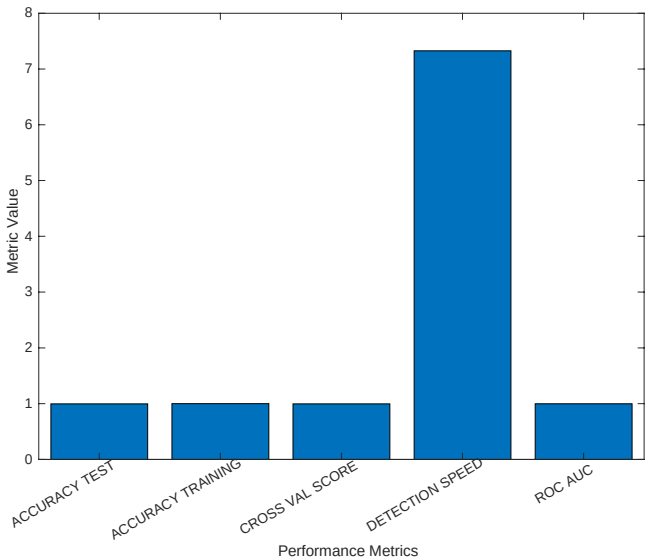

Fig. 11. Accuracy, cross-validation score and detection speed for Random Forest

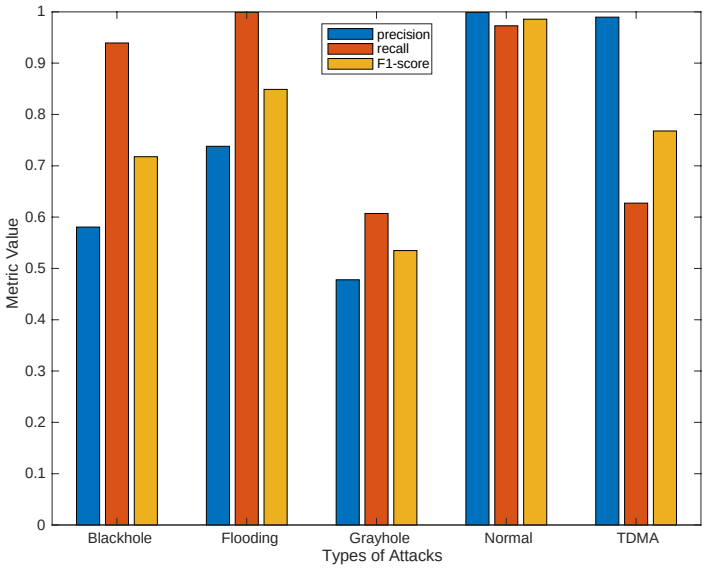

Fig. 12. Precision, Recall, and F1-score of the various attacks for Naïve Bayes

Fig. 13 shows the performance of the Naïve Bayes algorithm across key metrics, including Accuracy, ROC-AUC, Cross-Validation Score, and Detection Speed. The algorithm achieved high accuracy for the training and test datasets, with values exceeding 0.95 and a strong ROC-AUC score of nearly 1.0, indicating excellent classification reliability. Additionally, Naïve Bayes exhibited a notable detection speed of less than one minute, showing its computational efficiency and suitability for real-time intrusion detection in Wireless Sensor Networks. These results highlight Naïve Bayes as a lightweight yet effective algorithm for rapid and reliable classification tasks.

Fig. 14 presents the performance of the Stacking Classifier in detecting and classifying various attack types in the WSN-DS dataset evaluated through Precision, Recall, and F1-Score. The classifier achieved outstanding results for Normal, Flooding, and TDMA attacks, with all metrics exceeding 0.98, indicating its robustness and effectiveness for these attack types. However, the Stacking Classifier struggled with Blackhole and Grayhole attacks. Precision and Recall values for Grayhole were the lowest among all attack types, at approximately 0.87 and 0.89, respectively, while Blackhole attacks showed slightly reduced Recall at 0.91.

These results highlight the difficulty of the classifier in consistently detecting and classifying more complex or overlapping attack patterns. This performance emphasizes the need for further tuning or hybrid approaches to improve the Stacking Classifier's ability to handle nuanced attack behaviours while maintaining its high accuracy for simpler patterns.

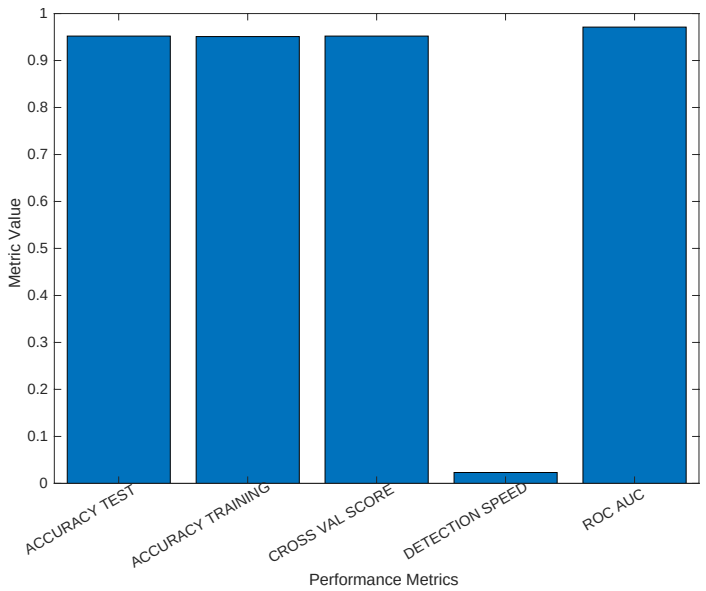

Fig. 13. Accuracy, cross-validation score and detection speed for Naïve Bayes

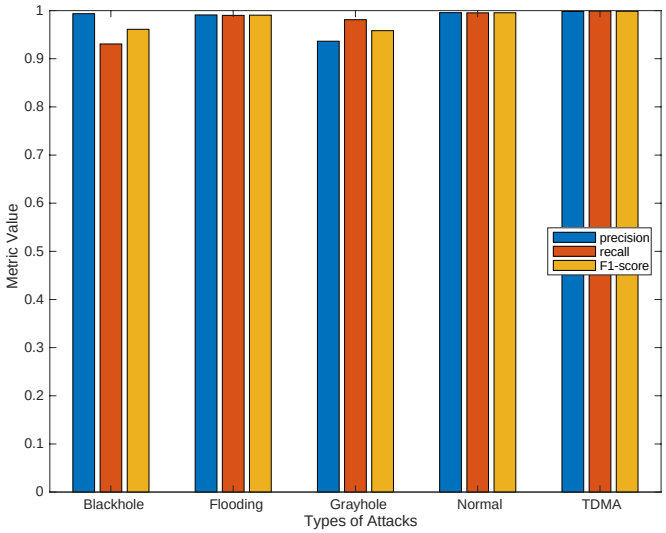

Fig. 14. Precision, Recall, and F1-score of the various attacks for Stacking

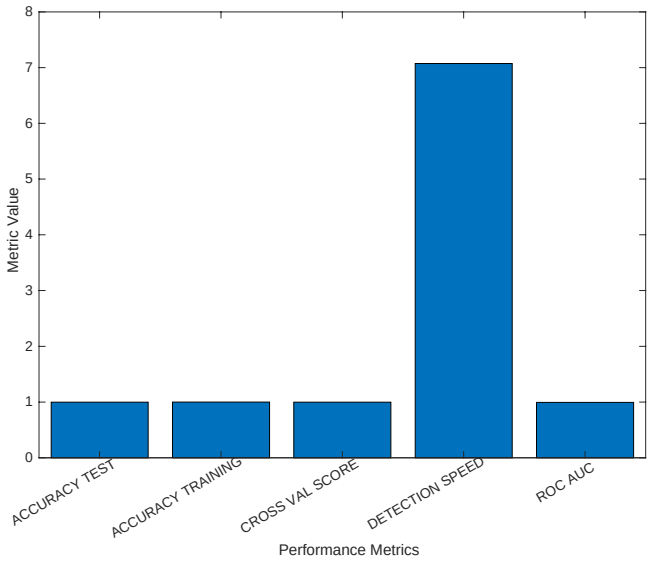

Fig. 15. Accuracy, cross-validation score and detection speed for Stacking

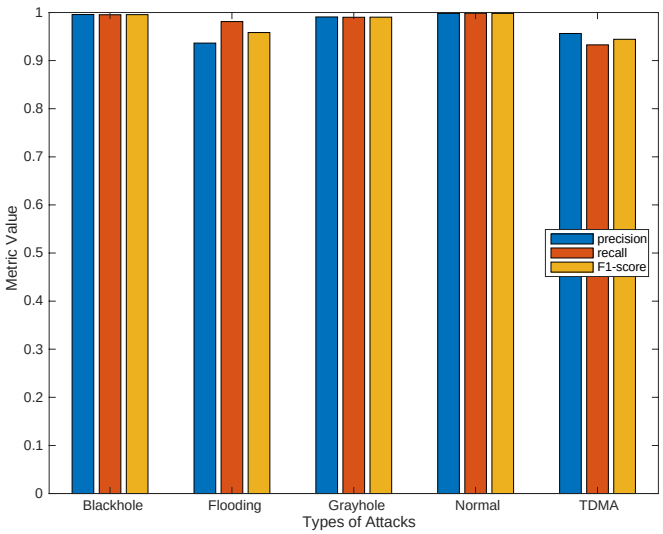

Fig. 16. Precision, Recall, and F1-score of the various attacks for AdaBoost

Fig. 15 illustrates the performance of the Stacking Classifier across key metrics, including the Accuracy, Cross-Validation Score, ROC-AUC, and Detection Speed. The classifier demonstrated excellent performance, achieving high accuracy values for training and testing and a robust ROC-AUC close to 1.0, reflecting its reliability in classification tasks. The detection speed of the Stacking Classifier was measured at seven minutes, demonstrating its efficiency in handling complex intrusion patterns in the WSN-DS dataset. Furthermore, as part of the stacking ensemble, AdaBoost contributes to accurately classifying various attack types, emphasizing the ensemble's overall strength in managing diverse and nuanced attack behaviours.

Fig. 16 illustrates the performance of the AdaBoost algorithm in detecting and classifying various attack types in the WSN-DS dataset, evaluated using the precision, Recall, and F1-Score. The algorithm demonstrated exceptional performance across all attack types, with metrics consistently exceeding 0.93. Even for TDMA attacks, which recorded the lowest recall value, the classification rate remained high, at 93%. These results underscore AdaBoost's ability to quickly and accurately identify diverse attack patterns, making it highly suitable for real-time intrusion detection in Wireless Sensor Networks. Its robust performance across all metrics highlights its adaptability to various attack complexities. The performance metrics for AdaBoost on the dataset are shown in Fig. 17, and they are almost equivalent to 1, or 100%, with a detection speed of 1.6 minutes.

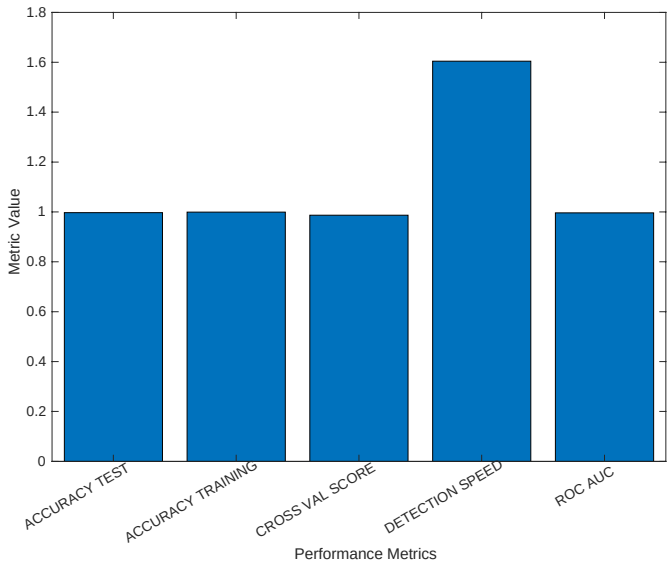

Fig. 17. Accuracy, cross-validation score and detection speed for AdaBoost

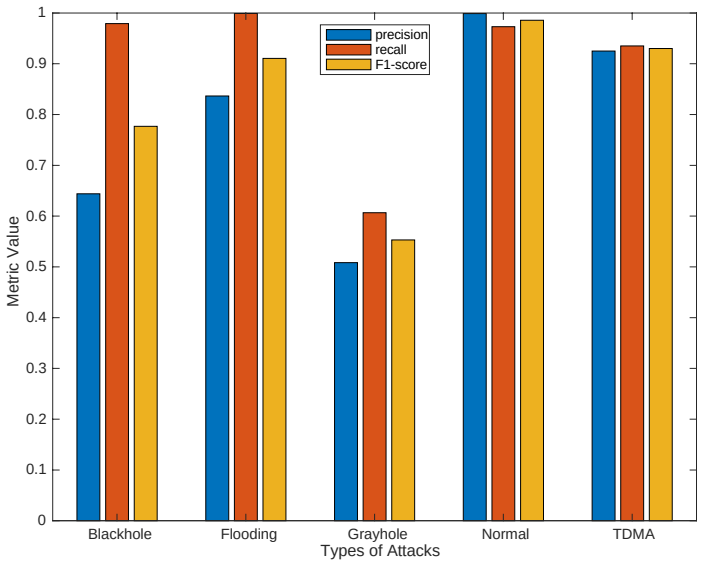

Fig. 18. Precision, Recall, and F1-score of the various attacks for Multi-layer Perceptron

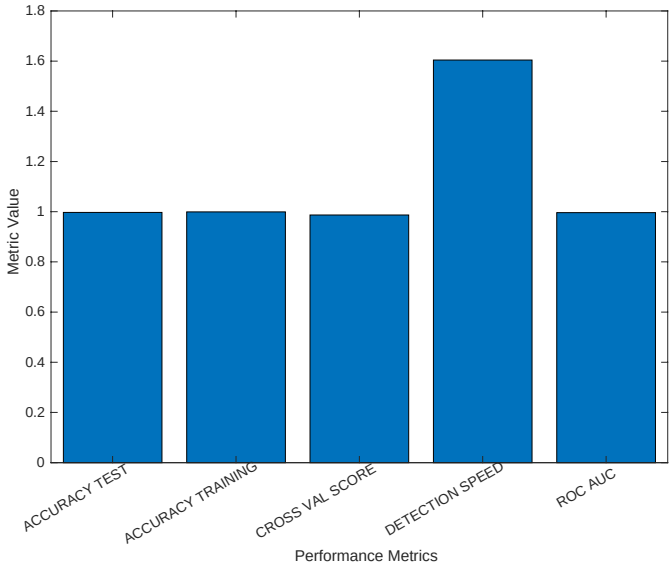

Fig. 19. Accuracy, cross-validation score and detection speed for Multi-layer Perceptron

Fig. 18 illustrates the performance of the Multi-layer Perceptron (MLP) algorithm in classifying various attack types within the WSN-DS dataset, evaluated through precision, Recall, and F1-Score. The algorithm performed exceptionally well detecting Normal, Flooding, TDMA, and Blackhole attacks, with metrics consistently greater than 0.90. However, MLP struggled to accurately classify grayhole attacks, achieving an accuracy value below 60%. This indicates a significant challenge in recognizing the nuanced patterns associated with grayhole attacks, suggesting additional tuning or hybrid approaches to improve the classification performance for this attack type.

Fig. 19 demonstrates the performance of the Multi-layer Perceptron (MLP) algorithm across key metrics, including the Accuracy, Cross-Validation Score, ROC-AUC, and Detection Speed. The algorithm performed fairly well, with Accuracy and ROC-AUC values exceeding 0.85, indicating reliable classification capabilities. The detection speed of approximately 1.3 minutes further highlights the computational efficiency of the MLP, making it suitable for near-real-time intrusion detection. These results suggest that while MLP performs well overall, additional optimizations may be required to enhance its performance for more complex attack scenarios in Wireless Sensor Networks.

5. Discussion

5.1 Model Performance and Generalization

The results demonstrate that the machine learning (ML) algorithms employed in this study achieved exceptional performance in both the training and testing phases, indicating their ability to generalize well to unseen data. Most models achieved training accuracies close to 100% with comparable test accuracies, indicating minimal overfitting. For instance, ensemble methods, such as random forest, AdaBoost, and stacking classifiers, consistently deliver near-perfect accuracy and cross-validation scores. These results align with [12] and [14], who reported the reliability of ensemble methods for intrusion detection in Wireless Sensor Networks (WSNs).

In contrast, K-Nearest Neighbors (KNN) exhibited unique performance characteristics, particularly with an average detection speed of 86.2 minutes. Although KNN achieved high accuracy metrics, its prolonged detection time stems from its computational intensity during runtime, a limitation observed in other studies, such as [13]. This tradeoff between accuracy and computational efficiency limits KNN's suitability of KNN for real-time applications. Although algorithms such as Naïve Bayes and Decision Tree demonstrated faster detection speeds (0.023 and 0.025 min, respectively), their reduced precision and F1-scores suggest susceptibility to false classifications. In contrast, AdaBoost achieved a near-ideal balance between speed (1.604 min) and accuracy (99.7%), maintaining high ROC-AUC and cross-validation scores. This equilibrium is particularly valuable in WSN contexts, where minimising false positives and ensuring consistent detection reliability are as critical as achieving a low latency. Therefore, the slightly higher computational cost of AdaBoost is justified by its superior detection stability and generalisation capability, making it more suitable for real-world WSN deployments.

5.2 Computational Efficiency and Detection Speed

Detection speed has emerged as a key performance metric for evaluating the feasibility of algorithms in real-time intrusion-detection scenarios. Among the models being assessed, AdaBoost achieved the fastest detection speed of 1.6 minutes, followed closely by the decision tree and Multi-layer Perceptron (MLP), making AdaBoost a strong candidate for time-sensitive environments. These findings are consistent with those of [18], who highlighted the efficiency and

accuracy of AdaBoost in resource-constrained environments. However, ensemble methods such as Random Forest and Stacking Classifiers recorded slower detection speeds of 7.3 and 7.1 minutes, respectively.

Although these algorithms excelled in Precision, Recall, and F1-Score metrics, their computational overhead underscores the need for optimization in latency-critical scenarios. This observation aligns with [17] and [30], who noted the tradeoffs between detection speed and classification performance in ensemble models.

5.3 Handling Imbalanced Data

The imbalanced class distribution within the WSN-DS dataset poses a significant challenge to model development, particularly for minority attack types. The application of Borderline SMOTE for oversampling minority classes improved classification performance for complex attack types such as Grayhole and Flooding. For example, ensemble methods such as AdaBoost and Random Forest achieved near-perfect F1-Scores of 1.0 for these attack types, confirming the utility of such techniques in addressing class imbalance. These findings align with observations by [17] and [31], who emphasized the importance of effective data pre-processing for handling class imbalance.

On the other hand, algorithms such as Naïve Bayes struggled to maintain high performance on imbalanced data, recording the lowest Precision and Recall values for grayhole attacks. This limitation reflects the algorithm's sensitivity to skewed class distributions, as [32] noted. The impact of Borderline SMOTE and undersampling was assessed by comparing the class-specific recall and F1-scores before and after resampling, revealing significant improvements for minority attack types. Nevertheless, the generation of synthetic data via SMOTE may pose a limited risk of overfitting, particularly when minority samples are scarce. This risk was mitigated through the implementation of 10-fold cross-validation and will be further examined in future studies using original, unbalanced datasets for performance validation.

5.4 Attack Type-Specific Performance

The algorithms exhibit varying levels of effectiveness in detecting specific attack types. Normal, Blackhole, and TDMA attacks were consistently classified with high-precision, Recall, and F1-Score values across most algorithms, demonstrating their robustness for these attack types.

However, Grayhole and Flooding attacks posed more significant challenges, particularly for Naïve Bayes and MLP, which recorded F1-Scores below 0.8 for these classes. These findings align with prior research by [25], highlighting the complexity of detecting subtle and overlapping attack patterns in WSN traffic.

5.5 Practical Implications and Recommendations

The results emphasize the importance of selecting appropriate algorithms for intrusion detection in WSNs and balancing the detection speed, accuracy, and computational efficiency. While ensemble methods such as random forest and stacking classifiers excel in classification performance, their relatively slower detection speeds may limit their applicability in real-time scenarios. In contrast, AdaBoost and Decision Tree demonstrated a more favourable balance between speed and accuracy, making them more suitable for real-time deployment.

Furthermore, addressing data imbalance through techniques such as Borderline SMOTE significantly enhances the detection of minority-class attacks such as Grayhole and Flooding. The findings of this study underscore the need for further exploration of hybrid and adaptive models that optimize the detection speed and accuracy across diverse attack scenarios, a recommendation echoed by [12] and [26]. By identifying efficient, lightweight, and accurate ML algorithms tailored to the constraints of WSNs, this study contributes to the advancement of intrusion detection technologies in these networks.

Future research should focus on optimizing the computational efficiency of ensemble methods and enhancing the detection of complex attack patterns through advanced data pre-processing and model-tuning techniques.

6. Limitations, Implications, and Future Directions

This section highlights the study's limitations, broader implications, and potential directions for future research to advance intrusion detection systems in Wireless Sensor Networks (WSNs).

6.1 Limitations

Although this study provides valuable insights into the comparative performance of machine learning algorithms for intrusion detection, it has certain limitations.

1. *Dataset Dependency:* The WSN-DS dataset used in this study is a simulated dataset that, while suitable for benchmarking, may not fully represent the diverse and dynamic nature of real-world WSNs. Real-world datasets can introduce additional variabilities and challenges.
2. *Static Network Assumptions:* The evaluation assumes a static network topology, neglecting the potential impact of node mobility, which is often present in real-world WSN applications. Node mobility can significantly influence detection performance and computational efficiency.

3. *Limited Attack Scope:* This study focuses on Denial of Service (DoS) attacks, including Blackhole, Grayhole, Flooding, and TDMA scheduling attacks. Other intrusion types, such as Sybil or routing attacks, were not considered, which could limit the generalizability of the findings.
4. *Algorithm Configurations:* While hyperparameter tuning was conducted, the configurations used for algorithms, such as Random Forest and Stacking, may not be optimal for every WSN environment, leaving room for further fine-tuning.
5. *Energy Consumption:* The study did not evaluate the energy consumption of the algorithms, which is a critical metric for WSNs because of their resource-constrained nature.

This study employed an NS-2–simulated WSN-DS dataset to ensure controlled experimentation and reproducibility. However, it does not encompass the complex environmental dynamics typical of real-world WSNs. Variables such as node mobility, link instability, interference, and environmental noise can significantly affect detection performance in actual deployments. The findings presented should be considered as baseline results obtained under controlled conditions. Future research will build upon this study by integrating real-world WSN datasets and conducting testbed experiments, thereby allowing for a more thorough assessment of the model's robustness, scalability, and adaptability within practical operational constraints. Although this study evaluated algorithmic efficiency by utilising detection time as a computational metric, it did not include a direct assessment of energy consumption, which is crucial for practical implementation in wireless sensor networks (WSN). Future research will incorporate energy usage profiling for each algorithm to elucidate the relationship between detection performance and energy cost, thereby facilitating more sustainable intrusion detection in resource-constrained networks.

6.2 Implications and Applications

The findings of this study have several implications for the design and implementation of intrusion-detection systems in WSNs.

1. *Algorithm Selection for Intrusion Detection:* The AdaBoost algorithm emerged as the best performer, with accuracy, precision, Recall, and F1-Scores close to 1 and a detection speed of 1.6 minutes. This makes it a strong candidate for real-time, resource-constrained environments.
2. *Balancing Accuracy and Efficiency:* While algorithms such as Random Forest and Stacking offer excellent accuracy and robustness, their longer detection times highlight the need for tradeoffs between detection speed and computational efficiency. This insight is critical for deploying intrusion detection systems in time-sensitive applications.
3. *Application in Critical WSN Domains:* The results can guide the deployment of machine learning-based intrusion detection systems in critical applications such as healthcare, environmental monitoring, and smart infrastructure, where the timely detection of intrusions is paramount.
4. *Data Imbalance Handling:* Using techniques such as Borderline SMOTE to address data imbalance demonstrated its effectiveness in enhancing the performance of algorithms such as Random Forest and ANN. This reinforces the importance of data pre-processing in intrusion detection systems.

6.3 Future Directions

Future research should build on the findings of this study to address its limitations and explore additional avenues for improving intrusion detection in WSNs.

1. Real-World Dataset Utilisation and Validation

Future research should investigate real-world Wireless Sensor Network (WSN) datasets that encompass various intrusion types and environmental conditions to validate the models developed using the NS-2 simulated dataset. A pertinent research question is as follows: To what extent do machine learning-based intrusion detection algorithms generalise when applied to non-simulated, real-world WSN environments? Methodologically, this could involve benchmarking existing models against field-collected data from deployed sensor networks in smart agriculture, the industrial Internet of Things (IoT), or environmental monitoring systems.

2. Dynamic and Scalable Network Topologies

Although the present study utilised static simulations, future research should explore the algorithm’s performance in dynamic network topologies characterised by node mobility and fluctuating communication links. This line of enquiry can address questions such as the following: How does node mobility influence the detection latency and accuracy of IDS models in dense WSN environments? Conducting scalability experiments through expanded NS-3 or hybrid emulation frameworks will be essential to evaluate the robustness of the algorithm under conditions of high-density traffic and evolving attack complexities.

3. Energy-Efficient Algorithm Design

Given the resource constraints inherent in Wireless Sensor Networks (WSNs), future research should incorporate energy consumption profiling into the evaluation of intrusion detection system (IDS) performance. A pertinent question in this context is as follows: What is the trade-off between detection accuracy and energy efficiency among lightweight and ensemble-based algorithms? Methodologies such as energy-aware optimisation and hardware-level measurements can offer valuable insights into the sustainability of models for long-term deployments.

4. Integration of Advanced and Evolving Attacks

Future research should incorporate advanced attack types, such as wormhole, Sybil, sinkhole, and routing attacks, to assess the adaptability of the models. This would address the question of whether hybrid machine-learning-based intrusion detection system (IDS) frameworks can maintain accuracy when confronted with multi-vector and evolving adversarial threats. The methodology may involve adversarial learning techniques and dynamic attack simulations within wireless sensor network (WSN) emulators.

5. Hybrid and Adaptive Algorithm Development

Integrating the strengths of algorithms, such as the adaptability of AdaBoost, robustness of Random Forest, and nonlinear learning capabilities of Artificial Neural Networks (ANN), can result in hybrid Intrusion Detection System (IDS) models that effectively balance speed, accuracy, and resource consumption. Future research should investigate adaptive meta-learning approaches that dynamically adjust model weights in response to network behaviour.

6. Real-Time Testbed Implementation and Validation

Validating the proposed algorithms on real-world Wireless Sensor Network (WSN) testbeds or Internet of Things (IoT) edge platforms, such as Raspberry Pi clusters or Arduino-based nodes, is essential to corroborate the simulation outcomes. This study explores the impact of communication delays, packet loss, and hardware constraints on the performance of an Intrusion Detection System (IDS) in live deployments. This step is crucial for bridging the gap between the controlled simulation results and the actual operational performance.

7. Context-Aware and Self-Optimising IDS Design

The forthcoming generation of Intrusion Detection System (IDS) models should integrate context awareness, enabling dynamic adjustments to detection thresholds and algorithmic configurations in response to the network state, energy levels, or traffic intensity. Future research should investigate whether context-driven adaptation can enhance the longevity and resilience of IDS in heterogeneous Wireless Sensor Network (WSN) environments. These systems may utilise reinforcement learning or rule-based adaptive mechanisms for intelligent intrusion mitigation.

7. Conclusion

This study evaluated the performance of multiple machine learning algorithms for intrusion detection in Wireless Sensor Networks (WSNs), utilizing the Network Simulator-2 (NS-2) simulated WSN-DS dataset containing Denial of Service (DoS) attacks. The algorithms investigated included K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Naive Bayes, AdaBoost, Stacking, and Multi-layer Perceptron (MLP). The results highlight AdaBoost as the most efficient algorithm, achieving near-perfect metrics for accuracy, precision, Recall, and F1-score, with an exceptional detection speed of 1.6 minutes. AdaBoost is a strong candidate for real-time intrusion detection in resource-constrained WSN environments. Stacking and Random Forest demonstrated robust performance with similarly high accuracy and reliability but required marginally longer detection times of 7.1 and 7.3 minutes, respectively. These results underscore the tradeoff between the detection speed and the computational demands of ensemble models.

On the other hand, despite its classification accuracy, KNN exhibited significant inefficiency with an average detection speed of 86.2 minutes, rendering it impractical for real-time applications in WSNs. This study provides valuable insights into selecting machine learning algorithms for intrusion detection in WSNs, emphasizing the importance of balancing detection accuracy with computational efficiency. The findings suggest that while algorithms such as AdaBoost and Stacking are suitable for deployment in real-world scenarios, further optimization of resource-intensive models, such as KNN, could enhance their applicability. Future research should focus on developing lightweight optimization techniques and exploring hybrid models to improve intrusion detection systems' robustness, speed, and energy efficiency in WSNs, ensuring enhanced network security and resilience against evolving cyber threats.

References

[1] K. S. Adu-Manu, 'A Study into Lifetime Maximization of Wireless Sensor Networks for Water Quality Monitoring', University Of Ghana, 2019.

[2] K. S. Adu-Manu, F. A. Katsriku, J.-D. Abdulai, and F. Engmann, 'Smart River Monitoring Using Wireless Sensor Networks', *Wirel Commun Mob Comput*, vol. 2020, 2020, doi: 10.1155/2020/8897126.

[3] K. S. Adu-manu *et al.*, 'Review Article WSN Architectures for Environmental Monitoring Applications', *J Sens*, vol. 2022, 2022.

[4] K. S. Adu-manu, F. Engmann, G. Sarfo-kantanka, G. E. Baiden, and B. A. Dulemordzi, 'Review Article WSN Protocols and Security Challenges for Environmental Monitoring Applications : A Survey', *J Sens*, vol. 2022, 2022.

[5] Z. Liu, G. Mohiuddin, J. Zheng, M. Asim, and S. Wang, 'Intrusion detection in wireless sensor network using enhanced empirical based component analysis', *Future Gener. Comput. Syst.*, vol. 135, pp. 181–193, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:250534078

[6] D. Manivannan, 'Recent endeavors in machine learning-powered intrusion detection systems for the Internet of Things', *J. Netw. Comput. Appl.*, vol. 229, p. 103925, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:270662368

[7] H. Kheddar, Y. Himeur, and A. I. Awad, 'Deep transfer learning for intrusion detection in industrial control networks: A comprehensive review', *J. Netw. Comput. Appl.*, vol. 220, p. 103760, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:258291793

[8] H. Yao, Y. Yang, X. Fu, and C. Mi, 'An Adaptive Sliding-Window Strategy for Outlier Detection in Wireless Sensor Networks for Smart Port Construction', *J Coast Res*, vol. 82, no. 82, pp. 245–253, 2018, doi: 10.2112/SI82-036.1.

[9] M. S. Alsahli, M. M. Almasri, M. Al-Akhras, A. I. Al-Issa, and M. Alawairdhi, 'Evaluation of Machine Learning Algorithms for Intrusion Detection System in WSN', *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 617–626, 2021, doi: 10.14569/IJACSA.2021.0120574.

[10] T. T. Lai, T. P. Tran, J. Cho, and M.-S. Yoo, 'DoS attack detection using online learning techniques in wireless sensor networks', *Alexandria Engineering Journal*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:265431580

[11] R. Ahmad, R. Wazirali, and T. Abu-Ain, 'Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues', *Sensors*, vol. 22, no. 13, p. 4730, 2022, doi: 10.3390/s22134730.

[12] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, 'Network intrusion detection system: A systematic study of machine learning and deep learning approaches', *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[13] S. Otoum, B. Kantarci, and H. T. Mouftah, 'A Novel Ensemble Method for Advanced Intrusion Detection in Wireless Sensor Networks', *IEEE International Conference on Communications*, vol. 2020-June, 2020, doi: 10.1109/ICC40277.2020.9149413.

[14] H. N. M. Akpinar, S. Duran, R. I. Eser, and S. Dogru, 'Detection of DoS Attacks in WSNs by using Machine Learning Models', *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–8, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:273377458

[15] Z. Ahmad, A. S. Khan, W. S. Cheah, J. bin Abdullah, and F. Ahmad, 'Network intrusion detection system: A systematic study of machine learning and deep learning approaches', *Transactions on Emerging Telecommunications Technologies*, vol. 32, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:225153435

[16] S. Otoum, B. Kantarci, and H. T. Mouftah, 'A Novel Ensemble Method for Advanced Intrusion Detection in Wireless Sensor Networks', *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:220890659

[17] H. M. Saleh, H. Marouane, and A. Fakhfakh, 'Improves Intrusion Detection Performance In Wireless Sensor Networks Through Machine Learning, Enhanced By An Accelerated Deep Learning Model With Advanced Feature Selection', *Iraqi Journal For Computer Science and Mathematics*, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:272773125

[18] S. Swami, P. Singh, and S. S. Chauhan, 'Design and Analysis of an Integrated Rule-Based and Machine Learning System for DoS Attack Detection in Wireless Sensor Networks', *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, pp. 1046–1052, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:271406287

[19] I. M. Almomani, B. Al Kasasbeh, and M. T. Al-Akhras, 'WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks', *J. Sensors*, vol. 2016, pp. 4731953:1-4731953:16, 2016, [Online]. Available: https://api.semanticscholar.org/CorpusID:39586351

[20] N. M. Alruhaily and M. L. Dina, 'A Multi-layer Machine Learning-based Intrusion Detection System for Wireless Sensor Networks', *International Journal of Advanced Computer Science and Applications*, vol. 12, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:235252595

[21] H. Liu and B. Lang, 'Machine learning and deep learning methods for intrusion detection systems: A survey', *Applied Sciences (Switzerland)*, vol. 9, no. 20, 2019, doi: 10.3390/app9204396.

[22] C. Iwendi, J. H. Anajemba, C. N. Biamba, and D. Ngabo, 'Security of Things Intrusion Detection System for Smart Healthcare', *Electronics (Basel)*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:236223965

[23] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, 'WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks', *J Sens*, vol. 2016, 2016, doi: 10.1155/2016/4731953.

[24] N. M. Alruhaily and D. M. Ibrahim, 'A Multi-layer Machine Learning-based Intrusion Detection System for Wireless Sensor Networks', *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, pp. 281–288, 2021, doi: 10.14569/IJACSA.2021.0120437.

[25] T. Kim, L. F. Vecchietti, K. Choi, S. Lee, and D. Har, 'Machine Learning for Advanced Wireless Sensor Networks: A Review', *IEEE Sens J*, vol. 21, no. 11, pp. 12379–12397, 2021, doi: 10.1109/JSEN.2020.3035846.

[26] A. Abdollahi, K. Rejeb, A. Rejeb, M. M. Mostafa, and S. Zailani, 'Wireless sensor networks in agriculture: Insights from bibliometric analysis', *Sustainability (Switzerland)*, vol. 13, no. 21, 2021, doi: 10.3390/su132112011.

[27] S. S. Band *et al*., 'When Smart Cities Get Smarter via Machine Learning: An In-Depth Literature Review', *IEEE Access*, vol. 10, pp. 60985–61015, 2022, doi: 10.1109/ACCESS.2022.3181718.
[28] M. Pundir and J. K. Sandhu, 'A Systematic Review of Quality of Service in Wireless Sensor Networks using Machine Learning: Recent Trend and Future Vision', *Journal of Network and Computer Applications*, vol. 188, no. August 2020, p. 103084, 2021, doi: 10.1016/j.jnca.2021.103084.
[29] K. A. Alaghbari, M. H. M. Saad, A. Hussain, and M. R. Alam, 'Complex event processing for physical and cyber security in datacentres - recent progress, challenges and recommendations', *Journal of Cloud Computing*, vol. 11, no. 1, 2022, doi: 10.1186/s13677-022-00338-x.
[30] K. Sedhuramalingam and N. Saravanakumar, 'A novel optimal deep learning approach for designing intrusion detection system in wireless sensor networks', *Egyptian Informatics Journal*, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:272810002
[31] E. El Ahmar, A. Rachini, and H. Attar, 'Cybersecurity Enhancement in IoT Wireless Sensor Networks using Machine Learning', *WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS*, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:273383622
[32] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, 'Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review', *Procedia Comput Sci*, vol. 171, no. 2019, pp. 1251–1260, 2020, doi: 10.1016/j.procs.2020.04.133.

Authors' Profiles

Samuel Mends: Holds MSc in Computer Science and is an experienced IT professional and co-founder of SmS Inc, where he has been driving innovation and systems integration for over 13 years. He holds a BSc in Computer Science from Valley View University and is a certified Cisco Network Associate from the Kofi Annan Centre of Excellence in ICT (AITI-KACE). Samuel has served in key technical roles, including as a Systems Engineer at the Ministry of Finance and during his national service at the Ministry of Defence**.** He brings strong expertise in enterprise routing and switching configuration, with a proven track record in deploying scalable network solutions across both public and private sectors.

Kofi Sarpong Adu-Manu: Associate Professor of Wireless Communication Networks at the University of Ghana. With over 50 peer-reviewed publications and 1,500+ citations, his research spans Wireless Sensor Networks, Artificial Intelligence, Cybersecurity, IoT, Emerging Technologies, Computer Science Education, and EdTech. He consults for World Bank, UNICEF, UNESCO, and the Commonwealth of Learning on educational technology. He served as lead judge for the National Stemnnovation Contest and supports tech-based learning through CENDLOS under Ghana's Ministry of Education. At the University of Ghana, he serves on multiple committees and is a Programme Assessor for Ghana Tertiary Education Commission. He is also a member of the Internet Society Ghana Chapter, and the Chartered Institute of Computing and Information Technology. Prof. Adu-Manu founded *Ladies in Tech Ghana* to empower girls in tech fields at secondary and tertiary levels. A mentor, leader, and tech enthusiast, he is passionate about inclusive digital transformation.

How to cite this paper: Samuel Mends, Kofi Sarpong Adu-Manu, "Evaluating Machine Learning Efficacy for DoS Intrusion Detection in Wireless Sensor Networks", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.1, pp. 1-23, 2026. DOI:10.5815/ijwmt.2026.01.01