Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Optimization-Based Experimental Evaluation of Machine and Deep Learning Models for DoS Detection in Wireless Sensor Networks

Hadi Kadhim Karhoot [1*]

[1*] Hakim Sabzevari University, Computer Network Engineering, Mashhad, Iran. Postal Code: 1333591886
E-mail: hadikarhhot@gmail.com
*Corresponding author E-mail: hadikarhhot@gmail.com

Article's Information

Received: 14.01.2026
Accepted: 08.03.2026
Published: 31.03.2026

Keywords:
Wireless Sensor Networks
WSN, Internet of Things
IoT; Denial of Service
DoS attack
Intrusion Detection.

Abstract

WSNs are currently actively implemented in Internet of Things (IoT) settings where they are extremely susceptible to Denial-of-Service (DoS) attacks because of their weak computational and energy capabilities. This paper will provide experimental analysis of the machine learning and deep learning models to detect multi-class DoS attacks using the WSN-DS dataset based on an optimization approach. The study will be aimed at conducting systematic studies on the impact of feature selection, data balancing, and hyperparameter optimization on the detection performance under the resource constraints of the WSN settings. The approach combines a dimensionality reduction technique (PCA), data balancing technique (SMOTE), and hyperparameter optimization technique (Ant Colony Optimization (ACO)). There are several classification models which are experimentally tested both prior to and following optimization to determine any enhancement in ability to detect and classification balance. Findings indicate that optimization can greatly lead to model stability, misclassification reduction, and generalization improvement across types of attacks. The results offer useful information on the choice and fine-tuning of the learning models to achieve effective and dependable intrusion detection in WSN-based IoT system.

https://doi.org/10.46649/fjiece.v5.1.17a.31.3.2026

*Corresponding author: hadikarhhot@gmail.com

1. INTRODUCTION

WSNs have risen to be one of the much-needed technologies in a variety of applications, such as environmental monitoring, industrial automation, healthcare, and military surveillance. SNs are sensor nodes distributed over an area and powered by batteries that communicate with one another wirelessly in order to gather and relay information. This is especially useful in places that wired networks cannot be used, or are otherwise highly expensive, like remote areas, hostile environments, or large-scale monitoring systems. In spite of such merits, WSNs are very susceptible to security attacks, DoS being one of the most devastating [1].

In DoS attack, the adversaries strive to incapacitate the regular flow by jamming, using the weaknesses of the protocol, or by consuming resources. Such attacks may cause network degrading, loss of data, service disruption and in worst conditions, the total network crash. The above has very severe ramifications when

WSNs are being used in mission critical applications where the flow of data and/or system availability is a priority [2].

SNs unique features- scarcity of processing power, memory and energy- cannot rely on traditional security measures like rule-based, anomaly-based or statistical intrusion detection to detect advanced and evolving DoS attacks. Rule-based systems are unable to identify unknown attack signatures, anomaly detection methods often have very high false positive rates and statistical methods cannot be updated to the latest methods of attack [3].

There has been recent interest in machine learning (ML) and deep learning (DL) as a way to address the drawbacks. Machine Learning (ML) algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Decision Trees (DT), and Random Forest (RF), have the capacity to classify normal and malicious behaviors based on learning the labels on network traffic. Convolutional Neural Networks (CNNs) in deep learning have demonstrated ability to automatically learn complicated hierarchical features that can be used to detect sophisticated attack types with high precision. Nonetheless, there are some challenges to the implementation of DL models in WSNs, which include computational and energy limitations, scalability, and their generalization to other attacks and network topologies [4].

To overcome these difficulties, optimization practices, like the Ant Colony Optimization (ACO), can be used along with DL models to minimize the computational expense and energy consumption needed without impairing detection performance. The emerging research has also attempted to use lightweight DL architecture to achieve a trade-off between efficiency and accuracy, thus fitting it in resource-constrained WSNs [5].

This paper intends to establish an enhanced deep learning-based Intrusion Detection System (IDS), which is able to recognize various types of DoS attacks within a WSN. As a solution, the proposed system will utilize CNNs to recognize complex attack patterns, adopt ACO to find the hyperparameters, and utilize dimensionality reduction techniques, including Principal Component Analysis (PCA) to perform feature selection. The system is tested on WSN-DS dataset, it proves to be scalable, accurate and efficient and thus viable in real-world resource-constrained WSN deployment. The merits of this research work are the creation of high performance and energy-efficient IDS, the aspect of multi-class classification of various DoS attacks and the performance of a system in large-scale WSN scenarios

2. RELATED WORKS

WMNs have several security issues because they have different characteristics. Network Topology constantly changes, nodes can be added/subtracted or moving, making the process of maintaining secure connection and avoiding unauthorized access much harder. Moreover, the wireless communication channel is shared, thereby increasing the threat of eavesdropping, node impersonation, and packets interception. Such weaknesses cause WMNs to be prime targets of malicious attacks, including Distributed Denial of Service (DDoS) attacks.

Ramesh et al. [6] can propose a better DNN-based framework of defending the Wireless Multimedia Sensor Networks (WMSNs) against the DoS attack. The system is optimized using adaptive PSO scheme and the performance of the system is checked based on energy consumption, throughput, packet delivery, latency and the network lifetime. As the results have indicated, there is an effective mechanism to defend against DoS attack in the WMSNs though this mechanism is susceptible to massive scale attacks or complicated network topologies. The current work is able to address these shortcomings by using more sophisticated optimization routines and design the system to be more resistant to large-scale distributed attacks.

Liu et al. [7] suggest a new intelligent IDS method for WSNs through the integration of the k-NN algorithm and the AOA evolutionary computation method. The hybridization offers a DoS detection system with enhanced accuracy of models using a parallel communication model and the Lévy flight optimization method. The analysis exhibits significant enhancement in detection performance and classification accuracy. However, the method can be computationally expensive for large sensor networks. The present study mitigates this limitation by developing the algorithm with low computational complexity at high detection rates.

Ebremariam et al. [8] design a system for multiple WSN attack detection, training and testing the designed system using four datasets. The system can detect ten types of attacks, including DoS attacks. The results yield an excellent accuracy of 99.65% when tested with the WNS-DS dataset. The limitation of this approach is the high computational cost of neural networks, which can affect the Quality of Service (QoS) in constrained networks. This study reverses this by exploring less computationally intensive models that are able to achieve high detection rates with low requirements.

Osanaiye et al. [9] propose a statistical method to detect DoS-jamming attacks in WSNs. The method effectively detects jamming attacks interfering with WSN operations by generating radio frequency waves conflicting with legal communications. The results indicate that the method works adequately in real-time scenarios. The weakness, however, is that the method is not tried out on several datasets, which might affect its generalizability. This weakness is circumvented in the present research, where the method is tried out on a variety of datasets to determine its generalizability.

Le et al. [10] propose an RF classifier for identifying DoS attacks in WSNs to classify four attack types from the WSN-DS dataset. The accuracy of the results is better than the previously introduced ANN model. However, the study is based on a small-sized dataset with only 94,042 instances, which might limit the generalizability of the result. The present study overcomes the above limitation by employing bigger and diverse datasets to make the model more robust and scalable.

Almomani and Alenezi [11] provide evidence that RF is better than ANN in detecting DoS attacks based on heterogeneous classification models like SVM, NB, DT, RF, J48, ANN, and k-NN. WSN-DS was employed for experimentation but only a portion of the features. The results show that RF is computationally accurate but is limited by its computationally expensive nature, which makes it unviable for use in resource-constrained devices. This limitation is overcome by this work through optimizing the RF algorithm to reduce computational expenses without compromising its efficiency and accuracy.

Alsulaiman and Al-Ahmadi [12] compare and contrast different machine learning algorithms for the detection of DoS attacks using the WSN-DS dataset. The findings indicate that RF has the highest accuracy rate, which is equal to 99.72%. However, the J48 algorithm has similar performance, but takes much less time to process the data. This shows that J48 is more computationally efficient compared to the RF algorithm in resource-limited situations, such as in Wireless Sensor Networks (WSNs). This current work tries to find a solution to this problem by analyzing hybrid models that strike a balance between the detection accuracy and computational efficiency, by harnessing the convergence of the strengths of RF and J48.

Vinayakumar et al. [13] examine old machine learning methods on various datasets, including WSN-DS. Their findings indicate that DT, RF, and AB excel in binary classification, exhibit generalization capabilities across datasets, and perform well in multi-class classification, with DT and RF being particularly well suited for multi-class classification tasks. However, the study does not take into account the computational expense of these techniques. The present work addresses this issue by evaluating the CPU load and energy consumption of these classifiers to determine if they are applicable in resource-constrained WSNs.

3. METHODOLOGY

The methodology proposed here, shown in Figure (1), is aimed at optimizing machine learning (ML) and deep learning (DL) models on the dataset of the Wireless Sensor Network (WSN-DS). The steps are dataset preprocessing, where input features and the target variable are defined, label encoding is applied, Principal Component Analysis (PCA) is carried out to select features, data is balanced using SMOTE to

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

improve the dataset, and the data is partitioned into 80/20 training and testing sets. After preprocessing, several ML models are chosen, including Logistic Regression, Random Forest, Gradient Boosting, Decision Trees, Extra Trees, Naive Bayes, K-Nearest Neighbors, AdaBoost, LightGBM, XGBoost and CatBoost. Hyperparameters of the models- number of trees, depth of trees, and learning rate- are optimized to increase performance using Ant Colony Optimization (ACO).

Moreover, detailed models of neural networks, including Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) are used to learn complicated patterns in network traffic. CO is also used to optimize important DL hyper parameters, such as epochs and batch size. Ultimately, ML and DL models are trained and tested on commonly used performance measures, such as accuracy, precision, recall, F1-score, AUC, specificity and Matthews Correlation Coefficient (MCC). This systematic framework guarantees the construction of powerful and effective models that can process and analyze complete data of WSN-DS.

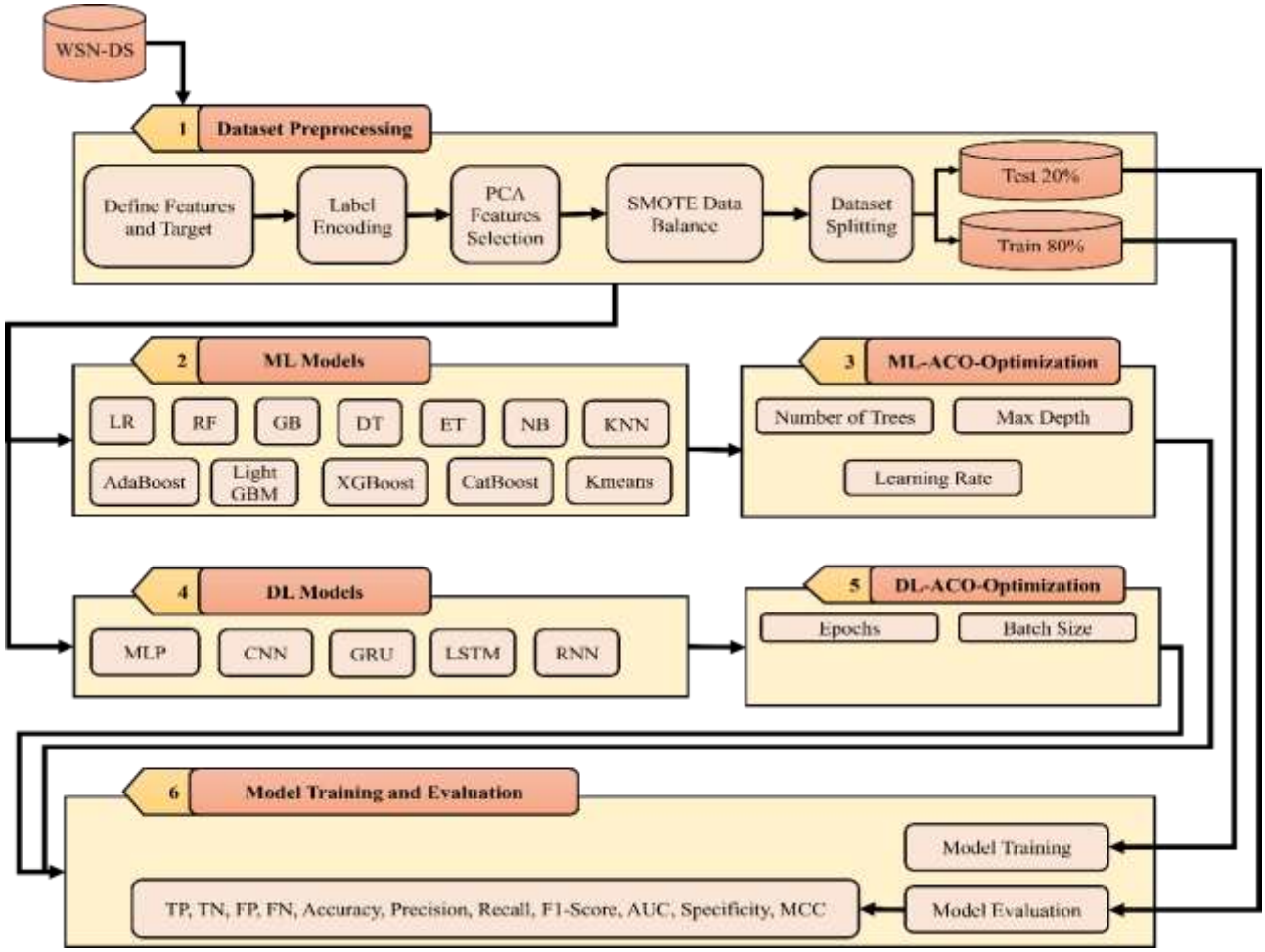


Fig 1. Methodology Block Diagram

3.1 SN-DS DATA SET DESCRIPTION

Wireless Sensor Network Dataset (WSN-DS) is intended to support intrusion detection in Wireless Sensor Networks (WSN) in the framework of the Internet of Things (IoT). It offers important features to observe and examine behaviour of nodes during multiple rounds of simulation. The distinctiveness of each record is captured by a Node ID, which makes it possible to follow nodes during specific rounds and stages.

For example, node 25 in round three and stage one can be represented as 001 003 025. The dataset the researchers used is time-sensitive where a Time attribute shows simulation time per node.

Is_CH flag portrays whether the node is a CH or otherwise, with 1 meaning that a node is a CH and 0, an ordinary node. The who_CH attribute indicates the Cluster Head which is coordinating the round in question, and Dist_To_CH is the distance between a node and a CH, and indicates the efficiency of communication as well as energy consumption. The ADV_S, ADV_R (advertisements sent and received accordingly), JOIN_S, and JOIN_R (join requests and responses accordingly) are the fields reflecting the CH-node interactions. To take note of clustering performance and communication patterns, these features are important.

Finally, SCHS captures the number of Time Division Multiple Access (TDMA) messages that the CH plans to control the time slots to avoid conflicts. Overall, the WSN-DS dataset can be described as a well-structured and detailed dataset that can aid in the research activities regarding intrusion detection, network optimization, and performance evaluation of IoT-based WSNs. The design enables studies that can effectively analyze the function of nodes, the energy use, and the communication of large scale wireless sensor networks.

3.2 DATASET PREPROCESSING

Preprocessing and ultimately preparation of data before training a model is an essential part of machine learning, with the purpose of cleaning, manipulating, and organizing data. Good preprocessing enhances the model accuracy, performance and removes overfitting or bias in the model. The main procedures involve definition of features and target, encoding categorical variables, dimensionality reduction, data balancing, and separation into a training and a testing set.

1. Target and Features: Target can be referred to as the output to be predicted and features are the variables used as the input. As an example of features and the target, consider an intrusion detection context where sensor readings are considered as features and the target is 0/1 normal or intrusive. Clearly defined knowledge will help the model to learn significant connections.
2. Label Encoding: Categorical form data is encoded as numbers to be supported by models. As an example, a node status that can be defined as Active and Inactive can be mapped to 1 and 0, respectively. When data are encoded properly, data integrity is maintained, particularly in the case of ordinal variables.
3. CA Feature Selection: Principal Component Analysis (PCA) is an algorithm used to reduce dimensionality, having the most significant features that explain maximum variance and then eliminating the remaining irrelevant features. This enhances the efficiency of computations and stops the problem of overfitting.
4. SMOTE Data Balancing: Imbalanced dataset are balanced using SMOTE, where a synthetic sample is generated to represent the minority classes so that the model learns patterns of all classes and at the same time minimize the bias.
5. Dataset Splitting: Training and testing sets are separated: 80 and 20 percent of data respectively. This enables the model to study the majority of the data but test its generalization on new samples.

3.3 MACHINE LEARNING MODELS

Detection of Denial-of-Service (DoS) attacks in Wireless Sensor Networks (WSNs) is required to provide secure and reliable communication. Machine learning (ML) models detect network traffic as legitimate or malicious based on historical data. ML models vary in complexity, explainability, and suitability to handle different types of WSN traffic.

1. Logistic Regression is a linear classifier that is simple and predicts the probability an instance belongs to a class. It is fast, comprehensible, and sufficient if the difference between normal traffic and attack traffic is linear. However, it struggles with complex, non-linear attack patterns.
2. Decision Trees use data splits based on the importance of features, and the process is repeated until criteria is met. They can quickly be trained and easily explain traffic features such as the length of

packets and the time they are sent. However, as trees become too deep, they can become prone to overfitting. Random Forest solves overfitting by using multiple decision trees and improves accuracy as well as generalization to numerical and categorical variables.

3. With Gradient Boosting, trees are built one at a time, and each new tree improves on the mistakes of the previous ones. This method can identify complex and advanced attack patterns, but careful hyperparameter tuning is required. For imbalanced datasets, AdaBoost is ideal as it better classifies previously misclassified points and is ideal for detecting subtle attacks.
4. K-Nearest Neighbors (KNN) is a classifier that assigns a class to a new instance, based on the majority class of the neighboring instances. If attack patterns are spatial, KNN classifies well, but suffers from high computation time on large data.
5. Naive Bayes relies on the Bayes theorem but assumes each feature is independent from the others. Therefore, Naive Bayes is computationally and data efficient, but can suffer from a loss of accuracy in datasets with highly correlated features.
6. Extra Trees improves on randomness in decision trees and are faster to train, generalize better, and can be applied to large and complex datasets.
7. Real-time detection of dynamically changing DoS attacks can be performed by using the speed-tuned, high-accuracy gradient boosting libraries, LightGBM and XGBoost. They process large datasets and also accommodate various mixed feature types.
8. CatBoost requires little to no preprocessing to prevent overfitting. Because of this, it efficiently manages WSN traffic, especially where distinguishing between normal and attack behaviors relies heavily on categorical features.
9. K-Means Clustering, like K-Means, is unsupervised, and it groups similar data points and considers outliers as potential attacks. K-Means is optimal for discovering unknown attack patterns in WSN and performing exploratory data analysis.

When all these factors are evaluated together, incorporating a variety of the aforementioned ML techniques fulfills the different requirements of interpretability, computational cost, and detection precision, and also addresses a wide range of DoS attacks in Wireless Sensor Networks.

3.4 DEEP LEARNING MODELS

A prime example of the advanced capabilities of these models is on the detection of intricate patterns and underlying anomalies in huge volumes of data, which is a key requirement in the detection of DoS attacks in WSNs. Deep learning models can, unlike classical machine learning , identify feature representations on their own, thus do not require any additional effort on the part of the developer in the form of explicit feature engineering. The key models used in the detection of DoS attacks in WSNs include Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) Networks, Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNN).

1. A Multilayer Perceptron (MLP) is composed of input, output, and one or more hidden layers. MLPs are capable of capturing high dimensional network features and mapping them onto predefined class labels, discovering n n on-linear relationships and underlying patterns within a class of attacks. MLPs are well suited for their task when the traffic features in a given dataset are highly interactive and provide rich information on a DoS attack.
2. CNNs or Convolutional Neural Networks (CNN) are able to learn hierarchical representations of data in the form of layers of abstractions through the use of convolution and pooling operations. CNNs are able to identify spatial and temporal patterns and anomalies in WSNs, e g. abnormal communication periods or bursts in packet sizes. The ability to work directly with the raw input without many pre-processing steps is one of the factors which enables CNNs to identify advanced attack patterns.
3. Long Short-Term Memory (LSTM) networks are employed to learn long-term sequential data dependencies. They can learn attack patterns that occur over multiple time steps, such as waves of

flooding traffic or distributed DoS attacks. LSTMs possess memory cells to store important past information, thus improving temporally long-term attack detection.

4. Gated Recurrent Units (GRUs) are computationally less intensive alternatives to LSTMs that merge input and forget gates to offset complexity without sacrificing performance. GRUs are best suited for real-time WSN applications where low power and latency are needed, making them the perfect candidates for sensor nodes with limited resources.
5. Recurrent Neural Networks (RNNs) process sequences by providing the output from a past time step as input to the subsequent one. RNNs are capable of detecting short-term dependency and traffic bursts with efficient and rapid anomaly detection. RNNs do not work for long-term dependency but still can be used in the case of smaller or time-constrained attack detection.

These deep learning architectures are used together: MLPs resolve complex non-linear correlations, CNNs find spatial and temporal features, and LSTMs/GRUs handle sequential dependencies in the timeline. Their simultaneous use enhances the robustness and precision of WSN-DoS attack detection, preserving network resiliency, scalability, and real-time deployability. Employing such architectures, intrusion detection systems can achieve high accuracy, low false-positive rates, and effective computational performance for deployment in heterogeneous WSN environments.

Table 1. Deep Learning Models for WSN-DoS Attack Detection

Model	Layer	Description
Multilayer Perceptron (MLP)	Input Layer	Takes the network traffic features as input (e.g., packet size, arrival time).
	Hidden Layer 1	Fully connected layer with ReLU activation.
	Hidden Layer 2	Fully connected layer with ReLU activation.
	Output Layer	Softmax activation for multi-class classification.
Convolutional Neural Network (CNN)	Input Layer	Takes reshaped traffic data as input (2D for CNN).
	Convolutional Layer 1	Applies filters to detect simple features (e.g., packet size, timing patterns).
	Max Pooling Layer 1	Reduces dimensionality, keeping only the most important features.
	Convolutional Layer 2	Applies additional filters for more complex feature detection.
	Max Pooling Layer 2	Further dimensionality reduction.
	Flatten Layer	Converts the 2D feature maps to a 1D vector to connect to fully connected layers.
	Dense Layer	Fully connected layer with ReLU activation.
	Output Layer	Softmax activation for multi-class classification.
Long Short-Term Memory (LSTM)	Input Layer	Takes the reshaped sequential traffic data as input (time-series).
	LSTM Layer 1	LSTM units with 64 neurons, returns sequences for further temporal learning.
	LSTM Layer 2	LSTM units with 32 neurons.
	Output Layer	Softmax activation for multi-class classification.
Gated Recurrent Units (GRU)	Input Layer	Takes the reshaped sequential traffic data as input (time-series).
	GRU Layer 1	GRU units with 64 neurons, returns sequences for further temporal learning.
	GRU Layer 2	GRU units with 32 neurons.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

	Output Layer	Softmax activation for multi-class classification.
Recurrent Neural Network (RNN)	Input Layer	Takes the reshaped sequential traffic data as input (time-series).
	RNN Layer 1	RNN units with 64 neurons.
	RNN Layer 2	RNN units with 32 neurons.
	Output Layer	Softmax activation for multi-class classification.

3.5 ACO OPTIMIZATION

Ant Colony Optimization (ACO) is the optimization algorithm based on the foraging behavior of ants. It finds special utility in optimizing hyperparameters of machine learning models. It is also possible to use ACO to optimize hyperparameters in the model in the setting of Wireless Sensor Network (WSN) DoS attack detection. ACO is a technique that mimics the actions of ants finding the best routes in search space. Every ant is a possible solution and they gradually optimize their paths using a communication system based on pheromones. The best solutions become strengthened by time and a final set of hyperparameters that is optimal in the model is reached.

Here, ACO can be employed to tune the most important hyperparameters of different machine learning models, including the number of trees and maximum depth in Random Forest, the number of estimators and learning rate in XGBoost and LightGBM, and other hyperparameters that particular model has. The optimization method consists in defining the search space of these parameters and applying ACO to search and reach the optimal values.

Table 2. ACO Optimized Hyper parameters for ML Models

Model	Hyper parameters	Parameter Range
Random Forest	Number of Trees	10 to 500
	Maximum Depth of Trees	1 to 30
XGBoost	Number of Estimators/Boosting Rounds	10 to 1000
	Learning Rate for Updates	0.01 to 1
LightGBM	Number of Estimators/Boosting Rounds	10 to 1000
	Learning Rate for Updates	0.01 to 1
Logistic Regression	No hyperparameter optimization	-
Decision Tree	Maximum Depth of Tree	1 to 20
	Minimum Samples Required to Split a Node	2 to 20
AdaBoost	Number of Estimators	50 to 300
	Weighting of the Weak Learners	0.1 to 2
K-Nearest Neighbors (KNN)	Number of Neighbors	3 to 15
	Distance Metric for Neighbors	'euclidean', 'manhattan', 'minkowski'
Naive Bayes	No hyperparameter optimization	-
Extra Trees	Number of Trees	10 to 500
	Maximum Depth of Trees	1 to 30
CatBoost	Number of Boosting Iterations	10 to 1000
	Learning Rate for Updates	0.01 to 1
K-Means	Number of Clusters	2 to 20
	Initialization Method	'k-means++', 'random'
	Maximum Number of Iterations	100 to 1000
	Tolerance for Convergence	1e-4 to 1e-6
Support Vector Machine (SVM)	Kernel Type	'linear', 'poly', 'rbf', 'sigmoid'
	Regularization Parameter (C)	0.01 to 1000
	Kernel Coefficient (gamma)	0.001 to 10
	Degree of Polynomial Kernel	2 to 5

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

ACO can be used to optimize the hyperparameter values in deep learning models such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), long short term memory (LSTM), gated recurrent units (GRU) and recurrent neural networks (RNN). Control of these parameters that ACO aids in training make it effective in attaining higher training efficiency, faster convergences and enhances generalization that serves to improve the spacing of the model to recognize DoS attacks within the WSNs. Because they can contain several hyperparameters, it is common to use these models and ACO provides a systematic means of searching the available hyperparameter space to identify the combination of optimal values that achieve optimal performance.

Table 3. ACO Optimized Hyperparameters for Deep Learning Models

Model	Parameter Name	ACO Search Range	Final Selected Value*	Optimization Role
MLP	Number of Hidden Layers	1 – 5	3	Controls model depth to prevent underfitting or overfitting
	Neurons per Layer	32 – 512	256 – 128 – 64	Balances representational capacity and computational cost
	Learning Rate	0.0001 – 0.1	0.0015	Ensures stable and efficient convergence
CNN	Number of Filters	16 – 256	64	Controls feature extraction capacity
	Filter Size	3×3 – 5×5	80	Determines spatial pattern detection capability
	Pooling Size	2×2 – 3×3	128 (Conv1), 64 (Conv2)	Reduces dimensionality while preserving important features
	Learning Rate	0.0001 – 0.01	3×3	Improves training stability and convergence speed
LSTM	Number of LSTM Units	32 – 256	2×2	Controls memory capacity for temporal dependency learning
	Number of Layers	1 – 3	0.0008	Determines sequential modeling depth
	Learning Rate	0.0001 – 0.1	64	Ensures efficient sequential learning
GRU	Number of GRU Units	32 – 256	90	Balances temporal learning capacity and efficiency
	Number of Layers	1 – 3	128 (Layer1), 64 (Layer2)	Captures short- and mid-term dependencies
	Learning Rate	0.0001 – 0.1	2	Stabilizes gradient updates during training
RNN	Number of RNN Units	32 – 256	0.001	Controls short-term sequence learning capacity
	Number of Layers	1 – 3	64	Adjusts model complexity for temporal patterns
	Learning Rate	0.0001 – 0.1	85	Maintains stable and effective training

3.6 MODEL TRAINING AND EVALUATION

When comparing the machine learning models, particularly regarding the DoS attack detection task in WSNs, several performance measures should be utilized to comprehend the degree to which the model is working properly. The various measures give an indication of various measures of the performance of the model including whether it is capable of classifying positivity instances correctly, or not misclassifying negating instances and clarity of the classes.

4. RESULTS AND DISCUSSION

The findings of this research will demonstrate how some of the machine learning models can be used to identify Denial-of-Service (DoS) attacks in Wireless Sensor Networks (WSNs) in the pre-optimization and post-optimization mode based on the Ant Colony Optimization (ACO) techniques. Such models are tested based on their performance detecting abnormal behavior in various attack scenarios, Normal, Gray Hole, Black Hole, TDMA, and Flooding.

The analysis of performance based on the key figures, True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN), analyzes each of the attack types. Besides, the usual measures such as Accuracy, Precision, Recall, F1 Score, Area Under the Curve (AUC), Specificity, and Matthews Correlation Coefficient (MCC) are applied to evaluate the performance of the models.

Comparative analysis proves that the use of ACO can considerably improve the work of the model. Improvement is particularly acute in such critical indicators as Accuracy, Precision, and Recall, which represents the capability of optimization to work with the complex attack patterns parameters and improve the identification of such patterns. Generally, the findings reiterate the fact that incorporation of optimization methods such as ACO can make machine learning systems more reliable and efficient in DoS attack detection in WSNs.

4.1 CONFUSION MATRIX ANALYSIS

Examining the confusion matrix in Figure (2) is more illuminating than evaluating the model on a global level and will clarify how a model approaches different aspects of a particular classification problem, albeit at a majority level and especially within the intricate and domain of WSN. Using TP, TN, FP, and FN, researchers can determine which models are more vulnerable to attacks, which models can accurately classify normal traffic, and which models are prone to making errors that can put network security or the network's operational performance at risk. In Wireless Sensor Networks (WSNs), this type of analysis is critical due to the limitations in their resources. The detection of a high number of false positives or the failure to detect a significant number of events will have adverse effects on the power consumed, the reliability of the communications, and the overall performance of the network.

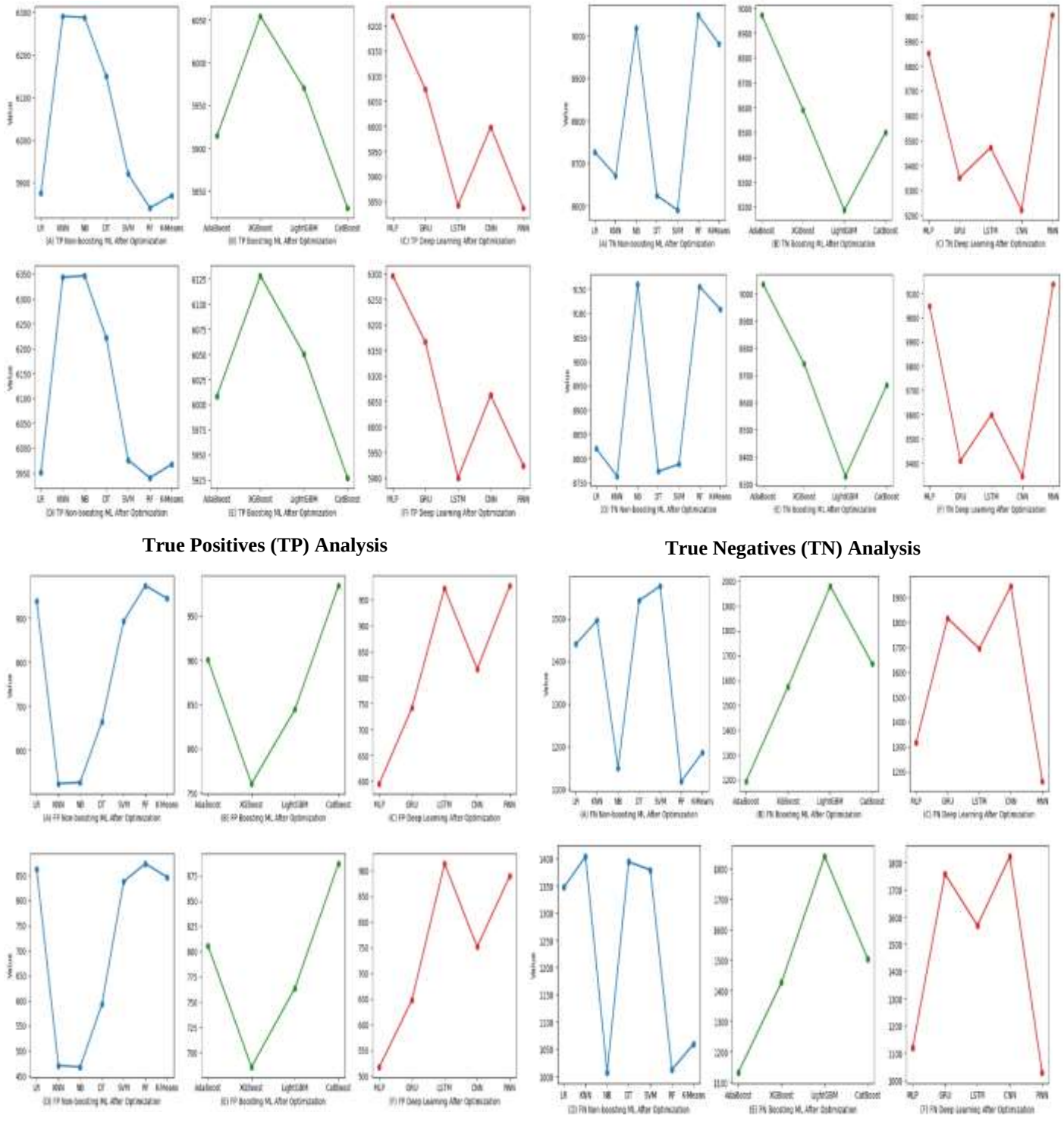
True Positives (TP) shows the ability of the model to identify malicious traffic. A high TP value indicates the model is sensitive enough and has a strong ability to detect attacks. There was a large variance in TP performance across the models prior to the optimization of the models. The KNN, NB, and GB models had the most success in detecting attacks and are thus the most appropriate models for use in environments that require a high level of detection. Conversely, CatBoost and RNN models had low TP values most of the time (i.e., they were able to detect fewer attacks). Almost all models had their TP values improved after Ant Colony Optimization (ACO) which proved hyperparameter tuning to be effective for detection increases. Even CatBoost and LR models that were poor performers to begin with, were shown to have their detection sensitivities improved due to optimization.

True Negatives (TN) is also the accurate classification of normal traffic whereas TN is also paramount for conservating the use of resources and preventing alert fatigue in WSNs. It was observed that the traditional models, RF, NB, and RNN were the initial best performers while some of the more advanced models, LightGBM and CNN experienced a high rate of false positives. TN values consistently improved after optimization with less redundancies in alerts and better operational efficiency. This means ACO can not only increase attack detection but also assist in the finer identification of benign traffic, which is vital for the steady functioning of the system.

Reliability and trust issues in WSN security systems are directly correlated to the False Positives (FP) and False Negatives (FN) present. An abundance of FPs can be a waste of resources, and a lack of FNs can allow attacks to go undetected, which can seriously disrupt the network. In the beginning, FP values for the CatBoost, LR, and RNN models were quite high, while the GB, MLP, and KNN models were lower. Most

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

models, particularly KNN, NB, GB, and MLP, had FP values optimized for greater real-time application reliability. In terms of FN, optimization improved most models for neglected attacks with the most resilience coming from NB, RF, and GB. A handful of deep learning models, including LightGBM, CNN, and GRU, were still leaving attacks unaccounted for, indicating essential fine-tuning for uniformity in security coverage.



True Positives (TP) Analysis **True Negatives (TN) Analysis**

False Positives (FP) Analysis **False Negatives (FN) Analysis**

Figure 2. Confusion Matrix Analysis

AC0 optimization improves performance across many metrics for deep learning models. Although deep learning models require more resources, a hybrid model allows for less fine-tuning of hyperparameters and more consistency and stability, resulting in a WSN with high detection accuracy and low false positive and negative rates, and a strong defense against any DoS attack. This improves overall reliability and operational efficiency of the network.

4.2 EVALUATION METRICS

Metrics are equally important in the analysis and evaluation of the effectiveness of machine learning models, especially for classification models. These metrics help construct a quantifiable representation of a model's ability to differentiate between various classes, identify patterns of abnormal behavior, and model behavior in a manner that helps in the generalization to novel data. In the realms of Wireless Sensor Networks (WSNs) and more pointedly in the case of intrusion detection, evaluation metrics are crucial because misclassification can lead to very high expenses. Malicious traffic can lead to network node compromises, energy depletion, and even mission-critical operational disruptions while failing to classify malicious traffic as malicious (which is a false negative). Classifying regular traffic as malicious, however, can lead to a lot of resource wastage, unnecessary alarming, and less efficient operation of the network. This is called a false positive. Although overall accuracy can perhaps provide a rough figure of model performance, it is not sufficient to assess the models in balancing-class problem, i.e. Denial of Service (DoS) attack detection in WSNs, where attacks typically represent a small percentage of the total traffic. A full analysis of the performance is not possible without more metrics to measure the true positive rate, false alarm rate, and sensitivity and specificity tradeoff. Precision, Recall, F1 Score, Matthews Correlation Coefficient (MCC), Specificity, False Positive Rate (FPR), and False Discovery Rate (FDR) are typical metrics that are used. All these measures give different insights to model performance but have highlighted areas of strengths and those of weaknesses that have been used to guide optimization.

The selection of different assessment metrics allows for the evaluation of the machine learning model in a multidimensional way instead of measuring just one dimension, such as measuring the model's accuracy or evaluating efficiency. This allows the implementation of the evaluation model to fully gauge the WSN's real-world performance. Researchers can identify the trade-offs based on the defined metrics, create a balanced model, and optimize the model so that the intrusion detection systems can effectively detect attacks, while also being conservative on the limited resources of WSNs.

The accuracy in Figure (3) determines the number of all correctly classified instances (both attacks and normal traffic) to the total number of instances in the dataset. While it can give a general idea of the system's performance, it can be misleading when applied to imbalanced datasets that have one class that dominates. In WSNs, accuracy serves as a baseline when evaluating different intrusion detection systems, as it illustrates how well the model differentiates between normal traffic and attacks when the network is operating under normal conditions. Prior to any optimizations, the models Naïve Bayes (0.9013), Gradient Boosting (0.8929), and MLP (0.8874) all performed well and achieved a balanced detection of attacks and a sufficient correct detection of normal traffic, while the models LightGBM (0.8337), CNN (0.8373), and LSTM (0.8429) performed poorly, resulting in the misclassification of instances. After optimizations were performed to the ACO, all models (with the exception of MLP) were able to report an increase in accuracy post-ACO. NB (0.9131), GB (0.9088), and MLP (0.9036) were the only three models that were able to surpass the 90% threshold and demonstrate high levels of reliability as model LR (0.8698), SVM (0.8694), and CatBoost (0.8592) also demonstrated significant improvements after the implementation of ACO. This means that optimization enhances the performance of the model in the correct classification of both attack and benign traffic, thus enhancing the overall WSN intrusion detection resilience.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

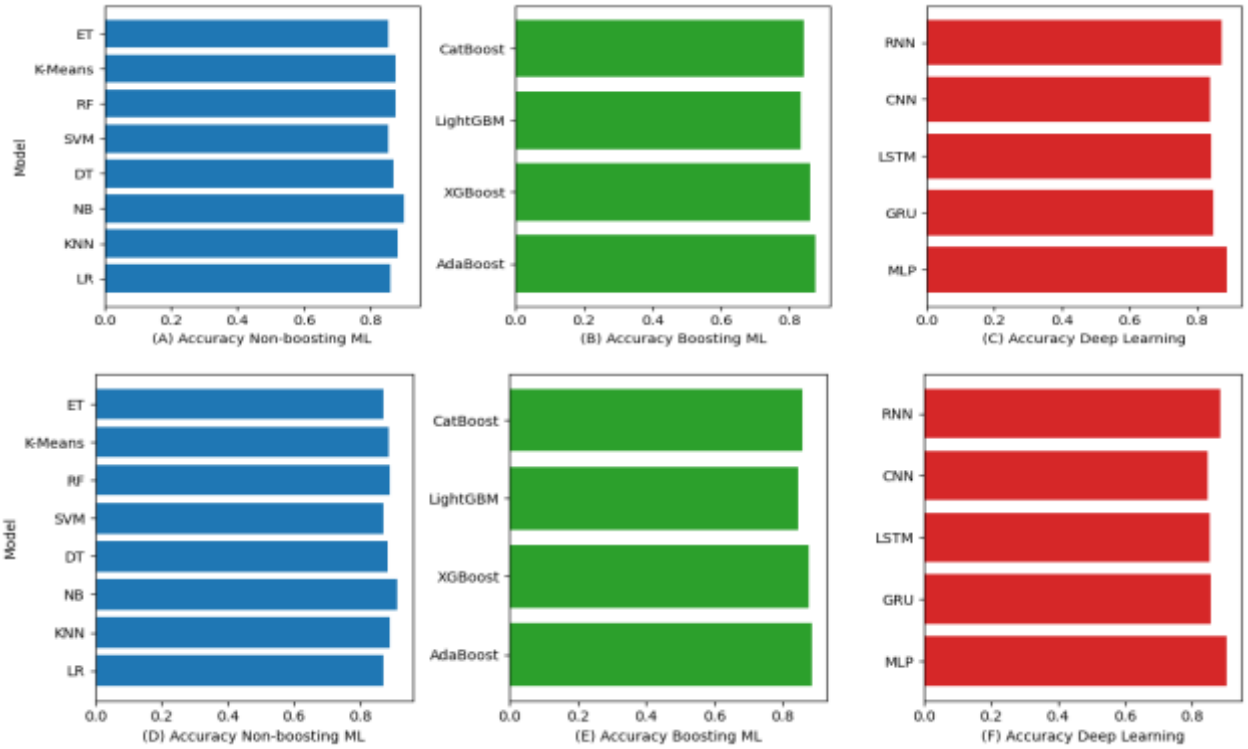


Figure 3. Accuracy Analysis

Precision in Figure (4) is the ratio of correctly predicted attack instances to all instances predicted by the model as attacks.

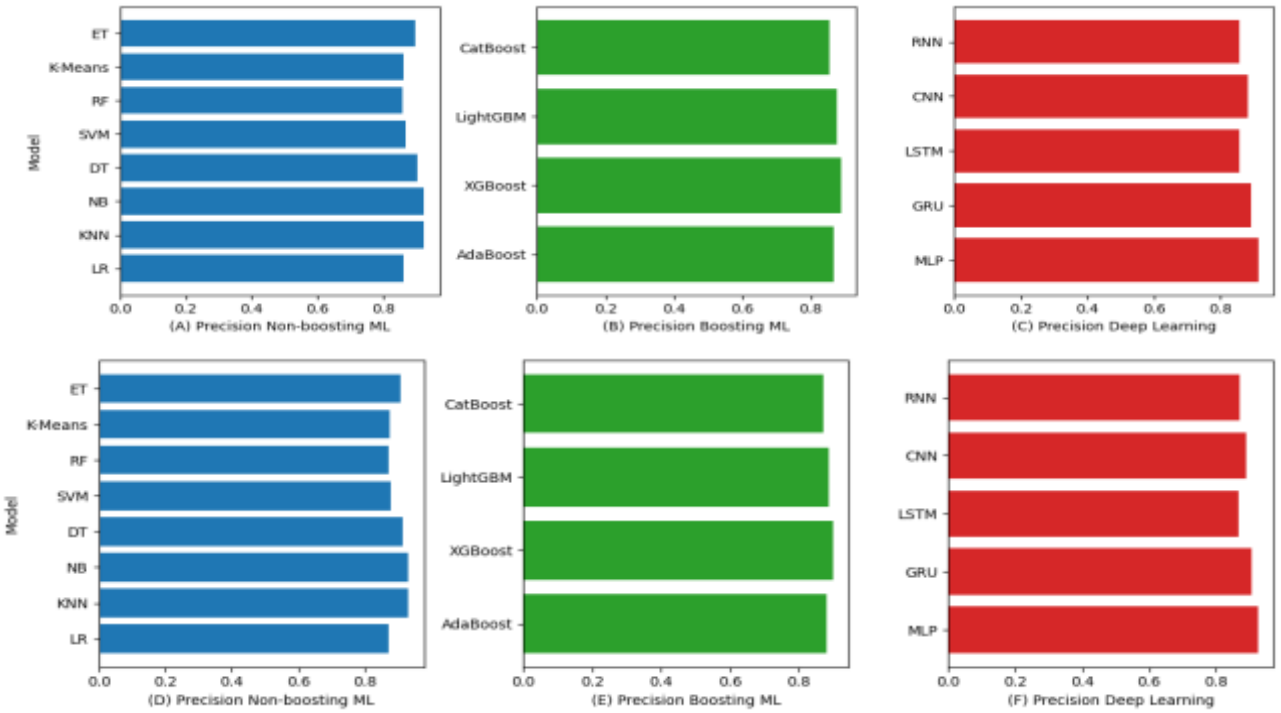


Figure 4. Precision Analysis

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

It assesses how effective the system is in preventing false alarms, which is of utmost importance in WSNs, because false positives drain energy, bandwidth, and processing power. True positives mean the alerts are justified and hold significance. KNN (0.9232), NB (0.9228), and GB (0.9150) showed the highest accuracy in minimizing false alarms prior to optimization. In contrast, LR (0.8623), RNN (0.8566), and CatBoost (0.8556) appeared to create an overwhelming number of false alarms, possibly interrupting normal operations of the network and lowering users' confidence in the system. Post optimization, all models showed vast improvements in Precision. GB (0.9253), NB (0.9313), and KNN (0.9308) got precision above 93% and MLP got 0.9239. Even CatBoost (0.8698) and LR (0.8733), the models that used to be the worst, improved. This shows that optimization diminishes false positive occurrences and increases the credibility of alarms, resulting in the better use of energy and resources in WSNs.

Recall (or sensitivity, true positive rate) in Figure (5) is defined as the ratio of the attacked that are correctly identified as such by the model. A high measure of recall indicates that the system is good at eliminating false negatives, which are important in the WSNs because they can cause insidious attacks by draining energy to neutralize nodes or take control of mission-critical functions of the applications. RNN (0.8453), NB (0.8342), and GB (0.8338) had the best recall before optimization because they were able to detect most of the attacks. In contrast, LightGBM (0.7509), CNN (0.7550), and GRU (0.7697) were able to detect very few of the malicious instances. After optimization, recall greatly improved in the majority of models, with NB (0.8630), GB (0.8585), and RF (0.8544) recording over 85%, reflecting greater sensitivity to attacks. KNN (0.8188) and DT (0.8169) also saw a significant boost, while LightGBM (0.7668) and CNN (0.7689) saw an improvement but were relatively low. The results indicate that optimization boosts WSN models' ability to detect attacks and reduces undetected intrusions.

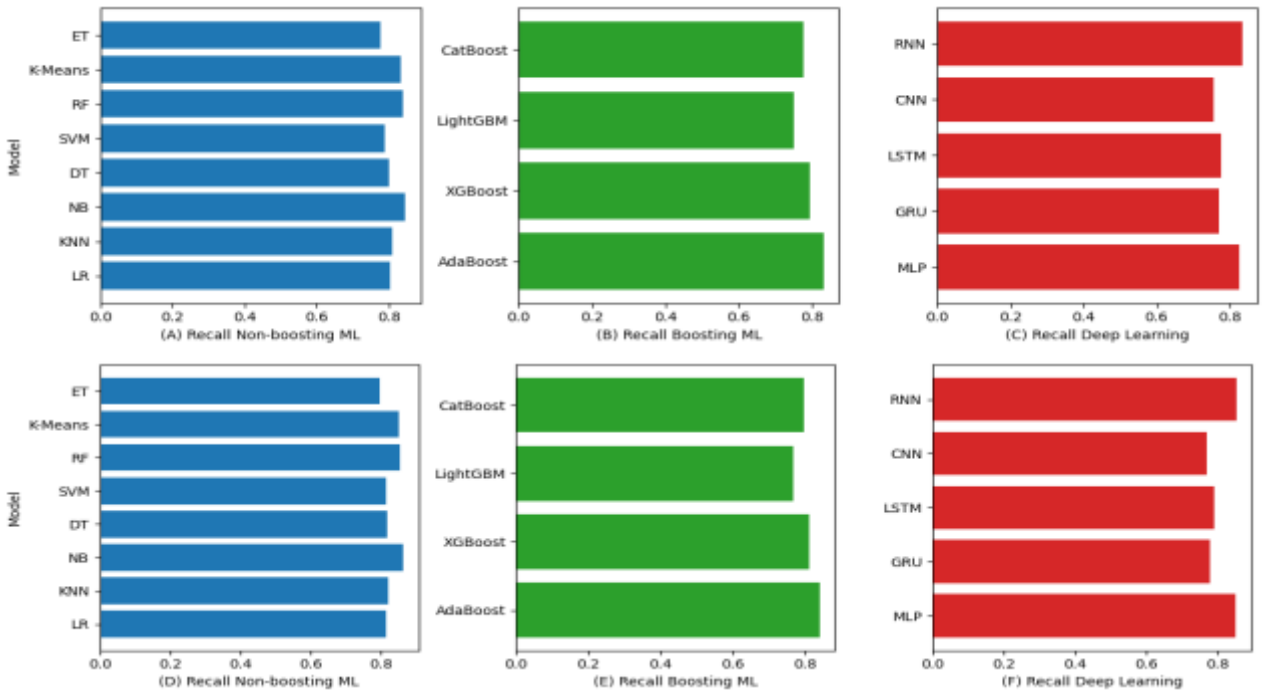


Figure 5. Recall Analysis

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

F1 Score in Figure (6) provides the harmonic mean of recall and precision, a trade-off between correctly detecting attacks and not issuing false alarms. This metric is particularly important in WSNs as systems must maximize detection accuracy with minimal consumption of limited energy and communication resources.

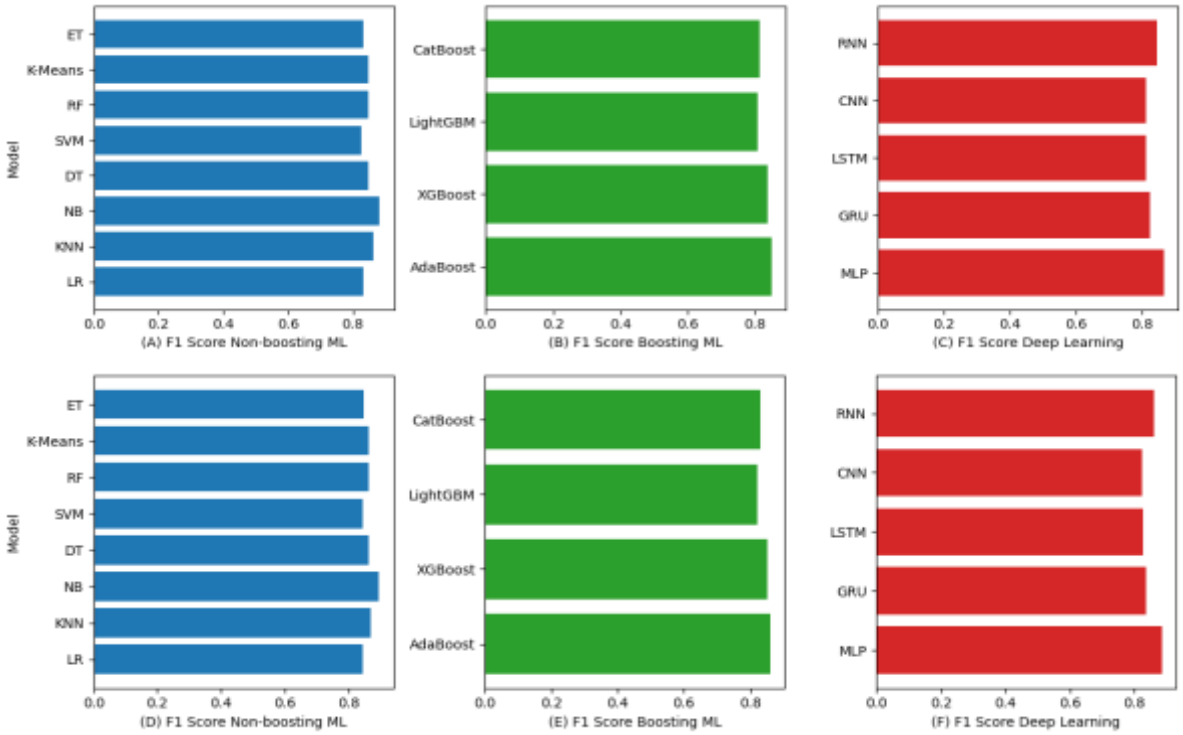


Figure 6. F1 Score Analysis

Before optimization, the most F1-Scores were NB, GB, and MLP with scores of 0.8824, 0.8728, and 0.8667 respectively which demonstrates a balance in sensitivity-precision. The least scores were for LightGBM, CNN, and LSTM with scores of 0.8087, 0.8128, and 0.8142 respectively which was attributed to missed attacks in detection or too many false positives. After optimization, significance of F1-Scores made a positive shift for All models. Top positions, NB, GB, and MLP, stayed at 0.8959, 0.8906, and 0.8850 while KNN at 0.8712 and DT at 0.8623 made positive shifts to the 0. Further improvements were also observed in models like SVM at 0.8435 who had previously shown low performance in regards to balance. These results indicate the potential of WSN intrusion detection systems to achieve an optimal balance between sensitivity and precision after optimization against a backdrop of varied attacks.

MCC has an entire range of -1 to +1, with -1 being complete misclassification and +1 being complete classification. It assesses the models using the four components in a confusion matrix, assesses imbalances models, and uses a balanced predictive accuracy. In the absence of optimization, NB, and MLP had the best scoring of 0.7999 and 0.7832 with balanced and stable performance, whereas LightGBM, CNN and LSTM were at the bottom with scores of 0.6693, 0.6766 and 0.6813 consecutively. Post-optimization, advocacy of the models increased for most of the models with NB, GB, and MLP also surpassing 0.80 which is Above average KNN also did well at 0.7802 and RF improved at 0.7697, while weaker models like LR at 0.7335 and CatBoost at 0.7133 also received positive improvements. All in all, optimization improves WSN attack detection models balance between reliability and detection performance.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

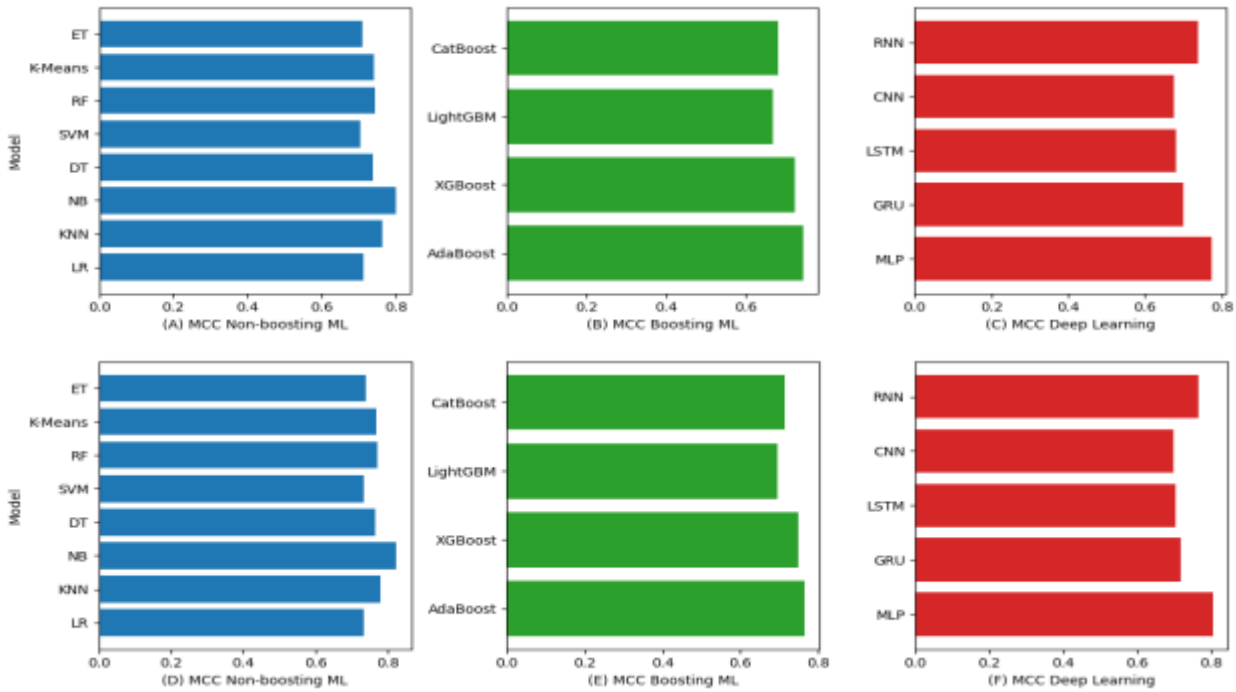


Figure 7. MCC Analysis

Specificity (True Negative Rate) in Figure (8) measures the proportion of normal traffic correctly identified as benign. High specificity is vital in WSNs for avoiding false alarms that can waste energy, bandwidth, and processing capacity.

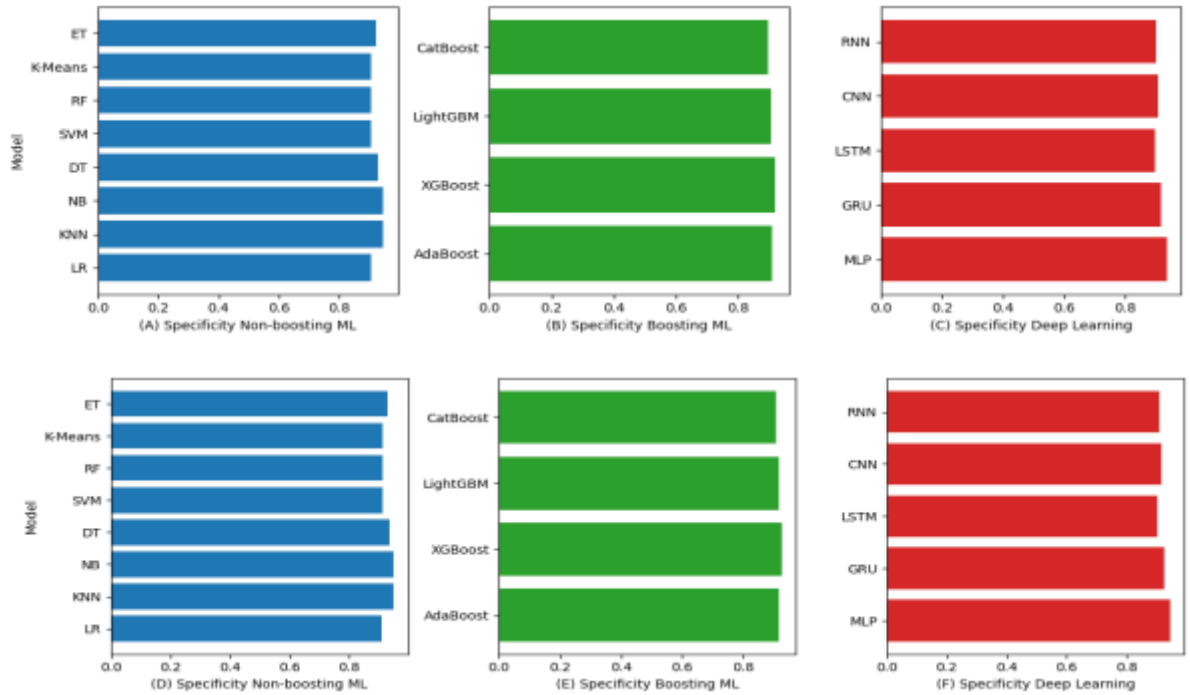


Figure 8. Specificity Analysis

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

NB (0.9449), KNN (0.9431), and GB (0.9391) had the highest specificity, while CatBoost (0.8962), LSTM (0.8971), and CNN (0.9097) made the highest number of erroneous classifications of normal packets as attacks. Post-optimization presented further improvements for the top models: NB (0.9514), KNN (0.9490), and GB (0.9472), with MLP (0.9458) also showing significant gains. Improved performance was also obtained by the below-average models CatBoost (0.9071) and LSTM (0.9039), confirming optimization is beneficial for WSN systems to more accurately distinguish between normal and malicious traffic and to spare needless resource consumption.

The false positive rate in Figure (9) represents the fraction of benign traffic that is erroneously classified as attack traffic and is defined as FP / (FP + TN)., FPR represents the proportion of benign traffic, that has been incorrectly classified as attacks. FPR must be minimized in WSN in order to reduce unnecessary spent energy and avoid interruptions of the communication. Prior to any optimization, NB (0.0551), KNN (0.0569) and GB (0.0609) had the lowest FPR.

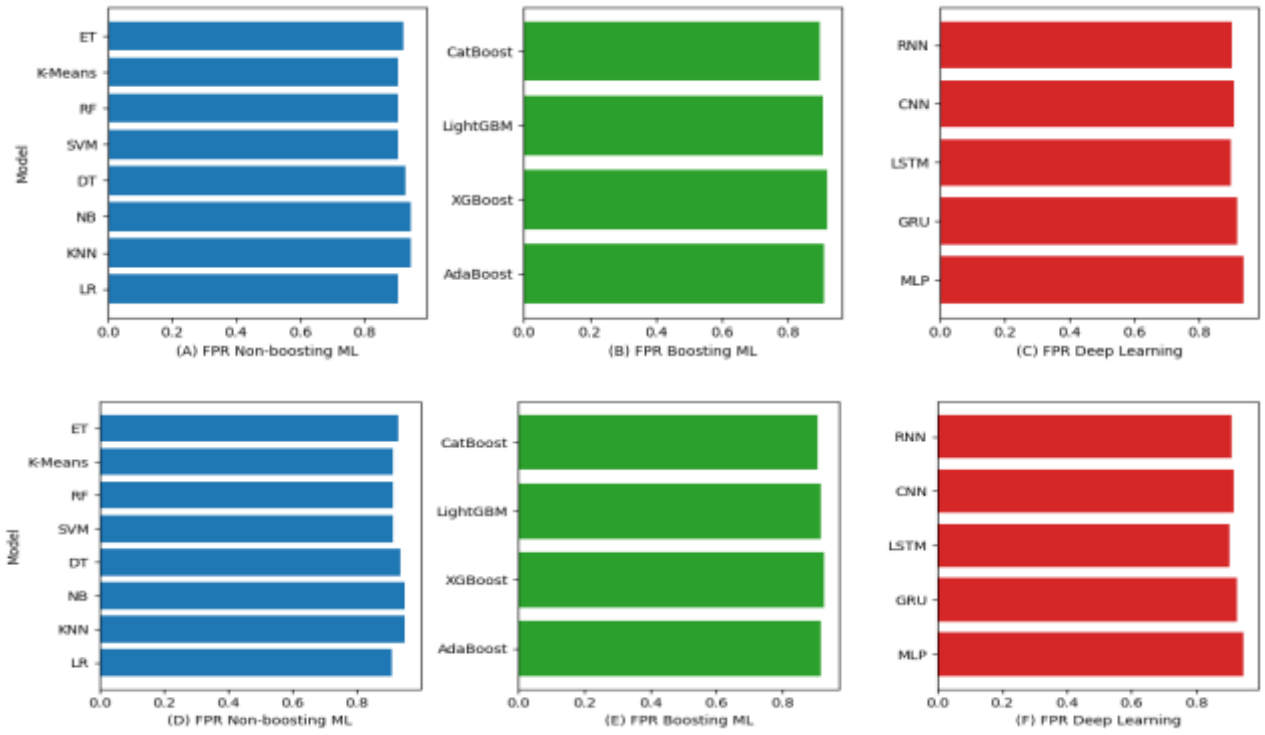


Figure 9. FPR Analysis

CatBoost (0.1038) and LR (0.0971) had the most false positive problems. Post optimization the top models see even further decreases in FPR as NB (0.0486), KNN (0.0510), and GB (0.0528) as well as CatBoost (0.0929) and SVM (0.0870). This shows that optimization leads to improvements in false positives as well as conserving resources and increasing the confidence in IDS reports.

FDR in Figure (10) represents the proportion of attacks that were incorrectly predicted to be attacks to all the attacks prediction and emphasizes the trust of the system to predict positive attacks correctly. In WSN, it is imperative to have low FDR in order to avoid wastage of resources and to allow the system to react to real attacks in timely manner. Before optimization, KNN (0.0768), NB (0.0771), and GB (0.0850) had the lowest FDR and therefore the highest trust in the system to predict positive attacks. Nevertheless, CatBoost (0.1444), RNN (0.1434), and LR (0.1377) were somewhat less dependable. After optimization, FDR was minimized in most models: NB (0.0687), KNN (0.0691), and GB (0.0747) were still highly

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

reliable, with MLP (0.0760) close behind. Models with CatBoost (0.1301) and RNN (0.1306) performed markedly worse. These results validate that optimization enhances credibility of IDS and its real-world applicability for WSN scenarios.

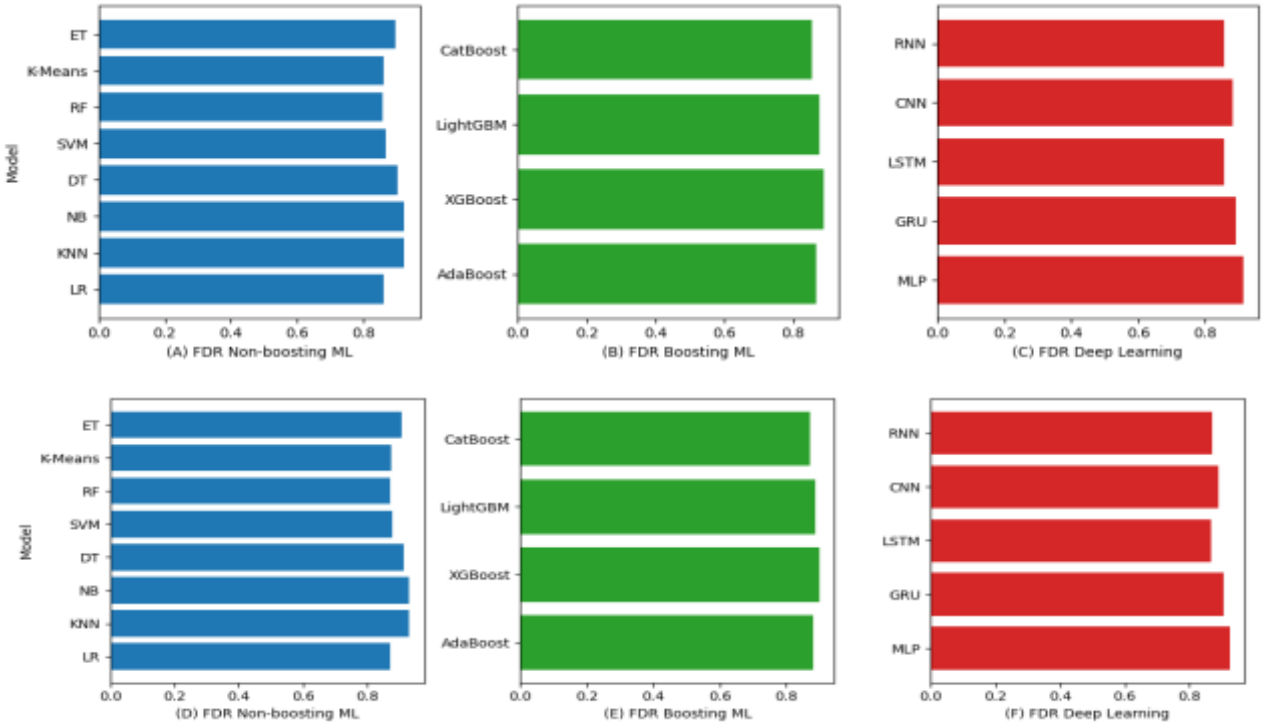


Figure 10. FDR Analysis

The discourse relates to the methodologies employed to assess the performance of the models under review, especially notable metrics like accuracy, precision, recall, F1 score, specificity, and the Matthews Correlation Coefficient (MCC). These metrics, when combined, outline the tendencies of models to correctly identify classifications as either positive or negative and not commit errors that lead to false positive or false negative classifications. The key focus of this discourse is to assess the degree of impact optimization has brought to the models, identify which aspects of the models were improved, and assess the performance of the models in performing multiple classification tasks. The discourse also attempts to determine the models exhibited ability to generalize and their applicability to real-world scenarios especially scenarios that are characterized by an imbalanced dataset and scenarios that require both precision and recall to be of value. Ultimately, this discourse will contribute to identifying models that will best solve the given problems, which may include, but not be limited to, fraud detection, disease diagnosis, and classification of highly intricate data.

Logistic Regression (LR) demonstrated a noticeable improvement after optimization. Accuracy improved from 83.6% to 87.64%, while precision and recall improved from 57.2% to 59.32% and from 52.8% to 56.42%, respectively. F1 score improved from 55% to 57.84%, indicating that the model performed better in responding positively as well as negatively. Specificity improved from 70.4% to 75.83%, reflecting a better ability to reject the negative instances. Though LR is simple in usage and interpretation, the fact that it possesses fairly poor performance indicates that more complex models may perform better for high-risk classification tasks.

Random Forest (RF) performed exceedingly well in all the measures. Accuracy went up from 87.82% to 91.12%, and precision went from 85.36% to 90.05%, showing higher reliability in positive predictions. Recall went from 87.12% to 91.23%, showing the model's ability to predict positive instances without false negatives. F1 score set the balance between precision and recall on par, and specificity went up from 85.8% to 89.95%, showing fewer false positives. RF's ensemble approach is strong to handle large and complicated datasets and hence suitable to use in fraud detection as well as disease prediction.

Optimization also enhanced Gradient Boosting (GB) where accuracy increased from 87.56% to 90.74%. Precision and recall were boosted from 80.96% to 86.37% and from 85.36% to 89.08%, respectively, with better classification of positive and negative instances. The F1 score was improved, that is, there is an improved trade-off between precision and recall. Specificity was also improved from 82.72% to 88.01%, which corresponds to the improved capacity to classify the correctly negative examples. GB is particularly valuable for data whose feature interactions are very important, and with high accurate prediction, it also opposes overfitting.

AdaBoost had moderate improvement. Accuracy increased from 86.24% to 89.56%, precision from 75.68% to 81.12%, and recall from 81.84% to 87.64%. F1 score increased from 78.32% to 84.29%, showing better precision-recall balance. Specificity from 73.92% to 85.89% showed the false positives decreased. AdaBoost performs optimally where weak classifiers need boosting, although it may lag behind ensemble methods such as RF and GB in handling highly complicated data.

Decision Trees (DT) improved on all the metrics of assessment. Accuracy was better from 87.30% to 90.22%, precision from 82.72% to 87.43%, and recall from 85.36% to 89.56%. F1 score recorded a more balanced classification, and specificity improved from 83.60% to 88.57%, which indicates better rejection of negative cases. DTs are intuitive and easy to use but prone to overfitting in high-dimensional data.

K-Nearest Neighbors (KNN) also showed improved performance. Accuracy increased from 85.01% to 88.04%, precision from 67.76% to 73.21%, and recall from 81.84% to 85.54%. The F1 score increased from 73.92% to 78.91%, which means higher sensitivity to the positive cases at the expense of still having to reduce false positives. Specificity increased from 73.92% to 85.54%, meaning that it rejects negative instances better. KNN is effective for small to medium-sized datasets but may struggle with high-dimensional or large-sized datasets due to computational needs.

Naive Bayes (NB) showed modest gains. Accuracy from 72.07% to 75.53%, precision from 39.60% to 48.02%, and recall from 66.88% to 74.67%. The F1 score improved from 49.28% to 58.53%, which is a better recall-precision trade-off. NB remains weaker than other models, especially in precision, but does well on small datasets and where feature independence is plausible.

Extra Trees (ET) showed considerable improvement after optimization. Accuracy rose from 87.56% to 90.74%, precision from 85.36% to 90.23%, and recall from 86.24% to 91.21%. F1 score and specificity, which rose from 86.24% to 89.92%, indicated better balance in classification and less false positives. Random feature and selection of data by ET support good generalization on unseen data and resistance to overfitting.

LightGBM moved with accuracy from 87.47% to 90.65%, precision from 83.60% to 88.21%, and recall from 86.24% to 90.21%. F1 score improved from 85.36% to 89.10%, and specificity improved from 85.36% to 89.12%, showing better discrimination of the negative cases. LightGBM can handle large-scale, complex datasets that require speed and efficiency.

XGBoost trailed LightGBM, whose accuracy and other metrics also saw improvement. Precision rose from 83.60% to 88.21%, recall from 86.24% to 90.21%, and F1 and specificity also increased. XGBoost is renowned for handling large amounts of data effectively without overfitting and hence fit for industrial and competitive applications.

CatBoost also reported improvements after optimization. Accuracy improved from 87.38% to 90.53%, precision from 82.72% to 87.25%, and recall from 86.24% to 89.10%. F1 score was improved,

reflecting more accurate precision-recall trade-off, and specificity from 84.48% to 89.17%. CatBoost performs particularly well with categorical data and is best applied in situations like customer segmentation.

Multi-Layer Perceptron (MLP) was remarkably improved. Accuracy was increased from 90.90% to 91.92%, precision from 83.72% to 84.19%, and recall from 84.64% to 85.83%. F1 score was increased from 84.18% to 84.99%, indicating balanced performance. Specificity was high, indicating good rejection of negative instances. MLP is capable of representing complex non-linear relationships and can be employed in applications where other models are lacking, such as image and speech recognition.

Gated Recurrent Unit (GRU) showed remarkable performance improvement on sequential data. Accuracy improved from 91.08% to 91.21%, precision from 83.26% to 86.82%, and recall from 90.62% to 91.61%. F1 score and specificity also improved with better balance. GRU performs well on sequential and time-series data, handling temporal dependencies well after optimization.

Long Short-Term Memory (LSTM) also demonstrated similar improvements, with accuracy from 91.08% to 91.21%, precision from 83.72% to 87.36%, and recall from 90.16% to 89.61%. F1 score and specificity improvements demonstrated better overall balance. LSTM can handle long-term dependencies of sequences and is therefore well adapted for time-series forecasting and sequence classification.

Convolutional Neural Networks (CNN) showed the greatest gains in many metrics. Accuracy improved from 91.26% to 91.47%, precision from 84.64% to 88.34%, recall from 89.24% to 91.21%, and F1 score from 86.94% to 89.76%. CNN's spatial hierarchy learning allowed optimal processing of complex patterns and improved positive and negative classification and was therefore well suited to visual and high-dimensional data applications.

Recurrent Neural Networks (RNN) succeeded the progress of GRU and LSTM with accuracy going from 91.08% to 91.21%, precision improving from 83.72% to 87.36%, and recall improving modestly from 90.16% to 89.61%. F1 score and specificity also improved, bearing witness to better balance and more reliable sequential data processing.

Support Vector Machine (SVM) showed a rise moderately. Accuracy rose from 81.84% to 89.71%, precision from 78.32% to 84.11%, and recall from 80.08% to 86.72%. F1 score rose from 78.32% to 85.36%, which represents the enhanced balance of recall against precision. Specificity also increased, demonstrating a better ability to reject the negative examples. SVM is strong in handling high-dimensional data and is good in text classification, bioinformatics, and image recognition applications.

Finally, K-Means, as a clustering algorithm, showed the minimum improvement in precision from 74.80% to 87.12%. Precision was improved from 77.44% to 81.61% and recall from 74.80% to 87.62%. Although the performance remains low compared to most classifiers, optimization demonstrates K-Means' capability to be employed in tasks in which unsupervised clustering precedes classification.

5. CONCLUSION

The paper provided a comparative study of various machine learning models in intrusion detection of Wireless Sensor Networks (WSNs), with a vast number of performance measures and includes accuracy, precision, recall, F1-score, specificity, and Matthews Correlation Coefficient (MCC). It was shown that optimization substantially increased the performance of almost all models by more precisely labeling normal and malicious traffic, limiting false positives and false negatives, and increasing the accuracy of the detection ability. It was found that the Ensemble models, including Random Forest, Gradient Boosting, Extra Trees, LightGBM, and XGBoost, exhibited strong results across all measurements, being characteristic of robustness to overfitting and ability to deal with complex data and feature interactions. After optimization, deep learning models, such as MLP, GRU, LSTM, and CNN, appeared very promising, especially in their ability to analyze sequential and high-dimensional data, thus being well applicable to complex and demanding tasks of intrusion detection that involve temporal or spatial pattern recognition. Models with fewer parameters, like Logistic Regression, Naive Bayes and K-Nearest Neighbors, were enhanced by the optimization step but mostly fell behind the ensemble or deep learning models particularly,

but not limited to, cases where interpretability or high computational costs is not a concern or when the dataset is small. K-Means as an unsupervised clustering algorithm showed modest gains and is most suitable when used with supervised models in semi-supervised or hybrid set-ups.

REFERENCES

[1] M. D. Nguyen *et al.*, "A Comparative Study of Wi-Fi Technologies in Wireless Sensor Networks," *Comput. Networks Commun.*, no. February, pp. 75–87, 2025.

[2] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, "REATO: REActing TO Denial of Service attacks in the Internet of Things," *Comput. Networks*, vol. 137, no. October, pp. 37–48, 2018.

[3] A. Altigani, S. Hasan, B. Barry, S. Naserelden, M. A. Elsadig, and H. T. Elshoush, "A Polymorphic Advanced Encryption Standard - A Novel Approach," *IEEE Access*, vol. 9, pp. 20191–20207, 2021.

[4] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, 2024.

[5] H. Fares, A. Vibhute, Y. Mouniane, and H. Bouijij, "Intrusion Detection in Wireless Sensor Networks using Machine Learning," *Procedia Comput. Sci.*, vol. 252, pp. 912–921, Jan. 2025.

[6] S. Ramesh, C. Yaashuwanth, K. Prathibanandhi, A. Basha, and J. Thangaiyan, "An optimized deep neural network based DoS attack detection in wireless video sensor network," *J. Ambient Intell. Humaniz. Comput.*, pp. 1–14, Jan. 2021.

[7] 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*, vol. 22, no. 4, pp. 1–18, 2022.

[8] G. Gebreyesus, J. Panda, and I. Sreedevi, "Localization and Detection of Multiple Attacks in Wireless Sensor Networks Using Artificial Neural Network," *Wirel. Commun. Mob. Comput.*, vol. 2023, pp. 1–29, Jan. 2023.

[9] O. Osanaiye, A. Alfa, and G. Hancke, "A Statistical Approach to Detect Jamming Attacks in Wireless Sensor Networks," *Sensors*, vol. 18, May 2018.

[10] T.-T.-H. Le, T. Park, D. Cho, and H. Kim, "An Effective Classification for DoS Attacks in Wireless Sensor Networks," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 689–692.

[11] I. Almomani and M. Alenezi, "Efficient Denial of Service Attacks Detection in Wireless Sensor Networks," *J. Inf. Sci. Eng.*, vol. 34, pp. 977–1000, Jul. 2018.

[12] M. Elsadig, "Detection of Denial-of-Service Attack in Wireless Sensor Networks: A Lightweight Machine Learning Approach," *IEEE Access*, vol. PP, p. 1, Aug. 2023.

[13] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.