Journal of Wireless Networks and Communication Systems
Vol. 1, Issue. 3, Jul-Sep 2025, pp. 48~61
ISSN: 3107-6432

Anomaly Detection in Wireless Sensor Networks Using Hybrid Machine Learning Models

Raghu M[1], Mr.T. Selvaraj[2]
[1]Department of Electrical & Electronics Engineering,
PARK College Engineering and Technology, Coimbatore, India.
[2]Assistant Professor, Department of Electrical and Electronics Engineering,
PARK College of Technology, Coimbatore, India.

Article Info

Article History:
Received Jun 19, 2025
Revised Jul 20, 2025
Accepted Aug 24, 2025

Keywords:
Wireless sensor networks
Anomaly Detection
Machine Learning
Principal Component Analysis
Graph Neural Network
Autoencoder

ABSTRACT

Nowadays, WSNs, or Wireless sensor networks, have become one of the most widely used wireless technologies for the purpose of sensor communication. WSNs are often designed for specialized applications, involving tracking or monitoring, in indoor or outdoor settings where battery capacity is a major concern. Over the past few years, numerous routing schemes have been developed to resolve these challenges. Currently, researchers use a variety of machine learning (ML) approaches to identify anomalies in WSN. The study presents an Anomaly Detection in Wireless Sensor Networks Using Hybrid Machine Learning Model (ADWSN-HMLM). The ADWSN-HMLM approach undergoes data preprocessing, feature extraction, detection, and classification. The benchmark IDS dataset is used to test the experimental results of the ADWSN-HMLM approach. The simulation outcomes showed that the ADWSN-HMLM approach outperformed other approaches. The infrastructure of WSN and the security issues they encounter are conveniently referenced in this study. Along with discussing the difficulties and suggested solutions for enhancing sensors' capacity to recognize threats, attacks, risks, and malicious nodes through their capacity to learn and self-develop using ML techniques, this study also explores the potential benefits of ML approaches in lowering the security costs of WSN across a number of domains. In addition to 98% for regular traffic, the detection accuracy for scheduling, grayhole, flooding, and blackhole assaults is 97.59%, 96.95%, 96.03%, and 97.05%, respectively. These findings demonstrate that the ADWSN-HMLM methodology can offer the WSN effective anomaly recognition.

Corresponding Author:

Raghu M,
Department of Electrical & Electronics Engineering,
PARK College Engineering and Technology, Coimbatore, India.

1. INTRODUCTION

Sensor nodes (SNs) serve as the primary backbone of wireless sensor networks (WSNs), which are among the most widely used wireless communication technologies [1]. In terms of configuration, WSNs can have hundreds to thousands of sensors, either homogeneous or heterogeneous. The majority of WSNs are made for a certain purpose, and typically, their SNs have some fundamental features, including communication, processing, sensing, and computation. Radio frequency electromagnetic pulses are used mostly for communication with nearby nodes. In addition, a base station (BS), which serves as a centralized node from which the SNs send their monitored data, is normally located at a designated spot in the WSN architecture [2].

WSNs are application-based communication systems, meaning that the communication network is constructed and sensors are placed in the monitoring area based on a particular application. For the purposes of tracking and monitoring, WSNs are typically placed in open areas. The rubber business, the chemical industry for harmful gas monitoring, and patient health monitoring are a few examples of monitoring applications [3]. Additionally, WSN technology can be utilized for tracking applications like tracking people, pets, and wild creatures. Micro-electromechanical systems (MEMS) have been included in recently released devices. WNS technology has evolved significantly in recent years, offering more creative platforms for more economical and effective communication networks.

Due to their numerous real-time applications, including battlefields, forest fire monitoring, healthcare, essential military surveillance, and building security monitoring, WSNs are currently one of the most popular study topics [4]. These applications are designed with the assumption that all nodes are trustworthy and cooperative. This isn't the case in real-time deployments, though, as SNs are subject to various incursions and attacks that can seriously impair the network's ability to function properly and reduce system performance.

Unfortunately, it might be challenging to protect this kind of network from different malicious attack activities, particularly when the nodes are composed of low-cost electronic devices with inadequate hardware capabilities [5]. Radio frequency information can be detected, decoded, and transmitted by SNs. There are very few base stations (BS) nodes in WSNs. Front-line tracking, compound smoke testing, weather monitoring, and monitoring in clinics (patients at risk) are only a few of their many applications.

Furthermore, WSNs are generally helpful for tracking and inspecting in difficult-to-reach places where human intervention is either impossible or not possible. Gaseous tension and concoction vapor observations are used for checking, and tracking is used for both human and animal tracking. In WSN, anomaly detection is a major challenge to guarantee security and stop malicious attacks. Currently, researchers use a variety of ML techniques to identify anomalies in WSN [6].

In a WSN, energy and security pose significant issues and are mutually exclusive. Battery drain will increase as security complexity rises. Given their detrimental effects on one another, security and energy usage rank among the most significant issues facing WSNs [7]. The requirement for lowering security and energy usage is one of the issues that modern research in this area is tackling because of the difficult conditions in which these sensors can function. Data encryption between two communication devices (two nodes) and related processes, including encryption and key exchange, are also regarded as conventional. Furthermore, ML approaches do not require human involvement, which is consistent with the nature of WSNs. The computational and resource constraints of nodes, as well as

the requirement for sizable data sets for learning, are the two primary obstacles to ML in WSNs. Additionally, ML algorithms are quite effective at identifying suspicious nodes and analyzing packets as they move between WSN nodes.

The research's primary contribution is as follows:

- Increased Productivity: Industrial WNs can precisely monitor and analyze data to spot possible problems and provide innovative, more effective methods to boost production efficiency thanks to ML approaches.
- Predictive Maintenance: By identifying irregularities in the data gathered from industrial wireless sensor networks, ML algorithms are able to forecast when maintenance is required. It lowers the expenses and downtime related to unforeseen repairs.
- Increased Safety: Algorithms that use ML are able to identify possible risks in operational settings and notify operators when necessary to take preventative action.
- Automation: By automating procedures and tasks, industrial wireless sensor networks can lower labor costs while improving accuracy.
- Better Quality: Before a product is sent, ML algorithms can find and identify flaws, which lowers return costs and boosts customer satisfaction.

The study presents an Anomaly Detection in Wireless Sensor Networks Using Hybrid Machine Learning Model (ADWSN-HMLM). The ADWSN-HMLM approach undergoes data preprocessing, feature extraction, detection, and classification. The benchmark IDS dataset is employed to test the experimental results of the ADWSN-HMLM method. The simulation results showed that the ADWSN-HMLM approach outperformed other approaches. The infrastructure of WSN and the security issues they encounter are conveniently referenced in this study. Along with discussing the difficulties and suggested solutions for enhancing sensors' capacity to recognize threats, attacks, risks, and malicious nodes through their capacity to learn and self-develop using ML algorithms, this study also explores the potential benefits of ML algorithms in lowering the security costs of WSN across a number of domains.

2. RELATED WORKS

Talukder et al. [6] use the WSN-DS and TON-IoT datasets to assess a new hybrid ML model that integrates PCA for dimensionality reduction and KMeans-SMOTE (KMS) for data balancing. To improve detection efficiency and accuracy, the model uses gradient boosting methods like XGBoost (XGBC) and classifiers like Random Forest Classifier (RFC) and DT Classifiers. The suggested hybrid (KMS + PCA + RFC) strategy performs exceptionally well on the WSN-DS dataset. It surpasses conventional data balancing methods based on SMOTE, TomekLink, and Generative Adversarial Networks. This hybrid technique offers scalable and reliable anomaly recognition while addressing issues with class imbalance and high dimensionality. The presented method is appropriate for real-time applications since it shortens training and prediction times, according to complexity analysis.

Salman et al. [7] introduced two dissimilar approaches. LSTM-deep network layers were integrated with a proprietary CNN in the first model. In order to create an ANN, the second model was constructed around all fully connected layers, or dense layers. In contrast to the Logistic Regression method (LR), which demonstrated exceptional results.

Altulaihan et al. [8] present an IDS defense system that uses intrusion detection and ML to reinforce IoT infrastructure security against DoS attacks. The suggested IDS continuously scans network traffic for departures from typical characteristics using anomaly recognition. Four dissimilar supervised classification models— SVM, DT, RF002C, and KNN —were employed. Furthermore, two dissimilar feature selection (FS) models were utilized—the Genetic Algorithm (GA) and the Correlation-based FS (CFS) model—and assessed how well they performed. We also applied the IoTID20 dataset to train our model for identifying unusual activities in IoT networks. When RF and DT classifier models were trained using selected features by GA, improved outcomes were obtained.

Mishra Jain [9] investigates enhanced anomaly and threat detection in IoT networks using Kolmogorov-Arnold Networks (KANs). The implementation uses the Reflectional Switch Activation Function (RSWAF) in conjunction with the Gaussian Radial Basis Function (RBF) for real-time applications. The network's capability to capture local nonlinear interactions is made possible by RBFs, which enhance the model's performance and computational efficiency. Faster learning and inference are made possible by the computationally effective activation mechanism offered by the RSWAF. Our tests show that the faster-KAN implementation considerably cuts down on training and inference durations while retaining robustness and excellent accuracy in identifying anomalies and attacks.

Elsadig [10] clarifies WSN limitations, vulnerabilities, and security risks, emphasizing DoS attacks. Recent methods for detecting DoS attacks have been carefully examined, exposing both their strengths and weaknesses. This offers insightful information about the state of recent studies in this area. In order to detect DoS assaults in WSNs, this work suggests a lightweight ML detection method based on a DT algorithm with the Gini FS method. The suggested method was trained and tested on an improved version of the author's WSN-DS dataset. When compared to RF, XGBoost, and KNN classifiers, the suggested method has demonstrated good performance, with an accuracy rate of 99.5% with minimal overhead. In terms of processing time, our suggested method performs noticeably better than FR, XGBoost, and KNN. Although RF obtained a slightly better accuracy, the suggested method significantly outperformed RF, which is crucial for satisfying WSN restrictions.

Gowdhaman and Dhanapal [11] suggested two methods—Data Distribution Based Concept Drift Detection and Error Rate Based Concept Drift Detection—and inspected their effects. Moreover, K-Means Clustering in conjunction with sliding window-based data collection and drift analysis has been utilized to upgrade training datasets and decrease data size. In order to discover anomalies, we employed the SVM classifier. Based on statistical testing, we have started retraining the model. The severity of concept drift in the datasets has been examined using detection accuracy, Kappa statistics, and KL-Divergence. The SVM has demonstrated improved classification accuracy following the implementation of the suggested method.

Behiry and Aly [12] suggest a clever hybrid architecture that uses AI and ML to detect and stop intrusions, improving the security of WSNs. Singular Value Decomposition (SVD) and PCA are two feature reduction approaches used in the study. Intrusion detection systems and network traffic classification are presented with the introduction of the Synthetic Minority Oversampling Technique for data balancing. Using both complete and reduced feature sets, the study assesses the experimental measure of a DL-based FFNN method. Additionally, a comparison with benchmark ML techniques is carried out. The proposed method accomplishes very well, attaining excellent intrusion detection accuracy and dependability for WSNs. The paper advances WSN security by outlining the system configuration and parameter settings.

Bukhari et al. [13] suggest a new architecture for intrusion detection in WSNs that combines a stacked CNN with SCNN-BiLSTM. Federated Learning (FL) is used in this model to improve detection efficiency and protect privacy. The suggested approach takes a novel approach by enabling several SNs to work together to train a central global model without disclosing personal information, allaying privacy worries. By carefully analyzing both local and temporal linkages, the DL methodology successfully detects complex and unidentified cyberthreats. Our suggested model showed better detection rates for intricate and unknown threats than conventional Artificial DNN (ADNN) approaches, greatly enhancing IDS performance. On both datasets, the model significantly decreased false positives and negatives while achieving a noteworthy classification accuracy. Our study demonstrates how DL and FL can improve WSN security and privacy. In addition to making it easier to identify sophisticated cyberthreats, the proposed architecture serves as an example of how DL methods can be used to strengthen an anamoly recogniton system while protecting user privacy.

3. PROPOSED METHODOLOGY

The study presents an ADWSN-HMLM approach. The ADWSN-HMLM approach undergoes data preprocessing, feature extraction, detection, and classification. The overall architecture of ADWSN-HMLM approach is demonstrated in Figure 1.

3.1 Data Pre-processing

To normalize or standardize the range of features in a dataset, feature scaling is a critical pre-processing stage. By bringing all features to a comparable scale, it ensures that no feature takes over the learning process just because it is more significant. $\mathrm{Z-score}$ normalization and $\mathrm{Min-Max}$ scaling are the two popular feature scaling techniques.

3.1.1. Min-Max Scaling

The features are scaled to a specified range, typically between zero and one, using min-max scaling. The following is the formula for Min-Max scaling:

$$\mathrm{X_{Scaled}} = \frac{\mathrm{X}-\mathrm{X_{min}}}{\mathrm{X_{max}}-\mathrm{X_{min}}} \quad (1)$$

In Eq. (1) X and X scaled are the original and the scaled feature value, $\mathrm{X_{min}}$ and $\mathrm{X_{max}}$ denotes the minimal and maximal values of the feature in the dataset.

All feature values are transferred to a range between zero and one thanks to min-max scaling. It works well when the features have a restricted range and the feature distribution is approximately linear.

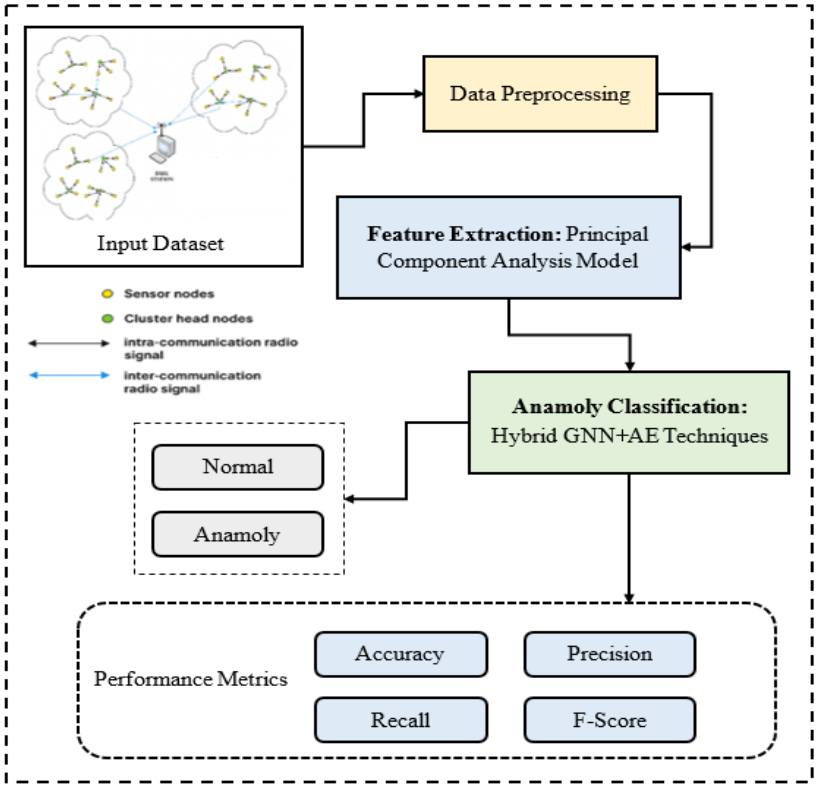

Figure 1. Overall architecture of ADWSN-HMLM approach

3.1.2. Z-score Scaling

Z-score scaling, sometimes referred to as standardization, modifies the characteristics to have a standard deviation of one and a mean of zero.

$$X_{scaled} = \sigma X - \mu/\sigma \quad (2)$$

The scaled feature value is denoted by X scaled, where X represents the original feature value. When the feature distribution is unbounded and unaffected by outliers, Z-score scaling can be helpful. The model can comprehend how well every feature contributes to the data distribution because it preserves the relative relationships between feature values. The data's properties and the algorithm being used determine which feature scaling technique is best. Min-Max scaling maintains the original data range and works well when the features have distinct minimal and maximal values. In contrast, Z-score scaling is robust to outliers and is typically chosen when the mean and variance are interpretable and the feature distribution is not constrained. It is crucial to remember that feature scaling, which is usually carried out during the data preparation stage prior to feeding the data into ML approaches, should be implemented independently for every feature in the dataset. By enhancing the convergence of gradient-based optimization techniques, mitigating the effects of numerical instability, and facilitating a significant comparison of various features within the dataset, feature scaling contributes to enhanced model accuracy and performance.

Wiener Filtering

Wiener filtering is carried out by the function wiener, which is built in. Wiener filtering is an adaptive method that estimates noise reduction based on local variance and modifies its filtering procedure accordingly. The Wiener filter achieves an excellent balance between edge preservation and

noise reduction when its window size is 5x5. The Wiener filter performs exceptionally well when applied to Gaussian noise and is advantageous for a range of sensor noise levels [14].

3.2 Feature Extraction using PCA

By extracting features from the original data while keeping variance information in the original data, PCA is a feature extraction technique that lowers the data dimensionality. By determining the correlation between data points and eliminating those with a strong correlation, the main goal is to accomplish dimensionality reduction.

Dimensionality reduction reduces the impact of duplicate information in subsequent feature learning and allows for the conversion of multiple data attributes into a small number of data attributes without significantly sacrificing critical information.

Dimensionality reduction permits the conversion of numerous data attributes into a limited number of data attributes without appreciably compromising important information, and thus lessens the influence of redundant information in later feature learning. To prevent some larger data points from producing significant mistakes, the standardization process brings all data points' sizes into the same range. The following is the standardizing formula. Covariance matrix calculation: The covariance matrix of the data points is computed to determine the correlation between them [15].

$$X_{new} = \frac{X_i - \mu}{\sigma} \qquad (3)$$

Determine the primary components of the data by computing the eigenvalues of the covariance matrix and the associated eigenvectors.

$$Cov = [Cov_{21}Cov11\ Cov_{M2}Cov_{22}Cov12 \ldots Cov_{MM}Cov_{2M}Cov{:}\ 1M] \qquad (4)$$

To create a vector matrix, feature vectors are ordered in rows from top to bottom based on the magnitude of feature values; the order from top to bottom denotes diminishing relevance. The first n rows of the directional matrix create a new matrix when the data dimension is decreased to n dimensions.

3.3 Anomaly Classification and Detection

We outline the different elements of our neural network study in this section. Before describing the GNNs we use in our research and the explicit form of the AE loss function, we provide a quick overview of their conceptual structure.

3.3.1. Graph Neural Network

WSN anomaly classification uses GNN, which are models that can extract characteristics from graph-structured data. They extend CNNs built-in inductive biases, such as shared weights and local connections, to variable length and potentially non-Euclidean input. In the following paragraphs, we first provide a broad overview of the paradigm before going into more detail on the two particular forms that we employ in our work. The features of the i^{th} node at the l^{th} timesteps are represented by $h_i^{(l)}$ in the following (similar to a layer in the typical ANNs). $\mathcal{N}(i)$ refers to the set of nodes interconnected to node i, and $e_{ij}^{(l)}$ represents the characteristics of the edge joining nodes i and j. We assume that $l = 0$and $h_i^{(0)} = x_i$ for the input layer. MPNNs are made up of a graph reading layer and a message passing phase [16].

$$m_{ij}^{(l)} = M^{(l)}(h_i^{(l)}, h_j^{(l)}, e_{ij}^{(l)}) \quad (5)$$

This determines the message m_{ij} for the node-to-node edge. Graph convolutions get their name from the fact that the message function is typically a multilayer perceptron (MLP) that is shared by all of the edges. After calculating the messages between all connected nodes for each timestep (or layer), an aggregation function is used to update each node's features.

$$h_i^{(l+1)} = (h_i^{(l)}, \{m_{ij}^{(l)} | j \in \mathcal{N}(i)\}) \quad (6)$$

This vector can serve as the final output in supervised learning and be fed into an MLP or used to minimize the loss function. Nevertheless, graph-level readouts are not used in a GAE to maintain the graph structure until the output is finished. Typically, GAE are made to categorize nodes or edges, with an emphasis on learning local properties of large graphs. However, the network must learn both local properties and global graph structures because our objective is to classify tiny networks.

To resolve this, we incorporate an edge-reconstruction network into the decoder, which enables our network to learn graph structures by recreating the complete graph. The blue box indicates the boundaries of the edge-reconstruction network. The chapters that follow go into further depth about these. The encoder and decoder are both part of the network. To rebuild the multidimensional edge information, we use an edge reconstruction network in the decoder.

3.3.2. Autoencoder

Neural networks known as AE convert an input space to a bottleneck dimension, or the latent dimension, and then back to the input space. We combine the input node attributes with the multi-dimensional edge information using graph convolutions. Thus, the physics information encoded in our 3D edge feature is learned by our network. Edge-convolution, which has demonstrated exceptional performance in supervised learning scenarios, is used in the timestep till we reach the latent space.

Each edge feature is reconstructed individually from the shared block by these three structurally identical blocks. The edge information is reconstructed as three adjacency matrices using an inner product layer. The following paragraphs describe these three elements as well as the makeup of the loss function. The first layer uses an MLP, known as the edge function F_w, to construct a weighted graph convolution using the node and edge characteristics as input. This uses the edge features as input and translates them to a $m \times n$ dimension, where n indicates the dimension of the updated node features and m is the dimension of the input node. The component-wise multiplication of the form is carried out by

$$ab_{m_{ij}^{(1)}} = abF_e(e_{ij}) \times ab\tilde{h} \quad (7)$$

In Eq. (7), a and b are the matrix's indices. The input node characteristics, $h_j^{(0)}$, are repeated n times to create a $d^{ab}\tilde{h}$. The aggregation step adds up the index of the matrix after taking the mean of ${}^{ab}m_{ij}^{(1)}$over all neighboring nodes j:

$$bh_i^{(1)} = \sum_a mean_{j \in \mathcal{N}(i)}(\{{}^{ab}m_{ij}^{(1)}\}) \quad (8)$$

The edge convolution operation is the foundation of our architecture. The dimensions of the original and updated node characteristics are determined by two linear layers, Θ_w and Φ_w, which have the same input and output dimensions.

$$m_{ij}^{(l)} = \Theta_w(h_j^{(l)} - h_i^{(l)}) + \Phi_w(h_i^{(l)}) \quad (9)$$

$$ah_i^{(l+1)} = \max\{{}^a m_{ij}^{(l)}\} \quad (10)$$

$$j \in \mathcal{N}(i) \quad (11)$$

In order to reconstruct the edge features from the node features of the final edge convolution output, the edge-reconstruction network employs an inner product layer. For every edge, the inner product corresponds to the two node indices. The layer creates a symmetric $N \times N$ matrix, where N represents the amount of nodes in the graph, because our graphs are undirected. Therefore, its constituents are

$$\widehat{A} = h_i . h_j \quad (12)$$

$$L_{node} = \sqrt{\sum_{ia} \frac{(\hat{x} - x_i^a)^2}{N \times 5}} \quad (13)$$

Where $\hat{x}$ and x_i^a are the input and reconstructed node features, correspondingly, and a and i are the node-feature indexes.

$$L_{edge} = \sum_a \sqrt{\sum_{ij} \frac{(\widehat{A} - A_{ij}^a)^2}{N \times N}} \quad (14)$$

Where i and j are node indices and a is the edge-feature index. The input adjacency matrix is A_{ij}^a, while the rebuilt adjacency matrix is $\widehat{A}$.

$$L_{auto} = \lambda_{node} L_{node} + \lambda_{edge} L_{edge} \quad (15)$$

In order to give the integrated node characteristics the same weight as each individual edge feature—which carries more pertinent physics information—we select $\lambda_{node} = 0.3$ and $\lambda_{edge} = 1$.

4. EXPERIMENTAL OUTCOMES

As indicated in Table 1, the stimulation results of the ADWSN-HMLM algorithm are evaluated using the WSNDS dataset, which consists of 528161 samples and five classes. A suitable feature-selection technique was used to create an improved version of this dataset. The ML models examined in this article were trained and tested using both the original and improved versions of the WSN-DS dataset.

Table 1. Details of the dataset

Classes	**No. of Samples**
Normal	470066
Flooding	30049
Blackhole	15596
Grayhole	4612
Scheduling	7838
Total No. of Samples	**528161**

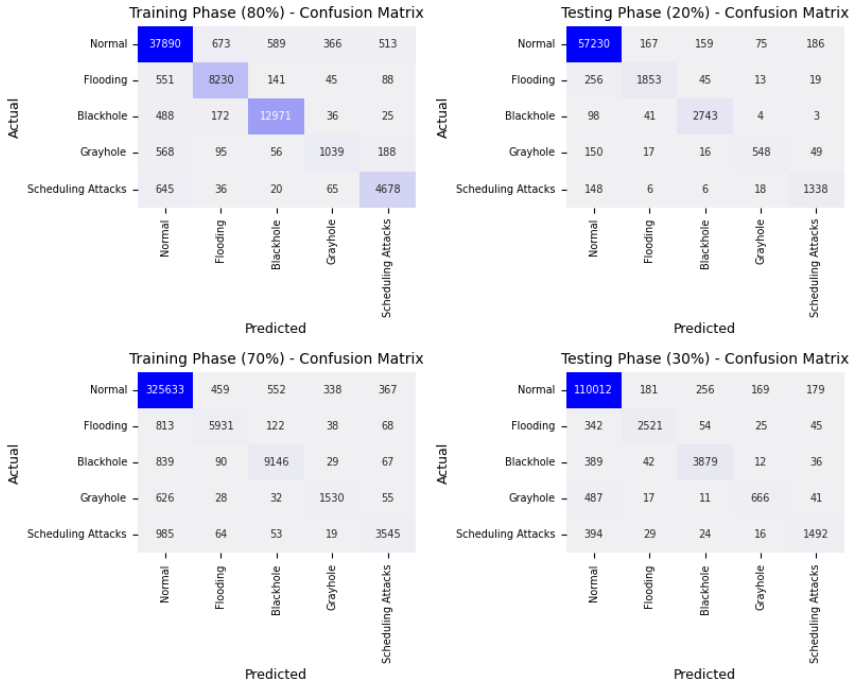

Figure 2. Confusion matrices of the ADWSN-HMLM method on 80:20 and 70:30 TRPH/TSPH

Fig. 3 shows the confusion matrices generated by the ADWSN-HMLM method on 80:20 and 70:30 of TRPH/TSPH. Five classes have been detected and classified based on the experimental value. Table 2 and Figure 3 analyze the anomaly recognition outcomes of the ADWSN-HMLM method using 80:20 and 70:30 TRPH/TSPH.

The outcomes highlighted that the ADWSN-HMLM method effectually identified five types. Under 80% of TRPH, the ADWSN-HMLM technique accomplishes average $accu_y$, $sens_y$, $spec_y$, and F_{score} of 97.83%, 97.90%, 96.93%, and 97.92%, respectively. Followed by 20% of TSPH, the ADWSN-HMLM method attained average $accu_y$, $sens_y$, $spec_y$, and F_{score} of 96.12%, 97.29%, 94.93% and 99.41%, correspondingly. With 70% of TRPH, the ADWSN-HMLM system obtains average $accu_y$, $sens_y$, $spec_y$, , and F_{score} of 96.14%, 95.80%, 97.94%, and 95.77%, correspondingly. Followed by 30% of TSPH, the ADWSN-HMLM method accomplished average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 95.03%, 95.04%, 96.63%, and 95.07%, correspondingly.

Table 2. Anomaly recognition outcome of ADWSN-HMLM method on 80:20 and 70:30 of TRPH/TSPH

Class	**$Accu_y$**	**$Prec_n$**	**$Reca_l$**	**F_{Score}**
Training Phase (80%)				
Normal	95.90	98.22	96.70	98.96
Flooding	96.76	97.59	97.16	96.87
Blackhole	97.83	96.83	98.76	98.76
Grayhole	96.62	98.84	89.88	89.45
Scheduling	95.95	87.92	78.63	76.86

Average	**97.83**	**97.90**	**96.93**	**97.92**
Testing Phase (20%)				
Normal	98.19	97.72	95.18	88.44
Flooding	98.12	96.86	94.68	86.36
Blackhole	88.54	89.43	79.68	87.93
Grayhole	97.33	97.65	88.45	79.85
Scheduling	86.42	78.56	77.87	96.74
Average	**96.12**	**97.29**	**94.93**	**99.41**
Training Phase (70%)				
Normal	97.81	96.07	98.41	96.52
Flooding	94.46	95.52	97.46	95.01
Blackhole	95.65	95.60	99.92	97.45
Grayhole	94.82	86.56	88.80	89.84
Scheduling	89.73	79.64	87.31	74.76
Average	**96.14**	**95.80**	**97.94**	**95.77**
Testing Phase (30%)				
Normal	96.48	95.05	96.68	95.73
Flooding	95.57	95.03	96.57	94.41
Blackhole	88.86	76.46	86.70	88.62
Grayhole	79.66	84.23	97.45	79.84
Scheduling	95.93	89.88	79.67	76.05
Average	**95.03**	**95.04**	**96.63**	**95.07**

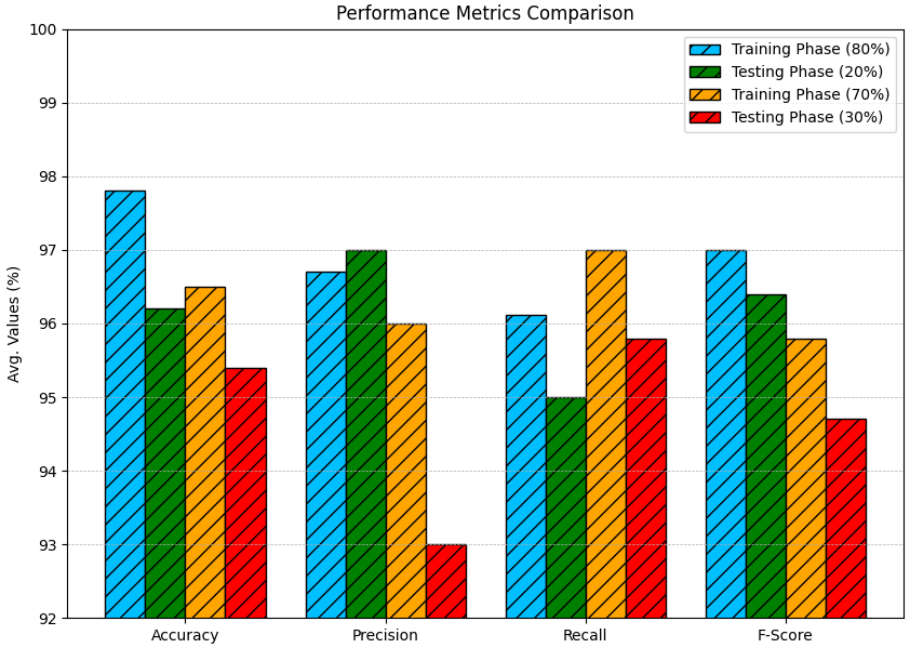

Figure 2. Average of HDBODR-DLCS technique at 80:30 and 70:30 of TRPH/TSPH

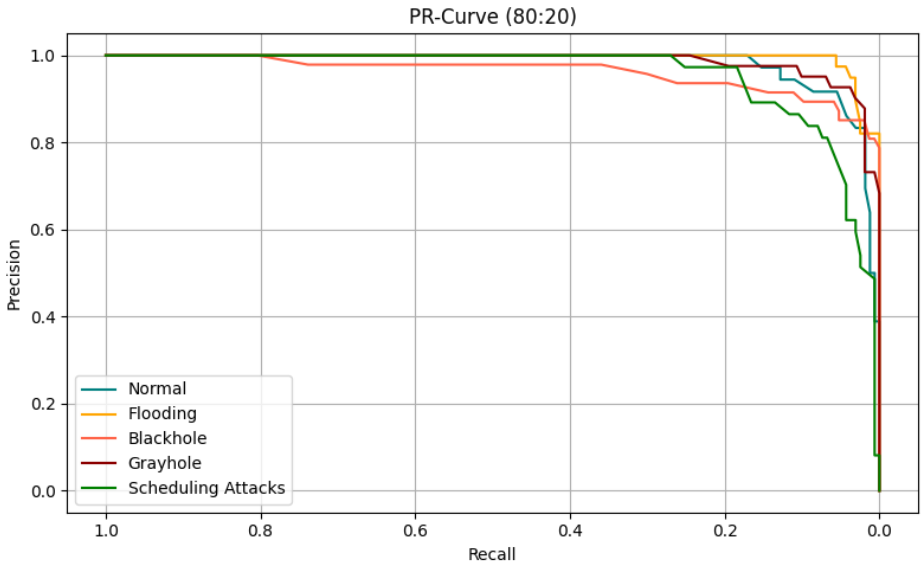

Figure 3. PR analysis of the ADWSN-HMLM approach on 80:20 of TRPH/TSPH

The results verify that, with respect to the PR curve in Figure 3, the ADWSN-HMLM approach under 80:20 of TRPH/TSPH consistently achieves high PR values throughout each class. These outcomes demonstrate how well the model can distinguish between different classes, underscoring its effectiveness in class detection. Additionally, we displayed ROC curves produced by the ADWSN-HMLM approach on 80:20 of TRPH/TSPH in Figure 4, which illustrates its capacity for class differentiation. This curve offers valuable insight into how the trade-off between TPR and FPR varies at different thresholds and classification epochs. The outcomes demonstrate the model's effectiveness in solving various classification problems by highlighting its classification performance across classes.

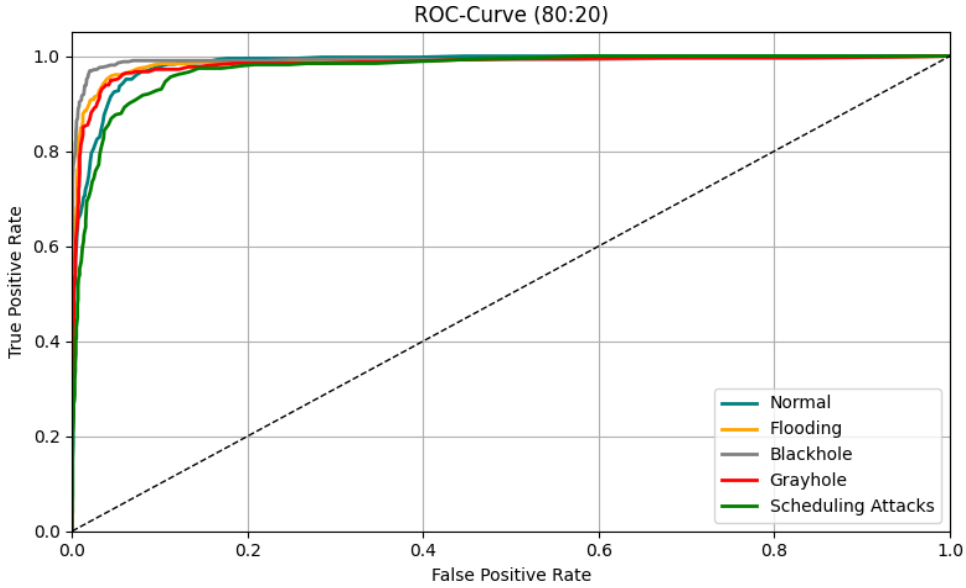

Figure 4. ROC analysis of the ADWSN-HMLM method at 80:20 of TRPH/TSPH

A comparison analysis ensured the improved performance of the ADWSN-HMLM approach, as illustrated in Figure 5. The outcomes show that the KNN, SVM, and ID-GOPA strategies have achieved unquestionably better results, while the LSTM model procedures have demonstrated the lowest performance. In the meantime, outcomes from the RKOA-AEID approach are almost optimum. Lastly, with improved $accu_y$ of 97.33%, $sens_y$ of 87.07%, $spec_y$ of 96.31%, and F_{score} of 97.85%, the ADWSN-HMLM approach demonstrates its superiority.

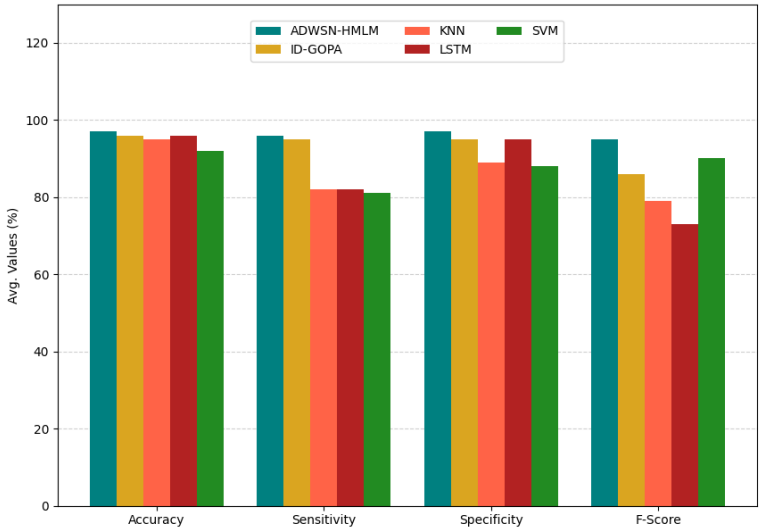

Figure 5. Comparative analysis of the ADWSN-HMLM method with other ML approaches

5. CONCLUSION

In this study, we presents an ADWSN-HMLM approach. The ADWSN-HMLM approach undergoes data preprocessing, feature extraction, detection, and classification. The benchmark IDS dataset is utilized to test the experimental results of the ADWSN-HMLM method. The simulation results showed that the ADWSN-HMLM approach outperformed other approaches. The infrastructure of WSN and the security issues they encounter are conveniently referenced in this study. Along with discussing the difficulties and suggested solutions for enhancing sensors' capacity to recognize threats, attacks, risks, and malicious nodes through their capacity to learn and self-develop using ML algorithms, this study also explores the potential benefits of ML algorithms in lowering the security costs of WSN across a number of domains. In addition to 98% for regular traffic, the detection accuracy for scheduling, grayhole, flooding, and blackhole assaults is 97.59%, 96.95%, 96.03%, and 97.05%, respectively. These findings demonstrate that the ADWSN-HMLM methodology can offer the WSN effective anomaly detection.

REFERENCES

[1] Mittal, M., De Prado, R.P., Kawai, Y., Nakajima, S. and Muñoz-Expósito, J.E., 2021. Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks. Energies, 14(11), p.3125.

[2] Haque, A., Chowdhury, N.U.R., Soliman, H., Hossen, M.S., Fatima, T. and Ahmed, I., 2023, September. Wireless sensor networks anomaly detection using machine learning: a survey. In Intelligent Systems Conference (pp. 491-506). Cham: Springer Nature Switzerland.

[3] Sharma, T., Balyan, A. and Singh, A.K., 2024. Machine learning-based energy optimization and anomaly detection for heterogeneous wireless sensor network. SN computer science, 5(6), p.751.

[4] Srivastava, A. and Bharti, M.R., 2023. Hybrid machine learning model for anomaly detection in unlabelled data of wireless sensor networks. Wireless Personal Communications, 129(4), pp.2693-2710.

[5] Mohan, S., Manke, A., Verma, S. and Baskar, K., 2024. Machine learning at the edge: GANs for anomaly detection in wireless sensor networks. In Enhancing Security in Public Spaces Through Generative Adversarial Networks (GANs) (pp. 305-317). IGI Global Scientific Publishing.

[6] Talukder, M.A., Khalid, M. and Sultana, N., 2025. A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction. Scientific Reports, 15(1), p.4617.

[7] Salman, E.H., Taher, M.A., Hammadi, Y.I., Mahmood, O.A., Muthanna, A. and Koucheryavy, A., 2022. An anomaly intrusion detection for high-density internet of things wireless communication network based deep learning algorithms. Sensors, 23(1), p.206.

[8] Altulaihan, E., Almaiah, M.A. and Aljughaiman, A., 2024. Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms. Sensors, 24(2), p.713.

[9] Mishra, S. and Jain, U., 2025. Secure IoT sensor networks through advanced anomaly detection with Kolmogorov–Arnold Networks (KANs). Microsystem Technologies, pp.1-11.

[10] Elsadig, M.A., 2023. Detection of denial-of-service attack in wireless sensor networks: A lightweight machine learning approach. IEEe Access, 11, pp.83537-83552.

[11] Gowdhaman, V. and Dhanapal, R., 2022. An intrusion detection system for wireless sensor networks using deep neural network. Soft Computing, 26(23), pp.13059-13067.

[12] Behiry, M.H. and Aly, M., 2024. Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods. Journal of Big Data, 11(1), p.16.

[13] Bukhari, S.M.S., Zafar, M.H., Abou Houran, M., Moosavi, S.K.R., Mansoor, M., Muaaz, M. and Sanfilippo, F., 2024. Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. Ad Hoc Networks, 155, p.103407.

[14] Omar, A. and Abd El-Hafeez, T., 2024. Optimizing epileptic seizure recognition performance with feature scaling and dropout layers. Neural Computing and Applications, 36(6), pp.2835-2852.

[15] Saheed, Y.K., Abdulganiyu, O.H. and Ait Tchakoucht, T., 2023. A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures. Journal of King Saud University-Computer and Information Sciences, 35(5), p.101532.

[16] Atkinson, O., Bhardwaj, A., Englert, C., Ngairangbam, V.S. and Spannowsky, M., 2021. Anomaly detection with convolutional graph neural networks. Journal of High Energy Physics, 2021(8), pp.1-19.