Advanced Optimization of RPL-IoT Protocol Using ML Algorithms

Mansour Lmkaiti[1], Ibtissam Larhlimi[2], Maryem Lachgar[3], Houda Moudni[4], Hicham Mouncif[5]
LIMATI Laboratory-Polydisciplinary Faculty, University Sultan Moulay Slimane, Morocco[1,2,3,5]
TIAD Laboratory, Faculty of Sciences and Technology , University Sultan Moulay Slimane, Morocco[4]

Abstract—**This study explores the transformative potential of machine learning (ML) algorithms in optimizing the Routing Protocol for Low-Power and Lossy Networks (RPL), addressing critical challenges in Internet of Things (IoT) networks, such as Expected Transmission Count (ETX), latency, and energy consumption. The research evaluates the performance of Random Forest, Gradient Boosting, Artificial Neural Networks (ANNs), and Q-Learning across IoT network simulations with varying scales (50, 100, and 150 nodes). Results indicate that tree-based models, particularly Random Forest and Gradient Boosting, demonstrate robust predictive capabilities for ETX and latency, achieving consistent results in smaller and medium-sized networks. Specifically, for 50-node networks, Neural Networks achieved the best performance with the lowest latency (2.43862 ms) and the best ETX (5.29557), despite slightly higher energy consumption. For 100-node networks, Q-Learning stood out with the lowest energy consumption (1.62973 J) and competitive ETX (2.70647), though at the cost of increased latency. In 150-node networks, Q-Learning again outperformed other models, achieving the lowest latency (0.68 ms) and energy consumption (2.21 J), though at the cost of higher ETX. Neural Networks excel in capturing non-linear dependencies but face limitations in energy-related metrics, while Q-Learning adapts dynamically to network changes, achieving remarkable latency reductions at the cost of transmission efficiency. The findings highlight key trade-offs between performance metrics and emphasize the need for algorithmic strategies tailored to specific IoT applications. This work not only validates the scalability and adaptability of ML approaches but also lays the foundation for intelligent and efficient IoT network optimization, laying the groundwork for future advancements in sustainable and scalable IoT networks.**

Keywords*—*IoT; RPL; machine learning; routing efficiency; energy consumption; expected transmission count; network optimization; Artificial Intelligence (AI)

I. Introduction

The Internet of Things (IoT) has revolutionized modern technological paradigms, enabling seamless connectivity and interaction between billions of devices, ranging from consumer electronics to industrial machinery [1]. This explosion of interconnected systems has catalyzed advancements in diverse fields, including smart cities, healthcare, agriculture, and industrial automation [2]. At the core of IoT systems lies the Routing Protocol for Low-Power and Lossy Networks (RPL), a pivotal framework designed to support the unique challenges of Wireless Sensor Networks (WSNs) operating under constrained resources [3]. Despite its widespread adoption, RPL's default mechanisms often fall short in optimizing key performance metrics such as ETX, latency, and energy consumption—metrics critical for ensuring scalability and reliability in large, dynamic IoT environments [2],[4].

The rapid growth of IoT devices has increased the complexity of network management, particularly in dynamic topologies with resource constraints. Traditional heuristic and rule-based optimization methods, while effective in static environments, lack the adaptability and precision needed to address the complex challenges of real-world IoT deployments. These limitations underscore the necessity for innovative solutions that leverage data-driven insights to optimize network behavior dynamically and efficiently [5].

Machine learning (ML), with its unparalleled ability to analyze complex datasets and extract actionable insights, emerges as a transformative solution for optimizing IoT networks. Unlike conventional methods, ML models can adapt to evolving network conditions, predict performance trends, and optimize resource allocation intelligently. Supervised learning techniques such as Random Forests and Gradient Boosting have demonstrated exceptional accuracy in predicting ETX and latency by effectively capturing feature interactions and avoiding overfitting. Similarly, Artificial Neural Networks (ANNs) excel in modeling non-linear dependencies within high-dimensional datasets, enabling precise energy consumption predictions. Reinforcement learning approaches like Q-Learning introduce an adaptive framework, allowing IoT networks to learn optimal routing policies through continuous interaction with the environment [6], [7].

This study delves into the application of ML techniques for enhancing RPL [26] performance in IoT networks, focusing on the evaluation of Random Forests, Gradient Boosting, ANNs, and Q-Learning across varying network sizes. By analyzing the scalability and adaptability of these models through extensive simulations, this work offers a comprehensive comparison of their strengths and limitations. The results demonstrate that ML-based optimization not only improves energy efficiency and reduces latency but also enhances network reliability, laying the groundwork for the development of intelligent, self-optimizing IoT systems.This research connects theory with practice, demonstrating how machine learning can revolutionize IoT network [28] management. The insights gained provide a roadmap for leveraging ML to overcome the critical challenges of energy efficiency, scalability, and adaptability in RPL-based IoT systems, thereby setting a benchmark for next-generation IoT deployments.The remainder of this paper is structured as follows: Section II presents related works, Section III formulates the problem statement, and Section IV describes the machine learning [29] algorithms applied. Section V discusses experimental results, and Section VI concludes the paper with key findings and future directions.

II. Related Work

The Internet of Things (IoT) [2] has emerged as a transformative domain, enabling interconnected devices to communicate, sense and transmit data across diverse environments. At the heart of this ecosystem lies the Routing Protocol for Low-Power and Lossy Networks (RPL), established by the Internet Engineering Task Force (IETF), which serves as a standardized protocol to facilitate efficient data exchange in IoT networks [4]. RPL effectively addresses key limitations of IoT devices, such as constrained computing power, limited memory capacity, and high energy consumption. However, optimizing RPL to enhance energy efficiency, reduce latency, and extend network lifetime remains an ongoing research challenge. In this context, machine learning (ML) [7] has emerged as a powerful tool for addressing these challenges, providing a robust framework for analyzing complex data patterns, enabling predictive insights, and supporting adaptive decision-making.

Supervised learning models, such as Random Forests and Gradient Boosting [19], have shown exceptional promise in predicting key network metrics like ETX and latency, leveraging ensemble approaches to improve routing decisions while avoiding overfitting. Similarly, Artificial Neural Networks (ANNs) [20], with their ability to model non-linear dependencies, have demonstrated potential in optimizing energy consumption and resource allocation in IoT environments. Reinforcement learning techniques, such as Q-Learning, introduce a dynamic approach to routing optimization, enabling IoT[24] systems to autonomously learn and adapt to changing network conditions while balancing exploration and exploitation. Empirical studies confirm the effectiveness of ML algorithms in reducing energy consumption while maintaining communication performance. Approaches such as dynamic sleep scheduling and intelligent data aggregation have been shown to significantly extend network lifetimes, while deep learning architectures have excelled in extracting intricate patterns from high-dimensional datasets, enhancing network performance.

The comparative evaluation of ML algorithms [6] highlights their unique capabilities, applications, strengths, and limitations, providing a comprehensive understanding of their utility in optimizing IoT networks. Random Forests [23], robust and interpretable ensemble methods, are particularly effective in avoiding overfitting and are widely applied in domains like astronomy and energy prediction [8], [9], [10], [11]. However, they can occasionally underperform compared to Gradient Boosting in tasks requiring higher precision [7]. Decision Trees, valued for their simplicity and transparency, are effective in applications such as medical diagnostics and crop disease classification [8], [9], [10], [12], [13], but their propensity to overfit makes them less reliable as standalone models [1], [6], [7], [14], [15]. Gradient Boosting, a sequential learning method, achieves high accuracy by iteratively correcting errors, making it well-suited for diverse tasks [8], [9], [10], [16], [17], though its computational demands and tuning complexity can pose challenges [1], [2], [4], [6], [18], [19], [20], [21], [22]. Q-Learning, a reinforcement learning technique, excels in dynamic decision-making environments such as IoT network optimization [8], [9], [10], while Neural Networks demonstrate unparalleled ability to handle high-dimensional and non-linear data patterns, making them indispensable for complex domains like image recognition and medical diagnostics [8], [9], [10], [23], [24]. However, their reliance on large datasets and high computational requirements can limit their application in resource-constrained settings [25].Together, these observations underscore the transformative potential of machine learning in IoT network optimization [27], emphasizing the importance of aligning algorithmic selection with specific application requirements. By dynamically evolving and adapting to real-time data, ML algorithms are positioned at the forefront of contemporary IoT [30] research, paving the way for intelligent, sustainable, and resilient network systems that address the pressing demands of scalability, efficiency, and adaptability.

III. Problem Statement

In this section, we formulate the problem of optimized routing RPL-based IoT network[30] considering the following metrics:ETX, the latency (LT) and the energy consumption (EC). The objective function integrating these criteria is defined as follows.

$$Minimize\ F = w_1.ETX + w_2.LT + w_3.EC \quad (1)$$

Where w_1, w_2 and w_3 are weights assigned to ETX, LT and EC respectively.

A. Define the Metrics

ETX measures the number of expected transmissions, including retransmissions, required to successfully deliver a packet over a link.

$$ETX_{ij} = \frac{1}{P_{ij}.P_{ji}}$$

Where P_{ij} is the probability of successful packet transmission from node i to node j, and P_{ji} is the probability of successful acknowledgment.
LT represents the time required for a packet to travel from the source to the destination.

$$LT_{ij} = d_{ij} + \sum_k ProcessingTime_k$$

Where d_{ij} is the propagation delay between nodes i and j, and the sum represents the processing delays at intermediate nodes.
EC is the of energy consumed to transmit a packet from the source to the destination.

$$EC_{ij} = TE_{ij} + \sum_k ProcessingEnergy_k$$

Where TE_{ij} is the energy consumed for transmission between nodes i and j, and the sum represents the energy consumed at intermediate nodes for processing.

B. Formulate the Constraints

The connectivity constraint ensures that the selected path maintains network connectivity.

$$\sum_{j \in N} x_{ij} = 1, \quad \forall\ i\ \in\ N$$

Where xij is a binary variable indicating whether the link between nodes i and j is part of the path (1) or not (0). The

Loop-Free constraint ensures that routing path does not exceed the available energy at any node.

$$\sum_{j \in N} x_{ij} = 1, \ \ \forall \ i \ \in \ N$$

The energy constraint ensures that the energy consumption does not exceed the available energy at any node.

$$EC_{ij} \ \leq \ E_i, \ \ \forall \ i \ \in \ N$$

Where E_i is the available energy at node i.

C. Optimization Problem Formulation

$$Minimize\, F = \sum_{i,j \in E} (w_1.ETX_{ij}+w_2.LT_{ij} + w_3.EC_{ij}).x_{ij}$$

Subject to:

$$\sum_{j \in N} x_{ij} = 1, \ \ \forall \ i \ \in \ N$$

$$x_{ij} \ + \ x_{ji} \ \leq \ 1, \ \ \forall \ i, j \ \in \ N$$

$$EC_{ij} \ \leq \ E_i, \ \ \forall \ i \ \in \ N$$

$$x_{ij} \in 0, 1$$

IV. Machine Learning Algorithms

The application of Machine Learning (ML) in IoT networks represents a paradigm shift in optimizing routing protocols, particularly the Routing Protocol for Low-Power and Lossy Networks (RPL) [3]. Unlike traditional heuristic-based approaches, ML algorithms provide data-driven solutions that dynamically adapt to the ever-changing conditions of IoT networks. By leveraging the vast amounts of data generated within IoT systems, these algorithms can predict network behaviors, optimize performance metrics, and enable intelligent decision making.

In the context of RPL optimization [26], ML techniques offer significant advantages in addressing critical metrics such as Expected Transmission Count, latency, and energy consumption. Various ML models have been successfully applied, each offering unique strengths and capabilities:

1) Random forests: This ensemble learning method combines the predictive power of multiple decision trees to deliver robust and accurate results. Random Forests are particularly effective in predicting ETX and latency while avoiding overfitting. Their ability to generalize well across diverse datasets makes them a reliable choice for IoT network optimization, especially in scenarios with high-dimensional data [8], [9], [10], [11], [29], [29].

2) Gradient boosting: As a sequential ensemble technique, Gradient Boosting iteratively refines weak models to achieve high accuracy. Its ability to capture complex interactions between features allows it to excel in predicting network performance metrics. Gradient Boosting has demonstrated remarkable efficiency in balancing ETX, latency, and energy consumption, making it a powerful tool for IoT network optimization [8], [9], [10], [16], [17].

3) Artificial Neural Networks (ANNs): Renowned for their capacity to model non-linear relationships, ANNs are well-suited for analyzing and predicting energy consumption patterns in IoT networks. Their multi-layered architecture enables them to learn intricate patterns and dependencies in the data, providing insights that drive more efficient routing and resource management [8], [9], [10], [23], [24].

4) Q-Learning: A reinforcement learning approach, Q-Learning introduces an adaptive mechanism for optimizing routing decisions. By interacting with the network environment, Q-Learning dynamically learns the best routing policies while balancing exploration and exploitation. This makes it highly effective in minimizing latency and energy consumption in dynamic IoT scenarios [8], [9], [10].

Machine learning algorithms stand out for their scalability and flexibility, making them well-equipped to handle the complexities of RPL-based networks. Simulations conducted on IoT networks of varying sizes (50, 100, and 150 nodes) have consistently highlighted the superior performance of ML models [27], [29] compared to traditional optimization techniques. For instance, Random Forests and Gradient Boosting have been shown to maintain balanced performance across metrics, while Q-Learning offers exceptional adaptability in dynamic environments.

ML algorithms enhance predictive capabilities and adaptive decision-making, optimizing key network metrics while fostering the development of intelligent and resilient IoT systems [26], [28]. The integration of these algorithms into RPL-based networks underscores their critical role in advancing the state-of-the-art in IoT optimization, paving the way for more efficient and sustainable IoT deployments. Through rigorous evaluation and continuous improvement, ML techniques are poised to revolutionize IoT network management, addressing the challenges of energy efficiency, latency reduction, and enhanced connectivity in real-world applications.

A. Dataset Configurations for Machine Learning Algorithms Simulations

In this section, we delve into the application of various machine learning algorithms—Random Forests, Gradient Boosting, Artificial Neural Networks (ANNs), and Q-Learning—to optimize routing within IoT networks. These algorithms are tested across different network configurations, with simulations conducted on data sets of varying sizes representing IoT networks with 50, 100, and 150 nodes. Each configuration is designed with specific training and testing data shapes to accurately reflect the complexity and scale of the simulated network environment. Table II provides an overview of the data set shapes used for each simulation size, ensuring a comprehensive evaluation of the machine learning models across diverse network scenarios.This table provides an overview of the datasets used to evaluate the performance of various machine learning algorithms across different network sizes. The number of features and instances in the training and testing sets reflects the complexity of the network configurations, ensuring a thorough analysis of each algorithm's predictive accuracy, adaptability, and scalability. This structured evaluation highlights the capacity of machine learning techniques to address the diverse challenges inherent in IoT networks, including energy efficiency, latency reduction, and routing optimization.

TABLE I. SHAPES OF DATA SETS FOR DIFFERENT SIMULATIONS

Simulation	Train Data Shape	Test Data Shape
50 nodes	(440, 7)	(110, 7)
100 nodes	(80, 3)	(20, 3)
150 nodes	(3539, 4)	(885, 4)

B. Implementation of Machine Learning Algorithms

This section presents a machine learning framework for minimizing transmission and energy costs in IoT networks. Algorithm 1 uses Q-Learning for multi-objective optimization, balancing exploration and exploitation with parameters like learning rate (α), discount factor (γ), and exploration rate (ϵ). Q-values are iteratively updated using ETX, latency, and energy consumption, ensuring robust and efficient routing policies.

Algorithm 1 employs a Random Forest model to predict ETX, latency, and energy consumption in IoT networks. It includes data preprocessing, dataset splitting, and training regressors for each metric. Performance is evaluated using R², MSE, and MAE, demonstrating the model's robustness for IoT optimization.

Algorithm 1 Random Forest for Multi-Objective Optimization

Input : Dataset with features: X;
Target variables: $Y_{ETX}, Y_{Latency}, Y_{Energy}$;
Number of estimators N_{trees};
Maximum depth of trees $MaxDepth$;
Metrics: $ETX, Latency\ (ms), Consumed\ Energy\ (J)$;

Output : Predicted values for metrics: $\hat{Y}_{ETX}, \hat{Y}_{Latency}, \hat{Y}_{Energy}$;
Model performance scores (e.g., R^2, MSE).

Begin Random Forest Algorithm

/* Data Preprocessing */
Split dataset into training and testing sets:
$X_{train}, X_{test}, Y_{train}, Y_{test} \leftarrow TRAIN\_TEST\_SPLIT(X, Y)$;

foreach $metric \in \{ETX, Latency, Energy\}$ **do**
  /* Train Random Forest Regressor for each metric */
  Initialize model: $RF_{metric} \leftarrow RANDOM\_FOREST(N_{trees}, MaxDepth)$;
  Train model: $RF_{metric}.FIT(X_{train}, Y_{train}[metric])$;

  /* Evaluate the model */
  $\hat{Y}_{metric} \leftarrow RF_{metric}.PREDICT(X_{test})$;
  Compute performance metrics: R^2, MSE, MAE;

/* Return Results */
Return $\hat{Y}_{ETX}, \hat{Y}_{Latency}, \hat{Y}_{Energy}$;
End Random Forest Algorithm

Algorithm 2 uses Gradient Boosting to optimize ETX, latency, and energy consumption in IoT networks. It involves preprocessing, dataset splitting, and training with hyperparameters ($N_{estimators}$, η, $MaxDepth$). Performance is evaluated using R², MSE, and MAE, ensuring accurate predictions and robust optimization.

Algorithm 2 Gradient Boosting for Multi-Objective Optimization

Input : Dataset with features: X;
Target variables: $Y_{ETX}, Y_{Latency}, Y_{Energy}$;
Number of estimators $N_{estimators}$;
Learning rate η;
Maximum depth of trees $MaxDepth$;
Metrics: $ETX, Latency\ (ms), Consumed\ Energy\ (J)$;

Output : Predicted values for metrics: $\hat{Y}_{ETX}, \hat{Y}_{Latency}, \hat{Y}_{Energy}$;
Model performance scores (e.g., R^2, MSE).

Begin Gradient Boosting Algorithm

/* Data Preprocessing */
Split dataset into training and testing sets:
$X_{train}, X_{test}, Y_{train}, Y_{test} \leftarrow TRAIN\_TEST\_SPLIT(X, Y)$;

foreach $metric \in \{ETX, Latency, Energy\}$ **do**
  /* Train Gradient Boosting Regressor for each metric */
  Initialize model: $GB_{metric} \leftarrow GRADIENT\_BOOSTING(N_{estimators}, \eta, MaxDepth)$;
  Train model: $GB_{metric}.FIT(X_{train}, Y_{train}[metric])$;

  /* Evaluate the model */
  $\hat{Y}_{metric} \leftarrow GB_{metric}.PREDICT(X_{test})$;
  Compute performance metrics: R^2, MSE, MAE;

/* Return Results */
Return $\hat{Y}_{ETX}, \hat{Y}_{Latency}, \hat{Y}_{Energy}$;
End Gradient Boosting Algorithm

Algorithm 3 employs a Neural Network to optimize ETX, latency, and energy consumption in IoT networks. The process includes data preprocessing (normalization and dataset splitting) and initializing neural networks for each metric with parameters such as layers (L), neurons ($N_{neurons}$), activation function (f), optimizer, learning rate (η), epochs (E), and batch size (B). Training utilizes backpropagation and gradient descent, with predictions evaluated via R², MSE, and MAE. By modeling nonlinear relationships, the algorithm ensures accurate predictions. The results include predictions and evaluation scores, demonstrating the neural network's effectiveness for IoT multi-objective tasks.

Algorithm 3 Neural Network for Multi-Objective Optimization

Input : Dataset with features: X;
Target variables: $Y_{ETX}, Y_{Latency}, Y_{Energy}$;
Neural network structure: Number of layers L, Neurons per layer $N_{neurons}$;
Activation function f;
Optimizer $Optimizer$ with learning rate η;
Number of epochs E, Batch size B;
Metrics: $ETX, Latency\ (ms), Consumed\ Energy\ (J)$;

Output : Predicted values for metrics: $\hat{Y}_{ETX}, \hat{Y}_{Latency}, \hat{Y}_{Energy}$;
Model performance scores (e.g., R^2, MSE).

Begin Neural Network Training

/* Data Preprocessing */
Normalize input features: $X \leftarrow NORMALIZE(X)$;
Split dataset into training and testing sets: $X_{train}, X_{test}, Y_{train}, Y_{test} \leftarrow TRAIN\_TEST\_SPLIT(X, Y)$;

foreach $metric \in \{ETX, Latency, Energy\}$ **do**
  /* Neural Network Initialization */
  Build model: $NN_{metric} \leftarrow INITIALIZE\_NN(L, N_{neurons}, f, Optimizer, \eta)$;
  Train model: $NN_{metric}.FIT(X_{train}, Y_{train}[metric], epochs = E, batch\_size = B)$;

  /* Evaluate the model */
  $\hat{Y}_{metric} \leftarrow NN_{metric}.PREDICT(X_{test})$;
  Compute performance metrics: R^2, MSE, MAE;

/* Return Results */
Return $\hat{Y}_{ETX}, \hat{Y}_{Latency}, \hat{Y}_{Energy}$;
End Neural Network Training

Algorithm 4 applies Q-Learning to optimize routing in IoT networks by minimizing ETX, latency, and energy consumption. The process begins with initializing the Q-table ($Q(s, a)$) to zero. Over $MaxEpisodes$, the algorithm explores the state space (S) using an ϵ-greedy policy to balance exploration and exploitation. Actions (a) are executed, resulting in state transitions (s') and rewards (r). Q-values are updated using the Bellman equation with learning rate (α), discount factor (γ), and maximum future rewards. After training, the optimal routing policy (π) is derived by selecting the action with the highest Q-value for each state. The algorithm outputs the optimized Q-table and routing policy, demonstrating Q-Learning's effectiveness in improving routing efficiency and reducing energy consumption and latency in IoT networks.

Algorithm 4 Q-Learning for Routing Optimization in IoT Networks

Input : State space S;
Action space A;
Learning rate α;
Discount factor γ;
Exploration probability ϵ;
Maximum episodes $MaxEpisodes$;
Metrics: $ETX, Latency\ (ms), Consumed\ Energy\ (J)$;

Output : Optimized Q-table Q;
Optimal routing policy π.

Begin Q-Learning Algorithm

/* Initialize Q-table */
$Q(s, a) \leftarrow 0, \forall s \in S, a \in A$;

for $episode \leftarrow 1$ **to** $MaxEpisodes$ **do**
  Initialize state $s \leftarrow INITIAL\_STATE()$;
  while ***not terminal state*** **do**
    /* Choose action using ε-greedy policy */
    $a \leftarrow$ **if** $random() < \epsilon$ **then** $RANDOM\_ACTION()$ **else** $\arg\max_a Q(s, a)$;
    Execute action a, observe reward r and next state s';

    /* Update Q-value */
    $Q(s, a) \leftarrow Q(s, a) + \alpha\big[r + \gamma \max_a Q(s', a) - Q(s, a)\big]$;
    $s \leftarrow s'$;

/* Extract optimal policy */
$\pi(s) \leftarrow \arg\max_a Q(s, a), \forall s \in S$;

/* Return Results */
Return Q, π;
End Q-Learning Algorithm

V. Results and Discussion

A. Experiment Environment

The tests were conducted on a device with an Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 8 GB RAM, and a 64-bit Windows system. Python was used to implement categorization methods on Jupyter Notebook, with libraries such as pandas (1.5.3), Pulp (2.6.0), Deap (1.3.1), and seaborn. Dependencies and tools were managed using Anaconda, which facilitates the implementation and management of machine learning algorithms.

B. Performance of Algorithms Across All Simulations

This study simulated IoT networks with 50, 100, and 150 nodes to evaluate the impact of machine learning algorithms on network optimization. The algorithms effectively predicted ETX, latency, and energy consumption, allowing for performance comparisons. Tables II, III, and IV highlight the potential of machine learning in optimizing IoT networks and driving future advancements.

TABLE II. Comparison of Performance of Algorithms for Simulation 50

Algorithm	ETX	Latency (ms)	Energy Consumption (J)
Random Forest	5.63656	2.63308	2.62552
Decision Trees	5.34587	2.52793	2.63175
Gradient Boosting	5.56853	2.64258	2.61170
Neural Networks	5.29557	2.43862	2.72135
Q-Learning	5.53092	2.65587	2.70280

TABLE III. Comparative Performance of Algorithms on the 100-node Simulation

Algorithm	ETX	Latency (ms)	Energy Consumption (J)
Random Forest	2.84176	64.19442	3.22043
Gradient Boosting	2.72260	63.31554	3.19787
Decision Trees	2.76172	66.95224	3.37378
Neural Networks	2.88632	43.04571	2.89517
Q-Learning	2.70647	1507.45000	1.62973

TABLE IV. Comparative Performance of Algorithms on the 150-nodes Simulation

Algorithm	ETX	Latency (ms)	Energy Consumption (J)
Random Forest	5.39482	2.51740	2.78374
Decision Trees	5.33565	2.44747	2.76888
Gradient Boosting	5.14449	2.83902	2.77862
Neural Networks	5.84716	2.52902	2.95446
Q-Learning	7.50000	0.68000	2.21000

C. Results of 50 Nodes

1) Random forest: The results in Fig. 1 reveal high accuracy for ETX and Latency, closely aligning with ideal predictions, but show challenges in predicting Consumed Energy, with greater deviations from actual values. This indicates a need for further tuning or advanced techniques, such as feature engineering or ensemble methods, to enhance energy prediction accuracy. The analysis highlights the model's strengths while identifying areas for improvement.

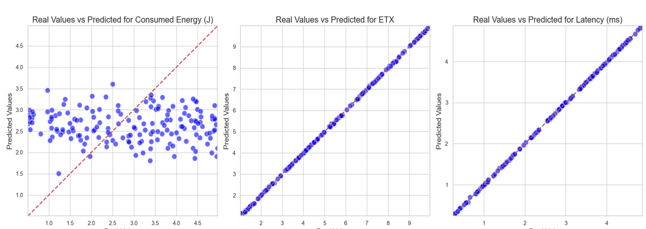

Fig. 1. Actual vs. Predicted values for Regression Analysis (RF).

2) Decision trees: Fig. 2 shows high accuracy for ETX and Latency but scattered deviations for Consumed Energy. Refinement through feature engineering, alternative algorithms, or more energy-focused data is needed for broader optimization.

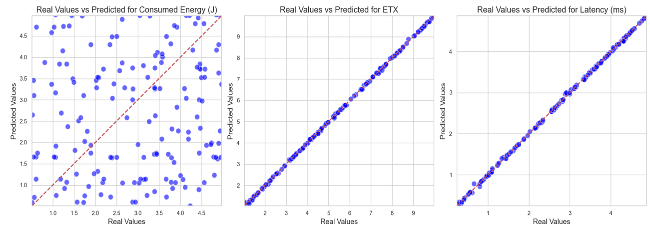

Fig. 2. Actual vs. Predicted values for Regression Analysis (DT).

3) Gradient boosting: The plots in Fig. 3 shows accurate ETX and Latency predictions, with tight clustering near the ideal line. Scattered Consumed Energy points suggest the need for refinement through feature engineering, non-linear models, or more data to improve energy prediction. Strengths and improvement areas are evident.

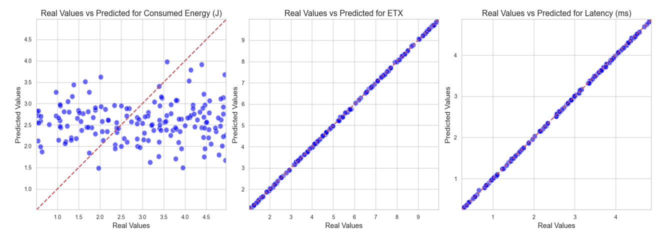

Fig. 3. Actual vs. Predicted values for Regression Analysis (GB).

Gradient Boosting curves in Fig. 4, 5 and 6 for a 50-node network show the trade-off between precision and recall as thresholds change. The F1 score peaks at the optimal threshold, balancing both. This flexibility allows optimization based on system priorities, enhancing detection or reliability.

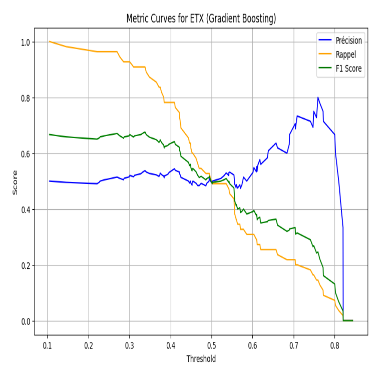

Fig. 4. Curves of ETX metric.

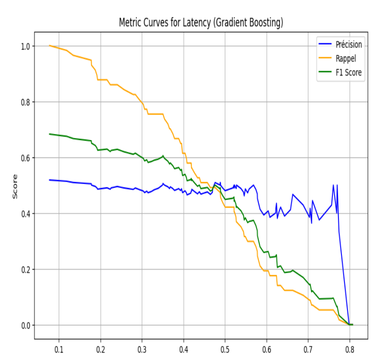

Fig. 5. Curves of the Latency metric.

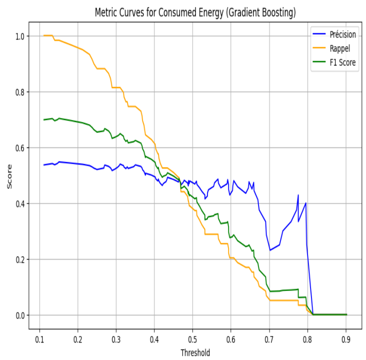

Fig. 6. Curves of the consumed energy metric.

4) Neural networks: The analysis in Fig. 7 shows strong ETX and Latency predictions but scattered Consumed Energy results. Minor ETX and Latency deviations suggest the need for better tuning, feature engineering, or more data to capture complex patterns. Strengths are evident, with areas for improvement noted.

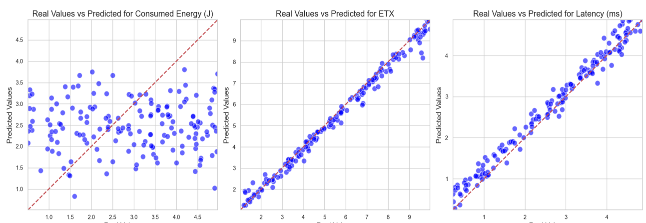

Fig. 7. Actual vs. Predicted values for Regression Analysis (Neural Networks).

5) Q-Learning: The confusion matrix in Fig. 8 and Fig. 9 shows that Q-learning effectively predicting low ETX and latency but struggling with consumed energy (Fig. 10) due to misclassifications. Refinement is needed to improve energy prediction accuracy.

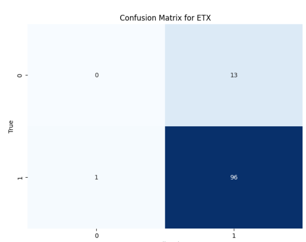

Fig. 8. Confusion Matrix of Etx.

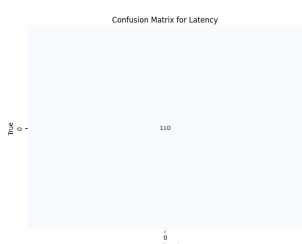

Fig. 9. Confusion Matrix of latency.

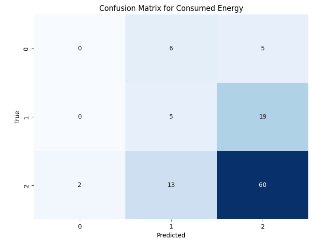

Fig. 10. Confusion Matrix of consumed energy.

6) Evolution of metrics: The graphs in Fig. 11 and Fig.12 shows ETX decreasing rapidly, Latency remaining stable, and Energy Consumption leveling off higher than ETX. This reflects a trade-off, with the system stabilizing at good connectivity, moderate latency, and low energy use.

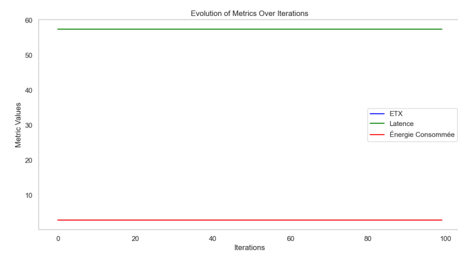

Fig. 11. Evolution of metrics over iterations.

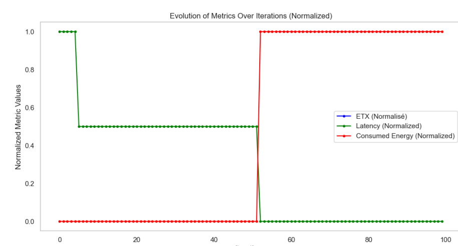

Fig. 12. (a) and (b) Evolution of Metrics over iterations (normalized).

D. Results of 100 Nodes

1) Random forest: The Random Forest model in Fig. 13 performs well for ETX and Latency with accurate clustering but struggles with Consumed Energy. Adding energy-specific features or advanced models could improve balance across metrics.

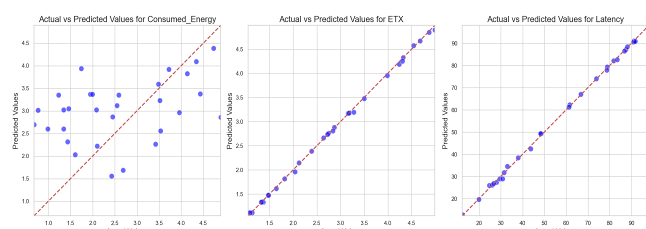

Fig. 13. Actual vs. Predicted values for Regression Analysis (RF).

2) Decision trees: The Decision Tree model n Fig. 14 performs well for ETX and Latency in a 100-node simulation but struggles with Consumed Energy due to non-linear limitations. Ensemble methods or energy-specific features could improve its balance across metrics.

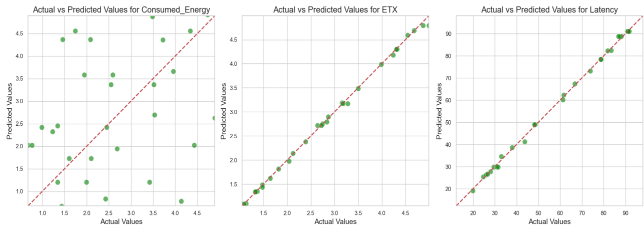

Fig. 14. Actual vs. Predicted values for Regression Analysis (DT)

3) Gradient boosting: Gradient Boosting excels in Fig. 15 with ETX and Latency but shows variance in Energy predictions, requiring refinements or energy-specific features for balanced performance.

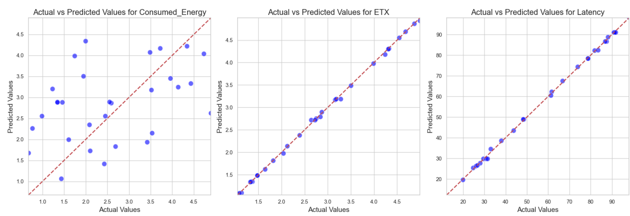

Fig. 15. Actual vs. Predicted values for Regression Analysis (GB).

4) Neural networks: The Neural Network performs in Fig. 16 well for Latency and moderately for ETX, but struggles with Consumed Energy, showing significant spread. Improvements may require tuning, targeted features, or hyperparameter optimization to achieve balanced accuracy across metrics.

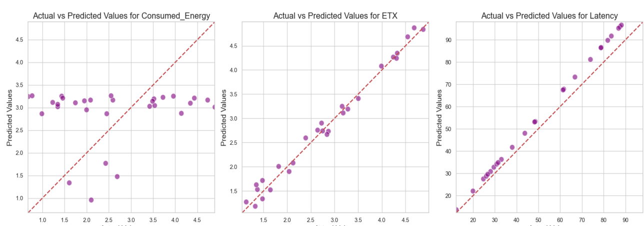

Fig. 16. Actual vs. Predicted values for Regression Analysis (Neural Networks).

The ROC curves and AUC in Fig. 17 values (0.87–0.92) show strong Neural Network performance in multiclass classification, with class 1 achieving the highest AUC (0.92). However, the lowest AUC (0.87) for class 0 indicates a performance gap. Model tuning or addressing class imbalances could improve discrimination for class 0.

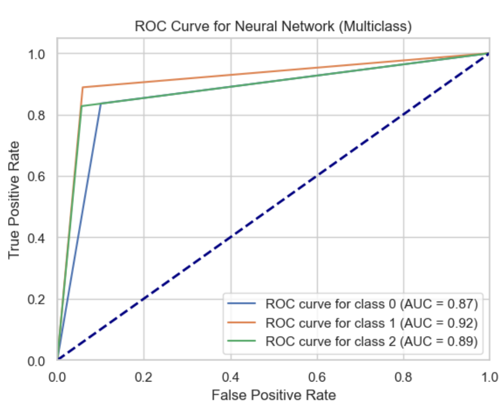

Fig. 17. ROC Curves of Neural Networks

5) Q-Learning: Fig. 18,19 and 20 shows that the model mainly classifies energy and ETX as "Medium," suggesting a learning bias. For latency, some distinction between "Medium" and "High" is observed, but no "Low" classifications appear, indicating possible limitations in detecting lower values.

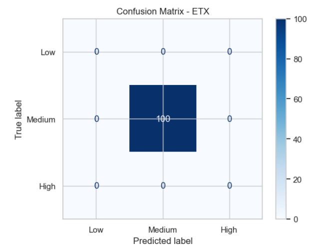

Fig. 18. Confusion Matrix of Q-learning(ETX).

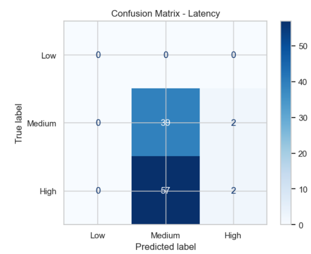

Fig. 19. Confusion Matrix of Q-learning(Latency).

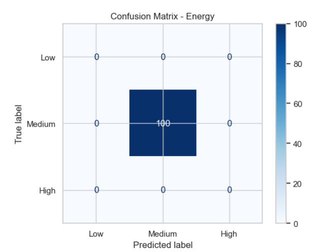

Fig. 20. Confusion Matrix of Q-learning(Consemd Energy).

E. 150 Nodes

1) Random forest: In the 150-node simulation, Fig. 21 show the Random Forest model excels in ETX and Latency, with predictions tightly clustered around the ideal line, reflecting effective feature representation. However, significant scatter in Consumed Energy predictions highlights challenges with complex patterns. Adding granular energy features or advanced ensemble methods could improve performance.

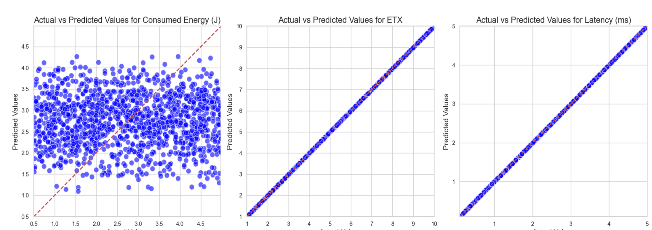

Fig. 21. Actual vs Predicted values for Regression Analysis (RF).

2) Gradient boosting: Fig. 22 shows that the Gradient Boosting model demonstrates high accuracy for ETX and Latency, with predictions closely aligning with the ideal line in a 150-node network, reflecting well-represented features. However, broad scatter in Consumed Energy predictions highlights challenges in modeling complexities, suggesting the need for energy-focused features or ensemble techniques.

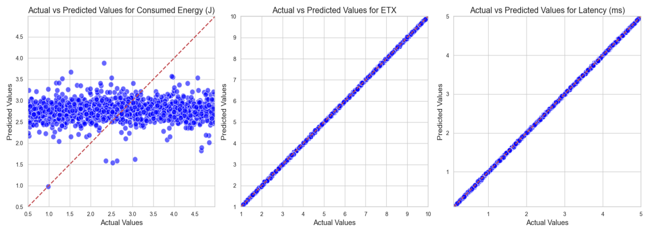

Fig. 22. Actual vs. Predicted values for Regression Analysis (GB).

3) Decision trees: Fig. 23 shows that the Decision Tree model performs well for ETX and Latency, with predictions aligning closely with the ideal line in a 150-node simulation, indicating ease of generalization. However, it struggles with Consumed Energy, showing broad scatter around the ideal line. Employing ensemble methods like Random Forest could enhance energy predictions.

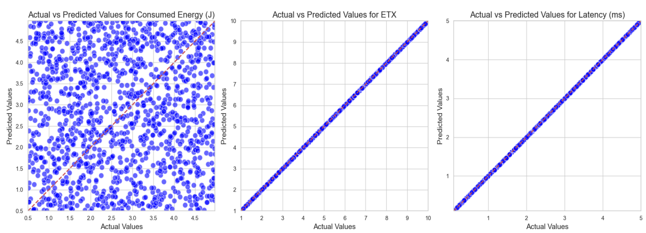

Fig. 23. Actual vs. Predicted values for Regression Analysis (DT).

4) Neural networks: Fig. 24 shows that the Neural Network model performs well for ETX and Latency, closely aligning with the ideal line in a 150-node network. However, its weak performance in Consumed Energy, marked by constant predictions, suggests oversimplification. Refining features, tuning the architecture, or applying regularization could improve sensitivity to energy variations.

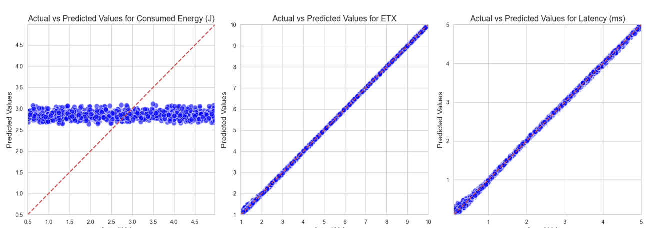

Fig. 24. Actual vs. Predicted values for Regression Analysis neural network.

5) Q-Learning: Fig. 25,26 and 27 shows that the confusion matrices indicate that the model excels at predicting the “No” class for ETX, Latency, and Consumed Energy, effectively identifying unaffected tasks. However, it struggles with the “Yes” class, potentially due to class imbalance or difficulty in capturing subtle differences. Further investigation is needed to identify the cause and enhance performance.

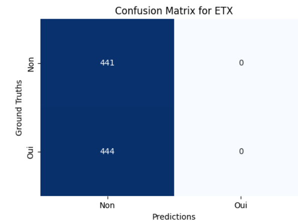

Fig. 25. Confusion Matrix of Q-learning(ETX).

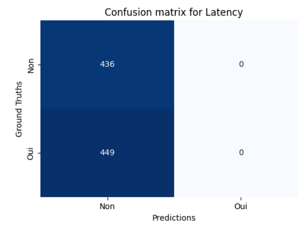

Fig. 26. Confusion Matrix of Q-learning(Latency).

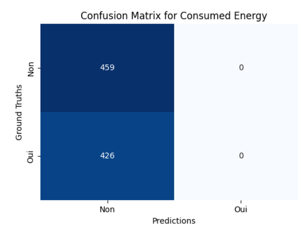

Fig. 27. Confusion Matrix of Q-learning(Consumed Energy).

Fig. 28, 29, and 30 compare ML algorithms in IoT based on ETX and latency, and energy consumption for 50, 100, and 150 nodes. Low ETX reflects better transmission efficiency. In 50-node networks, one algorithm shows high ETX, indicating challenges, while larger networks reveal variations, with some optimizing retransmissions. Latency is generally low in smaller networks, though inefficiencies appear for some algorithms. In larger networks, certain algorithms maintain low latency, while others increase due to overhead or longer paths. Energy consumption stabilizes in larger networks for most algorithms, though some struggle with scalability, highlighting differences in efficiency.

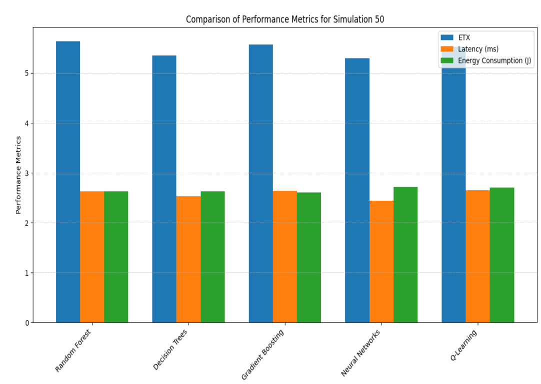

Fig. 28. Comparison for all simultaion 50,100,150 nodes with ML Algorithms.

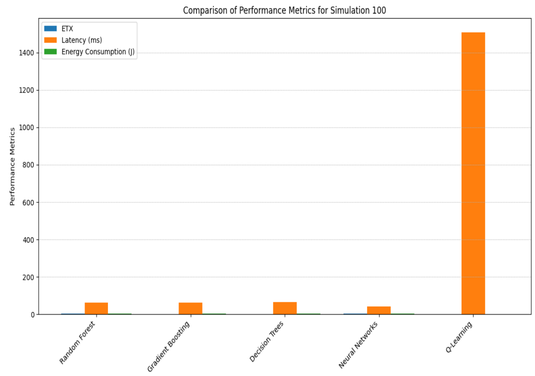

Fig. 29. Comparison for all metrics in 50 nodes with ML Algorithms.

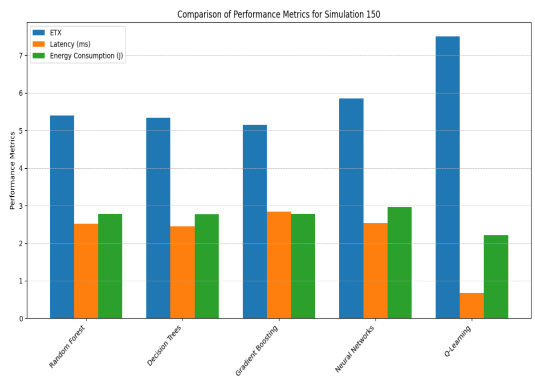

Fig. 30. Comparison for all metrics in 150 nodes with ML Algorithms.

VI. Discussion of the Results

This study provides a detailed evaluation of machine learning algorithms for optimizing IoT networks using the RPL protocol, focusing on ETX, latency, and energy consumption in simulations of 50, 100, and 150 nodes. Tree-based models like Random Forest and Gradient Boosting consistently demonstrated strong performance. For instance, Random Forest achieved an ETX of 5.63656, latency of 2.63308 ms, and energy consumption of 2.62552 J in the 50-node simulation, excelling in balancing accuracy and efficiency. Gradient Boosting performed well in 100-node simulations, with an ETX of 2.72260 and latency of 63.31554 ms, though energy predictions require further refinement.Energy consumption remains a challenging metric across all models. Neural Networks struggled significantly in the 150-node simulation, emphasizing the need for advanced architectures and tailored features. Q-Learning showed adaptability with low latency (0.68 ms) in the 150-node simulation but at the cost of higher ETX (7.50000), illustrating trade-offs between adaptability and transmission efficiency.These findings highlight the strengths of Random Forest and Gradient Boosting across multiple metrics, while Neural Networks and Q-Learning excel in specialized scenarios. The findings emphasize the need for algorithm selection based on network requirements and suggest exploring hybrid models to balance performance.While Q-Learning achieves low latency, it has a higher ETX, leading to potential inefficiencies in data transmission. Similarly, Neural Networks require significant computational resources, limiting their deployment in energy-constrained IoT environments.Incorporating real-world constraints into simulations could further enhance the practical applicability of these approaches, enabling more tailored optimization strategies for diverse IoT configurations.

VII. Conclusion

This study investigates how machine learning algorithms can optimize the RPL protocol in IoT networks. It focuses on three key performance metrics:ETX, latency, and energy consumption. By simulating networks of varying scales (50, 100, and 150 nodes), the research comprehensively evaluated the capabilities and limitations of several algorithms, including Random Forest, Gradient Boosting, Artificial Neural Networks (ANNs), and Q-Learning.The findings revealed that tree-based models, such as Random Forest and Gradient Boosting, excel in robustness and adaptability, showing exceptional predictive performance for ETX and latency in small and medium-sized networks. However, their energy consumption predictions require improvements, such as advanced feature engineering and enhanced ensemble techniques. Artificial Neural Networks, while capturing nonlinear dependencies effectively, struggled with energy consumption metrics in larger networks, underscoring the need for refined architectures and expanded datasets. Conversely, Q-Learning demonstrated remarkable adaptability, achieving significant latency reductions in larger networks, albeit at the cost of higher ETX, illustrating trade-offs between adaptability and transmission efficiency.A key insight from this research is the inherent trade-offs needed to address IoT network constraints. No single algorithm excels across all metrics, highlighting the necessity of hybrid approaches that combine the strengths of multiple models. For instance, integrating tree-based models for robustness with reinforcement learning techniques like Q-Learning for adaptability could lead to more efficient solutions in dynamic IoT environments. Beyond technical findings, this study establishes a foundation for future research, incorporating real-world constraints such as hardware limitations, dynamic network conditions, and application-specific requirements. Additionally, analyzing algorithm performance in real-time scenarios can refine their practical applicability and expand their utility.In conclusion, this work establishes a foundational methodology for IoT network optimization. By combining advanced simulations, machine learning models, and hybrid frameworks, this study paves the way for a new generation of intelligent, sustainable, and scalable IoT systems. The insights gained provide a robust foundation for future innovations in IoT networks, addressing challenges in performance optimization, energy efficiency, and resource management.

References

[1] S. Shaharuddin, K. N. A. Maulud, S. A. F. S. A. Rahman, A. I. C. Ani, and B. Pradhan, "The role of IoT sensors in smart building contexts for indoor fire hazard scenarios: A systematic review of interdisciplinary articles," *Phys. Rep. Rev. Sec. Phys. Lett.*, vol. 22, p. 10803, 2023.

[2] P. M. R., V. H. S., and S. J., "Holistic survey on energy-aware routing techniques for IoT applications," *Phys. Rev. E*, vol. 213, p. 103584, 2023.

[3] M. Lachgar, I. Larhlimi, A. Darif, H. Ouchitachen, and H. Mouncif, "Maximization of lifetime in wireless sensor networks using pattern search algorithm," in *Proc. Artif. Intell. Green Comput.*, vol. 806, pp. 138–148, 2023.

[4] M. Osman, J. He, N. Zhu, and F. M. M. Mokbal, "An ensemble learning framework for the detection of RPL attacks in IoT networks based on the genetic feature selection approach," *Phys. Rev. E*, vol. 152, p. 103331, 2024.

[5] S. Hakemi, M. Houshmand, and E. KheirKhah, "A review of recent advances in quantum-inspired metaheuristics," *Evol. Intell.*, 2022.

[6] H. Zou, S. Zeng, C. Li, and J. Ji, "A survey of machine learning and evolutionary computation for antenna modeling and optimization: Methods and challenges," *J. Eng. Appl. AI*, vol. 138, p. 109381, 2024.

[7] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *J. Stat. Mech. Theory Exp.*, vol. 110, p. 104743, 2022.

[8] H. Zou, S. Zeng, C. Li, and J. Ji, "Visualized simulating and improving particle swarm optimization," *J. Syst. Simul.*, 2007.

[9] M. Gabidolla and M. A. Carreira-Perpinan, "Low overhead routing in a lightweight routing protocol," in *Proc. 3rd Int. Conf. Artif. Intell. Comput. Vis. (AICV)*, 2023.

[10] K. Gao and Y. Wang, "A novel algorithm of machine learning: Fractional gradient boosting decision tree," in *Lecture Notes Inst. Comput. Sci. Soc. Inform. Telecommun. Eng.*, vol. 446, pp. 735–748, 2022. DOI: 10.1007/978-3-031-18123-8_58.

[11] F. D. R., F. D., and C. P., "Provenance-enabled packet path tracing in the RPL-based Internet of Things," *Sci. Rep.*, vol. 173, p. 107189, 2020.

[12] M. Gabidolla and M. A. Carreira-Perpinan, "Pushing the envelope of gradient boosting forests via globally optimized oblique trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 285–294, 2022. DOI: 10.1109/CVPR52688.2022.00038.

[13] F. Sadikoglu, B. Sekeroglu, and D. A. Ewuru, "Performance analysis of machine learning algorithms for medical datasets," in *Lecture Notes Netw. Syst.*, vol. 610, pp. 514–521, 2023. DOI: 10.1007/978-3-031-25252-5_68.

[14] S. A. Changazi, A. D. Bakhshi, M. Yousaf, S. M. Mohsin, S. M. A. Akber, M. Abazeed, and M. Ali, "Optimization of network topology robustness in IoTs: A systematic review," *IEEE Trans. Vis. Comput. Graph.*, vol. 250, p. 110568, 2024.

[15] A. Seyfollahi, M. Moodi, and A. Ghaffari, "MFO-RPL: A secure RPL-based routing protocol utilizing moth-flame optimizer for IoT applications," *Eur. Phys. J. B*, vol. 82, p. 103622, 2022.

[16] S. Narayana, L. Chennagiri, B. D. Kumar, S. K. R. Mallidi, and T. S. R. Sai, "Prediction of COVID-19 victim's well-being using extreme gradient boost algorithm," in *Proc. IEEE Int. Conf. Electron. Comput. AI*, pp. 958–963, 2023. DOI: 10.1109/ICECAA58104.2023.10212406.

[17] J. Zhang, Y. Wang, and L. Wu, "Stochastic particle swarm optimization algorithm," *Jisuanji Gongcheng Comput. Eng.*, 2006.

[18] Y. Yang, "Adaptive switching and routing protocol design and optimization in Internet of Things based on probabilistic models," *Phys. Rev. E*, vol. 5, pp. 204–211, 2024.

[19] B. A. Begum and S. V. Nandury, "Data aggregation protocols for WSN and IoT applications – A comprehensive survey," *Physica A*, vol. 35, pp. 651–681, 2023.

[20] S. Fortunato and M. Barthelemy, "An empirical assessment of ensemble methods and traditional machine learning techniques for web-based attack detection in Industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 103, no. 119, pp. 36–41, 2023.

[21] Š. Subelj and M. Bajec, "Towards developing a machine learning-metaheuristic-enhanced energy-sensitive routing framework for the Internet of Things," *Phys. Rev. E*, vol. 152, pp. 103331, 2024.

[22] A. Wakili, S. Bakkali, and A. E. H. Alaou, "Machine learning for QoS and security enhancement of RPL in IoT-enabled wireless sensors," *J. Sintl.*, vol. 5, pp. 100289, 2024.

[23] Z. Wen, H. Liu, J. Shi, Q. Li, B. He, and J. Chen, "ThunderGBM: Fast GBDTs and random forests on GPUs," *Mach. Learn. J.*, vol. 21, 2020.

[24] S. Narayana and V. Gopal, "Optimized routing in RPL-based IoT networks using machine learning," *IoT Edge Comput.*, vol. 15, pp. 512–523, 2023.

[25] P. S. Nandhini, S. Kuppuswami, and S. Malliga, "Energy-efficient thwarting rank attack from RPL-based IoT networks: A review," *J. MAPTR*, vol. 81, pp. 694–699, 2023.

[26] F.Medjek, D.Tandjaoui, and N. Djedjig, I.Romdhani "Multicast DIS attack mitigation in RPL-based IoT-LLNs," in *J.JISA*, 2021.

[27] R. Krishna and K. V. Prema, "Soybean crop disease classification using machine learning techniques," in *Proc. IEEE Int. Conf. Distrib. Comput. VLSI Electr. Circuits Robot.*, 2020.

[28] K. R. Venugopal and M. S. Roopa, "Intrusion detection model for IoT networks using graph convolution networks (GCN)," in *Proc. ICT Intell. Syst.*, vol. 361, pp. 1–12, 2023.

[29] S. Choudhary and S. Choudhary, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications," *Stud. Comput. Intell.*, 2019.

[30] P. S. Nandhini, S. Malliga, and V. Kuppuswami, "A systematic review of IoT routing algorithms based on performance criteria," *IoT J.*, 2023.