ML-Optimized RPL: Advanced Routing Protocol for Wireless Smart Grid

S Mahaboob Subhahan, Dr R Rajasekar
[1]*PG Scholar, Department of CSE, JNTUACEA, Anantapuramu, India*
[2]*Professor, Department of CSE, JNTUACEA, Anantapuramu, India*

Abstract-The Routing Protocol for Low Power and Lossy Networks (RPL) has become the de facto standard for routing in resource-constrained wireless networks, such as those found in smart grid and IoT environments. However, traditional RPL suffers from static decision-making, limited adaptability to dynamic conditions, and suboptimal parent selection strategies. This paper proposes ML-RPL, a machine learning-enhanced routing protocol that integrates a Naive Bayes classifier into RPL's parent selection mechanism. Unlike conventional approaches that rely on fixed metrics like ETX or hop count, ML-RPL leverages real-time network features— including RSSI, MAC losses, throughput, queue utilization, and node density—to estimate the probability of successful packet delivery through each neighbor. The routing decision is made by selecting the parent with the highest predicted reliability while maintaining compatibility with standard RPL control messages. A Python-based simulation environment was developed to evaluate the proposed approach across networks with 50, 100, and 200 nodes. Results demonstrate that ML-RPL outperforms standard RPL and RPL+ in terms of Packet Delivery Ratio (PDR) and end-to-end delay, especially under high-traffic and high-density conditions. The findings highlight ML-RPL's potential to enable intelligent, adaptive, and efficient routing in low-power wireless networks.

Keywords- Machine Learning, RPL, Wireless Sensor Networks, Low Power and Lossy Networks, Naive Bayes, Smart Grid Communication, Routing Protocols, Packet Delivery Ratio, End-to-End Delay, Network Optimization

I. INTRODUCTION

Communication technologies have become integral to the modernization of electrical infrastructure, particularly in the deployment of Smart Grids. These next- generation grids leverage advanced communication systems to ensure efficient, secure, and reliable grid operations. By incorporating devices capable of real-time communication, utility providers gain greater visibility and control over grid components, enabling smarter energy management and automation.

Smart grid communication architectures are typically divided into three key segments: the Home Area Network (HAN), Neighborhood Area Network (NAN) (also referred to as the Field Area Network (FAN)), and the Wide Area Network (WAN). HANs support communication among sensors and smart appliances within homes. These connect to smart meters, which transmit collected data across the NAN. NANs serve as intermediate networks connecting multiple smart meters, grid devices such as line sensors, smart switches, and EV chargers to centralized Data Aggregation Points (DAPs) or collectors. WANs link these DAPs with the utility control centers, ensuring end-to-end data flow and centralized command.

Among these, NANs are especially significant, as they carry substantial volumes of data and disseminate control signals across large areas. These networks may consist of thousands of nodes, and as such, reliable communication and efficient routing become primary concerns.

Both wired and wireless technologies are deployed in NANs, but wireless solutions are increasingly favored for their scalability, cost-effectiveness, and ease of deployment. Predominantly, two categories of wireless technologies are used:

1. RF Mesh Networks (e.g., IEEE 802.15.4, IEEE 802.11s), and
2. Low Power Wide Area Networks (LPWAN) such as Sigfox, LoRa, and NB-IoT.

A major challenge in mesh-based wireless networks lies in routing. The Routing Protocol for Low-Power and Lossy Networks (RPL) is a standardized protocol widely used in Wireless Smart Grid Networks (WSGNs). However, several studies have identified key limitations of RPL, including poor adaptability, inefficient resource utilization, and static decision-making that negatively impacts performance in dynamic environments.

To address these limitations, we propose ML-RPL, a novel routing strategy that integrates Machine Learning (ML) with RPL. Our approach employs the Naive Bayes classification algorithm to enhance routing decisions by predicting the probability of successful packet delivery based on multiple network features such as Hop Count, ETX, MAC Losses, RSSI, Throughput, Channel Utilization, Queue Utilization, and Node Density. By learning from past and current network behavior, ML-RPL dynamically adjusts path selection, enabling nodes to forward packets along routes with the highest delivery confidence.

Although Naive Bayes is used in this implementation due to its simplicity, computational efficiency, and suitability for real-time decision-making, the ML-RPL framework is extensible and can incorporate other ML algorithms in future work. This research demonstrates that even lightweight ML models can significantly improve routing reliability in LLNs compared to traditional protocols like RPL+ and MRHOF.

II. RELATED WORK

In recent years, there has been increasing interest in applying Machine Learning (ML) techniques to address core challenges in wireless networks, particularly within the domains of routing, congestion control, and traffic classification. These efforts have aimed to overcome the limitations of traditional routing protocols in Low Power and Lossy Networks (LLNs), especially RPL.

Low Power and Lossy Networks (LLNs) are characterized by constrained devices, limited computational capabilities, and unstable wireless links. The IPv6 Routing Protocol for LLNs (RPL) [1] has been standardized as the primary routing protocol for such environments, particularly over IEEE 802.15.4-based networks [2]. RPL constructs a Destination Oriented Directed Acyclic Graph (DODAG) using an objective function (OF) and static routing metrics, such as hop count and Expected Transmission Count (ETX). While RPL ensures loop- free, energy-efficient routing, its reliance on fixed metrics reduces adaptability under dynamic network conditions. This has motivated extensive research into incorporating Machine Learning (ML) techniques to enhance routing performance.

In the Vehicular Ad Hoc Network (VANET) domain, Boualouache et al. [6] proposed a decision tree–based multi- metric routing protocol that collected data from diverse urban simulations and applied regularization to select the most relevant routing metrics. While effective for metric selection, the study did not fully leverage the decision tree's native feature- ranking capabilities and lacked benchmarking against other VANET protocols. A follow-up study on the same dataset applied the CatBoost algorithm [6], demonstrating improvements in packet delivery ratio (PDR) and reduced control overhead. However, it suffered from transmission delays and was limited to vehicular environments, leaving its applicability to general LLNs unexplored.

In the smart grid domain, Galluccio et al. [4] surveyed communication challenges and standardization efforts, highlighting the need for Quality of Service (QoS)- aware routing. Chen et al. [9] later proposed a neural network–based congestion control framework for RPL in smart grid applications, enabling source nodes to decide on packet transmission based on real-time channel utilization and buffer occupancy. The approach improved throughput and latency but was evaluated on a simplified topology, which may not represent real-world deployment scenarios.

For Wireless Sensor Networks (WSNs), reinforcement learning has been applied to routing optimization. Taneja and Kushwaha [7] reviewed ML-based WSN routing protocols, including a Q-learning– driven parent selection mechanism that used rewards based on multiple network metrics. This method improved PDR and energy efficiency but required extensive state tracking, leading to scalability and overhead concerns.

Other enhancements to RPL have used deep learning models. Chen et al. [9] introduced an Adam Deep

Neural Network (ADNN) for dynamic parent selection based on packet features such as length, TTL, and payload content. While the results showed improved routing efficiency, the algorithm's grid-based hierarchy and adaptive timers increased complexity, limiting feasibility for resource-constrained devices. Similarly, Raza et al. [3] proposed a Random Forest–based parent selection approach, assigning weights to metrics like ETX, MAC losses, channel utilization, and throughput using feature importance. The use of static weight assignments, however, reduced adaptability under fluctuating traffic conditions.

Beyond domain-specific designs, broader ML-driven networking research has explored integrating software-defined networking (SDN) [5], mobile edge computing [8], and fog computing [12] to enhance decision-making in wireless networks. While promising, these methods often rely on computationally intensive models or large-scale data processing, which may not be viable for LLNs with strict resource constraints.

Summary of Limitations — From the reviewed literature, it is evident that ML- based routing solutions—including decision trees, ensemble methods, deep learning, and reinforcement learning—can enhance routing reliability and efficiency. However, the primary challenges remain high computational complexity, limited scalability, and reduced adaptability in highly dynamic environments. These factors make many existing approaches unsuitable for the constrained nature of LLNs, highlighting the need for lightweight, scalable, and adaptive routing enhancements that can operate efficiently on embedded systems.

III. MOTIVATION FOR INTELLIGENT PARENT SELECTION IN RPL

The Routing Protocol for Low Power and Lossy Networks (RPL) is a standardized distance-vector protocol designed for resource-constrained wireless networks such as wireless sensor networks (WSNs) and smart grids. RPL forms a topology known as a Destination-Oriented Directed Acyclic Graph (DODAG), where each node selects a preferred parent to forward its packets toward the root node or data sink.

Parent selection in RPL is guided by an Objective Function (OF), which defines the criteria and metrics used to evaluate potential routes. Common metrics include:

- Expected Transmission Count (ETX), which estimates link reliability,
- Hop Count, representing the number of intermediate nodes,
- Residual Energy, to prevent node overloading.

Among various objective functions, the Minimum Rank with Hysteresis Objective Function (MRHOF) is widely used. It selects parents based on minimizing a cost metric (such as ETX), combined with hysteresis to avoid frequent parent switching.

However, the default parent selection mechanism in RPL suffers from several limitations:

- Static Metric Dependency: RPL often selects parents using a single or limited set of metrics, ignoring other critical parameters like congestion, signal quality, or queue size.
- Lack of Adaptability: It does not adapt quickly to real-time changes in network conditions, such as link failures, interference, or traffic surges.
- Unbalanced Load Distribution: High-quality nodes often get selected frequently, leading to early energy depletion and node failures.
- Poor Performance in Dynamic Topologies: In networks with mobile nodes or frequent link variations, RPL's performance degrades significantly due to its rigid decision logic.

These challenges highlight the need for more intelligent and adaptive parent selection mechanisms in RPL. Instead of relying on fixed metrics or manual tuning, integrating machine learning enables each node to evaluate and rank potential parents using multiple dynamic features. In this context, lightweight and interpretable models such as Naive Bayes offer a promising solution by providing fast, probabilistic routing decisions based on real-time network conditions.

V. INTELLIGENT PARENT SELECTION USING NAIVE BAYES IN ML-RPL

In a typical RPL network, nodes become aware of their neighbors by receiving DIO (DODAG Information Object) messages. Each time a node receives a DIO message from a neighbor, it updates

its set of candidate parents and evaluates the most suitable parent based on predefined routing metrics.
Unlike conventional RPL, which combines metrics such as ETX, hop count, or energy levels into a single weighted score, ML- RPL takes a probabilistic approach. When a node receives a DIO, it uses a trained Naive Bayes model to evaluate how likely it is that the DIO-sender node can reliably forward packets. The input to the model includes multiple real-time metrics such as link quality, signal strength, throughput, and buffer occupancy. The output is a probability score indicating the likelihood of successful packet delivery through that neighbor.

Instead of calculating raw metric scores, each node interprets the ML model's output as a "cost" — a lower cost indicating a higher chance of reliable delivery. To determine the best route to the DODAG root, the node combines this cost with the advertised cost from each neighbor to reach the root. The neighbor offering the lowest total cost is selected as the new preferred parent.

However, to maintain stability and avoid unnecessary switching, ML-RPL employs a hysteresis mechanism. A node only changes its preferred parent if the new route provides a significantly lower cost compared to the current one. This threshold helps prevent frequent and inefficient parent switches in highly dynamic networks.

The entire process is summarized in a simple decision algorithm. A node evaluates each neighbor in its candidate parent set, calculates the total routing cost through each, and then selects the one with the minimum cost—provided it meets the threshold requirement for switching.

This parent selection strategy is the foundation of ML-RPL. It introduces intelligence into the routing decision by predicting delivery success rather than relying on static metrics. As a result, ML- RPL offers improved adaptability, reliability, and energy efficiency, especially in networks with fluctuating conditions.

VI. DATA COLLECTION AND ML MODEL SELECTION

A. Data Collection and Processing

To train and evaluate the proposed ML-RPL model, we developed a custom Python- based simulator tailored to mimic realistic smart grid network behavior. This simulator was designed to emulate low-power and lossy wireless environments, allowing fine- grained control over node behavior, network topology, traffic flow, and packet- level routing dynamics.

The simulator models a distributed smart meter network where each node generates and forwards packets to a central collector (DODAG root). During these routing operations, the system logs detailed transmission data at every hop. The recorded dataset captures a wide variety of real-time network conditions, making it well-suited for supervised machine learning.

The following routing metrics were collected as feature inputs:

- Hop Count: Number of hops from the source to the collector.
- Expected Transmission Count (ETX): Reliability score of the communication link.
- MAC Losses: Percentage of frames dropped or lost at the MAC layer.
- Node Density: Number of neighboring nodes within transmission range.
- Channel Utilization: Channel occupancy at the receiver's end.
- Throughput: Packet transmission rate measured at the receiver.
- Queue Utilization: Proportion of the buffer used at the receiving node.
- Received Signal Strength Indicator (RSSI): Measured signal strength from sender.

To ensure diversity in training data, we simulated a variety of traffic loads and node densities. Parameters such as packet generation intervals, buffer sizes, and interference levels were varied to simulate congestion, packet loss, and instability.

The raw output was structured using the Pandas library in Python. Each row of the dataset corresponds to a single hop-level transmission event, consisting of input features (routing metrics) and an associated label indicating whether the packet was successfully received (1) or dropped (0).

This structured dataset was then used to train the machine learning model that powers intelligent parent selection in ML- RPL.

B. ML Model Selection

The second step in the ML-RPL framework involves selecting a machine learning model capable of predicting whether a forwarded packet will be successfully received by a neighbor node. This is framed as a binary classification problem, where the objective is to learn from past routing metrics and accurately classify transmission outcomes as successful or unsuccessful.

For this purpose, the Naive Bayes algorithm was selected due to its simplicity, interpretability, and suitability for real-time, resource-constrained environments like Wireless Sensor Networks (WSNs). Naive Bayes is a probabilistic classification model based on Bayes' Theorem. It assumes that all input features are conditionally independent given the output class, which greatly simplifies the computation of prediction probabilities.

Despite its strong assumption of feature independence, Naive Bayes has proven effective in a variety of domains, particularly when the input features are informative and the model needs to operate with minimal computational overhead. This makes it an excellent fit for LLNs, where nodes typically have limited processing power and memory.

The core idea of Naive Bayes is to calculate the posterior probability of a transmission being successful, given the observed routing features such as ETX, RSSI, MAC losses, throughput, and others. During inference, each feature contributes independently to the final probability estimate, and the class with the highest posterior probability is selected as the prediction.

Naive Bayes also offers fast training and inference, allowing it to be easily retrained or updated with new data in adaptive routing scenarios. Additionally, its low model size and minimal RAM usage make it a practical option for deployment in embedded network environments.

In contrast to more complex models like CatBoost or deep neural networks, Naive Bayes requires no complex tuning or tree- based structures. Its lightweight and deterministic behavior ensures that predictions are fast, consistent, and explainable — all key advantages for real- time parent selection in ML-RPL.

VII. SIMULATION SETUP AND RESULTS

A. Simulation Environment

To evaluate the performance of the proposed ML-RPL protocol, we conducted simulations using a custom-developed Python-based network simulator designed specifically for low-power and lossy wireless environments such as smart grid and IoT networks.

The simulation scenario consisted of three different network sizes:

- 50 nodes (light traffic and low density),
- 100 nodes (medium scale), and
- 200 nodes (dense network and high traffic conditions).

Each node in the simulation acted as a smart device capable of generating, forwarding, and receiving packets within a mesh-based routing topology. One node in each simulation acted as the collector or DODAG root.

All simulations were executed for 90 minutes of network time per run. To ensure consistency and statistical reliability, 10 repetitions of each scenario were performed, and the results were averaged with 95% confidence intervals.

During simulation, key network dynamics were emulated:

- Dynamic routing updates using RPL, RPL+, and ML-RPL,
- Random traffic generation from all nodes,
- Varying network congestion and queue utilization based on node density and traffic intensity.

This environment enabled us to evaluate the adaptability and scalability of ML-RPL under increasingly complex network conditions without relying on application- specific traffic types.

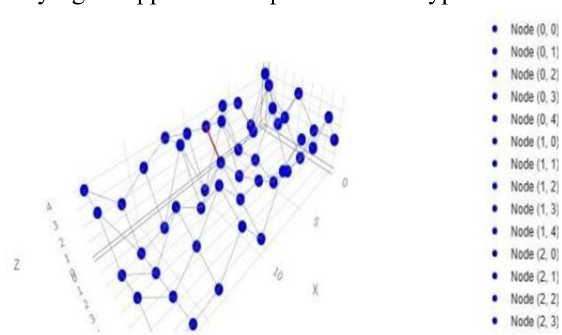


Fig 1 simulation setup

B. Simulation Results

1. Packet Delivery Ratio (PDR)

The simulation results show that ML-RPL consistently maintains a higher Packet Delivery Ratio (PDR) compared to both MRHOF and RPL+, especially as network density and traffic load increase.

In the 50-node simulation, all protocols performed relatively well under low traffic conditions. ML-RPL achieved 96% PDR, slightly ahead of RPL+ and significantly better than MRHOF.

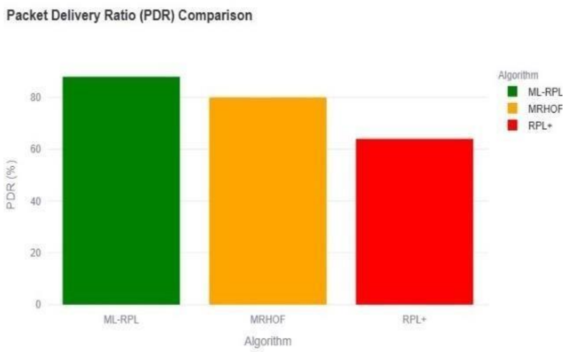


Fig 2 pdr result of 50 nodes

As the network scaled to 100 nodes, the increased routing complexity led to reduced PDR for all protocols. However, ML-RPL maintained a strong performance with 93%, while RPL+ and MRHOF dropped to 88% and 84%, respectively.

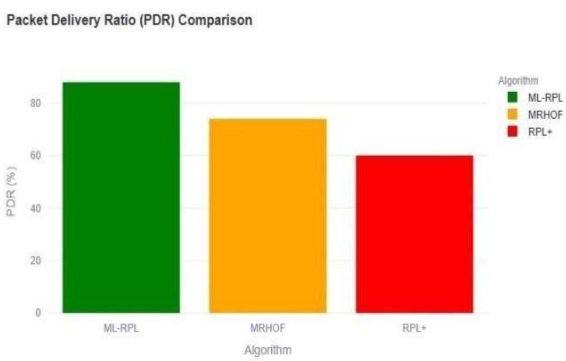


Fig 3 pdr result of 100nodes

In the 200-node scenario, where the network was highly loaded, PDR dropped across all routing schemes as expected. ML-RPL showed only a slight reduction, maintaining a PDR above 90%, whereas RPL+ and MRHOF experienced more pronounced drops. This demonstrates ML- RPL's ability to handle increased traffic without compromising delivery reliability.

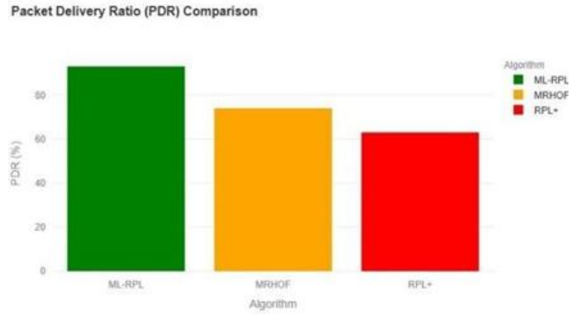


Fig 4 pdr result of 200 nodes

2. End-to-End Delay

Delay analysis also revealed valuable insights. Under low-load conditions (50 nodes), ML-RPL's delay was slightly better than RPL+, thanks to its smarter parent selection strategy. However, as the number of nodes increased, the difference in delay became more evident.

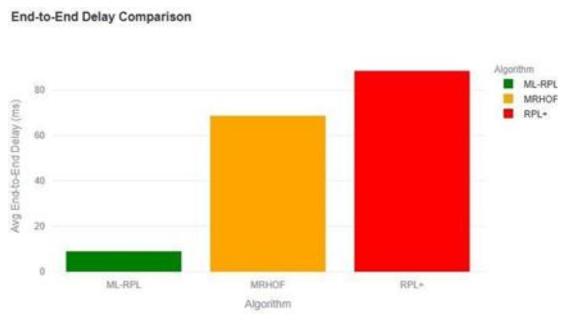


Fig 5 end to end result of 50 nodes

In the 100-node and 200-node networks, ML-RPL maintained lower end-to-end delay compared to both baselines, even though the improvement over RPL+ was narrower. All three protocols experienced an increase in delay due to higher contention, longer queues, and more complex routing decisions.

Nevertheless, ML-RPL consistently achieved faster data delivery, proving its ability to adapt under dynamic load by intelligently choosing parents that reduce congestion and avoid lossy paths.

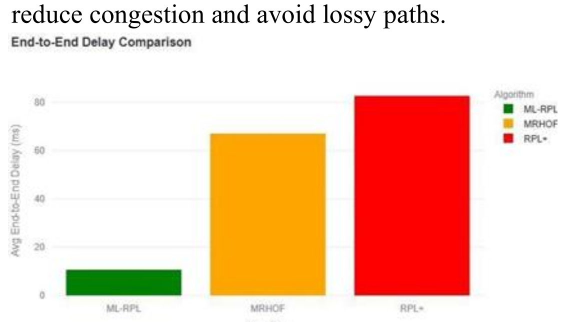


Fig 6 end to end result of 100 nodes

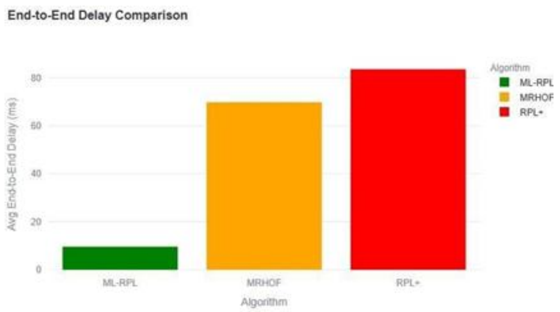


Fig 7 end to end result of 200 nodes

D. Deploying ML-RPL as a Streamlit Web Application

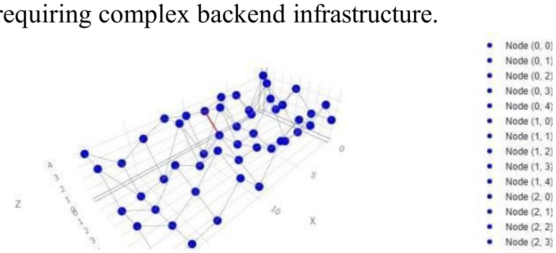
To enhance the accessibility and usability of the ML-Optimized RPL framework, the final model was deployed as an interactive web application using the Streamlit framework. This lightweight and intuitive interface allows users—such as researchers, network engineers, or students—to visualize, simulate, and evaluate routing performance in real-time using actual network configurations and routing features.

The web app enables users to:

- Input or simulate network conditions (e.g., node count, ETX, RSSI, queue status),
- Trigger the ML-RPL decision- making logic powered by the Naive Bayes classifier,
- View outputs like predicted packet delivery probability and selected parent nodes,
- Analyze routing outcomes and compare with traditional RPL results.

The interface includes real-time visualizations such as routing tables, metric graphs, and parent selection outcomes, helping users understand the behavior of the ML-enhanced routing strategy. The UI is designed to be simple, responsive, and informative, making ML- RPL accessible to both technical and non- technical users for experimentation and educational purposes.

The deployment via Streamlit also ensures easy portability and browser-based usage without requiring complex backend infrastructure.



VIII. CONCLUSION AND FUTURE WORK

This paper presented ML-Optimized RPL, an enhanced routing protocol for Low Power and Lossy Networks (LLNs) that integrates a Naive Bayes-based machine learning model into the parent selection mechanism of RPL. Unlike conventional protocols that rely on fixed metrics such as ETX and hop count, the proposed approach leverages a broader set of real-time features—including RSSI, MAC losses, queue utilization, throughput, and channel conditions—to make intelligent, adaptive routing decisions.

Through a Python-based simulation framework, ML-Optimized RPL was tested across varying network scales of 50, 100, and 200 nodes. Simulation results demonstrated that the proposed protocol consistently outperformed traditional RPL and RPL+ in terms of Packet Delivery Ratio (PDR) and end-to-end delay, particularly under high-traffic and dense network conditions. This validates the effectiveness of using lightweight machine learning models for real-time decision- making in resource-constrained environments.

Additionally, the deployment of a Streamlit-based web interface made the system more accessible and user-friendly, enabling interactive experimentation and visualization of routing decisions.

Future Work

In future iterations, the model can be extended to:

- Include online learning capabilities for dynamic environments,
- Incorporate other ML models (e.g., Random Forest, XGBoost) for performance comparison,
- Support multi-sink topologies or mobile node scenarios,
- Integrate energy-aware routing decisions for better battery management,
- And evaluate real-time deployment on physical IoT devices using hardware platforms like Raspberry Pi or ESP32.

REFERENCE

[1] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *IETF RFC 6550*, 2012.

[2] G. Montenegro *et al.*, "Transmission of IPv6

Packets over IEEE 802.15.4 Networks," *IETF RFC 4944*, 2007.

[3] M. Raza, A. Ahmad, and A. Anwar, "An optimized RPL protocol for improving routing in low power and lossy networks," *IEEE Access*, vol. 7, pp. 120019–120031, 2019.

[4] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Smart grid communications: Overview of research challenges, solutions, and standardization activities," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 21–38, 2013.

[5] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular Adhoc network with Fog computing," *IEEE Access*, vol. 5, pp. 25275–25287, 2017.

[6] A. Boualouache, A. Lakas, and B. Bensaou, "A survey on machine learning- based routing protocols for vehicular ad hoc networks," *Ad Hoc Networks*, vol. 99, 2020.

[7] P. Taneja and D. Kushwaha, "Machine learning-based routing protocols in WSNs: A review," *Journal of Network and Computer Applications*, vol. 137, pp. 12–25, 2019.

[8] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[9] Y. Chen, M. N. Mejri, and M. H. Yaghmaee, "A machine learning based framework for enhancing RPL protocol in smart grid," *Computer Networks*, vol. 193, 2021.

[10] A. Panarello, G. Merlino, F. Longo, A. Puliafito, and M. Villari, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Computing Surveys*, vol. 53, no. 6, 2020.

[11] M. Bacco, M. Colucci, and P. Cassarà, "A smart routing algorithm for wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018.

[12] M. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.