Mesopotamian journal of Cybersecurity
Vol.4,No.3, **pp**. 251–277
DOI: https://doi.org/10.58496/MJCS/2024/026; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Collaborative Intrusion Detection System to Identify Joint Attacks in Routing Protocol for Low-Power and Lossy Networks Routing Protocol on the Internet of Everything

Omar A. Abdulkareem[1,2,*], Raja Kumar Kontham[1], Farhad E. Mahmood[3]

[1] *Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam 530003, India*.
[2] *Directorate of Research and Development, Ministry of Higher Education and Scientific Research, Baghdad 10065, Iraq*.
[3] *Department of Electrical Engineering, University of Mosul, Mosul 41002, Iraq*.

ARTICLEINFO

Article history
Received 04 Oct 2024
Accepted 02 Dec 2024
Published 25 Dec 2024

Keywords
Intrusion detection system wormhole attack
RPL protocol rank attack
Internet of everything
Version number attack

ABSTRACT

The Routing Protocol for Low-Power and Lossy Networks (RPL) routing protocol is utilized in the Internet of Everything (IoE) is highly vulnerable to various collaborative routing attacks. This attack can highly degrade network performance through increased delay, energy consumption, and unreliable data exchange. This critical vulnerability necessitates a robust intrusion detection system. This study aims to enhance a Collaborative Intrusion Detection System (CIDS) for detecting and mitigating joint attacks in the RPL protocol, focusing on improving detection accuracy while minimizing network delay and energy usage. A series of algorithms and techniques are implemented, including Queue and Workload-Aware RPL (QWL-RPL) for congestion reduction, weighted random forward RPL with a genetic algorithm for load balancing, fuzzy logic for trust evaluation, and Light Gradient Boosting Machine (GBM) for attack detection. Additionally, Q-learning with a trickle-time algorithm is used to classify and manage joint attacks effectively. Numerical analysis indicates that the proposed approach performs better than existing methods in multiple metrics, including accuracy, energy consumption, throughput, control message overhead, precision, and computing time. By integrating these diverse techniques, the proposed CIDS offers a scalable and efficient solution to improve the security and performance of RPL-based networks in IoE environments, outperforming current approaches in detection accuracy and resource optimization.

1. INTRODUCTION

The Internet of Things (IoT) is a vast network of linked modules with minimal power consumption that serve as crucial elements in different industries, such as healthcare, public transit, manufacturing processes, and domestic automation [1], [2]. The Internet of Things is a ubiquitous network of various items such as sensors, cameras, and robots. These objects can communicate with each other using protocols from the Internet or other protocols' address systems [3]. Due to the inadequacy of technology and standards, the Internet of Things faces several obstacles. One of the most fundamental concerns is to provide and maintain the dependability of routing and information sharing [4]. The phrase IoE was invented to reference people's connection via a network, procedures, data, and objects in more semantic and useful ways than before. Although the IoT is a dynamic worldwide infrastructure focused on things, IoE builds an upper foundation above the IoT and deals with smart connections to networks and technology [5]. The IoT has become the most essential technology in recent years because of its low power & low-cost sensor technologies. The IoT simplifies everyday tasks such as home automation, smart healthcare, and smart transportation, among others. In which the IoT comprises resource-constrained sensor devices that have been connected over low-power wireless protocols which are Low-Power and Lossy Networks (LLN), in which the LLN has limited capacity and great latency due to its communication patterns [6]. RPL-based networks are subject to attacks on routing that are common in Wireless Sensor Networks (WSNs), as well as attacks that use RPL-specific features such as node rank & version number. Attacks on the RPL protocol that exploits network bandwidth waste reduce the efficiency of IoT networks [7]. The RPL protocol allows code updates to regulate network traffic, energy consumption [8], etc. Some explorers reveal strategies to extemporize network competency using the RPL interface for connected devices [9][23]. Security in information communication has been a long-standing concern. Considering the development of wireless technology and its wide adoption due to its simple and rapid implementation, as well as the inexpensive cost of actual network platforms, security flaws in the technology itself, and the potential for phishing and fraudulent activities by providing strategies to counteract such assaults [10]. However, transmitting RPL messages is critical to the network's performance.

*Corresponding author. Email: it@gmail.com

Some research has indicated that increasing fairness between nodes in terms of the transmission of RPL signals can increase network performance, such as route creation [11]. IDSs spontaneously monitor network traffic with a variety of frameworks and methods. They categorize the network traffic as normal or abnormal using various models and approaches. Regarding computer safety and non-violent support, users can be notified about network dangers using recognition and prediction systems. Upon receiving these alerts, the structure takes the necessary action on the other organizations [12]. Routing attacks, which are prevalent in WSNs, can affect RPL-based networks. Additionally, attacks that utilize RPL-specific attributes such as version numbers and node ranking can be used against them. Attacks against the RPL protocol for routing cause network congestion and affect the effectiveness of the IoT network [13][30]. In this regard, RPL is especially intended to reduce energy use. This procedure, which has been defined by the Internet Engineering Task Force Working Group, picks the most effective routes based on some specific criteria integrated into Objective Functions (OBF) [14]. It focuses on constructing the most efficient pathways among every other node and one or more root nodes. RPL is an anti-looping distance vector. It generates a Destination-Oriented Directed Acyclic Graph (DODAG) using node and link parameters such as Hop count (HC), Expected transmission count (ETX), energy, and Link Quality Level (LQL). These metrics are used to build an OBF. It is worth noting that the OBF is responsible for determining the node's chosen parent. As a result, the nodes proceeded to choose the optimum route until they reached their target. RPL has 2 standard OBFs. The principal is the Objective Function Zero (OBF0), which uses HC as a routing metric, and the second is the Minimum Rank with Hysteresis Objective Function (MRHOBF), which uses ETX as a travel measure [15]. As a result, IDS is used to defend information and communication networks [16]. The main issue with the RPL protocol is that it is susceptible to several routing attacks and does not take into account network layer security [17]. Because the RPL protocol has limited memory and resources, it was designed with basic security measures [18][34]. The purpose of the study is to improve the collaborative intrusion detection system (CIDS) to identify joint assaults in the RPL routing protocol inside the IoE, with an emphasis on enhancing detection accuracy while decreasing network delay and energy usage.

The main objective is to improve the CIDS to identify joint attacks in the RPL routing protocol within the Internet of everything that takes into account security issues, classification issues, network delay, etc. The primary objectives are to develop the trust calculation mechanism for neighboring nodes to improve the accuracy of intrusion detection in the RPL routing protocol. Furthermore, the analysis and investigation of strategies is to overcome the detrimental effects of increasing traffic volume and node density on protocol performance and maintaining throughput.

1.1 Motivation

The primary reasons for this study are listed below.

- As the use of IoT devices increases, network protocols must be safe and effective.
- Routing attacks can affect latency, energy efficiency, and data dependability in RPL (Routing Protocol for Low-Power and Lossy Networks).
- In complex IoE networks, current intrusion detection systems (IDS) have difficulty striking a balance between scalability, resource efficiency, and detection accuracy.

1.2 Challenges

The main challenges of this research are given below

- **Traffic Congestion:** Increased latency and energy usage are caused by high network traffic.
- **Load Imbalance:** Certain nodes experience resource loss as a result of uneven demand caused by an unequal node distribution.
- **Complexity of trust evaluation:** It can be difficult to determine a node's level of trustworthiness in dynamic networks.
- **Detection accuracy:** It is challenging to identify cooperative assaults in RPL with few false positives.
- **Energy and latency restrictions:** To preserve network performance, energy consumption and latency must be decreased.

1.3 Research Contributions

The main aim is to enhance the performance of the collaborative IDS to identify joint attacks in the RPL routing protocol on the Internet of Everything. The main contributions of this research are given below.

- To dynamically control traffic congestion, this study presents QWL-RPL, a unique protocol that integrates workload and queue information at the node level. In contrast to conventional RPL, which frequently has problems with traffic distribution and load balancing, QWL-RPL efficiently balances network load by using workload measurements and queue status in conjunction to make routing decisions. This is especially beneficial for high-density IoE systems, as it uniformly distributes traffic around the network, lowering average network latency, and increasing throughput.

- The work uses a genetic algorithm in conjunction with the WRF method to overcome the drawbacks of fixed-parent selection in RPL. By dynamically allocating communication demands, this combination improves network stability and avoids congestion at certain nodes. The system ensures that communication channels are selected as efficiently as possible by allocating weights according to each node's load and remaining capacity. For IoE networks that need reliable performance, this strategy in conjunction with genetic optimization enables the network to adjust to changes and maintain a balanced data flow.
- A fuzzy logic-based trust evaluation technique is incorporated in this work as a recognition of the intricacy of trust computations in dynamic IoT contexts. To provide a more precise trust score for each node, the fuzzy logic system evaluates several characteristics, including context information (CI), quality of communication (QoC) and quality of service (QoS). The system can more accurately identify possibly malicious nodes thanks to this strategy, which takes into account the inherent uncertainty in node behavior. In settings with frequent contacts with nodes and mobility, the result is a strong trust mechanism that improves the reliability of the detection.
- A strong yet computationally efficient model that is appropriate for resource-constrained IoT devices, the GBM algorithm is included in the research to effectively identify a variety of assaults (including rank and wormhole attacks). By prioritizing high-gradient data and ignoring less relevant data, the GBM method ensures minimal energy consumption and faster processing times, in contrast to standard detection models that may result in substantial computational and memory overhead. This enables real-time threat detection by the CIDS without putting undue strain on the constrained resources of IoT devices.
- To use the trickle-time method in conjunction with a Q-learning-based reinforcement learning model to tackle the problem of identifying joint assaults, including version number manipulation. Because of this combination, network behavior can be continuously learned, allowing the system to recognize and adjust to questionable patterns linked to coordinated assaults. By adjusting the frequency of message distribution, the trickle-time algorithm enables quick identification of network irregularities without taxing network capacity. The network's resistance to complex multi-vector assaults is increased by this integration, which also maintains a low overhead while greatly increasing the classification accuracy for coordinated attacks.
- Using several measures, including accuracy, energy consumption, throughput, precision, control message overhead, and calculation time, the study thoroughly evaluates the suggested CIDS. The results of simulations using the Cooja simulator on Contiki-3.x show that the suggested CIDS works better than current models, including SMTrust-RPL and SecRPL-MS, over a range of node densities. In addition to confirming the CIDS's exceptional accuracy and efficiency, these experimental validations demonstrate its suitability for high-density IoE situations with demanding security and performance standards.

1.4 Objectives

The following list the primary objectives of this study.

- To provide a scalable and reliable CIDS for RPL that reduces energy usage and network latency while improving detection accuracy.
- To improve load balancing and traffic control in RPL networks.
- Integrate criteria for trust assessment that increase the precision of detection.
- To efficiently categorize and counteract coordinated assaults without using up too many network resources.

1.5 Significance

The following lists the main significance of this work.

- The suggested solution fortifies IoE networks against coordinated and focused attacks.
- To maintain LLN installations in IoE, the system reduces energy consumption and latency.
- Shows robustness under growing network size and applies to a range of IoE applications (smart cities, healthcare, etc.).
- Extends device life and guarantees steady network performance by balancing network load and reducing control message overhead.

1.6 Paper organizations

The rest of this paper is divided into the following sections: Section 2 contains a review of the literature on previous research that is more relevant to our study. In addition to the primary issue statements addressed in earlier publications, the statements are enumerated. Section 3 contains a protocol, a mathematical representation, a pseudocode, and the research technique for the proposed study. The experimental findings and an evaluation of the ongoing and planned projects are provided in Section 4. In Section 5, the conclusion of the research is included.

2. LITERATURE SURVEY

This section provides a review of the literature on intrusion detection for joint attacks in the RPL routing protocol in IoE. The article [19] utilizes low-power loss networks for the multicast protocol for low-power and lossy networks (MPL) in this post before going into great detail to point out its functional flaws. Then, they offer many solutions that focus on various topics to solve such restrictions. The effectiveness of the suggested MPL enhancements was investigated through a comprehensive series of realistic modeling and tests. The collected findings demonstrate that their ideas perform better than the MPL protocol in end-to-end latency and retain the same level of DPR dependability while also outperforming it in terms of resource usage. However, in their research, they face challenges regarding security threats and the need to consume significant computing time and resource-constrained nodes.

The article in [20] presents verification and secure trust-based RPL routing in the mobile sink-supported Internet of Things (SecRPL-MS): proposed in this study. Initially, all IoT nodes within the system register with the safety entity through an enrollment procedure carried out by SecRPL-MS. In this study, the network's mobile sink is deployed to mitigate the IoT nodes' frequent deaths. Every grid member (GM) node that wishes to send data to the grid head (GH) node has to go through an authentication procedure. In RPL, secure routing is implemented using the sailfish optimization algorithm. Before sending its detected data to the GH node, each GM node encodes it using the prince method. The technique known as the quantum-inspired neural network (QINN) is used to choose the moving points to represent the mobile sink. However, in their research, they need to suggest a trust-based safe routing method based on an RPL-based IoT network and identify other security threats. They must identify additional security threats in RPL-based IoT networks and recommend a trust-based secure routing strategy in their study.

In this article [21] rank and black hole attacks in RPL while taking into account stationary and mobile IoT nodes. The carefully selected trust criteria and measurements, particularly movement-based metrics, are the foundation of the suggested Security, Mobility, and Trust-Based Model (SMTrust). Through simulated studies, the suggested approach is evaluated, and the results demonstrate that SMTrust outperforms current trust-based approaches in protecting RPL. However, in their research, they need to enhance power consumption and analyze end-to-end delay. The article [22] proposes RPL, the foundational technology for most IoT devices, and the Clone ID attack, a rarely studied identity attack. As a result, a strong AI-based security framework is proposed to combat identity theft attacks, which traditional applications tend to misidentify. Using samples from the RPL network as a baseline, unsupervised pre-training approaches are used to choose important features. After that, a dense neural network (DNN) is skilled in optimizing deep feature engineering to enhance classification outcomes and ward off malevolent traceability efforts. However, in their research, they want to improve the classification accuracy related to RPL attacks. This article [24] addresses identifying intrusions; this article combined the hierarchical semantic method with the neural network algorithm group method of data handling (GMDH). The detection of breaches may be greatly affected by the hierarchical semantic method, which is based on translating infiltrate values into interpretable numerical values and the identification of key variables in IoT infiltration. Another framework based on a neural network with hidden layers is generated via the GMDH algorithm. It also picks up lessons from the past and recognizes possible future invasions. The results of various techniques were contrasted with the results of the suggested model. However, in their research, the attack detection accuracy is low in the RPL protocol environment. The article [25] raises the security of the RPL routing system; this work builds an intelligent and lightweight IDS model called RPL Attacks based on Intrusion Detection for Efficient Routing (RAIDER). To address the lack of security around RPL, RAIDER uses simulation to study the effects of four RPL assaults. It also integrates an automata framework with the IDS nodes to examine the behavior of the nodes and minimize the influence of such attacks. Based on finite automata theory, the IDS nodes intermittently transplant the observed data as multiple states while keeping an eye on the network. RAIDAR identifies RPL attacks by basing its attack judgments on the context-aware attack making choices system's pre-estimated threshold for state changes. RAIDER maximizes the efficiency of RPL routing while using the least amount of energy. However, in their research, effective intrusion detection is detected, but not security attacks in the RPL network. The article [26] offers to improve the security of the RPL protocol, this study develops

an IDS based on deep cellular learning automata and semantic hierarchy. To make attack characteristics meaningful, a semantic hierarchy is used, and Deep Cellular Learning Automata (DCLA) is used to make the RPL protocol more secure. Five attack-related datasets have been employed in this instance: Darknet, version number, NSL-KDD, botnet, and distributed denial of service (DDoS). The suggested approach performs better than its alternatives, according to a comparison of the findings obtained from five data sets. However, in their research, they have challenges with resource constraints and decreasing data processing time.

The combination of fuzzy logic-based trust evaluation for enhanced detection reliability and queue and load-aware RPL (QWL-RPL) for dynamic traffic management, which are not frequently integrated into current RPL protocols. Furthermore, a unique method for managing resource limitations and detection accuracy in IoE networks is demonstrated by the use of light gradient boosting machine (GBM) for low-overhead attack detection and Q-learning with trickle-time algorithms for

real-time joint attack categorization. This set of methods, which are particularly designed for the RPL protocol in IoE, is both unique and significant improvement over earlier research that mainly focused on resource consumption optimization or single-attack identification.

2.1 Problem Statement

In this part, the explicit existing works and their solutions is compiled. This study also offers research solutions to the mentioned problems.

Specific Problem Definition: The article [27] presents a trust-based safe routing protocol and is suggested based on mobility. SMTrust aims to defend against black hole and RPL Rank attacks. Three distinct scenarios are used to evaluate the suggested protocol, including stationary and mobile nodes in an IoT network. The default RPL objective function, SecTrust, Dynamic Trust Calculation Mechanism (DCTM), Mobility and Residual Trust Support (MRTS), Minimum Rank with Hysteresis Objective Function (MRHOBF) and SMTrust are all contrasted. The primary concerns are discussed in greater detail below.

- However, in their research, they need to calculate the trust values of neighboring nodes.

The article [28] has presented a technique called RI-RPL, which is based on the creation of the RPL routing protocol and leverages RL to address them. RI-RPL is intended to be achieved in three broad steps. Routers are in line to optimize the RPL protocol in the initial stage, with an emphasis on the Q-learning algorithm. Deviations in the parental learning in various network settings receive assistance in the following phase, which is based on learning and convergence. Adjustments in leadership and oversight are coordinated during the third stage. This method was selected due to its ability to efficiently handle the intended problems without using excessive network resources for computations. The following is a list of the problems with this work.

- The throughput has shown a rising tendency as nodes and traffic rates rise; at traffic volume and a certain number of nodes, the throughput even begins to decline. The detrimental effects of more traffic and nodes on protocol performance are the cause of this problem.

This article [29] has provided a detailed analysis of rank assault, one of those dangers to RPL. A lightweight and effective strategy to reduce and localize the rank attack is proposed and evaluated, taking into account the limited resources of IoT devices. In particular, their method computes and verifies the validity of the advertised rank by using a new Echelon Metric-Based Objective Function (EMBOBF) instead of the standard RPL. In the RPL network architecture, the echelon value is decided additively by the root node and the accompanying parent node(s). Their method not only finds the assailant node or nodes, but also detaches them immediately. The primary concerns are discussed in greater detail below.

- However, in their research, their work will increase network delay and energy consumption during attack detection.

Article [31] offers an arrangement for network node placement based on the Multi-Sink Routing Protocol for Low-Power and Lossy Networks (RPL). The following is a list of problems with this work.

- It has been noted that the participating nodes are not uniformly distributed across the sink. Although some sinks have fewer nodes than others, other sinks have more. As a result, the load balance will increase in their research.

This article [32] presented that feedforward, and fuzzy neural networks are used to create a unique IDS that can identify routing attacks in WSNs. Research results show that, in contrast to other methods such as support vector machine (SVM), decision tree (DT), and random forest (RF) designs, the suggested model achieves an average detection rate and the highest detection accuracy. The primary concerns are discussed in greater detail below.

- However, in their research, the accuracy of attack detection is poor, and it has a rather lengthy computation time.

2.2 Research solution

The fuzzy logic used to calculate trust levels takes into account uncertainty, which is one of the most essential intrinsic qualities of trust. The innovative QWL-RPL protocol will minimize network congestion while improving average throughput latency. Light gradient boosting is used to identify rank and wormhole assaults in RPL, as well as Energy and Delay Aware Data Aggregation to reduce network latency and energy usage. Combining energy-aware data aggregation with light-gradient boosting allows for the detection of assaults with little latency and energy cost. The weighted random forward algorithm RPL protocol with Genetic algorithm might consider load balancing over RPL to disperse communication and messages to prevent congestion on one chosen parent, hence improving performance. The Q-learning strategy is used to spot the malevolent nodes in a version number assault. It detects and reduces overhead on RPL network nodes accurately. The trickling-time approach is employed to increase the attack detection accuracy. The combination of Q-Learning and trickle time allows for more accurate detection of version number attacks.

3. PROPOSED METHOD

The goal of the project is to enhance the overall performance of CIDS for joint attacks in the RPL routing protocol in the IoE. Fig. 1 represents the overall architecture of the proposed architecture. The Collaborative Intrusion Detection System (CIDS) for the RPL protocol in Internet of Things applications is shown in this picture with its tiered design. IoT nodes are at the base of a DODAG structure (Destination-Oriented Directed Acyclic Graph), sending data to specified parent nodes. The system incorporates cutting-edge protocols, including a Weighted Random Forward (WRF) mechanism for effective load balancing and Queue and Workload-Aware RPL (QWL-RPL) for traffic management. The system optimizes data routing according to node load and energy capacity using a genetic algorithm in conjunction with WRF. This is crucial in high-density Internet of Things networks. Fuzzy logic for trust assessment and light gradient boosting (GBM) to improve attack detection precision without taxing computer power further strengthen this configuration. This architectural design ensures that network security is improved, energy consumption is minimized, and traffic loads are dynamically balanced. The risks and demands unique to IoT networks are immediately addressed in real-time by this tiered approach, especially in the face of high traffic and a variety of attack vectors.
Important steps are taken as part of this process, including the following.

1. Traffic Congestion
2. Load Balancing
3. Trust evaluation
4. Joint Attack Classification
5. Network Delay and Energy Consumption
6. Classification of Joint Attacks

3.1 Traffic congestion

Initially, tree topology is constructed. This is the base structure that allows for efficient routing of data from numerous nodes to the central root node. Observing the behavior of the RPL protocol in numerous congested environments, it becomes obvious that traditional routing approaches struggled to maintain efficiency under high traffic volumes. To overcome this challenge, the novel queue and workload-aware RPL (QWL-RPL) is introduced. This novel protocol incorporates information from the queue and workload in the node-wise environment into its routing decisions, effectively distributing the traffic evenly across the network. Implementing QWL-RPL results in significant reductions in average throughput delay and overall network delay that demonstrate the efficiency of managing traffic congestion.

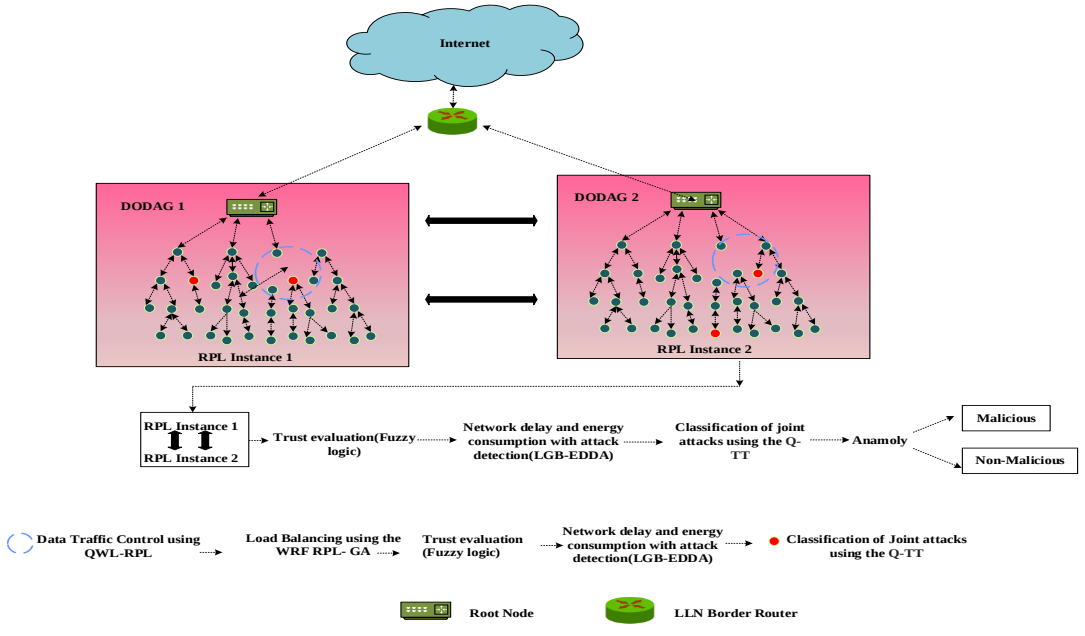

Fig. 1. Architecture of proposed collaborative intrusion detection system (CIDS).

3.1.1 Queue-and workload-aware RPL

In a variety of diverse network environments, the problem of load-traffic imbalance is resolved by the QWL-RPL enhanced routing protocol algorithm. Therefore, it is not appropriate for different network load traffic situations to maintain load and traffic balancing using the routing topology subtree. In this scenario, the workload and overhead statistics must be used for continuous traffic on each device or node. Because the network queue's memory is so small, a node or scheme's queue can only hold a maximum of 4 packets at once. When this packet or data is compared to each device's workload and overhead information, it is noticeably extremely small. Because of its subtree, which consists of all of its offspring nodes, each device or node's workload, and overhead information include control information and the real traffic load. Workload information and the number of traffic flows for each node are provided every ten seconds (numty). According to this notion, the node's workload is calculated from the total number of packets transferred over a 10-second transmission period. It also takes into account the type of packet that each node receives from its subtree and calculates the number of packets that the parent node generates. The workload calculation is defined by Algorithm 1. The workload and overhead information are used with the queue information that has a weight value as stated in Equation 1.

$$Rank = Rank(q) + WLMAC_{ty} + \propto Qbuflen \quad (1)$$

Where $Qbuflen$ is a list of packets in the waiting buffer queue, $Rank(q)$ is the rank of the parent device or node, and $WLMAC_{ty}$ is the total quantity of data or packets delivered or received at the physical layer throughout the most recent period. Next, as previously mentioned, the Predictable Transmission Count (PTC) node distributes control packets to every device or node at a predetermined, regular interval to verify the quality of the connection or link. The getting device or node then distributes the same control or probe packet to every device or node again, which causes the network to become more congested and causes a delay.

Algorithm 1: workload measurement

1. Start
2. Transmission of packet(numty)
3. Provide the last transmission clock time
4. Set event timer of 10 clock seconds
5. Workload ← workload + numty
6. If (event timer expired), then
7. workload← 0
8. reset event timer
9. end
10. Move to step 5

3.2 Load balancing

Following the mitigation of traffic congestion, the main goal is to improve the load balancing within RPL networks. By using the weighted random forward (WRF) algorithm RPL protocol with a Genetic algorithm can distribute communication and messages more efficiently and that prevents congestion at any single preferred parent node. The WRF algorithm considers multiple potential forwarders and assigns weights based on their current load and capacity, which ensures that the network communication load is balanced. This technique mitigates the risk of overloading specific nodes, which leads to a more stable and effective network performance. The Genetic algorithm is employed to optimize the selection of forwarders by evolving the weight distribution principles over time. This dynamic approach continuously improves the network's adaptability to changing conditions, and additionally enhances load balancing and overall network stability.

3.2.1 Weighted random forward algorithm

In high-traffic and high-data-demand situations, constrained sensor networks must operate at full volume to manage and complete the activities required by their design. As a result, the RPL protocol is designed for low data rate situations in which sensors exhibit persistent slumber since they do not require regular job processing. Thus, in high-demand situations, this protocol poses serious problems with managing and allocating the data load that the network experiences. In this kind of scenario, while the node's buffer capacity, energy consumption, network lifetime, and packet success rate, amongst other factors, are pretentious to a certain scope, it is necessary to address techniques to improve the operability of the WSN.
For these and other details, the WRF-RPL protocol is suggested using various message channel methods to load distribution, ensuring the effective use of the sensor network's energy. Candidate parents that are on the best path to connect the exchange of messages to an endpoint can be evaluated based on a measure that takes into account the energy remaining

of a node and the number of parent nodes it has. This characterization enables the weighted random choice method, which is integral to the proposed scheme, where each node is assigned a score that determines its significance in the network. The contributions and issues raised are salvaged for the suggested load-balancing strategy, which defines a novel network performance, concerning WRF-RPL operation. This measure is selected to take into account the candidate node's current energy values and its available delay options, as stated in Equation 2, which combines both principles.

$$metric_{evalf} = Q_{(balance\ energy)}(\%) * \left(Q_{(count\ of\ parent)}\right). \quad (2)$$

Q is defined as the element representing the current parent under analysis in Equation 2 of the metric calculation. Its attributes are $Q_{(balance\ energy)}$, which the node's remaining energy percentage is, and $Q_{(count\ of\ a\ parent)}$, which is the number of candidate parents a node has. Because the neighbor node studied has many paths to the sink, appropriate load distribution between the associates in the message tree may result from considering the $count\ of\ parent$ value. Regarding a node's energy consumption, the configuration of this value is left up to the researcher's judgment because the energy ceiling may change based on the node's capabilities. To evaluate potential parents and find upstream pathways, changes must be made to the DIO message that is broadcast. Information about the investigated node can be stored based on shared metrics, as explained. Regarding a node's energy consumption, the configuration of this value is left up to the researcher's judgment because the energy ceiling may change based on the node's capabilities. To locate upstream routes, changes to DIO message transmission are required due to the implementation of a new metric to evaluate prospective parents. As mentioned in Algorithm 2, it is feasible to save data related to the investigated node based on shared metrics.

Algorithm 2: Parent Set Construction

Data: node of parent Q related to a message DIO
Outcome:
Arrangement $set\ of\ parent, weight\ set\ z, arrival\ set$
updates $set\ of\ parent$
$\cup\ Q\ arrival\ set\ \cup actual\ time(segs)$

$$Q\ weights\ \leftarrow\ Q_{(count\ of\ parent)} * Q_{(balance\ energy)}$$
$$weight\ set\ \cup Q\ weight$$

Algorithm 2 explains the initial steps of the protocol upon receiving a DIO message and before examining potential parent candidates (Q). The retrieved attributes of the message are explained, including the arrival time and the value corresponding to the weight selection metric, which matches the selection metric of the assembled candidate parent in the array weight set. Equation 3 also explains how the protocol determines the value of the RANK, a parameter of the RPL execution.

$$RANK\ (m) = hops(m) + 1 \quad (3)$$

In Step 3, a node's rank (m) is calculated, which is the number of hops among its current position, and the sink node, which is the root of the RPL instance. Let hops be the number of nodes needed to exchange data with the gateway node n to exchange information with the gateway node, as shown in the equation hops(m)). In Algorithm 3, the steps for how a node uses the WRF protocol for next-hop selection are specified. The preferred parent in the WRF protocol is identified, covering the steps and choices made by a node participating in the RPL instance during the WRS process.

Algorithm 3: Weighted Random Selection

Input: parent set, weight set, arrival set, t: current time constant for parent selection
Output: preferred parent: chosen parent node for the net-hop transmission
Subsequent Event for Jump Selection
$S\ \leftarrow$ *Choose a random number between 1 and 100 to find the % of your choice.*
$every\ weight\ \leftarrow$ Addition of all weights in the "array
Weight set"
$Current\ time \leftarrow$ Analysis of time, according to the clock system.
$Prev\ \leftarrow\ 0$ Selection probability of preceding counts.
For each $Q_i \in parent\ set$,
$W_i \in$ weight set, $a_i \in$ arrival set do
If $\Delta\ t \geq$ (current time - a_i) then
If $\left(\frac{W_i}{all\ weights*100}\right) + prev\ \geq R\ \leq prev$ then

Preferred parent ← q_i
Returned preferred parent
Else
Prev ← $prev + \frac{w_i}{all\ weights*100}$
End
End
End
Return θ

These steps can be used to delimit Algorithm 3:

Step 1: Investigate and evaluate the potential parents who satisfy the at Δt requirement. Following this filter, each parent's weight is determined by adding together all of the potential parents' metrics and dividing it by their proportion of impact.

Step 2: A weighted random selection will be carried out after the parents' weights are known. To make this choice, a random coefficient is created that, when a candidate parent's range of impact is compared to the whole sum of its weights, indicates which percentage the candidate parent falls into.

Step 3: After the selected candidate parent is selected, it is kept on file for upcoming correspondence until the parent sets elements are received.

Through the distribution of the likelihood of choice among potential parents, the concept mitigates congestion caused by an inadequate load distribution. The convergence of choosing alternate routes with enough energy and a higher load relaying option (higher$Q_{(count\ of\ a\ parent)}$,) is the desired outcome. Regarding the actions that characterize the operation of the WRF-RPL protocol, Figure 2 illustrates the progression of the procedures that underpin the analysis and choices made by a sensor node when receiving a packet.

Figure 2 shows how crucial it is to receive the DIO packet and choose the chosen parent afterward. The flow chart illustrates how the WRF algorithm in the RPL protocol makes decisions, with the nodes selecting the parent nodes in real time according to their proximity to the destination and residual energy. The process begins by receiving DIO packets and using weights based on the traffic load, distance (rank), and current energy of each candidate node to choose possible parent nodes. Nodes occasionally re-evaluate the optimal parent in an iterative process to preserve routing efficiency. The connection strength (based on DODAG measurements) and the residual energy of each candidate are important parameters that affect the weighted random selection procedure. The latter procedure adheres to the condition that has a direct impact on the exchange of control messages over the network's lifetime. To choose the global parent and notify it via DAO messages, conditional involves determining whether, among the candidate's parents, a parent has been selected with a greater weight or importance utilizing Algorithm 3. If the contrary is true, the WRF-RPL protocol does not take any further action because the selected preferred parent will be based on interior information rather than topology. Compared to the published route, this uninformed trip is regarded a local or alternative route. Unlike the path described by the DODAG, which resembles a method of global data, this uninformed hop is regarded as an alternate or local route. To assist neighboring nodes in choosing the best parent based on parameters such as residual energy (RER), the DODAG root broadcasts a DODAG information object (DIO). By ensuring that nodes with greater energy reserves are selected, this energy statistic increases the lifespan of the network. Nodes use methods that optimize for both energy efficiency and low hop distance to determine their rank depending on how close they are to the root, taking into account energy and hop count. In this stage, the data packets from the child nodes are aggregated using compressed sensing (CS) theory at the parent node. The aggregated parent nodes send data in the direction of the DODAG root. By reducing duplicate data transmissions, this aggregation process, which is outlined in the text using formulas, ensures energy conservation and network latency reduction. The aggregated data is compiled by the DODAG root to provide a comprehensive representation of network information.

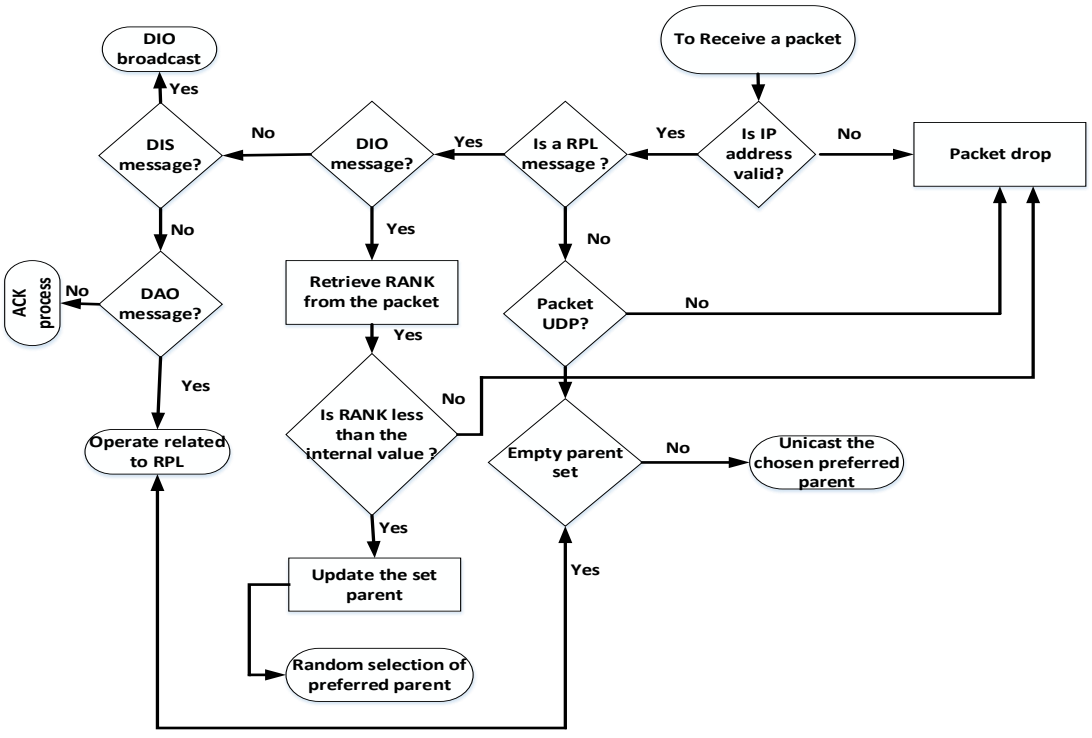

Fig. 2. Workflow of weight random forward (WRF) algorithm in RPL.

Increased network resilience and energy efficiency are made possible by this load-balancing technique. The WRF method extends the network lifetime and reduces latency while guaranteeing a high-quality, balanced traffic distribution by constantly adapting to network load and node availability.

3.2.2 Genetic algorithm

The structure and layout of the GA components for the distributed load-balancing clustering problem (DLBCP) are covered in this section. Genetic representation, population initialization, fitness function, selection scheme, crossover, and mutation are the main parts of GA procedures. To effectively execute GA, the GA components is designed using our domain expertise. In this study, the customized GA for the DLBCP is created. One feature of the clustering problem is that different *Ch* selection sequences will result in different clustering outcomes. This characteristic has been used in genetic demonstration to greatly improve the capacity of GAs for exploration. Since the DLBCP is a combinatorial optimization issue, the population initialization took this property into account. The number of members of the cluster and the degree of the node are used as parameters in the fitness function. Other environmental factors do not have a bearing on either of the two characteristics. As a result, the algorithms may function effectively in practical settings. Additionally, the crossover and mutation are precisely constructed so that, after operations, the chromosome does not include any duplicate node IDs. In conclusion, the development of GAs in this work has taken into account the domain expertise of the clustering issue.

a. Genetic representation

Our methods generate solutions that represent *Ch*, which are chosen from among all nodes in the network. To produce a random set of cluster heads, a random permutation (RP) of node IDs will be helpful. In this paper, a chromosome by RP of node IDs is represented. It is crucial to confirm that every chromosome has a unique node ID. Every chromosomal node ID is referred to as a gene.

b. Population Initialization

Every chromosome in a GA represents a possible solution. A specific number *p*, of chromosomes makes up the original population P. The appropriate permutation of the node IDs for every chromosome is chosen at random in our methods to investigate the variation in genes. The following process is used to create the first population.

1. Start(k=0)
2. Develop a chromosomeChr_k: To determine the appropriate Ch_k cluster headset, randomly permute the node IDs.
3. K is equal to $K + 1$. Proceed to Step 2 if $k < p$; if not, quit.

Consequently, $P = \{Chr_0, Chr_1, \ldots, Chr_{p-1}\}$ is the initial population. A combinatorial optimization issue with a discrete search space has been under investigation. As a result, it is extremely difficult, if not unbearable, to forecast the regions in which ideal solutions are most likely to occur. It is appropriate to initialize the population at random.

c. fitness function

It is important to appropriately assess the quality of a solution (fitness value), which is established by the fitness function. Our methods seek to identify the set of cluster heads that, when combined, can form a load-balanced cluster structure, meaning that each cluster head has the same cluster head degree, or serves an equal number of cluster members. The standard deviation of the cluster head degrees serves as our main yardstick for evaluating the quality of the solutions. Therefore, the option with the lowest standard deviation is selected from a group of potential solutions. The fitness value $E\ (Chr_i)$, which stands for chromosome Chr_i (which represents the cluster head CH_i), is as follows:

$$E(Chr_i) = \left(\sigma_{CH_i}\right)^{-1} = \sqrt{\frac{1}{n}\sum_{l=1}^{n}\left(f_l - \overline{f_{CH_l}}\right)^2}^{-1} . \quad (4)$$

d. Selection scheme

Selection contributes significantly to population quality by transferring high-quality chromosomes to the next generation. The fitness value is used to pick the chromosome. The straightforward and efficient pairwise choice of the tournament method without replacement is used.

e. Crossover and mutation

Two key genetic operators are crossover and mutation. With crossover, two-parent chromosomes can become two offspring chromosomes. Every gene found on each offspring chromosome comes from a separate location on each of the two parent chromosomes. A mutation modifies the values of several genes to produce an offspring chromosome from a single parent chromosome.

3.3 Trust evaluation

Once load balancing is achieved, the next step is to ensure the integrity of the network through trust evaluation. Trust calculation in RPL networks is inherently uncertain and challenging. **Fuzzy logic** is employed to overcome this issue. The fuzzy logic allows for more flexible and refined trust assessments by accommodating the inherent uncertainties in evaluating the node behavior and interactions. These results in more accurate and reliable trust metrics that help to identify and isolate potentially compromised nodes, thus improving the overall security of the network.

3.3.1 Fuzzy logic

A multistage fuzzy model called FDTM-IoT is used to assess the reliability of IoT devices. FDTM-IoT computes trustworthiness in three dimensions at the first fuzzy stage. Contextual data, QoS, and peer-to-peer communication quality (QPC) are taken into account. The model is comprehensive and dynamic due to the consideration of dimensions and methods of calculation and evaluation. The development of FDTM-IoT followed a tiered framework.

Each dimension therefore has a sub-dimension of its own. This arrangement results in a dynamic model. It is simple to add and remove more dimensions and subdimensions from this dynamic model. A separate fuzzy inference system is suggested for every dimension. The final fuzzy inference system receives input from fuzzy inference systems in all dimensions in the second fuzzy stage.

3.3.2 fuzzy inference system

Multiple input variables can be transformed into one using a fuzzy inference method. The idea that a variable may belong to a set that is between true and false is fundamental to fuzzy inference systems. Linguistic variables can be used by a fuzzy inference system. When words or sentences are used as input or output variables rather than numbers, they are known as linguistic variables. There are multiple steps in the fuzzy inference system.

- **Fuzzification:** Fuzzification is the process of transforming linguistic variables into explicit variables. The fuzzy specification specifies the degree of membership in fuzzy sets for the input variables.
- **Fuzzy inference:** Using inference methods, the inference engine assesses and infers the rules. Fuzzy inference computes the outcome after performing fuzzified input combinations.
- **Aggregation:** At this stage, the unification has been completed. Put another way, all values are united into one if an output is dependent on many rules.
- **Defuzzification:** The defuzzification unit transforms the output into an explicit or numerical value.

3.3.3 Trust Dimension

The quality of communication (QC) between two objects is assessed in this domain. This dimension evaluates the success rate of B in the relationship with A from the point of view of A, regardless of the outcome. This dimension is used to calculate and evaluate trust and takes into account the most recent direct QC observation, previous QC information, and indirect QC information (recommendations). The trust level derived from direct observations and information is known as the last direct QPC. Eq. (5) is used to compute the direct QC. Eq. (5) uses the terms c_i (belief rate), V (uncertainty), and D (certainty from A to B).

$$\begin{cases} DirT_{qc}^{A,B}(y) = \left(T_{qc_1}, \ldots., T_{qc_k}, v\right) \\ Where \\ T_{qci} = c_i \times D^{A,B}(y) \\ v = V^{A,B}(y) = 1 - D^{A,B}(y) \end{cases} . \tag{5}$$

The device's trust level, or historical QC, is determined by past data. Equation (6) is used to calculate the historical QC.

$$\left(DirT_d^{A,B}\right)_{historialQC} = \begin{cases} \dfrac{\sum_{t=1}^{m} e^{-\frac{t}{g}} \times \left(DirT_d^{A,B}\right)_{historialQCt}}{\sum_{t=1}^{m} e^{-\frac{t}{g}}} \\ Where \\ n = 3 \ for\ device\ constrined \\ n = 5\ for\ IoT\ devices \\ n = 7\ for\ powerful\ machines \\ G\ is\ the\ relative\ strength\ of\ the\ memory \end{cases} \tag{6}$$

Eq. (6) states that the trust level in the nth most recent relationship is $\left(DirT_d^{A,B}\right)_{historialQCt}$. G is used to provide the greatest significance and influence on the most current messages. The degree of trust acquired through referrals from neighbors is known as indirect QC. Recommendations are filtered using Equation (7) to eliminate harmful suggestions. Following the selection of the suggestions, Eq. (8) from my most recent publication is used to compute the indirect QC.

$$\begin{cases} if\ (dis \leq threshold_{Re\ d} \\ Re\ is\ acceptable \\ else \\ Re\ is\ not\ acceptable \\ Where \\ dis = \left(DirT_{qc_B^A} \neq 0\right) ? \mid Re\ dT_{qc}^{a,B,R} - DirT_{qc}^{A,B} \mid : \\ \left| Re\ d\ T_{qc}^{A,B,R} - \frac{\sum_{R=1}^{m} Re\ \ dT_{qc}^{A,B,R}}{m} \right| \\ and \\ threshold_{Re\ d} = 0.2; R\epsilon\{recommendation_r\}; \\ m\ is\ total\ number\ of\ recommendation \end{cases} \tag{7}$$

$$RedT_{qc}^{A,B,R} = \begin{cases} \dfrac{\sum_{r=1}^{m} w_R \times RedT_{qc}^{A,B,R}}{\sum_{r=1}^{m} w_r} \\ where\ w_r \epsilon [0,1]\ , w_r = DirT^{A,R} \end{cases} . \tag{8}$$

$RedT_{qc}^{A,B,R}$ in Eq. (8) denotes entity R's proposal to entity A about entity B. wr represents the credit assigned to each of the suggested entities by their track record and effectiveness. The value of one represents the complete and absolute trust that entity R has, and w_r is in the range of 0 and 1.

a) Quality of Service (QoS)

The term quality of service refers to the general assessment of a thing's total services rendered about its intended use. This dimension can be altered according to how trust is to be determined. Various QoS parameters are offered in the literature.

b) Contextual Information (CI)

Status and abilities that are acquired from context are referred to as contextual information. This dimension assesses the given data. The IoT ecosystem provides a wealth of contextual information. Among these, the most crucial information is regarding the movement of objects. One of the most crucial CIs that is dynamically generated in real-time is security capabilities. The intelligence of the device is another CI. Risk circumstances, energy status, environmental risk, and temporal status are further areas of corporate intelligence. The right parameters must be taken into account to calculate CI trust, depending on the purpose and context of the trust calculation.

c) Final trust

The final trust fuzzy inference system combines the computed trust in each dimension (the outputs of fuzzy systems in each dimension) to determine the final trust value. For this reason, as inputs enter the final fuzzy system, the outputs of fuzzy systems in every dimension are used. Fuzzy inference is then used to establish the final trust value.

3.4 Joint Attack Classification

➢ **Network Delay and Energy Consumption**

With trust evaluation placed, the attention turns to detecting joint attacks like rank attacks and wormhole attacks that affect the network delay and energy consumption. Consequently, light gradient boosting (LGB) was proposed to detect attacks effectively. The LGB is a machine learning capability that ensures the identification of the malicious with minimal computational overhead. Additionally, to minimize network delay and energy usage, energy and delay-aware data aggregation is integrated into the system. This technique involves two key processes.

- Parent Selection: The first process is the selection of parents, which utilizes the residual energy of the routing metric (RER) to select the most energy-efficient parent node for data transmission. This ensures that nodes with higher energy reserves are preferred, which persists in the overall network lifespan.
- Data aggregation: After the parent selection data aggregation takes place, which employs the compressed sensing (CS) theory at the parent node to efficiently associate the data packets from child nodes. It reduces the volume of transmitted data conserves energy and reduces the delay.

3.4.1 Light-Gradient Enhancement

This research proposes a multiclass classification approach to counteract rank and wormhole assaults in an RPL-based Internet of Things network. The dataset's benign, rank, and wormhole target classes" are classified using multiclass classification using the light gradient boosting machine model. Microsoft created the model in 2016 as a simplified version of the gradient boost technique for binary classification, incorporating exclusive feature clustering and 1-side sampling. By giving the high-gradient data instances a great priority and eliminating the inadequate gradient data instances, the one-side sampling technique known as Gradient-based One-Side Sampling (GOSS) achieves and preserves the precise data gain. The GOSS function's mathematical form is shown in equation (9). The estimated variance gains in the $A \cup B$ subgroup, described by A_l, is represented by $\widehat{W}_k$. A_r, B_l, and B_r in the formula, and $1 - a/b$ denotes the gradient sum's normalization coefficient.

The $\widehat{W}_k$ (d) is applied to discover the optimal split point for smart data set sampling, focusing on cases with notable gradients to improve model accuracy. This also helps to simplify things.

$$\widehat{W}_k(d) = \frac{1}{m}\left(\left(\sum_{y_i \epsilon A_l} g_i + \frac{1-a}{b}\sum_{y_i \epsilon B_l} g_i\right)\text{^}2 / m_l^k(d) + \left(\sum_{y_i \epsilon A_r} g_i + \frac{1-a}{b}\sum_{y_i \epsilon B_r} g_i\right)\text{^}2 / m_r^k(d)\right). \quad (9)$$

Second, by combining unique features into a single feature, the exclusive feature-bundling technique, or EFB, reduces complexity. The model's histogram-based techniques reduce memory usage and speed up training, which is beneficial for LLNs like the Internet of Things. As a result, this model is used to perform multiclass classification, which is then followed by fine-tuning and hyperparameter optimization.

3.4.2 Energy and Delay-Aware Data Aggregation

The Selection of the Suggested Energy and Delay Conscious Parent and Data Aggregation are the two steps involved in the Data Aggregation of RPL (EDADA-RPL). The residual energy (RER), a routing parameter, is used in the parent selection process to decide the optimal parent for data exchange. Data packets from the child nodes are combined by the data aggregation process in the parental node using the CS theory. Ultimately, the combined information travels from a parent that is lower to the DODAG root, or sinks. To get the original data, the DODAG root compiles the combined information and performs the reconciliation procedure.

a. Selection of parents

The DODAG Information Object (DIO) in EDADA-RPL is broadcast by the DODAG root node to the network's neighboring nodes. The participant node sends the DODAG Advertisement Object (DAO)communication to the parent or DODAG root. During the trickling interval, the child node receives the signal of the DODAG Advertisement Object-Acknowledgment (DAO-ACK) from the DODAG root or parent node. Depending on the remaining metric energy (RER) routing at the DODAG node level, the participating node selects the parent node. The RPL router's current energy availability is displayed as residual energy. The RER calculates the discrepancy between the node's present energy consumption and its initial energy consumption. Equation (10) provides the formula to calculate the remaining energy.

$$RER(Mi) = \frac{F_{initial} - F_{depleted}}{F_{initial}} \ . \tag{10}$$

b. Parent Rank Calculation

The rank shows the distance between the participant's node and the root of the DODAG. The parent node(y) rank and its value, Rank Increase Value, are used to calculate the node "y" rank. The residual energy and MinHop rank increase values are computed using the Rank Upsurge value. The rank calculation's default value for the MinHop Rank Increase variable is 256. Equations (11) and (12) provide the rank calculation.

$$Rank(y) = Rank(parent\ node(y)) + Rank\ increase\ value \tag{11}$$
$$Rank\ increase\ value = RER + min\ hop\ rank\ increase \tag{12}$$

Algorithm 4 provides the EDADA-RPL parent selection algorithm.

Algorithm 4: "EDADA-RPL parent selection"

Input: DIO-DODAG Information Object message containing routing information, DAO-DODAG Advertisement Object sent from child to parent, DAO-ACK-Acknowledgment of DAO received from parent, DIO_RER-Residual Energy Routing metric from DIO message

Output: Optimal Parent- Selected parent node for efficient energy

1. For the preferred ParentNode parentNode list, do
2. Calculate RER
3. $RER(Mi) = \frac{F_{initial} - F_{depleted}}{F_{initial}}$
4. compute the Rank(m)
$$Rank(y) = Rank(parent\ node(y)) + Rank\ Increase\ Value$$
5. Calculate the Rank-Upload Value
$$Rank\ increase\ value = RER + MinHop\ rank\ increase$$
6. If Best Parent Node>= Preferred Parent Node Then
$$Best\ ParentNode = Preferred\ ParentNode$$
7. End
8. While preferred Parent Node==Best Parent Node,

Source node=Preferred Parent node

9. End
10. End
11. Return Optimal Parent

3.4.3 Using CS Theory Data Aggregation in Parent Node

The data $e = \{e1, e2, \ldots en\}^T$ is transmitted by the sensor nodes to the parent node throughout the data aggregation process. The data are collected and aggregated by the parent node PN1, which then forwards the combined data packets to the DODAG root. A sparse matrix, or weighted sum of the random number multiplied by the sensor data 'e', is sent to each parent node.

Path 1 The values generated by node PN_1 are $r_{11} \times e_1$, as stated in Equation (13).

$$PN_1 = r_{11} \times e_1 \ , \tag{13}$$

where $e_1$the parent node 1 is(PN_1) aggregate information, and r_{11} denotes the PN_1 Value selection. Initially, the aggregated data $(r_{11} \times e_1)$ is sent to PN_2 by the parent nodePN_1. Second, in addition to gathering and combining data

from its child nodes, the parent node PN_2 also gathers data fromPN_1. The equation provides the mathematical representation (14).

$$PN_2 = r_{11} \times e_1 + r_{12} \times e_2 , \quad (14)$$

$$PN_3 = \sum_{i=1}^{k} r_{1i} \times e_i , \quad (15)$$

where r_{1i} is the random number of every parent node in path 1 and k denotes the number of parent nodes in each path. The data is transmitted from PN_1 to the DODAG root by the parent nodes in path 1. Equation (16) provides the observation matrix, which is the result of the DODAG root's collection of the aggregated information in path 1.

$$z_1 = \sum_{i=1}^{k} r_{1i} \times e_i \quad (16)$$

Where z_1 denotes the path 1 aggregated data collection by the DODAG root.
In the same manner, the data packetsz_i, where i = 1, 2… N is received by the DODAG root, which also gathers the aggregated information from M pathways. Thus, Equation (17) can be used to express the data aggregation process mathematically.

$$\begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ z_N \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{1M} \\ r_{12} & r_{22} & r_{2M} \\ r_{N1} & r_{N2} & r_{NM} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ z_N \end{bmatrix} . \quad (17)$$

Then, CS theory can be used to renovate the weighted total of combined records from the N path to the innovative data of the M node. As a result, the maximum number of data transmissions can be lowered ($N < M$). Algorithm 5 shows the data aggregation using the CSP theory.

Algorithm 5: Using CSP Data Aggregation

Input: Sensor data $e = \{e1, e2, \ldots en\}$
Output: Compressed data z

1. Calculate the data aggregation from $PN1$ to the DODAG root

$z_1 = \sum_{i=1}^{k} r_{1i} \times e_i$

2. Using the M pathways, calculate the aggregation of data in the DODAG root.

$$\begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ z_N \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{1M} \\ r_{12} & r_{22} & r_{2M} \\ r_{N1} & r_{N2} & r_{NM} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ z_N \end{bmatrix} . \quad (18)$$

3. Return the aggregated data z

3.4.4 Classification of Joint Attacks

Finally, it is introduced to classify and mitigate joint attacks like version number attacks using the Q-learning strategy. The **Q-learning**, reinforcement learning technique is used to identify the malevolent nodes with high accuracy while having minimal overhead on the network. This strategy continuously learns and adapts to network conditions to identify malicious behaviors indicative of version number attacks. To improve the detection accuracy, the **Trickle timer algorithm** is integrated. This algorithm regulates the frequency of control message dissemination, which ensures that the network can rapidly and accurately detect attacks. Integrating **Q-learning with the trickle-time algorithm** achieves enhanced accuracy in detecting the version number attacks efficiently to safeguard the network while maintaining a low overhead. This approach improves the RPL network which remains resilient and secure against joint attacks and promotes reliable communication within the Internet of Everything environment.

i. Q-learning

Q-RPL algorithm 6 illustrates our suggested method of detection. When the node receives the DIO packet, the Q-RPL Protocol operation is executed. Each node's information is kept in a static array variable to ensure data integrity. Additionally, a global variable $Qlist\ [MAX\_NODE]$ has been created. It contains the list of discount$_s$, together with the disregarding factor of the. Every node keeps track of the number of DIO communications it receives from other nodes in

120 seconds. The client uses UDP packets to transmit this data to the server once every 120 seconds. To choose the best routing routes based on past performance data, nodes utilize Q-Learning, a reinforcement learning model. Figure 3 illustrates the iterative process by which nodes modify their route choices in response to rewards from earlier activities. Every node monitors the total rewards earned from the routes it has traveled, adjusting its policy to prioritize routes that provide greater utility (low latency, energy efficiency, little interference). The system is self-learning; nodes automatically adjust to changing network conditions by exchanging Q-values, which enhances overall route selection and reduces error margins without requiring centralized management.

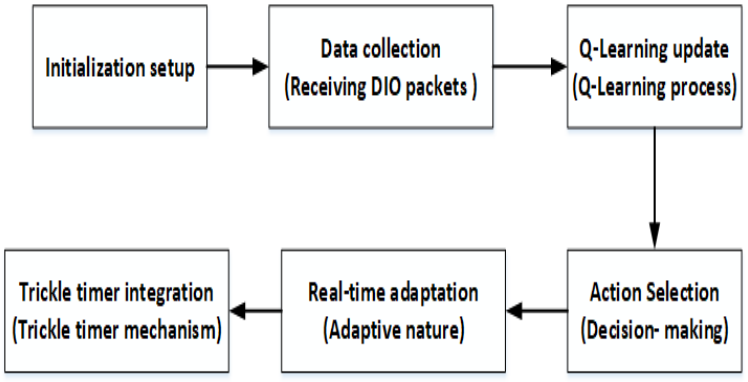

Fig. 3. Iterative Q-Learning Process for Route Selection in RPL.

Nodes can respond to changes and any interruptions in real time by using reinforcement learning. IoT networks are greatly impacted by this adaptive learning process, which improves network stability, especially in situations of congestion or targeted assaults.

Algorithm 6: Q-RPL: version number attack detection.

1. $Node_M, Qlist\ [MAX\_NODE]$ Global variable
2. dio_{count}, Ver_count static variable for each node
3. $\tau_i \leftarrow [ver_{num}, timestamp, node_{id}]$ global variables
4. $Procedures\ Q - RPLproc$
5. $\emptyset_{currentime} \leftarrow systemtime()$
6. If $\vartheta == 0$ then check if the list is empty
7. $\vartheta = \vartheta + 1$ Increment the variables
8. Call allocates list () procedure static procedure. Initialize the structure for the node.
9. End if
10. $DIO_{coming} = from$ get the node ID from where DIO coming
11. For $i \leftarrow 1$ to Max_NODE do
12. If $Node_S[\tau_i.from] == source_{ip}$ then
13. $\delta \leftarrow Node_S[DIO_{coming}].version$ get the current version of node
14. $\Delta \leftarrow dio.version$ get version from DIO message
15. If $\Delta - \delta > 0$ then the version changed
16. $Node_S[DIO_{coming}] = Node_S[i].dio_{count} + Node_S[i].ver_{count}$

Updating the node list value for the node for detection with the penalty

17. Else $Node_S[DIO_{coming}] = Node_S[i].\ dio_{count}$ updating the node list value without penalty
18. End if
19. End if
20. End for
21. End process

TABLE I. DESCRIPTIVE OF VARIABLES.

Variable Name	Description
$Node_M$	Nodes in the network
$Qlist$ [ ]	The incoming DIO count with penalties for each node is stored in a global list.
dio_{count}	Store count of DIO
ver_{count}	Increments when the version changes
Version number	keeps the most recent version of the node
τ_i	Stores the $timestamp,node_{-id}$
$\emptyset_{currentime}$	obtains the system time as of right now.
ϑ	To start the list
DIO_{coming}	Obtain the inbound DIO's Node-ID.
MAX_{NODE}	Total node count inside the system
δ	keeps the version number updated.
Q_table_server [][]	depicts the condition of the network using the DIO count.

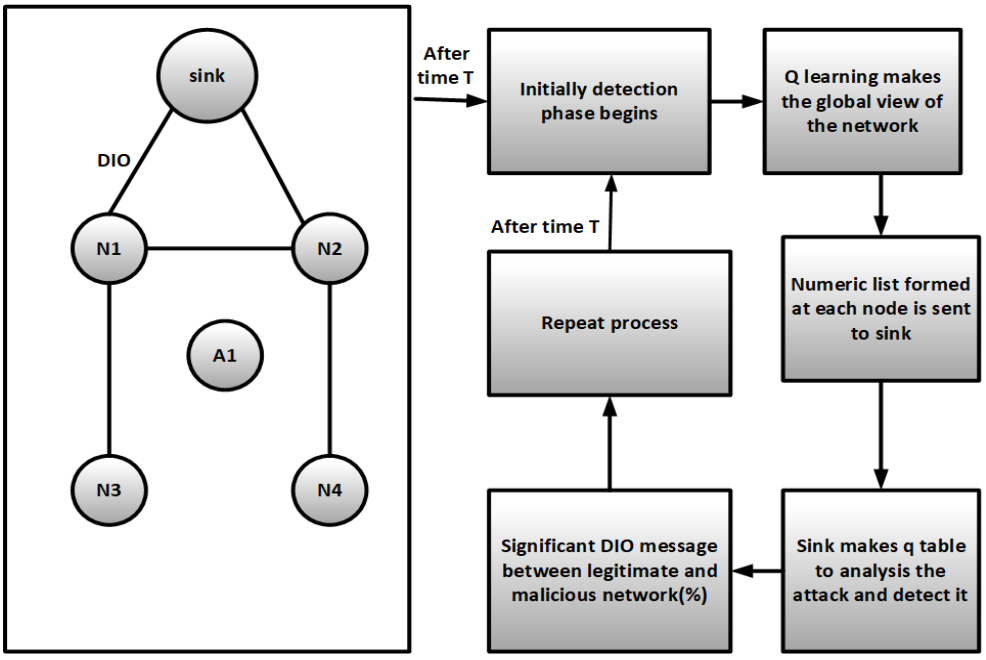

Fig. 4. Detection and Isolation of Version Number Attack.

The parameters and their descriptions used in the algorithms are shown in Table I. Following filtering the client data; the server creates a graph that shows the inbound and outgoing DIO messages among the nodes. While the server is the root node or a border router, the data collected from the client nodes are Z1 nodes. Figure 4 shows how version number assaults in RPL are identified and isolated by CIDS. By altering a node's version number, these attacks interfere with the construction of DODAG. Nodes keep a watch on the version history of the DIO messages sent by their parent nodes. In the CIDS, anomalies in version increments cause an alert, marking the questionable node for additional examination. Affected nodes are separated as soon as they are detected, preserving route integrity for the system. By using consistency thresholds to isolate compromised nodes, the detection system determines penalties for inconsistent version updates.

ii. Trickle-timer algorithm

The goal of the trickling timer method is to modify the DIO transmission frequency in response to changes in the network. The standard states that when a network discrepancy is found, the communication rate of the DIO messages increases. The

identification of network loops, a node failing or entering the network, and other events are among the inconsistent notifications. The trickling timer algorithm's brief flow is explained as follows:

1. Set J to a value in $[J_{min}, J_{max}]$ to begin the initial time slot
2. The initial slot by environment d=0,$s_{timer} = [\frac{J}{2}, J]$ is a point taken arbitrarily in the interval, and each time slot steps at J.
3. When the trickle time obtains consistent messages let d+=1
4. At the timer the trickle timer checks if there is d<k and only permits packets to be directed if d<k
5. When J expires, make J*2, if J*2>J_{max} , set J = J_{max}
6. The trickling timer will reset itself if it detects a conflicting message.

Algorithm 7: trickle time based on RL.

Input: $\leftarrow J_{min}, d_k \leftarrow k, t_n \leftarrow 0, d_n^m \leftarrow 0, m \leftarrow 1, incon_n^{m-1} \leftarrow 0, reward \leftarrow 0, \Delta Q \leftarrow 0$, Q table$\leftarrow 0$

1. $Random\ time: use\quad s_n^m = \left[t \times \frac{j}{m}, (t+1) \times J/m\right]$ *to* estimate s_n^m
2. $Receive\ consistent\ DIO\ then: d_n^m +=1$
3. $Receiving\ an\ inconsistent\ DIO\ then: J \leftarrow J_{min}, DIOSent_n \leftarrow 0, d_n^m \leftarrow 0, m \leftarrow 1, DIOcount_n^m \leftarrow 0, incon_n^{m-1} \leftarrow 1,$
4. while *the random time* s_n^m *expires* : do
5. select *a random number* among [0, 1] $(rand)$ *to* exposure *and use*
6. if $rand \leq explore$ then
7. if $d_n^m < d_k$ then
8. $DIO\ transfer(t_1), DIO_n^{sent}$ ++
9. else
10. $DIO\ suppression(s_1)$
11. end if
12. else
13. $Use\ a_{current}^{optimal} = \arg max Q(t,a)$ *to* choose *the best action that has led to the highest cumulative reward in the past*
14. end if
15. end while
16. while the time interval expires, do
17. $Calculate\ s_n^m$ regarding *to* $s_n^m = \begin{cases} 1 - incon_n^m & if\ t = t_0 \\ incon_n^m & if\ t = t_1 \end{cases}$
18. $Calculate\ \Delta Q\ according\ to$ $\Delta Q(t,a) = \{s_n^m(t,a) + \tau \times \max(t,a)\} - \Delta Q(t,a)$
19. $Update\ the\ Q-table\ according\ to$ $Q^{new}(t,a) = Q(t,b) + b \times \Delta Q(t,a)$
20. $Update\ j: j \longleftarrow j * 2$
21. if j > $J_n ay$ then
22. J $\longleftarrow J_n ay$
23. end if
24. if $DIOCount_n^m = 0$ then
25. $d_k \leftarrow k$
26. else
27. $Calculated_k\ using$ $$d_k = \frac{\sum_{j=1}^{m} DIOcount_n^m}{m}$$
28. end if
29. m+ = 1
30. $DIOCount_n^m += d_n^m$
31. $incon_n^{m-1} \longleftarrow 0$
32. end while

Where,
$s_n^m = reward$

$\Delta Q(t, a)$ is a better learning estimate for a certain pair of states and actions. Algorithm 7 demonstrates the trickle time-based RPL algorithm.
Version number assaults produce significant congestion and interfere with network routes. For IoT applications with strict reliability requirements, this detection method is crucial because it isolates rogue nodes, maintaining route stability and guaranteeing continuous data transfer.

4. EXPERIMENTAL RESULTS

This section presents the experimentation analysis and performance evaluation of the suggested study plan. This part is divided into three subsections: research overview, comparative analysis, and simulation study.

A. Simulation Setup

To simulate the proposed research method, Contiki-3.x with Cooja simulator is utilized. This tool has an efficient network topology and provides all specifications for the proposed technique. Table II indicates the system specifications. Table III shows the simulation parameter.

TABLE II . SYSTEM SPECIFICATION.

Software specification	OS	Ubuntu 20.04
	Network simulator	Contiki-3.x with Cooja simulator
Hardware specification	RAM	4 GB
	hard disk	500 GB

TABLE III. SIMULATION PARAMETER.

Parameters		**Descriptions**
Network parameters	IoT nodes	100
	Routers(sink)	4

B. Comparative analysis

This section contrasts the suggested approach with several existing ones, including Secure trust-based RPL routing in the mobile sink (SecRPL-MS)[20], security, mobility, trust-based model(SMTrust)[21], RPL attacks based on intrusion detection for effective routing (RAIDER) [25], flexible trickle algorithm based on RPL (RPL-FL) [33] assesses its efficiency using performance metrics like Number of nodes vs. accuracy (%), Number of nodes vs. precision (%), Number of nodes vs. energy consumption (mW), Number of nodes vs. throughput (%), Number of nodes vs. control message overhead, Number of nodes vs. time computing (s).

a. Number of Nodes vs. accuracy (%)

In a CIDS for detecting joint assaults in the RPL routing protocol, the number of nodes and accuracy (%) are correlated as follows:

$$\text{Accuracy}(\%) = \alpha \times \log(\text{number of nodes}) + \beta \quad (18)$$

Here

α, β is a constant

TABLE IV. NUMERICAL OUTCOMES OF ACCURACY (%).

(x-axis) – Number of Nodes	**Accuracy (%)- (y-axis)**		
	SecRPL-MS	**SMTrust-RPL**	**Proposed**
20	30	32	35
40	42	45	46
60	57	62	66
80	81	86	90
100	90	95	98

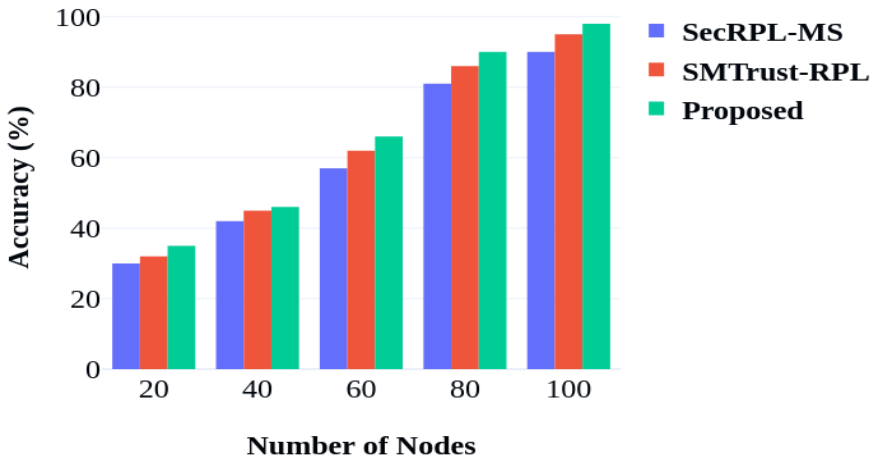

Fig. 5. Accuracy vs. Number of Nodes in CIDS.

Fig 5 and Table IV represent the number of nodes vs. accuracy and the numerical outcomes of accuracy (%). The accuracy for three routing protocols, SecRPL-MS, SMTrust-RPL, and the proposed method, was analyzed along various numbers of nodes. The proposed approach consistently outperforms the other two achieving 35% accuracy with 20 nodes and peaking at 98% with 100 nodes. In the comparison, SMTrust-RPL reached 95% at 100 nodes, while SecRPL-MS had the lowest performance starting at 30% with 20 nodes and reaching 90% with 100 nodes. This demonstrates the superior performance of the proposed approach across all the tested node quantities. The accuracy of the suggested CIDS protocol is much greater than the others; it starts at 35% with 20 nodes and reaches 98% with 100 nodes. Advanced approaches such as QWL-RPL for traffic management and the GBM algorithm for low computing burden during attack detection are responsible for the accuracy increases. In a variety of node settings, the suggested method maintains good accuracy. For real-world IoT applications, where scalability and dependability are crucial, particularly in hostile contexts where coordinated attacks may occur, this high accuracy rate is essential.

b. Number of Nodes vs. Precision (%)

In a CDIS for detecting joint assaults in the RPL routing protocol, the number of nodes and accuracy (%) have the following relationship:

$$\text{Precision}(\%) = \tau \times \log(\text{number of nodes}) + \gamma \tag{19}$$

τ, γ is a constant

TABLE V. NUMERICAL RESULTS OF PRECISION (%).

(x-axis) – Number of Nodes	Precision (%)- (y-axis)		
	SecRPL-MS	SMTrust-RPL	Proposed
20	20	23	25
40	39	42	45
60	69	72	76
80	81	85	90
100	85	90	95

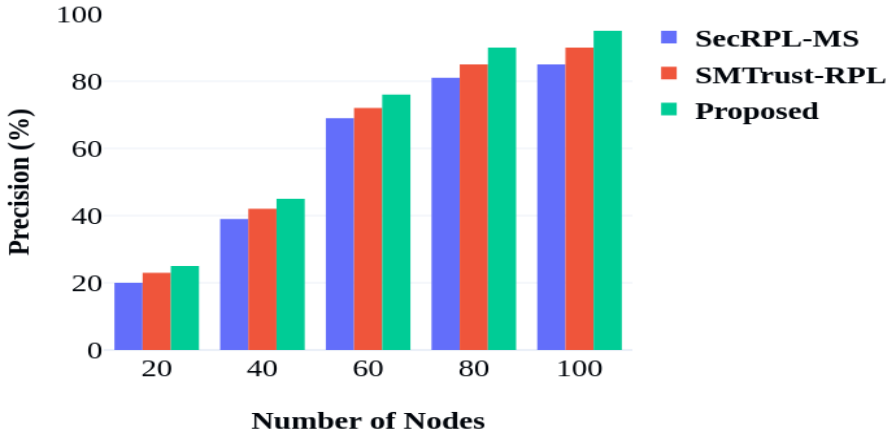

Fig. 6. Precision vs. Number of Nodes in CIDS.

Fig 6 and Table V represent the number of nodes vs. precision (%) and the numerical outcomes of the precision (%). The precision of SecRPL-MS, SMTrust-RPL, and a proposed approach was assessed over varying Nodes. The suggested approach consistently exhibited superior precision that starts at 25% with 20 nodes and reaches 95% at 100 nodes. SMTrust-RPL followed closely beginning at 23% and achieving 90% precision by 100 nodes. SecRPL-MS had the lowest precision starting at 20% and the maximum reach out is 80% with 100 nodes. This demonstrates that the suggested approach offers the highest precision across all the nodes. Because of its low-latency classification made possible by GBM and integrated fuzzy logic trust evaluation, the suggested CIDS shows exceptional precision, reaching 95% at 100 nodes. Here, precision is essential since it gauges how well the system can distinguish between malicious and benign nodes while reducing false positives. To prevent false alarms, which can waste network resources and affect dependability, CIDS must have a high accuracy rate. Because of its increased accuracy, the suggested approach may function dependably across extensive IoT networks, facilitating precise and consistent intrusion detection.

c. Number of Nodes vs. Energy consumption (mW)

In CDIS for detecting collaborative assaults in the RPL routing protocol, the number of nodes and energy usage (mW) are correlated as follows:

$$Energy\ consumption\ (mW) = \emptyset \times Number\ of\ nodes + \theta \quad (20)$$

$\theta, \emptyset$ is a constant

TABLE VI. NUMERICAL RESULTS OF ENERGY CONSUMPTION (mW).

(x-axis) – Number of Nodes	Energy consumption (mW) -(y-axis)		
	SecRPL-MS	SMTrust - RPL	Proposed
20	5	4	3
40	6	5	4
60	7	6	5
80	9	8	6
100	10	9	7

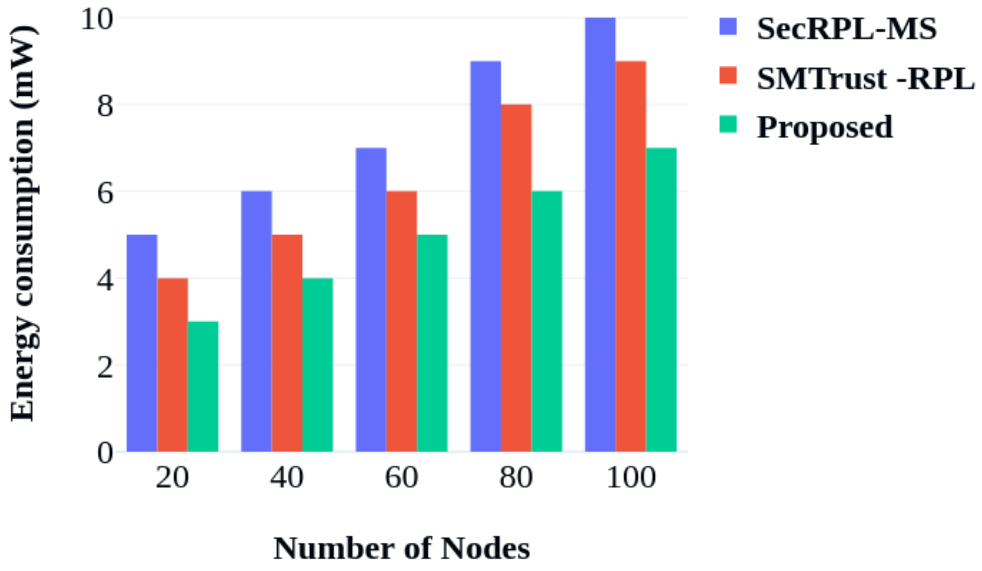

Fig. 7. Energy consumption vs. Number of Nodes in CIDS.

Fig. 7 and Table VI represent the number of nodes versus energy consumption (mw) and the numerical results of energy consumption (mw). The energy consumption of SecRPL-MS, SMTrust-RPL, and the suggested approach were analyzed on different nodes. The suggested method shows a lower energy consumption that starts at 2 mW with 20 nodes and gradually increases to 7 mW at 100 nodes. The SMTrust-RPL exhibits moderate energy consumption that starts with 2 mW and increases to 9 mW by 100 nodes. SecRPL-MS had the highest energy consumption, beginning at 1mW and reaching 10mW at 100 nodes. This recommends that the proposed approach is more energy efficient compared to the other 2 protocols as the number of nodes upsurges.

Energy consumption with the suggested approach peaks at 7 mW with 100 nodes after beginning at 3 mW with 20 nodes. This energy efficiency results from efficient attack detection with low resource requirements and intelligent traffic management via load balancing. Effective energy use is essential for Internet of Things devices running on limited power. The suggested CIDS system is ideal for long-term IoT deployments because of its low energy consumption, which preserves longer network uptime while striking a balance between high detection precision and lower energy consumption.

d. Number of nodes vs. throughput (%)

For detecting joint assaults in the RPL routing protocol, the number of nodes and throughput (%) have the following relationship:

$$\text{Throughput}(\%) = \tau \times \log(\text{number of nodes}) + \gamma \quad (21)$$

τ, γ is a constant

TABLE VII. NUMERICAL RESULTS OF PERFORMANCE (%).

(x-axis) – Number of nodes	Throughput (%) - (y-axis)		
	SecRPL-MS	SMTrust-RPL	Proposed
20	10	15	20
40	15	30	40
60	40	50	60
80	70	80	86
100	90	95	96

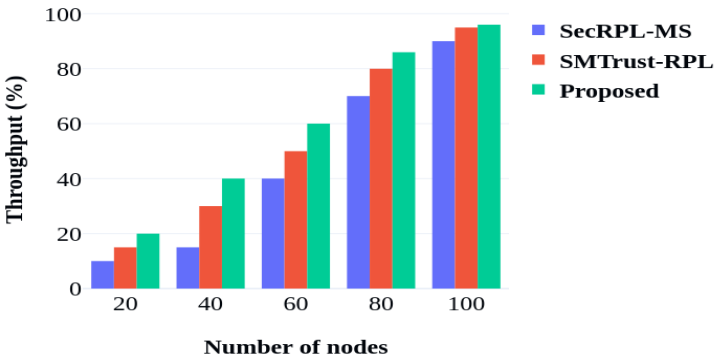

Fig 8. Throughput vs. Number of Nodes in CIDS.

Fig 8 and Table VII show the number of nodes vs. throughput (%) and the numerical outcomes of throughput (%). The throughput of SecTrust, SMTrust-RPL, and the suggested method was evaluated across several nodes. The suggested approach consistently shows the highest throughput starts at 20% at 20 nodes and reaches 96% at 100 nodes. SMTrust-RPL followed beginning at 15% and reaching out 95% at 100 nodes. SecTrust had the lowest throughput starting at 10% and achieving 90% at 100 nodes. This demonstrates that the proposed approach delivers the highest throughput across all tested nodes compared to other methods.

The suggested strategy makes use of data aggregation techniques and improved load distribution using WRF-RPL to sustain good throughput even as the number of nodes rises. This procedure increases the rate of successful data packet delivery, minimizes duplicate data transfers, and saves energy. For Internet of Things applications that need to exchange data in real time, high throughput is crucial. The efficiency of the suggested strategy in managing massive IoT data needs without sacrificing speed or reliability is confirmed by its capacity to maintain throughput under heavy load situations.

e. Number of Nodes vs. control message overhead

For detecting joint assaults in the RPL routing protocol, the number of nodes and control message overhead have the following relationship:

$$Control\ message\ overhead = \alpha \times \log(\text{number of nodes})^2 + \beta \quad (22)$$

TABLE VIII. NUMERICAL OUTCOMES OF CONTROL MESSAGE OVERHEAD.

(x-axis) – Number of nodes	control message overhead- (y-axis)		
	RAIDER	SMTrust-RPL	Proposed
20	1000	1200	1500
40	1500	2400	2600
60	2000	3400	3700
80	2500	3500	3900
100	3000	3600	4000

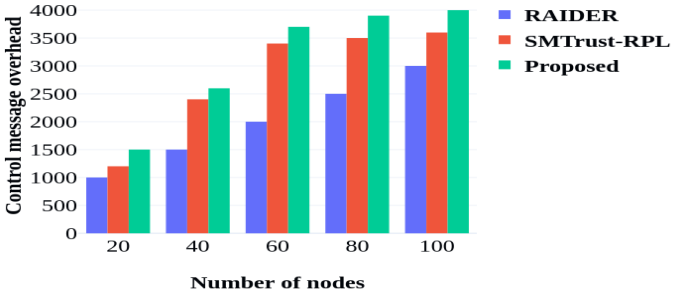

Fig. 9. Control Message Overhead vs. Number of Nodes in CIDS.

Fig 9 and Table VII show the number of nodes vs. control message overhead and the numerical outcomes of control message overhead. The control message overhead of RAIDER, SMTrust-RPL, and the proposed approaches was assessed across different nodes. The suggested method with the highest overhead starts with 1500 messages at 20 nodes and rises to 4000 messages at 100 nodes. The SMTrust-RPL shows the moderate overhead begins at 1200 messages and reaches out to 3600 messages by 1000 epochs. The RAIDER had the lowest overhead starting with 1000 messages and increasing to 3000 messages at 100 nodes. This indicates that while the RAIDER and the SMTrust-RPL have lower control message overhead, the proposed work remains the most effective in terms of overhead across all the nodes.

Due to its many security features, such as real-time attack categorization and dynamic learning for coordinated attack detection, the suggested CIDS has a larger control message overhead than RAIDER and SMTrust-RPL. At 20 nodes, the overhead is 1500 messages; at 100 nodes, it is 4000. Energy and resource use are impacted by control message overhead. The suggested CIDS delivers better accuracy and precision at the expense of increased overhead. In networks where security and resistance to sophisticated assaults are more important than cutting costs, this trade-off is advantageous.

f. Number of Nodes vs. time computing(s)

For detecting joint assaults in the RPL routing protocol, the number of nodes and time computing (s) have the following relationship.

$$time\ computing(s) \quad \alpha \times \log(\text{number of nodes})^2 + \tag{23}$$

TABLE IX. NUMERICAL OUTCOMES OF TIME COMPUTING (s).

(x-axis) – Number of Nodes	Time computing(s)- (y-axis)		
	RPL-FL	SMTrust-RPL	Proposed
20	23	21	20
40	48	45	40
60	70	66	60
80	89	80	75
100	120	110	80

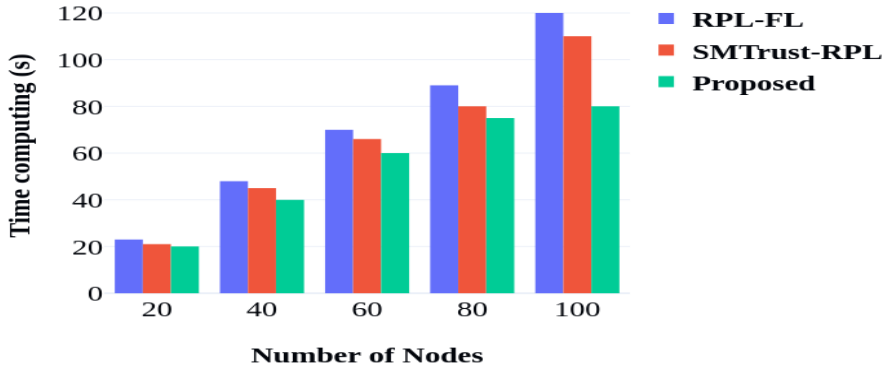

Fig. 10. Computation Times vs. Number of Nodes in CIDS.

Fig 10 and Table IX show the number of nodes vs. time computing(s) and the numerical outcomes of time computing(s). the time computing of RPL-FL, SMTrust-RPL, and the suggested method was measured over different nodes. The suggested approaches consistently show the shortest computing time starting at 20s with 20 nodes and increasing to 80 seconds with 100 nodes. The SMTrust-RPL had a moderate computing time starting at 21s and reaching 110s by 100 nodes. RPL-FL shows the longest computing time begins at 23s and escalates at 100 nodes. This demonstrates that the proposed approaches are the most effective across all the tested nodes. Minimal calculation times are demonstrated by the suggested CIDS, ranging from 20s at 20 nodes to 80s at 100 nodes. The use of adaptive learning techniques (Q-learning) and lightweight detection algorithms (like GBM) that lower processing demands is what makes this efficiency possible. Since computation time is essential for real-time applications, the suggested method's shorter computation time guarantees that the network can react to threats instantly. Quick response is crucial in time-sensitive situations, such as healthcare IoT systems, and this feature facilitates the adoption of CIDS. Table X represents the comparison of the proposed work with the existing work.

TABLE X. COMPARISON OF PROPOSED WORK VS EXISTING WORK.

Metric	Proposed CIDS	SMTrust-RPL	SecRPL-MS	RPL-FL	RAIDER
Accuracy (%)	35 - 98	32 - 95	30 - 90	-	-
Precision (%)	25 - 95	23 - 90	20 - 85	-	-
Energy Consumption (mW)	3 - 7	4 - 9	5 - 10	-	-
Throughput (%)	20 - 96	15 - 95	10 - 90	-	-
Control Message Overhead	1500 - 4000	1200 - 3600	-	-	1000 - 3000
Computation Time (s)	20 - 80	21 - 110	-	23 - 120	-

C. Discussion

Effective in resource-constrained contexts, the suggested Collaborative Intrusion Detection System (CIDS) for a large-scale Internet of Everything (IoE) network was created with computational complexity and scalability in mind. The system achieves processing efficiency by combining Q-learning with a Light Gradient Boosting Machine (GBM). GBM only considers high-gradient data, minimizing computational effort by eliminating less important aspects and using histogram-based methods to speed up processing and reduce memory requirements both of which are critical for IoT devices with limited resources. Being a model-free technique, Q-learning further enhances efficiency by enabling nodes to base their routing and detection choices on cumulative rewards that have been saved, eliminating the need for constant computation or large data storage.

Queue and Workload-Aware RPL (QWL-RPL), which controls load distribution across nodes to avoid congestion, improves system scalability. Together with Weighted Random Forward (WRF) routing, which is optimized using a genetic algorithm to guarantee that no one node is overloaded, this load-balancing technique maintains performance even as the network grows. Furthermore, as the number of nodes increases, a Trickle timer in conjunction with Q-learning modifies the frequency of control messages in response to network fluctuations, protecting bandwidth and preventing over-communication. By combining data at intermediate nodes, the Energy and Delay-Aware Data Aggregation (EDADA) feature makes an additional contribution. This lowers transmission collisions and conserves energy, two important factors in large-scale deployments with frequent data exchanges.

EDADA's compressed sensing theory, which reduces redundant data transmission and energy usage per node, is included in CIDS, which was designed with resource limits in mind. The decentralized processing strategy of the system, which fits well within the memory and processing capacity constraints of IoT nodes, removes the requirement for centralized computing by allowing each node to independently carry out routing and trust evaluations based on local information. In addition to distributing energy consumption evenly among nodes, load balancing prolongs the network's operating life by minimizing battery drain on any one node. All things considered, CIDS successfully strikes a compromise between high accuracy, scalability, and energy and computational economy, making it a reliable option for protecting large IoE networks when resources are limited.

The increased control message overhead produced by the Collaborative Intrusion Detection System (CIDS) is one of the study's main limitations. Even though this cost helps the RPL protocol identify joint assaults more accurately, it can strain network resources, especially in large-scale IoE installations. In contexts with limited resources, this additional cost may result in increased energy consumption and decreased network efficiency, which would impact network performance as a whole.

5. CONCLUSION

The improved method for improving security and efficiency in RPL-based IoE networks, EDADA-RPL, is presented. To maintain high throughput and accuracy while lowering network latency and energy consumption, EDADA-RPL combines energy-efficient data aggregation with workload-aware and queue-aware routing. Important additions include the use of Q-learning with a trickle timer to dynamically adjust to changing attack patterns like rank and wormhole assaults, and the incorporation of fuzzy logic for trust evaluation, which improves the system's capacity to detect and isolate compromised nodes. A graph for the following metrics that include the Number of nodes vs. accuracy is plotted, Number of nodes vs. energy consumption (mW), Number of nodes vs. throughput, Number of nodes vs. control message overhead, Number of nodes vs. precision, Number of nodes vs. time computing (s). Our approach performance is examined using numerical analysis, demonstrating that it performs better than the current methodologies across all measures. Finally, our methods perform better than the existing works.

Implications of this study suggest that IoE networks using the proposed CIDS will experience a more resilient RPL protocol, leading to sustained network performance even under complex attack scenarios. These findings offer practical applications in smart cities, healthcare IoT, and other critical infrastructures where secure, reliable communication is essential. Future work could focus on further optimizing control message overhead and validating the CIDS's performance across diverse IoT environments, thereby broadening its applicability.

Conflicts of interest

The author has no conflicts of interest relevant to this article.

Funding

The absence of funding details in the author's paper suggests that the research was entirely self-funded.

Conflicts Of Interest

The paper states that there are no personal, financial, or professional conflicts of interest.

References

[1] A. Jamalipour and S. Murali, "A taxonomy of Machine-learning-based Intrusion Detection Systems for the Internet of Things: A survey," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9444–9466, 2021.

[2] N. Alfriehat, M. Anbar, S. Karuppayah, S. D. A. Rihan, B. A. Alabsi, and A. M. Momani, "Detecting Version Number Attacks in Low Power and Lossy Networks for Internet of Things Routing: Review and Taxonomy," IEEE Access, 2024.

[3] P. D. Acevedo, D. Jabba, P. Sanmartin, S. Valle, and E. D. Nino-Ruiz, "WRF-RPL: Weighted Random Forward RPL for High Traffic and Energy Demanding Scenarios," IEEE Access, vol. 9, pp. 60163–60174, 2021.

[4] P. Shahbakhsh, S. H. Ghafouri, and A. K. Bardsiri, "RAARPL: End-to-end Reliability-Aware Adaptive RPL Routing Protocol for the Internet of Things," International Journal of Communication Systems, vol. 36, no. 6, e5445, 2023.

[5] V. C. Farias da Costa, L. Oliveira, and J. de Souza, "Internet of Everything (IoE) Taxonomies: A Survey and a Novel Knowledge-based Taxonomy," Sensors, vol. 21, no. 2, p. 568, 2021.

[6] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a Revolutionary Approach for Future Technology Enhancement: a Review," Journal of Big Data, vol. 6, no. 1, pp. 1–21, 2019.

[7] R. Sahay, A. Nayyar, R. K. Shrivastava, M. Bilal, S. P. Singh, and S. Pack, "Routing Attack-Induced Anomaly Detection in IoT Network using RBM-LSTM," ICT Express, 2024.

[8] H. S. Kim, J. Paek, D. E. Culler, and S. Bahk, "PC-RPL: Joint Control of Routing Topology and Transmission Power in Real Low-Power and Lossy Networks," ACM Trans. Sens. Netw., vol. 16, no. 2, pp. 1–32, 2020.

[9] S. Garg, D. Mehrotra, H. M. Pandey, and S. Pandey, "Static to Dynamic Transition of RPL Protocol from IoT to IoV in Static and Mobile Environments," Cluster Computing, vol. 26, no. 1, pp. 847–862, 2023.

[10] M. Zaminkar, F. Sarkohaki, and R. Fotohi, "A method based on Encryption and Node Rating for Securing the RPL Protocol Communications in the IoT Ecosystem," International Journal of Communication Systems, vol. 34, no. 3, e4693, 2021.

[11] S. T. Liu and S. D. Wang, "Improved Trickle Algorithm Toward Low Power and Better Route for the RPL Routing Protocol," IEEE Access, vol. 10, pp. 83322–83335, 2022.

[12] B. Varastan, S. Jamali, and R. Fotohi, "Hardening of the Internet of Things by using an Intrusion Detection System based on Deep Learning," Cluster Computing, pp. 1–24, 2023.

[13] J. Wu, Y. Wang, H. Dai, C. Xu, and K. B. Kent, "Adaptive Bi-Recommendation and Self-Improving Network for Heterogeneous Domain Adaptation-Assisted IoT Intrusion Detection," IEEE Internet of Things Journal, 2023.

[14] I. Zaatouri, N. Alyaoui, A. B. Guiloufi, F. Sailhan, and A. Kachouri, "Design and Performance Analysis of Objective Functions for RPL Routing Protocol," Wireless Personal Communications, vol. 124, no. 3, pp. 2677–2697, 2022.

[15] K. A. Darabkh and M. Al-Akhras, "Improving Routing Protocol for Low-Power and Lossy Networks over IoT Environment," in 2021 30th Wireless and Optical Communications Conference (WOCC), 2021, pp. 31–35.

[16] A. Thakkar and R. Lohiya, "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges," Archives of Computational Methods in Engineering, vol. 28, no. 4, pp. 3211–3243, 2021.

[17] A. Bang and U. P. Rao, "Performance Evaluation of RPL Protocol Under Decreased and Increased Rank Attacks: A Focus on Smart Home use-Case," SN Computer Science, vol. 4, no. 4, p. 329, 2023.

[18] D. Paganraj and M. Chelliah, "DE2RA-RPL: Detection and Elimination of Resource-related Attacks in IoT RPL-based Protocol," *The Journal of Supercomputing*, pp. 1–31, 2024.

[19] I. E. Lakhlef, B. Djamaa, M. R. Senouci, and A. Bradai, "Enhanced Multicast Protocol for Low-power and Lossy IoT Networks," IEEE Sensors Journal, 2024.
[20] B. Rakesh, "Novel Authentication and Secure Trust-based RPL Routing in Mobile Sink Supported Internet of Things," Cyber-Physical Systems, vol. 9, no. 1, pp. 43–76, 2023.
[21] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, M. Humayun, A. O. Ibrahim, and A. Abdelmaboud, "A Trust-based Model for Secure Routing against RPL Attacks in Internet of Things," Sensors, vol. 22, no. 18, p. 7052, 2022.
[22] C. D. Morales-Molina et al., "A Dense Neural Network Approach for Detecting Clone ID Attacks on the RPL Protocol of the IoT," Sensors, vol. 21, no. 9, p. 3173, 2021.
[23] J. F. Yonan and N. A. A. Zahra, "Node Intrusion Tendency Recognition Using Network Level Features Based Deep Learning Approach," BJN, vol. 2023, pp. 1–10, Jan. 2023.
[24] M. Shirafkan, A. Shahidienjad, and M. Ghobaei-Arani, "An Autonomous Intrusion Detection System for the RPL Protocol," Peer-to-Peer Networking and Applications, vol. 15, no. 1, pp. 484–502, 2022.
[25] D. B. Gothawal and S. V. Nagaraj, "An Intelligent and Lightweight Intrusion Detection Mechanism for RPL Routing Attacks by Applying Automata Model," Information Security Journal: A Global Perspective, vol. 32, no. 1, pp. 1–20, 2023.
[26] M. Shirafkan, A. Shahidinejad, and M. Ghobaei-Arani, "An Intrusion Detection System using Deep Cellular Learning Automata and Semantic Hierarchy for Enhancing RPL Protocol Security," Cluster Computing, vol. 26, no. 4, pp. 2443–2461, 2023.
[27] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, M. S. Hossain, and A. Yassine, "Trust and Mobility-based Protocol for Secure Routing in Internet of Things," Sensors, vol. 22, no. 16, p. 6215, 2022.
[28] N. Zahedy, B. Barekatain, and A. A. Quintana, "RI-RPL: A New High-quality RPL-based Routing Protocol using Q-learning Algorithm," The Journal of Supercomputing, vol. 80, no. 6, pp. 7691–7749, 2024.
[29] A. O. Bang and U. P. Rao, "EMBOF-RPL: Improved RPL for Early Detection and Isolation of Rank Attack in RPL-based Internet of Things," Peer-to-Peer Networking and Applications, vol. 15, no. 1, pp. 642–665, 2022.
[30] A. S. Bin Shibghatullah, "Mitigating Developed Persistent Threats (APTs) through Machine Learning-Based Intrusion Detection Systems: A Comprehensive Analysis," SHIFRA, vol. 2023, pp. 17–25, Mar. 2023, doi: 10.70470/SHIFRA/2023/003.
[31] M. Nazaralipoorsoomali, P. Asghari, and S. H. H. S. Javadi, "Performance Improvement of Routing Protocol for Low-power and Lossy Networks Protocol in an Internet of Things-based Smart Retail System," International Journal of Communication Systems, vol. 35, no. 10, e5166, 2022.
[32] M. Ezhilarasi, L. Gnanaprasanambikai, A. Kousalya, and M. Shanmugapriya, "A Novel Implementation of Routing Attack Detection Scheme by using Fuzzy and Feed-Forward Neural Networks," Soft Computing, vol. 27, no. 7, pp. 4157–4168, 2023.
[33] H. Lamaazi and N. Benamar, "A Novel Approach for RPL Assessment based on the Objective Function and Trickle Optimizations," Wireless Communications and Mobile Computing, vol. 2019, no. 1, p. 4605095, 2019.
[34] L. Hussain, "Fortifying AI Against Cyber Threats Advancing Resilient Systems to Combat Adversarial Attacks," EDRAAK, vol. 2024, pp. 26–31, Mar. 2024, doi: 10.70470/EDRAAK/2024/004.