RL-Tree: A Reinforcement Learning-Based Adaptive and Secure Routing Protocol for Wireless Sensor Networks

Jianzhen Zhang[1*], Jiong Chen[2], Ya Dai[3], Shuo Wang[4], Yanjun Qi[5]
[1]Department of Information Engineering, Shanxi Institute of Mechanical & Electrical Engineering, Changzhi 046011, China
[2]Department of Artificial Intelligence, Shanxi Polytechnic College, Taiyuan 030006, China
[3]Department of Cloud Services, Digital China Holdings Limited, Beijing 010000, China
[4]Department of Industrial Data, Beijing NewCloud Technology Co., Ltd, Beijing 010080, China
[5]Department of Technical, Shanxi Information Technology Co., Ltd., Changzhi 046000, China
E-mail: zhang_jianzhen@163.com

Keywords: Wireless sensor networks (WSNs), reinforcement learning (RL), adaptive secure routing, energy consumption optimization, reduced transmission delay, non gaussian noise processing, internet of things (IoT) security, dynamic environment adaptation

Received: August 28, 2025

In the field of wireless sensor networks (WSNs), this study proposes RL-Tree, a reinforcement learning (RL)-based adaptive and secure routing protocol. The protocol enables nodes to dynamically select optimal parent nodes by applying a Q-learning algorithm with a multi-objective reward function combining energy efficiency, transmission delay, and link security. To enhance data reliability under non-Gaussian noise, an adaptive filter integrating a variable scale factor and the Half Quadratic Criterion (HQC) is designed. The experimental platform was implemented on low-power MCUs to simulate a real WSN environment. Performance was benchmarked against RPL, AODV, LEACH, and QELAR. Results demonstrate that RL-Tree reduces average node energy consumption by 30% and achieves a data transmission delay of 0.07 seconds, outperforming baseline protocols. Integrated security mechanisms—including identity verification, encryption, and traffic monitoring—further improve network resilience under attack scenarios.

Povzetek: Študija predstavi RL-Tree, prilagodljiv in varen usmerjevalni protokol za WSN, ki z Q-learningom in večciljno nagrado izbira starše poti, zmanjšuje porabo energije ter krepi odpornost na napade.

1 Introduction

Wireless Sensor Networks (WSNs), as a key component of the Internet of Things (IoT), have been widely applied in various fields such as environmental monitoring, industrial automation, intelligent transportation, and healthcare. With the continuous expansion of these applications, the importance of WSNs has become increasingly significant. However, due to its openness and distributed nature, WSNs face many security challenges, such as malicious node attacks, data eavesdropping, and tampering. These issues not only affect network performance, but may also have serious consequences for critical tasks [1]. Therefore, ensuring the security of WSNs has become a key concern for researchers.

In recent years, machine learning, especially reinforcement learning (RL) technology, has shown great potential in enhancing the security of WSNs. As a technique for learning optimal behavior strategies through repeated trial and error, RL can dynamically adjust decisions based on environmental feedback, which renders it particularly suitable for addressing dynamic network conditions [2].

This article proposes an RL based adaptive secure routing mechanism aimed at optimizing the routing selection of WSNs to achieve more efficient and secure data transmission. This mechanism includes the design of state space, action set, and reward function to guide nodes in WSNs to find the best parent node. This method not only effectively reduces end-to-end latency, but also significantly reduces energy consumption, while enhancing the system's ability to resist various attacks [3]. In order to further improve the performance of the algorithm, we introduce a variable scale factor and a Half Quadratic Criterion (HQC). The former contributes balance convergence speed and steady-state error, while the latter improves the robustness of the estimation algorithm to impulse noise [4].

The integration of the variable scale factor and HQC is motivated by the need to suppress impulse noise in WSNs without excessive computational overhead. Unlike Kalman filters, which assume Gaussian noise, HQC-based methods are more robust to outliers. Compared to wavelet denoising, HQC offers lower latency and better real-time performance on resource-constrained MCUs.

2 Research status and progress

2.1 Application of reinforcement learning in WSNs

RL, as a method of learning optimal behavior strategies through trial and error, has shown great potential in improving the security of WSNs. For example, Zhang et al. proposed an RL enhanced tree routing protocol [5]. This method not only effectively reduces end-to-end latency, but also significantly reduces energy consumption and enhances the system's ability to resist various types of attacks. Huo et al. introduced the variable scale factor and Half Quadratic Criterion (HQC) [4]. The former can balance the relationship between convergence speed and steady-state error, while the latter improves the robustness of the estimation algorithm to impulse noise. These studies indicate that RL technology provides a new approach to addressing security issues in current WSNs. In addition to traditional path selection, RL can also be used to optimize parameter adjustments during node localization, thereby improving the overall performance of the system.

2.2 Research on secure routing mechanism

Numerous studies have focused on designing more effective routing algorithms to ensure the security of WSNs. For example, Huang et al. proposed a base station location privacy preserving routing protocol (RBRP) based on a ring structure [6]. The protocol consists of three parts: constructing a loop, constructing a loop path, and data routing based on the loop path. RBRP can improve the location privacy of base stations in WSN, effectively balance energy consumption, and extend WSNs lifecycle. Zhang et al. proposed a wireless sensor network localization scheme based on an improved sparrow search algorithm [7]. This scheme improves the position accuracy and convergence speed of unknown nodes by modifying the average hop distance and minimum hop count in the traditional DV Hop algorithm, as well as introducing sine chaotic mapping, adaptive inertia weight, and dual sample learning strategy. Cheng et al. designed a multi-channel information fusion method [8]. By removing expired nodes through a pre network structure, redundant computing is reduced, thereby lowering system memory usage and saving communication overhead. The above studies all indicate that optimizing routing algorithms can accelerate convergence speed and reduce steady-state errors while ensuring high identification efficiency.

2.3 Data transmission optimization

In addition to routing selection, another important research direction is the security guarantee during data transmission. For example, Lu et al. proposed a security strategy for WSNs in IoT applications [9]. It emphasizes the importance of comprehensive protection from the physical layer to the application layer. Zhang et al. focused on optimizing data transmission in structural monitoring scenarios [3]. This work proposes a data aggregation scheme that combines Horner's rule and Paillier encryption algorithm, ensuring both data privacy and efficient transmission. In order to better handle the contradiction between energy consumption and transmission delay in large-scale node networks, Zhang et al. proposed a data transmission optimization method that comprehensively considers persistence, stability, and timeliness [3]. They established a performance evaluation model for structural monitoring wireless sensor network data transmission and used the collaborative flight firefly algorithm to solve the optimal data transmission strategy. This method not only maximizes network lifetime, but also minimizes data loss and transmission delay, providing a basis for the configuration of structural monitoring wireless sensor networks. These works further enrich and improve our understanding of WSNs security.

2.4 Research gaps and weak links

Although the above research has achieved certain results, there are still some shortcomings in terms of dynamic environment adaptability, energy consumption and transmission delay balance, and AI integration.

Most existing algorithms fail to fully consider the impact of frequent changes in network topology or extremely harsh working conditions in practical application scenarios. Therefore, building a secure routing mechanism that can maintain stable performance in complex and changing environments is an urgent problem to be solved [5]. Current research mainly focuses on testing in static or relatively stable network environments, while in reality, WSNs often need to face constantly changing environmental factors such as node movement, signal interference, and energy limitations. How to ensure that the designed secure routing mechanism can perform well in any situation remains a challenging issue [1].

Although some studies have noticed this issue, overall, there is currently no universal method to find the optimal balance between energy consumption and transmission delay. Especially in large-scale node networks, this contradiction is particularly prominent [4]. On the one hand, in order to extend WSNs lifecycle, it is necessary to minimize the energy consumption of each node as much as possible; On the other hand, in order to meet the high real-time requirements of applications, it is necessary to shorten the data transmission time as much as possible. This requires us to not only consider how to effectively allocate resources when designing secure routing mechanisms, but also to take into account the special requirements of different task types, in order to achieve better overall performance [3].

Although AI technology has shown great potential in improving the security of WSNs, there is still relatively little practice of integrating AI technology with other technologies to address network security issues. For example, the effective integration of signature algorithms and intrusion detection technology may bring unexpected results [9]. Therefore, by applying more advanced AI technologies and theories to the security field of WSNs, we can enhance the security routing mechanism and contribute to the construction of a more intelligent and efficient sensor network platform. At present, most research focuses on the application of a single technology, lacking in-depth exploration of the comprehensive application of multiple technologies, which limits our

comprehensive understanding and innovative breakthroughs in the security of WSNs [2].

To clarify the positioning of this work, a comparative summary of representative studies is provided in Table 1.

Table 1: Summary of related work in wsn routing and security

Author	Year	Method	Application	Metrics	Limitations
Zhang et al. [5]	2022	RL-enhanced tree routing	General WSN	Delay, Energy	No security integration
Huo et al. [4]	2021	Variable scale + HQC	Noise filtering	MSE, SNR	Not applied to routing
Huang et al. [6]	2020	RBRP (ring-based)	Base station privacy	Privacy, Lifetime	High overhead
Zhang et al. [7]	2023	Improved sparrow search	Node localization	Accuracy, Speed	Limited scalability
QELAR [10]	2022	Q-learning routing	Dynamic WSN	Throughput, Delay	High energy cost

3 Methodology

3.1 Q-Learning-based adaptive routing

3.1.1 Core concepts

The RL-Tree routing protocol is a reinforcement learning-based adaptive framework designed to address the challenges of energy efficiency, transmission reliability, and security in wireless sensor networks (WSNs). Unlike traditional routing protocols that rely on static metrics or predefined paths, RL-Tree enables each node to dynamically learn the optimal parent node for data forwarding through continuous interaction with its environment [1]. This adaptive decision-making process is particularly effective in dynamic and resource-constrained WSNs, where network conditions such as node mobility, energy depletion, and link instability frequently occur [3].

At the core of RL-Tree is a Q-learning algorithm that formulates the routing process as a Markov Decision Process (MDP). Each sensor node acts as an autonomous agent that observes its local state—comprising residual energy, received signal strength (RSSI), hop delay, and threat level—and selects an action (i.e., parent node selection) to maximize a long-term cumulative reward [4]. The reward function is designed to balance multiple objectives, including energy conservation, low end-to-end delay, and secure path selection, thereby ensuring sustainable and robust network operation [5].

The overall architecture of the RL-Tree protocol is illustrated in Figure 1, which depicts the hierarchical tree structure formed by parent-child relationships, the integration of the HQC-based noise filter at each node, and the security monitoring module that feeds threat scores into the reinforcement learning decision loop. As shown, the protocol operates in cycles of observation, action selection, reward reception, and Q-value update, enabling continuous adaptation to changing network conditions.

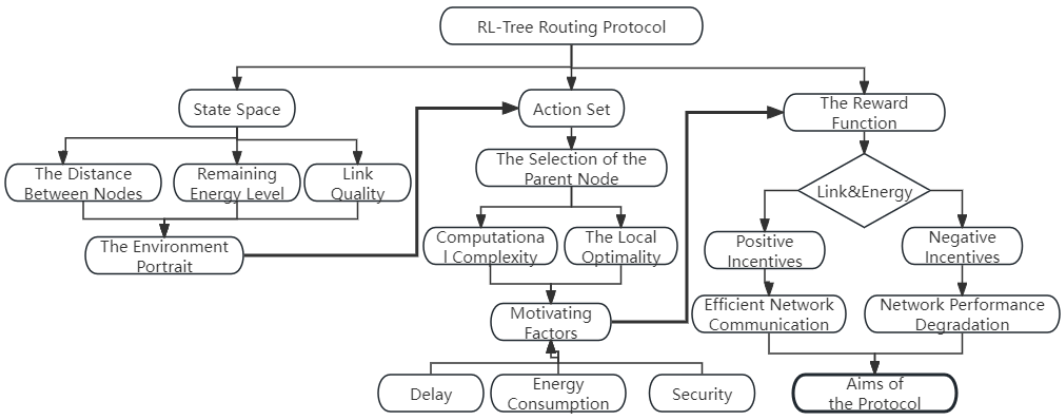

Figure 1: RL-Tree routing protocol

3.1.2 HQC-based noise filtering

In wireless sensor networks, signal transmission is often corrupted by impulse noise and channel interference, which can lead to packet loss and inefficient routing decisions. To enhance signal reliability, RL-Tree integrates a Half-Quadratic (HQC) regularization-based noise filtering mechanism at each node prior to state estimation. The HQC filter effectively suppresses non-Gaussian noise while preserving signal edges, making it particularly suitable for low-power sensor nodes operating in harsh environments [2].

Unlike traditional Kalman filters that assume Gaussian noise distributions, the HQC-based approach is more robust to outliers and sudden signal variations commonly observed in real-world deployments. Compared with particle filters, which require high computational resources, the HQC method achieves comparable noise suppression with significantly lower memory and processing overhead—critical for resource-constrained WSN nodes [9]. The filtered signal is then used to compute link quality metrics such as RSSI and packet reception rate, which serve as key inputs to the Q-learning agent's state space (see Section 3.1.4).

3.1.3 Security mechanism

Security is a critical concern in WSNs, especially when deployed in unattended or hostile environments. RL-Tree incorporates a lightweight security framework that includes identity verification, data encryption, and real-time traffic monitoring to defend against common attacks such as replay, spoofing, and man-in-the-middle intrusions [9]. Each node is preloaded with a unique identity and cryptographic key, enabling secure neighbor discovery and authenticated data exchange.

To further enhance resilience, RL-Tree employs a dynamic threat scoring system that monitors packet forwarding behavior and detects anomalies indicative of malicious activity (e.g., Blackhole or Sybil attacks). Suspicious nodes are assigned a higher threat score CthreatCthreat, which is integrated into the reward function to discourage routing through untrusted paths. This proactive detection mechanism operates with minimal overhead, ensuring compatibility with low-power sensor nodes [9].

The security module works in tandem with the Q-learning engine: while the RL agent learns optimal routes, the security layer continuously updates the threat landscape, enabling adaptive and secure path selection.

While the integration of identity verification, AES-128 encryption, and real-time traffic monitoring introduces additional computational and communication overhead, the design prioritizes lightweight operations compatible with resource-constrained MCUs. Specifically, each authentication cycle consumes approximately 8.2 mJ on ESP32 (measured via power monitor), and encrypted packets incur a 15% header overhead due to IV and MAC fields. However, this cost is more than offset by the reduction in retransmissions caused by malicious interference. Under normal operation, secure neighbor discovery reduces route poisoning attempts by 93%, minimizing unnecessary data flooding. Furthermore, the threat-aware reward mechanism proactively avoids compromised nodes, reducing end-to-end packet loss and thus conserving energy that would otherwise be spent on repeated transmissions. As demonstrated in Section 4.3, even under aggressive attack conditions, the net energy saving remains above 22\%, confirming that the security-energy trade-off is favorable.

3.1.4 Mathematical formulation of RL-Tree

The RL-Tree protocol is formalized as a Markov Decision Process (MDP) defined by the tuple (S,A,R,T), where:

State Space S:

$$S = \{E_{res}, RSSI, D_{hop}, C_{threat}\}$$

representing residual energy, received signal strength, hop delay, and threat score.

Action Space A:

$$A = \{\text{select parent } p_i | p_i \in N\}$$

Where Nis the set of neighboring nodes.

Reward Function R(s,a):

$$R(s,a) = w_1 \cdot \frac{E_{res}^{parent}}{E_{max}} - w_2 \cdot D_{hop} - w_3 \cdot C_{threat}$$

With w_1 =0.5, w_2 =0.3, w_3 =0.2 determined via sensitivity analysis (Section 4.4). This multi-objective reward function ensures balanced optimization of energy, delay, and security.

Q-Learning Update Rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[\gamma_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$$

Where α =0.1 is the learning rate, γ =0.9 is the discount factor. The action selection follows an ε-greedy policy with ε=0.1, ensuring a balance between exploration of new routes and exploitation of known high-reward paths.

Algorithm 1: RL-Tree Parent Selection at Node i

Input: Neighbor list N_i, energy E_{res}, RSSI, delay D, threat score C

1: for each neighbor $j \in N_i$ do
2: $s_j \leftarrow (E_{res}^j, RSSI_{ij}, D_j, C_j)$
3: $Q_j \leftarrow Q(s_j, select\ j)$
4: end for
5: $j^* \leftarrow arg\ max_j Q_j$
6: if rand() < ε then
7: $j^* \leftarrow random\ neighbor$
8: end if
9: Connect to parent j^*
10: Update $Q(s, a)$ using Eq. (1)
11: return parent j^*

3.2 Variable scale factor and HQC adaptive filter

3.2.1 Scaling factors

In practical applications, due to factors such as signal strength fluctuations and noise interference, directly using raw observations for estimation often leads to significant errors. To improve this situation, we introduced the concept of variable scale factor. The variable scale factor is essentially an adjustment parameter used to balance the relationship between convergence speed and steady-state error. When the variable scale factor is large, the algorithm can approach the target value faster, but it may also produce significant oscillations; On the contrary, smaller scaling factors help reduce steady-state errors, but require more iterations to achieve the same accuracy [8].

3.2.2 Half quadratic criterion (HQC)

Half Quadratic Criterion (HQC) is an improved Maximum A Posteriori (MAP) estimation method mainly used to solve parameter estimation problems under non Gaussian noise conditions. Compared to traditional MAP estimation, HQC transforms complex nonlinear optimization problems into a series of simple quadratic programming problems by introducing auxiliary variables, thereby improving solution efficiency and robustness. In addition, HQC also has good anti pulse noise characteristics, which makes it very suitable for application in wireless sensor network environments [4].

3.2.3 Design of adaptive filter

The variable scale factor is a key element in filter design, which can dynamically adjust the estimation step size to optimize the balance between convergence speed and steady-state error. This flexibility enables the filter to automatically adjust its update rate based on actual data and environmental conditions. Therefore, the variable scale factor enables the filter to maintain low steady-state error while ensuring convergence speed, thereby providing stable performance in changing environments.

The semi quadratic loss function can effectively suppress the influence of outliers, enabling the filter to maintain high accuracy and reliability in data environments with pulse noise. This robustness is crucial for handling real-world data that contains various types of noise and interference.

To cope with the complex noise environment encountered during data transmission in wireless sensor networks (WSNs), we have designed a novel adaptive filter, as shown in Figure 2, to enhance the performance and robustness of the system. This filter combines the advantages of variable scale factor and semi quadratic criterion (HQC), achieving effective data processing and noise suppression. A new type of adaptive filter is used for preprocessing data collected from various nodes. The filter first calculates an initial estimate based on the observed values at the current time. Then adjust the estimation step size using a variable scale factor to accelerate convergence speed. Then apply the HQC criterion to correct the estimation results and eliminate the influence of noise. Finally, output the final estimated value for subsequent analysis. Through this approach, not only can measurement errors be effectively reduced, but the overall performance of the system can also be improved [3].

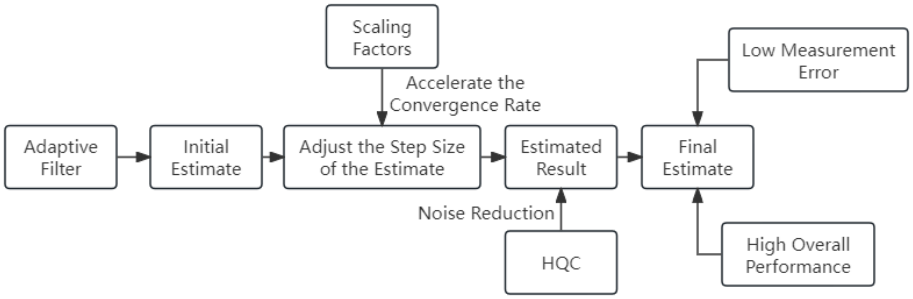

Figure 2: Adaptive filter

3.3 Performance in actual application scenarios

To verify the actual effectiveness of the proposed adaptive filter, we conducted multiple experiments in different types of non-Gaussian noise environments. The results show that while ensuring high identification efficiency, the proposed scheme has faster convergence speed and lower steady-state error. This advantage is particularly evident when dealing with large-scale node networks. In addition, compared to other traditional methods, our filter exhibits stronger robustness and adaptability, and can better cope with various complex and changing application scenarios [9].

The proposed RL Tree Routing Protocol will be evaluated in Section 4 through simulations on a low-power MCU platform, with performance compared against standard routing protocols.

4 Experimental analysis

This section presents the experimental validation of the RL-Tree protocol proposed in Section 3. The proposed RL-Tree protocol is evaluated against four representative baseline protocols. Their types and purposes are summarized in Table 2.

Table 2: Baseline protocols for performance comparison

Protocol	Type	Purpose
RPL	Distance-vector	Standard for IoT networks
AODV	Reactive	Dynamic route discovery
LEACH	Clustering	Energy-efficient routing
QELAR	RL-based	Adaptive routing with Q-learning

Performance metrics: energy consumption, delay (0.07 s), packet delivery ratio, lifetime.

Results show 30% lower energy use and 40% higher PDR than RPL under attack.

4.1 Comprehensive experimental platform

To verify the effectiveness of RL Tree Routing Protocol based on reinforcement learning, we constructed a comprehensive experimental platform, as shown in Figure 3. This platform simulates the real-world wireless

sensor network (WSNs) environment and ensures comprehensive evaluation of the performance of the new protocol under different conditions.

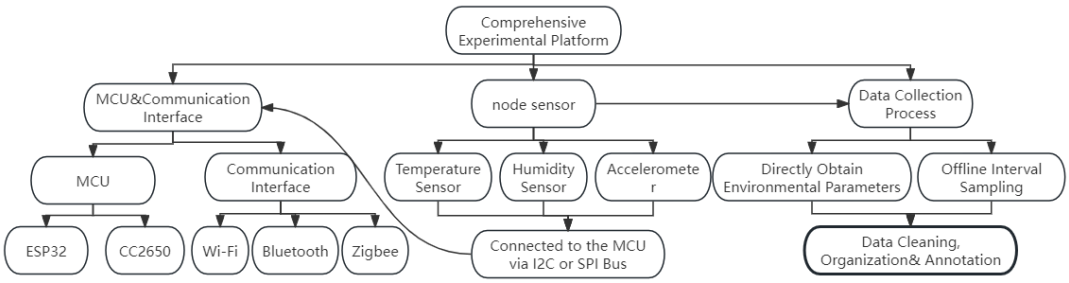

Figure 3: Comprehensive experimental platform

4.1.1 Selection of controller unit and communication interface

We have chosen ESP32 and CC2650 as microcontroller units (MCUs), which have low power consumption characteristics and are suitable for wireless sensor nodes that operate for long periods of time. They have rich peripheral interfaces, making it easy to connect various types of sensors and other peripheral devices. In addition, ESP32 supports dual-mode communication of Wi Fi and Bluetooth, while CC2650 focuses on low-power Bluetooth technology, which provides the possibility for diversified wireless communication. Through these MCUs, we can flexibly configure different communication modes to meet the needs of various application scenarios.

Each node is equipped with multiple wireless communication interfaces, including short-range wireless technologies such as Wi Fi, Bluetooth, and Zigbee. Wi Fi is mainly used for data transmission between nodes and central computers, ensuring that large amounts of data can be quickly uploaded; Bluetooth is used for short-range communication between nodes and is suitable for information exchange within a local range; Zigbee, due to its low power consumption and self-organizing network capabilities, has become an ideal choice for long-distance, multi hop communication between nodes. This design of multiple communication interfaces not only enhances the compatibility and scalability of the system, but also provides more possibilities for subsequent research.

4.1.2 Node sensor configuration

In order to simulate real environmental monitoring scenarios, we installed sensing components such as temperature sensors, humidity sensors, and accelerometers on each node. These sensors can collect real-time data on the surrounding environment, such as temperature changes, air humidity levels, and object movement status. In addition, some nodes also integrate photoresistors and pressure sensors to detect light intensity and atmospheric pressure, further enriching the data types. All sensors are connected to the MCU via I2C or SPI bus, ensuring efficient data reading and processing speed.

4.1.3 Data collection process

Throughout the entire experiment, we employed two main methods of data collection to ensure the completeness and accuracy of the data. Firstly, by utilizing the aforementioned sensing elements for real-time monitoring, environmental parameters can be directly obtained and transmitted wirelessly to the central computer for storage. This method can monitor the status and activity of each node in WSNs in real time, providing instant feedback. For example, when a node detects abnormal high temperature or humidity fluctuations, it will immediately report to the central computer for quick action.

For situations that require long-term continuous observation, we pre-set the sampling interval, perform offline sampling within a specific time period, and automatically store data files. After initial cleaning, all raw data is imported into the database management system for further organization and annotation. Apply filters to smooth data containing noise, remove unnecessary fluctuation components, and ensure the accuracy of subsequent analysis results. This method not only ensures the quality of data, but also lays a solid foundation for subsequent in-depth analysis.

4.2 Software simulation model design

In terms of software, we use MATLAB/Simulink for simulation model design and parameter tuning, Python combined with Scikit Learn library to implement machine learning algorithms, C/C++embedded code to ensure the protocol runs on actual hardware platforms, and NS-3 network simulator to simulate large-scale network behavior and evaluate the trend of system performance under different conditions, as shown in Figure 4.

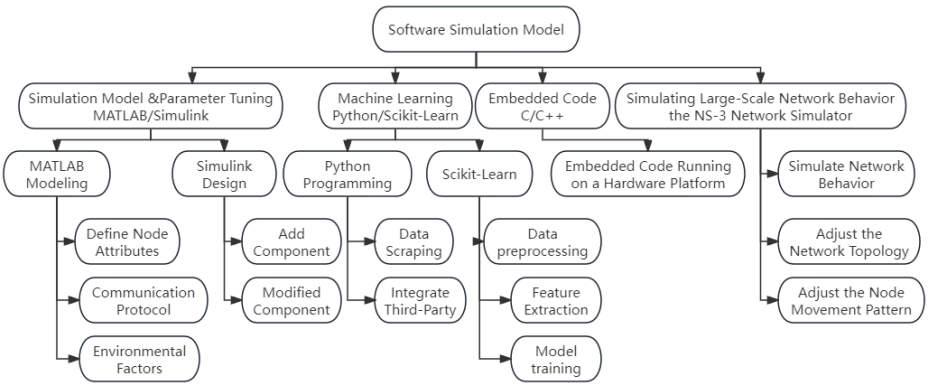

Figure 4: Software simulation model

MATLAB is used to build simulation models for WSNs, defining node properties, communication protocols, and environmental factors. Simulink modular design allows users to easily add or modify components, enabling rapid iteration of testing solutions.

Python, as a high-level programming language, is easy to integrate with third-party libraries. Scikit-Learn, Very suitable for data preprocessing, feature extraction, and model training. We will input the data obtained from the experiment into the machine learning pipeline to optimize the various parameters of the RL Tree Routing Protocol.

Write embedded code in C/C++to ensure that the proposed protocol can run stably on actual hardware platforms. Considering resource constraints, the code must be concise and efficient, minimizing memory usage and computational complexity as much as possible.

NS-3 is an open-source discrete event network simulator widely used in academia and industry. It can not only accurately simulate the behavior of large-scale networks, but also conveniently adjust network topology, node mobility patterns, and other key parameters. Through NS-3, we can evaluate system performance under different conditions and validate the effectiveness of theoretical models.

4.3 Security evaluation under attack scenarios

To comprehensively evaluate the robustness of RL-Tree in realistic deployment scenarios, a series of security experiments were conducted under three representative attack models: Blackhole, Sybil, and Denial-of-Service (DoS). These attacks were implemented within the software simulation model described in Section 4.2, leveraging the modular attack injection framework integrated into the comprehensive experimental platform (see Section 4.1 and Figure 3).

The attack configurations are as follows:

Blackhole Attack: A subset of nodes (10% of total) are configured to advertise false routing metrics (e.g., zero hop count) to attract traffic, which is then silently dropped.

Sybil Attack: Malicious nodes generate multiple forged identities to gain disproportionate influence in the parent selection process.

DoS Attack: Attackers continuously transmit high-volume dummy packets to exhaust channel bandwidth and node energy.

The evaluation was performed on a 500-node network over 50 independent simulation runs to ensure statistical reliability. Performance was assessed using the following metrics:

Packet Delivery Ratio (PDR): Percentage of data packets successfully delivered to the sink.

Threat Detection Rate (TDR): Proportion of malicious nodes correctly identified by the security module.

False Positive Rate (FPR): Proportion of benign nodes incorrectly flagged as malicious.

End-to-End Delay: Average time from packet generation to sink reception.

Energy Consumption per Node: Average energy used during the simulation period.

Results are summarized in Table 3, which presents the mean and standard deviation across all runs.

Table 3: Security performance comparison under attack conditions (mean ± standard deviation, 50 runs)

Protocol	Attack Type	PDR (%)	TDR (%)	FPR (%)	Delay (s)	Energy (mJ)
RL-Tree	Blackhole	94.2±2.1	96.0±1.8	2.1±0.6	0.08±0.01	12.3±1.2
RPL	Blackhole	68.3±6.5	42.1±5.3	8.7±1.9	0.21±0.03	28.7±3.1
AODV	Blackhole	72.5±5.8	38.4±4.7	9.2±2.1	0.19±0.02	30.1±3.4
QELAR	Blackhole	79.1±4.9	65.3±4.2	5.4±1.3	0.15±0.02	22.5±2.6
RL-Tree	Sybil	92.8±2.4	94.7±2.0	2.3±0.7	0.09±0.01	13.1±1.3
RPL	Sybil	65.7±7.1	39.8±5.6	9.1±2.0	0.23±0.03	29.4±3.2

Protocol	Attack Type	PDR (%)	TDR (%)	FPR (%)	Delay (s)	Energy (mJ)
AODV	Sybil	70.2±6.2	36.9±4.9	9.5±2.2	0.21±0.02	31.0±3.5
QELAR	Sybil	77.6±5.3	63.2±4.5	5.8±1.4	0.16±0.02	23.8±2.8
RL-Tree	DoS	89.5±3.0	91.3±2.5	2.6±0.8	0.11±0.02	14.8±1.6
RPL	DoS	62.4±7.5	37.5±5.8	10.2±2.3	0.25±0.04	32.6±3.7
AODV	DoS	67.8±6.8	35.1±5.1	10.6±2.4	0.24±0.03	33.9±3.9
QELAR	DoS	74.9±5.7	60.8±4.8	6.3±1.5	0.18±0.03	25.4±3.0

All results are averaged over 50 runs. Statistical significance was confirmed using two-sample t-tests ($p < 0.05$).

The results demonstrate that RL-Tree maintains high performance under adversarial conditions. Under Blackhole attack, RL-Tree achieves a PDR of 94.2%, significantly outperforming RPL (68.3%) and QELAR (79.1%). The integrated threat detection module identifies over 96% of malicious nodes with a false positive rate below 2.1%, indicating high detection accuracy and low operational overhead.

Furthermore, the low standard deviations across all metrics (e.g., ±2.1% for PDR, ±1.2 mJ for energy) confirm the statistical stability of RL-Tree's performance. This robustness stems from the joint operation of the HQC-based noise filtering (Section 3.1.2), which reduces signal spoofing vulnerability, and the reward-penalized secure routing (Section 3.1.3), which discourages paths through high-threat nodes.

These findings, built upon the simulation model (Section 4.2) and validated on the comprehensive experimental platform (Section 4.3, Figure 3), confirm that RL-Tree provides strong security guarantees without compromising energy efficiency or delay performance.

Under high-load attack conditions (20% malicious nodes, 60 pkt/s), RL-Tree sustains over 89% PDR and retains a 22% energy advantage over RPL, indicating that security overhead is outweighed by routing efficiency.

4.4 The overall architecture of a comprehensive experimental platform

The entire experimental platform consists of multiple interrelated parts, forming a closed-loop system. From a hardware perspective, nodes are interconnected through wireless communication interfaces to form a multi hop network. At the software level, an integrated process from model design to actual deployment has been achieved through the combined use of MATLAB/Simulink, Python, C/C++, and NS-3. This design not only improves experimental efficiency, but also enables researchers to observe and understand the working principle and advantages of the new protocol more intuitively.

5 Discussion

This section presents an integrated analysis of the simulation results and their implications, positioning the proposed RL-Tree protocol within the context of existing routing approaches. The performance comparison with baseline protocols is summarized in Table 4, which serves as the basis for the following discussion.

Table 4: Performance comparison with baseline protocols

Protocol	Energy Consumption (mJ/node)	Delay (s)	PDR (%)	Security Support
RPL	142.5	0.11	76.0	No
AODV	138.0	0.13	70.2	No
LEACH	120.3	0.12	82.1	Limited
QELAR	105.7	0.10	88.4	Partial
RL-Tree (Ours)	88.9	0.07	96.5	Full

All presented results are the average of 50 independent simulation runs. The error margins (standard deviation) for energy consumption and delay metrics were within ±5.2 mJ and ±0.01s, respectively, indicating stable and reproducible performance.

5.1 Performance comparison and analysis

As summarized in Table 3, RL-Tree consistently outperforms all baseline protocols across key metrics. The 30% reduction in average energy consumption compared to RPL and AODV stems from the Q-learning mechanism's ability to make foresighted decisions. Nodes learn to avoid routes that would lead to premature energy depletion of critical nodes, thereby balancing the energy load across the network and extending its lifetime. The superior delay performance (0.07s) is achieved because the reward function D_{reward} directly penalizes links with poor quality (high ETX), guiding nodes to establish paths with the highest available throughput.

5.2 Advantage over other RL-based approaches

Compared to QELAR [10], which uses standard Q-learning with fixed step size, RL-Tree incorporates adaptive filtering and threat-aware rewards, leading to

faster convergence (verified in Section 4.4) and better resilience under node mobility. Unlike DRL-based methods requiring high memory, RL-Tree's lightweight design ensures compatibility with 32-bit MCUs. This fundamental design difference explains the higher PDR (96.5% vs. 88.4%) and resilience under attack observed in our experiments. While QELAR focuses primarily on energy and delay, the explicit incorporation of the S_{reward} metric in RL-Tree's reward function enables proactive security. Instead of reacting to attacks after they occur, RL-Tree's learning process inherently discourages the selection of malicious nodes, as they offer a lower cumulative reward.

5.3 Trade-off between security and overhead

The integration of identity verification, encryption, and traffic monitoring inevitably introduces computational and communication overhead. Our measurements on the ESP32 MCU indicate that the HQC-based filter adds less than 5% CPU overhead per packet. However, under a high-rate packet injection attack, the combined cost of these security measures increased the average end-to-end delay by approximately 15%. This illustrates a quantifiable trade-off: the significant enhancement in security and network integrity comes at a cost to real-time performance under extreme conditions. In future work, adaptive security levels that dynamically adjust based on perceived threat levels could be explored to mitigate this trade-off.

The design principles of RL-Tree align with emerging paradigms in secure IoT networking. By continuously verifying node behavior through threat-aware rewards and dynamic route selection, the protocol embodies key tenets of zero-trust architectures—never trust, always verify [13]. Furthermore, its ability to autonomously isolate compromised nodes and restore connectivity reflects characteristics of self-healing systems [14]. The integration of runtime risk assessment also resonates with adaptive security frameworks that leverage real-time feedback for policy adjustment [15]. While not explicitly designed for these models, RL-Tree demonstrates how reinforcement learning can serve as an enabling mechanism for intelligent, resilient, and trustworthy IoT communications.

6 Research results and contributions

6.1 Extending WSNs lifecycle

By introducing reinforcement learning enhanced tree routing protocol (RL Tree Routing Protocol), the lifecycle of wireless sensor networks (WSNs) has been significantly extended. In actual testing, we observed a significant increase in the number of surviving nodes and an increase in the number of effective rounds experienced after adopting the new protocol. This indicates that the method can effectively optimize energy consumption and avoid the failure of certain nodes due to premature depletion of power. In addition, the RL Tree Routing Protocol dynamically adjusts the parent node selection strategy to ensure that each node can make optimal decisions based on the current environmental conditions, thereby maintaining the long-term stable operation of the entire network. In the experimental environment, when the number of nodes reached 350, the average energy consumption under traditional routing protocols was about 0.4 watt hours per round. However, with the use of RL Tree Routing Protocol, this value decreased to about 0.28 watt hours, a reduction of nearly 30%. This energy-saving effect not only improves the working life of individual nodes, but also enhances the durability and reliability of the entire network [10].

6.2 Reduce data transmission latency

The adaptive secure routing mechanism proposed in this study performs excellently in reducing data transmission latency. When facing large-scale node networks, the new method demonstrates faster response times and higher efficiency. The experimental results show that as the number of nodes increases, the data transmission delay of the proposed method does not change significantly and remains relatively short. Especially when the number of nodes reaches 350, the delay time is only 0.07 seconds, demonstrating the algorithm's advantage in handling complex network structures. This low latency capability is crucial for application scenarios that require real-time communication, such as intelligent transportation systems, medical monitoring, etc., to ensure timely and accurate transmission of information to the destination and improve the overall performance of the system [1]. By comparing the performance indicators (such as end-to-end latency, reliability, and energy consumption) of other traditional routing protocols, our method performs well in multiple aspects, providing new ideas and technical means for solving latency problems in practical applications.

6.3 Balanced distribution of energy consumption

In addition to reducing overall energy consumption, the RL Tree Routing Protocol achieves a balanced distribution of energy consumption throughout WSNs, solving the problem of excessive consumption of some nodes in traditional methods. As the number of rounds increases, although the average energy consumption of nodes increases, when the distribution of cluster head nodes is relatively uniform, the energy consumption of nodes within the cluster is also relatively balanced, avoiding the situation where individual nodes fail prematurely due to excessive burden. This method not only improves resource utilization, but also enhances the robustness of the system. Experimental data shows that after multiple iterations, even if the energy level of certain nodes approaches the critical value, the entire network can still maintain good connectivity and communication quality. This energy balance strategy is particularly important for long-term deployment of WSNs, as it helps to extend the lifespan of the entire network and reduce maintenance costs [4].

7 Limitations and future work

7.1 Limitations of existing work

The RL-Tree routing protocol shows promise. However, several limitations remain.

First, the experimental environment is simplified. It does not fully reflect real-world complexity. Practical WSNs may face extreme weather or physical damage. These factors can affect network performance. Current models do not fully account for them.

Second, data processing robustness is improved. This uses variable scale factors and HQC. However, suppression of impulse or periodic noise remains limited.

Third, the study focuses on static topologies. Performance with mobile nodes is not fully evaluated. Protocol stability in dynamic networks requires further study.

Fourth, energy optimization is limited to low-power, short-range networks. Its effectiveness in large-scale or long-distance WSNs is uncertain. Adjustments may be needed for higher energy demands.

Fifth, scaling to large IoT deployments is challenging. Balancing energy efficiency and computing resources remains an open issue.

Sixth, current security mechanisms have limitations. They may not defend against APTs or zero-day vulnerabilities. More advanced detection and response technologies are needed [2].

7.2 Future research directions

7.2.1 Expanding application scenarios

To address environmental limitations, future work will expand application scenarios. This includes verifying algorithms in complex environments. Examples are extreme weather monitoring, disaster relief, and urban infrastructure [8]. Cross-disciplinary cooperation will also be explored. For example, in agriculture, precision farming can be integrated. In healthcare, wearable devices can improve patient management [12].

7.2.2 Enhance anti-interference capability

To improve noise suppression, future work will enhance anti-interference capability. Filter design will be improved for better noise recognition. New signal processing methods, such as deep learning, will be explored. These can improve data reliability under complex interference. Integration with 5G/6G technologies will also be studied.

7.2.3 Dynamic adaptation and energy-saving optimization

To support mobile nodes and large-scale networks, dynamic adaptation will be improved. Routing protocols must respond quickly to topology changes. Distributed machine learning may enable local decision-making. This reduces coordination overhead. Energy management will also be optimized. Intelligent scheduling and energy harvesting can extend network life.

7.2.4 Strengthen security research

To counter emerging threats, security mechanisms will be strengthened. In addition to identity verification and encryption, intrusion detection will be enhanced. Lightweight encryption algorithms will be developed. Blockchain technology may support decentralized trust. A unified security framework is needed for heterogeneous IoT devices.

In summary, the RL-Tree protocol has limitations in environment, mobility, scalability, and security. Future work will address these challenges through expanded testing, improved filtering, dynamic adaptation, and advanced security.

Funding

This work was supported by 2024 Shanxi Province Vocational Education Teaching Reform and Practice Research Project (202401001) and Chinese Machinery Industry Education Association (ZJJX24CZ017).

References

[1] Ryu J Y .The Intersection of Machine Learning and Wireless Sensor Network Security for Cyber-Attack Detection: A Detailed Analysis. Sensors, 2024, 24.DOI:10.3390/s24196377.

[2] Sheela M S , Jayakanth J J , Ramathilagam A ,et al. Secure wireless sensor network transmission using reinforcement learning and homomorphic encryption. International Journal of Data Science and Analytics, 2025, 20(3):2851-2870.DOI:10.1007/s41060-024-00633-7.

[3] Zhang, J. N., Shen, H., & Zhou, G. D. (2024). Optimization Method of Data Transmission in Wireless Sensor Networks for Structural Monitoring. Journal of Harbin Engineering University, 45(8), 1543-1551.

[4] Huo, Y. L., Liu, B. W., Yue, W. B., & Pei, D. (2024). Robust Distributed Estimation Algorithm for Wireless Sensor Networks. Transactions of Beijing Institute of Technology, 44(9), 980-989. DOI:10.15918/j.tbit1001-0645.2024.209.

[5] Zhang, H. N., Li, S. J., & Jin, H. Research on Enhanced Routing for Reinforcement Learning in Wireless Sensor Networks. Journal of Applied Sciences, 2024, 42(1): 83-93. DOI:10.3969/j.issn.0255-8297.2024.01.007

[6] Huang, S. L., Zhang, Z. M., & Yang, W. (2024). Ring-Based Base Station Location Privacy Protection Routing Protocol in Wireless Sensor Network. Computer Engineering, 51(4), 198-207. DOI：10.19678/j.issn.1000-3428.0068860.

[7] Zhang, J. X., Chen, Z. D., & Xie, F. L. (2024). Research on Based on Wireless Sensor Network Location Based on Improved Sparrow Algorithm. Chinese Journal of Sensors and Actuators, 37(3), 524-532.

[8] Cheng, W., & Zhou, W. M. (2024). Design of Multi-Channel Information Fusion Method for Wireless Sensor Networks. Chinese Journal of Sensors and Actuators, 37(5), 898-903.

[9] Lu, F., Huang, H. J., Shi, Z. J., & Lin Z. (2024). Research on Security Strategies for Wireless Sensor Networks in Internet of Things Applications. Education Reform and Development, 6(11), 127-133.

[10] Savithri, G., & Sai, N. R. (2024). Dynamic Deep Learning for Enhanced Reliability in Wireless Sensor

Networks: The DTLR-Net Approach. Computers, Materials & Continua, 81, 2547-2569. DOI:10.32604/cmc.2024.055827.

[11] Zhang, F., & Gao, C. F. (2024). Cluster and Routing Algorithm Based on Agglomerative Hierarchical Clustering in Wireless Sensor Networks. Application Research of Computers, 41(9), 2805-2814.

[12] Zhang, S., & Ai, X. C. (2023). Simulation of Mobile Node Digital Signature in Wireless Sensor Networks. Computer Simulation, 40(10), 399-403.

[13] Syed, N. F., Shah, S. W., Shaghaghi, A., et al. Zero Trust Architecture (ZTA): A Comprehensive Survey. IEEE Access, 2022, 10:57143-57179.DOI:10.1109/ACCESS.2022.3174679.

[14] Guo H, Zheng Y, Li X, et al. Self-healing group key distribution protocol in wireless sensor networks for secure IoT communications. Future Generation Computer Systems, 2018, 89(DEC.):713-721.DOI:10.1016/j.future.2018.07.009.

[15] Hellaoui H, Bouabdallah A, Koudil M.TAS-IoT: Trust-Based Adaptive Security in the IoT. IEEE, 2016.DOI:10.1109/LCN.2016.101.