DQN-Raft+: A Deep Reinforcement Learning-Optimized Lightweight Consensus Algorithm for Secure Edge Storage in IoT Environments

Pengyuan Wang, Qiaonian Xu*, Huaiqing Zhang
School of Wuwei Vocation College School of Information Technology, Wuwei Gansu, 733000, China
E-mail: wwocxqn66@163.com
*Corresponding author

Keywords: internet of things, digitization, information security, privacy protection, deep learning

Received: April 14, 2025

The rapid development of the Internet of Things (IoT) has intensified security and privacy challenges across data generation, transmission, and storage. This study introduces a blockchain-based secure edge storage model tailored for IoT environments and presents a lightweight consensus algorithm, Deep Q-Network (DQN)-Raft+, which incorporates deep reinforcement learning. By combining the decentralized features of edge computing and blockchain, the model enables automated data access control through smart contracts. Furthermore, it optimizes leader node selection in the Raft consensus process using a DQN, formulating the consensus as a Markov Decision Process to enhance responsiveness and privacy protection in dynamic network conditions. Experiments were performed in a simulated environment using TensorFlow 2.6 and a MySQL database. The performance of DQN-Raft+ was compared against traditional consensus algorithms, including Proof of Work, Proof of Stake, Practical Byzantine Fault Tolerance, and Delegated Byzantine Fault Tolerance. Results indicate that DQN-Raft+ significantly reduces block generation delay (175.77 ms) and achieves a high privacy protection score (0.95). It also maintains a low data loss rate of 0.01%, demonstrating enhanced robustness and real-time capability. These findings indicate that DQN-Raft+ effectively strengthens data security and privacy in IoT systems, offering a technically sound and efficient mechanism for secure data exchange. The study provides both a theoretical framework and practical direction for future research in secure IoT deployment.

Povzetek: Študija uvaja DQN-Raft+, lahkoten konsenzni algoritem, optimiziran z globokim učenjem za izboljšano varnost podatkov in zaščito zasebnosti v IoT omrežjih z izboljšanjem latence in robustnosti.

1 Introduction

The rapid advancement of Internet of Things (IoT) technology has led to the widespread adoption of smart devices, increasing the frequency of digital information generation, transmission, and storage [1]. Concurrently, concerns regarding secure storage and privacy protection have become increasingly critical. According to Statista, the number of connected IoT devices worldwide is projected to reach billions in the near future. While this expansion enhances data accessibility, it also elevates the risk of breaches, tampering, and other security threats [2]. Traditional data storage solutions often fail to meet the stringent requirements for confidentiality and integrity in IoT environments [3]. In response, blockchain technology has emerged as a promising alternative due to its decentralized, tamper-resistant, and transparent nature. Numerous studies have examined blockchain's application in data protection, noting its potential to mitigate hacking attempts and unauthorized data alterations [4, 5]. Nonetheless, blockchain alone does not fully address privacy concerns. In complex IoT environments, ensuring data security while safeguarding user privacy remains a formidable challenge [6, 7]. Current privacy protection mechanisms are frequently limited by performance constraints. As data volume increases, computational overhead and response time often become significant barriers to efficiency [8, 9].

This study addresses the challenges of high latency and inefficient privacy protection in IoT environments. It investigates the following research question: In IoT scenarios with frequent terminal node state fluctuations, can a Deep Q-Network (DQN) be used to optimize the Raft consensus algorithm to reduce block generation latency and enhance the efficiency of data access control and privacy protection in edge blockchain systems? To explore this question, a blockchain-based secure storage model is deployed at the network edge. A lightweight consensus mechanism, DQN-Raft+, is developed by integrating deep reinforcement learning into the Raft protocol. The proposed approach is evaluated in terms of security, efficiency, and adaptability, demonstrating improved performance under dynamic and resource-constrained IoT conditions. This study introduces a secure digital information storage model for blockchain-based IoT environments and designs a lightweight consensus algorithm incorporating deep learning (DL) to enhance data privacy protection. The blockchain model improves data confidentiality and integrity, addressing the limitations of traditional storage methods in IoT systems. The lightweight consensus algorithm reduces computational overhead and boosts data processing efficiency, supporting the real-time handling of large-scale IoT data. This approach not only strengthens privacy protection but also enhances system performance, offering a novel framework and technical direction for secure storage and privacy management in IoT applications.

The remaining sections of this study are structured as follows. Section 2 reviews related work on secure information storage and privacy protection in IoT environments, with a focus on the strengths and limitations of existing consensus algorithms and DL-based privacy-preserving techniques. Section 3 describes the architecture of the proposed secure storage system, which integrates blockchain and edge computing. This section covers the data acquisition mechanism, blockchain framework, security model design, and the implementation of the DQN-Raft+ consensus algorithm. Section 4 presents the experimental setup and performance evaluation, including platform construction, data collection methods, and evaluation metrics. A comparative analysis is conducted across several dimensions-latency, privacy protection, and access control-highlighting differences between DQN-Raft+ and baseline algorithms in terms of system complexity, learning convergence, and data security. Section 5 concludes the study by summarizing key findings, discussing current limitations, and suggesting directions for future research.

2 Related work

The rapid advancement of information technology has heightened concerns regarding the secure storage and privacy protection of digital information, particularly within cloud computing and IoT environments. Khan and Por pointed out that the rise of information technology has exacerbated the challenges of secure storage, particularly in cloud computing and IoT environments, where the risks of data leakage and tampering have grown significantly [10]. To address these challenges, Pazhani et al. proposed an improved memory-efficient distributed algorithm architecture for implementing two-dimensional discrete wavelet transform, to mitigate the high computational cost associated with image and video compression. In portable devices and high-speed communication systems, conventional multiplier-based computation is no longer feasible due to constraints on chip area and power consumption. The distributed algorithm architecture replaces multipliers with shift operations and lookup tables, thereby improving computational speed and reducing power usage. However, the increase in filter coefficients often leads to significant expansion of the lookup table size, which remains a concern [11]. In parallel, Dwivedi et al. highlighted the necessity of encrypted data storage in edge computing environments to reduce security risks during data transmission [12]. Cao et al. observed that current privacy protection mechanisms primarily rely on encryption, access control, and anonymization. However, these approaches often struggle with high-dimensional and dynamic data, especially in IoT contexts, where the growing number of devices and data volume leads to reduced efficiency and reliability [13]. Wen et al. highlighted the ongoing challenge of balancing privacy protection with data usability in the process of data sharing and utilization [14]. In response, blockchain has gained attention for its decentralized, tamper-resistant, and transparent properties. Chen et al. noted its widespread application in the secure storage of digital information [15]. Ren et al. introduced a blockchain-based solution that uses smart contracts to automate data management and access control [16]. Furthermore, Emami et al. explored the role of consortium blockchains in enabling secure data sharing among multiple participants while protecting privacy [17]. In the domain of DL, Rodríguez et al. proposed its application in privacy protection by learning user behavior patterns. Their study suggested that DL-based mechanisms could enhance data analysis accuracy while reducing reliance on raw data [18]. Valencia-Arias et al. further investigated the integration of DL and blockchain, indicating that this combination improves security and flexibility in both privacy protection and data sharing [19].

Table 1 presents a structured comparison of representative approaches. This table illustrates the advantages and limitations of existing work-especially in relation to consensus algorithms and DL-based privacy protection mechanisms.

Table 1: Comparison of consensus algorithms and DL-based privacy protection approaches

Method Type	**Representative Model**	**Main Advantages**	**Main Limitations**	**DQN Integration Potential**
Consensus Algorithm	Raft	Simple design; low communication overhead	Poor fault tolerance; lacks adaptability in dynamic settings	Compatible
	Practical Byzantine Fault Tolerance (PBFT)	Strong consistency; tolerates Byzantine faults	High communication complexity; poor scalability	Difficult
	Delegated Byzantine Fault	Better performance than PBFT	Complex architecture; sensitive to	Theoretically

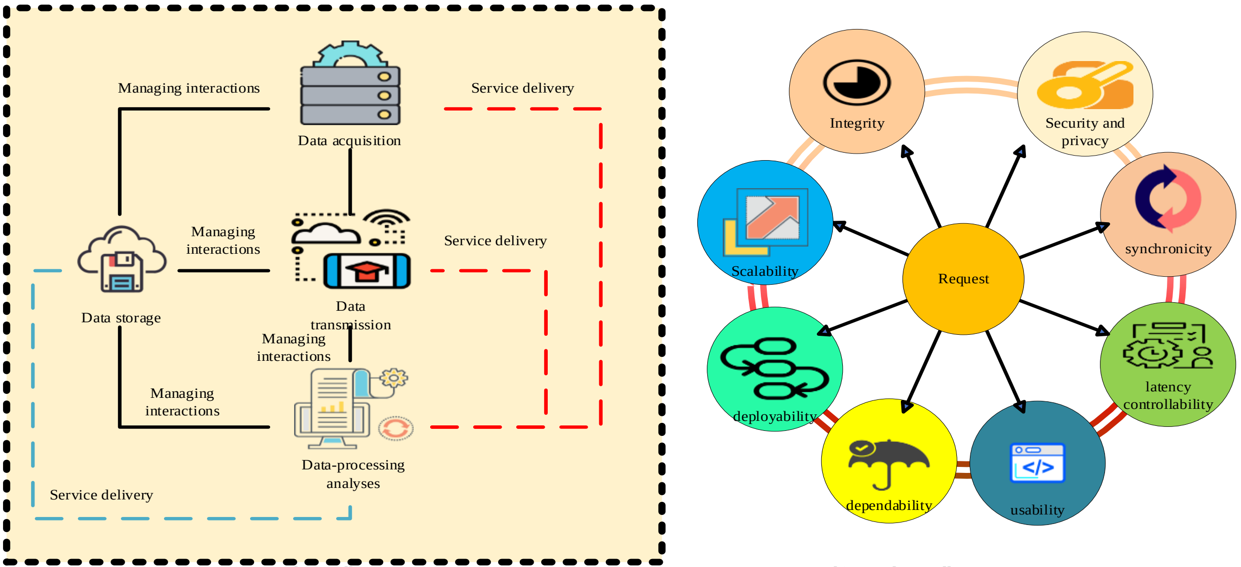
	Tolerance (DBFT)		node expansion	feasible
	Proof of Work (PoW)	High security; widely used	High energy usage; high latency	Not suitable
	Proof of Stake (PoS)	Energy-efficient; faster block generation than PoW	Lower decentralization; risk of centralization	Not suitable
DL Privacy Methods	CNN-based privacy model	Effective at capturing spatial features	Rigid architecture; limited generalization across diverse data	Partially compatible
	Recurrent Neural Network + Access Control	Suitable for modeling temporal privacy patterns	Poor scalability in real-time IoT environments	Compatible
	Federated Learning	Data remains local; strong privacy guarantees	High communication overhead; slow convergence	Can be combined with DQN
	DQN-based privacy scoring	Adaptive learning; responsive in dynamic settings	Complex modeling; high training cost	Core of this study

In summary, while substantial progress has been made in secure storage and privacy protection for digital information, key challenges persist. Most existing consensus algorithms lack the flexibility to handle dynamic IoT environments, and DL techniques, though effective in learning complex patterns, have not been fully integrated into consensus mechanisms. In particular, the application of DQNs in consensus optimization remains limited. Existing research often suffers from vague state space definitions and unstable training processes. To address these gaps, the DQN-Raft+ algorithm proposed in this study constructs a lightweight and self-adaptive consensus model, enhancing both privacy protection efficiency and system robustness in IoT environments.

3 The Resource Sharing System for Vocational Education Based on Blockchain and edge computing

3.1 IoT

IoT, a core technology linking the physical and digital worlds, is characterized by high-frequency, multi-source, and real-time data collection. These attributes create favorable conditions for integrating edge intelligence and blockchain technologies to address the demands of secure, efficient, and privacy-aware data storage [20]. To meet these requirements, this study develops a secure digital information storage model based on a collaborative architecture that combines blockchain with edge computing. Before detailing the model design, it is necessary to outline the typical data acquisition process in IoT environments. As shown in Figure 1, this process forms the theoretical and technical foundation of the proposed architecture [21].



(a) IoT data collection process

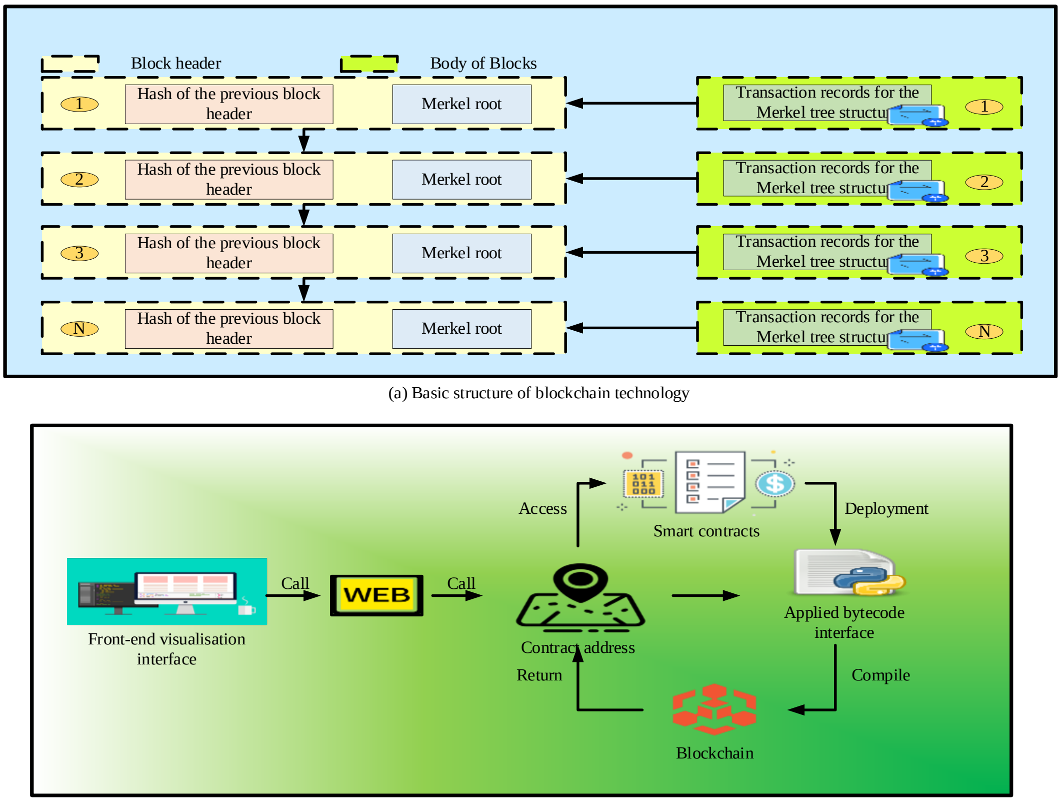
(b) IoT data collection requirements

Figure 1: Data acquisition and processing flow in an IoT environment

Figure 1 illustrates a standard data flow in IoT systems, including stages such as data sensing by edge devices, preprocessing, transmission, and storage. This workflow provides a reference framework for designing the blockchain–edge computing-based secure storage model.

3.2 Blockchain technology

Blockchain is a decentralized, distributed ledger technology that records and stores data securely and transparently. It organizes data into linked "blocks," with each block containing the hash of the preceding block, thereby forming an immutable chain. The core features of blockchain-transparency, immutability, and security-have enabled its widespread application in areas such as financial transactions, supply chain management, and identity authentication. Furthermore, blockchain supports the deployment of smart contracts, which automate transactions without intermediaries, reducing operational costs and improving efficiency [22]. The fundamental structure of blockchain and the design principles of smart contracts are presented in Figure 2 [23].


Block header
Body of Blocks
1
Hash of the previous block header
Merkel root
Transaction records for the Merkel tree structu
1
2
Hash of the previous block header
Merkel root
Transaction records for the Merkel tree structu
2
3
Hash of the previous block header
Merkel root
Transaction records for the Merkel tree structu
3
N
Hash of the previous block header
Merkel root
Transaction records for the Merkel tree structu
N
Access
Smart contracts
Deployment
Call
WEB
Call
Front-end visualisation interface
Contract address
Applied bytecode interface
Return
Compile
Blockchain

(a) Basic structure of blockchain technology

(b) Principles of smart contract development

Figure 2: Basic structure of blockchain and development principles of smart contracts

Cryptography serves as the cornerstone of blockchain security and privacy. Hash functions are used to maintain data integrity and prevent tampering, while public–private key cryptography supports identity authentication and transaction signing. Digital signatures ensure transaction authenticity, and both symmetric and asymmetric encryption schemes safeguard data confidentiality. Additionally, zero-knowledge proofs allow users to validate claims without exposing sensitive data. The integration of these cryptographic techniques significantly enhances the security and trustworthiness of blockchain-based data storage and sharing systems [24]. Key cryptographic mechanisms are shown in Figure 3 [25].

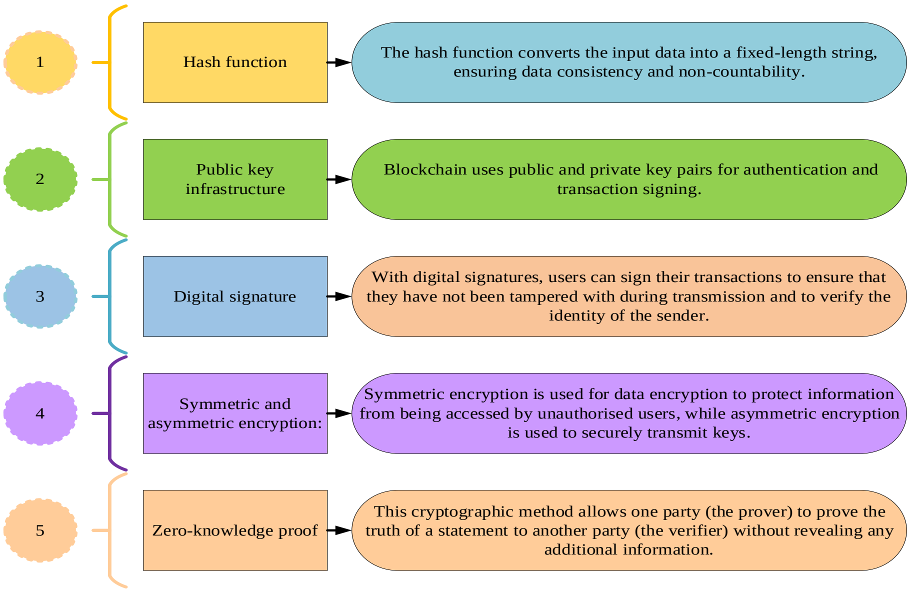

1
Hash function
The hash function converts the input data into a fixed-length string, ensuring data consistency and non-countability.
2
Public key infrastructure
Blockchain uses public and private key pairs for authentication and transaction signing.
3
Digital signature
With digital signatures, users can sign their transactions to ensure that they have not been tampered with during transmission and to verify the identity of the sender.
4
Symmetric and asymmetric encryption:
Symmetric encryption is used for data encryption to protect information from being accessed by unauthorised users, while asymmetric encryption is used to securely transmit keys.
5
Zero-knowledge proof
This cryptographic method allows one party (the prover) to prove the truth of a statement to another party (the verifier) without revealing any additional information.

Figure 3: Cryptographic techniques

3.3 Digital information security storage model for network edge-based blockchain IoT

This study proposes a digital information security storage model for blockchain-enabled IoT environments based on network edge computing. The model integrates the decentralized architecture of blockchain with the low-latency, high-responsiveness advantages of edge computing to ensure data security, availability, and privacy protection. The model architecture comprises three primary components: edge devices, edge computing nodes, and a blockchain network connected to cloud servers. At the edge device level-such as sensors, smart home systems, and industrial controllers-data is generated and preprocessed locally. This reduces latency and bandwidth usage by limiting the volume of raw data transmitted over the network. Data is then securely sent to edge computing nodes via encrypted communication protocols. Edge nodes play a key role in processing the data, including encryption, deduplication, and compression, to enhance both transmission efficiency and security. The processed data is subsequently uploaded to the blockchain for decentralized storage. Blockchain ensures data traceability and immutability, effectively preventing tampering and forgery while preserving data integrity. Each data block embeds corresponding access control policies to manage permissions precisely during access operations. To strengthen privacy protection, the model incorporates encryption techniques and multi-factor authentication. Authorized access is enforced through smart contracts, which automate access control, policy execution, and verification. These contracts not only streamline the access process but also create a secure environment for data sharing among heterogeneous devices and users. Moreover, the model adopts a modular architecture, allowing for dynamic scalability and adaptability. As the number of IoT devices and data volume continues to grow, the system can flexibly reallocate storage and computational resources to accommodate varying demands. Overall, this network edge-based blockchain IoT model significantly enhances digital information security while improving system performance and user experience. By implementing this framework, users can securely store, manage, and share data in a highly responsive and privacy-preserving environment, thereby supporting the sustainable development and application of IoT technologies. The structural design of the proposed model is illustrated in Figure 4.

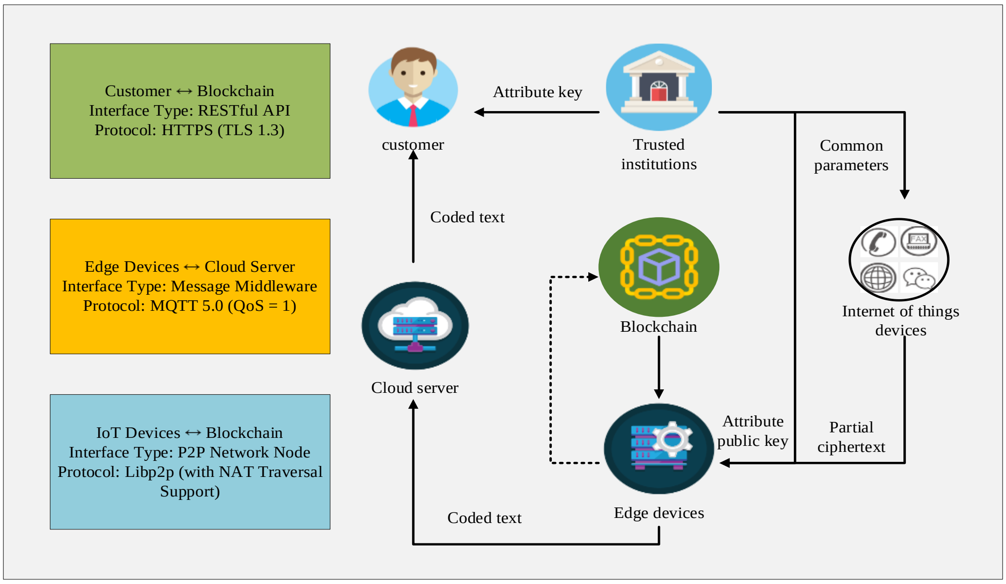

Customer ↔ Blockchain
Interface Type: RESTful API
Protocol: HTTPS (TLS 1.3)
Edge Devices ↔ Cloud Server
Interface Type: Message Middleware
Protocol: MQTT 5.0 (QoS = 1)
IoT Devices ↔ Blockchain
Interface Type: P2P Network Node
Protocol: Libp2p (with NAT Traversal Support)
customer
Attribute key
Trusted institutions
Common parameters
Coded text
Blockchain
Internet of things devices
Cloud server
Attribute public key
Partial ciphertext
Edge devices
Coded text

Figure 4: Structure of the blockchain-based digital information security storage model at the network edge

As shown in Figure 4, the blockchain-based information security storage model deployed at the network edge clearly defines the interfaces and protocols among its components through technical annotations. Clients interact with the blockchain via Representational State Transfer Application Programming Interface over Hypertext Transfer Protocol Secure and Transport Layer Security 1.3. Data requests are submitted using attribute-based encryption, and access rights are verified accordingly. Edge devices and cloud servers communicate via the Message Queuing Telemetry Transport 5.0 protocol to transmit fragmented and encrypted data. Reed-Solomon coding is used to improve transmission efficiency under low-bandwidth conditions. IoT devices act as lightweight blockchain nodes and participate in a lightweight PBFT consensus mechanism through the Libp2p-based peer-to-peer networking protocol, enabling block header synchronization and transaction validation. The model integrates several key security mechanisms, including trusted execution environments for key protection, zero-knowledge proofs to conceal transaction attributes, and randomized sampling verification to reduce the computational burden on edge nodes. Through layered encryption, protocol optimization, and lightweight consensus design, the model establishes a secure and efficient data storage and verification pipeline from edge to cloud.

On edge devices, data is encrypted using symmetric encryption algorithms. The ciphertext is computed as Equation (1):

$$C = E(K, D) \tag{1}$$

In Equation (1), C represents the encrypted ciphertext; D denotes the original data; E means the encryption function; K refers to the encryption key. To ensure data integrity and immutability, the hash value H and digital signature S of the data are generated as follows:

$$H = H(D) \tag{2}$$

$$S = Sign(K_{priv}, H) \tag{3}$$

Here, K_{priv} private key of the edge device used for digital signing. Each block B_i on the blockchain contains the following information:

$$B_i = (C, H, S, T, A) \tag{4}$$

In Equation (4), T is a timestamp indicating when the data is generated or stored, and A an access control policy specifying which users or devices can access the data. The access control policy A is represented as a set of conditions, where each condition C_i corresponds to a specific access rule. This can be expressed as Equation (5):

$$A = \{C_1, C_2, \dots, C_m\} \tag{5}$$

Smart contracts are used to automate data access management based on these conditions. The access control logic is defined by the function:

$$F(U, A) \to Access_{result} \tag{6}$$

F denotes the access control function of the smart contract, U signifies the user requesting access, and $Access_{result}$ indicates whether the user is granted access to the data. When a user requests data, the access rights are verified by invoking the smart contract, which can be expressed as Equation (7):

$$R = F(U, A)\text{if}R = \text{true} \to D = D_{retrieved} \tag{7}$$

In Equation (7), R represents the result of the permission check, and $D_{retrieved}$ denotes the raw data retrieved from the blockchain if access is granted.

To further clarify the integration of encryption, access control mechanisms, and the blockchain transaction process, this study designs the data on-chaining procedure to comprise five sequential stages: preprocessing, transaction generation, smart contract triggering, data on-chaining, and access verification. After data is encrypted by edge devices-as outlined in Equations (1) to (3)-and its hash and digital signature are generated, the edge computing nodes encapsulate the encrypted content into a transaction and broadcast it across the blockchain network. Consensus nodes subsequently verify the digital signature and access policy, and upon successful validation, the transaction is committed to the blockchain via the underlying consensus mechanism. Each transaction contains the fields cipher_data, hash_val, signature, timestamp, and access_policy, corresponding to the data block structure defined in Equation (4). This ensures the integrity of the data and the synchronous on-chaining of the associated access control policy. For access control enforcement, this study implements smart contracts using the Solidity programming language, deploying them on a private blockchain based on the Ethereum architecture. Ethereum is selected due to its mature smart contract ecosystem and strong support for complex access control logic. The access control smart contract is deployed on-chain and includes a core function, checkPermission(address user, bytes32 data_id), which evaluates a user's address U against the predefined access policy A, returning a Boolean value that determines whether access is permitted. Access verification is executed through an on-chain contract call. Smart contracts are automatically triggered upon a user's data request, eliminating the need for manual intervention or intermediary middleware. This mechanism facilitates autonomous enforcement of access control directly within the blockchain environment. The proposed framework achieves seamless integration of encryption, permission management, and blockchain-based auditability. Furthermore, it establishes a practical foundation for the future integration of privacy-enhancing technologies and secure multi-party data sharing in IoT environments.

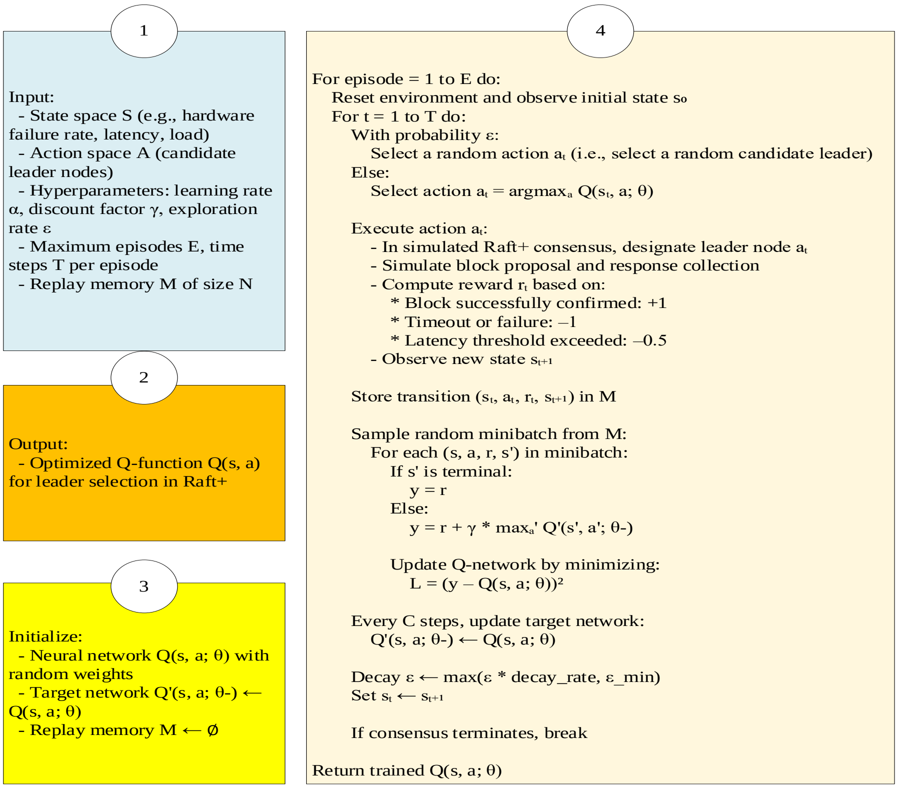
3.4 Design of lightweight consensus algorithms based on DL

DL, a subfield of machine learning, employs multi-layer neural networks to extract complex features from large-scale data and automate decision-making in intricate environments [26]. To enhance the efficiency and privacy-preserving capabilities of blockchain applications in IoT environments, this study proposes a lightweight consensus algorithm named DQN-Raft+, which integrates deep reinforcement learning with the classical Raft consensus protocol. Built upon the Raft algorithm, DQN-Raft+ introduces a policy-learning agent that utilizes a DQN to dynamically optimize the leader election process. In contrast to centralized scheduling strategies, DQN-Raft+ deploys lightweight proxy models in parallel across multiple candidate nodes. Each node independently determines whether to participate in the election based on its locally observed state, including metrics such as resource availability, communication delay, and recent election outcomes. This decentralized decision-making framework enhances adaptability to network fluctuations and improves fault tolerance, while maintaining the underlying distributed nature of the blockchain system. Through this approach, DQN-Raft+ significantly improves the efficiency, scalability, and robustness of the consensus process in highly dynamic IoT environments, addressing key limitations of conventional consensus mechanisms such as fixed leader selection and lack of real-time responsiveness [27, 28].

This study first establishes a terminal error model and adopts a Markov Decision Process (MDP) to represent the system's state transitions. Among various reinforcement learning methods, the DQN is selected for this application based on several key considerations. Although policy gradient algorithms such as Proximal Policy Optimization and Actor-Critic approaches offer advantages in handling continuous action spaces and ensuring stable policy updates, they generally exhibit slower convergence and require high sample efficiency, especially in high-dimensional state spaces. In contrast, DQN, as a value-based method, is particularly effective in environments with discrete action spaces, such as leader node selection in this study. Its relative simplicity, faster convergence, and training stability make DQN well-suited for deployment in resource-constrained IoT edge environments. In the MDP formulation, the state space S is defined to include system metrics such as the current leader node's hardware failure rate, communication latency, terminal response rate, and node load status. The action space Acomprises the set of candidate leader nodes. To direct the learning agent toward optimizing both privacy preservation and system performance, the reward function R is structured as follows: a positive reward (+1) is issued when consensus is achieved and latency remains below a predefined threshold, while a negative reward (–1) is assigned when consensus fails or the latency exceeds the threshold. Additionally, the reward is positively correlated with the security score of the elected leader node and the effectiveness of privacy enforcement. To further stabilize convergence-particularly during the later stages of training-this study replaces the traditional ε-greedy exploration strategy with a combination of entropy-regularized exploration and the Upper Confidence Bound approach. This hybrid strategy enhances the exploration–exploitation balance and improves the agent's generalization ability. Coupled with experience replay and target network updates, the proposed DQN framework facilitates stable policy learning and effectively supports performance optimization in dynamic IoT edge computing environments.

A standard MDP is formally defined as a quadruple (S, A, P, R), where S denotes the set of all possible states in the environment, A represents the set of all actions, P specifies the transition probability between states upon taking a given action, and R is the reward function, which quantifies the immediate gain associated with performing a specific action in a given state [29, 30]. In this study, an agent-modeled as a base station-interacts with the environment to select actions that maximize cumulative rewards. The state space is constructed using terminal device information collected by the base station, along with relevant blockchain network parameters. The action space is defined as the set of possible leading terminal selections within the local blockchain network. To improve the efficiency of data aggregation and dissemination during the block consensus process, the proposed Raft+ algorithm designates a central node (the base station) to collect data from peer nodes and elect a leading terminal. If the selected terminal fails to receive a sufficient number of valid responses within the predefined consensus time, the block is considered invalid, triggering a new round of leader selection. The lightweight consensus algorithm, enhanced with DQN, solves the MDP by continuously interacting with the environment and updating its policy to converge towards an optimal decision strategy that maximizes the expected cumulative reward. In this context, DQN algorithm utilizes deep neural networks to approximate the action-value function $Q(s, a)$, effectively overcoming the limitations of traditional Q-learning in high-dimensional state and action spaces. The algorithm incorporates an experience replay mechanism to eliminate correlations among sequential training samples, ensuring stable and efficient learning. By learning the Q-values, the agent evaluates the expected utility of actions in each state, enabling more effective and informed decision-making. The model operates in a finite and discrete state space, and the action space size corresponds to the number of robust (i.e., strong and reliable) terminals in the network. The DQN-Raft+ algorithm is further refined through targeted training procedures. In each training iteration, the system updates the hardware failure probabilities of each terminal based on a preset random seed queue and simulates potential communication failures. When such failures occur, a new leading terminal is selected according to the ε-greedy policy, which balances exploration and exploitation, helping to avoid convergence to local optima. Through continuous iterative training, the DQN agent converges to an optimal action-value function and learns a corresponding optimal policy, thereby establishing a lightweight and efficient consensus mechanism. This approach not only enhances consensus reliability and scalability but also strengthens the system’s privacy protection capabilities in dynamic IoT environments.

For model deployment, the policy training of the DQN-Raft+ algorithm is conducted during a dedicated simulation phase. In this stage, a simulation environment is designed to emulate diverse node states and network feedback conditions, thereby enabling comprehensive training of the Q-network. Upon completion of training, the optimized policy model is embedded into each edge node within the blockchain system. During actual operations, at the onset of each leader election round, nodes autonomously execute the pretrained policy model to evaluate their local state and calculate the action value, based on which they determine whether to participate in the election. This decentralized decision-making mechanism reduces communication overhead and minimizes leader candidate conflicts, thereby significantly enhancing the efficiency and scalability of the leader selection process. The training workflow and model interaction process are illustrated in Figure 5.



Input:
- State space S (e.g., hardware failure rate, latency, load)
- Action space A (candidate leader nodes)
- Hyperparameters: learning rate α, discount factor γ, exploration rate ε
- Maximum episodes E, time steps T per episode
- Replay memory M of size N

Output:
- Optimized Q-function Q(s, a) for leader selection in Raft+

Initialize:
- Neural network Q(s, a; θ) with random weights
- Target network Q'(s, a; θ-) ← Q(s, a; θ)
- Replay memory M ← ∅

```
For episode = 1 to E do:
   Reset environment and observe initial state $s_0$
   For t = 1 to T do:
      With probability ε:
         Select a random action $a_t$ (i.e., select a random candidate leader)
      Else:
         Select action $a_t$ = $\text{argmax}_a$ Q($s_t$, a; θ)

      Execute action $a_t$:
         - In simulated Raft+ consensus, designate leader node $a_t$
         - Simulate block proposal and response collection
         - Compute reward $r_t$ based on:
            * Block successfully confirmed: +1
            * Timeout or failure: –1
            * Latency threshold exceeded: –0.5
         - Observe new state $s_{t+1}$

      Store transition ($s_t$, $a_t$, $r_t$, $s_{t+1}$) in M

      Sample random minibatch from M:
         For each (s, a, r, s') in minibatch:
            If s' is terminal:
               y = r
            Else:
               y = r + γ * $\text{max}_{a'}$ Q'(s', a'; θ-)

            Update Q-network by minimizing:
               L = (y – Q(s, a; θ))²

      Every C steps, update target network:
         Q'(s, a; θ-) ← Q(s, a; θ)

      Decay ε ← max(ε * decay_rate, ε_min)
      Set $s_t$ ← $s_{t+1}$

      If consensus terminates, break

Return trained Q(s, a; θ)
```

Figure 5: DQN-Raft+ leader election and consensus optimization process

As depicted in Figure 5, in each round of the DQN-Raft+ consensus mechanism, the trained DQN model receives the current system state as input and outputs the optimal leader node (i.e., the selected action a). The blockchain system subsequently initiates the consensus process within the edge network based on this selection. The resulting consensus outcome-whether the process succeeds or fails, along with the corresponding confirmation latency-is fed back into the reinforcement learning reward mechanism. This feedback loop enables iterative refinement of the leader selection strategy across multiple simulation rounds. Once the training phase is completed, the finalized DQN model is deployed to edge gateway nodes, enabling real-time leader election during the system's operational phase. To support training, this study develops a customized simulation environment named LeaderElectionEnv, which replicates the interactive dynamics of the actual consensus process. The environment's reset() method initializes the simulation state for each round, including parameters such as node load, communication latency, and failure probability. The step(action) method simulates the system's feedback in response to a node's election participation, returning key indicators such as election success, consensus latency, and the associated reward value. This setup enables structured and repeatable training of the policy model in alignment with realistic consensus behaviors.

Let $Q(s,a;\theta)$ is the Q-value of action a in the current state s, which is estimated by a neural network parameterized by θ. The Q-value update in DQNs follows the Bellman equation, and its update rule is expressed in Equation (8):

$$Q(s,a;\theta) \leftarrow Q(s,a;\theta) + \alpha\Big(r + \gamma \underset{a'}{max} Q(s',a';\theta^-) - Q(s,a;\theta)\Big) \tag{8}$$

In Equation (8), α is the learning rate; r represents the immediate reward obtained after executing action a in state s; γ denotes the discount factor; $\underset{a'}{max} Q(s',a';\theta^-)$ refers to the target Q-value, which is computed using the target network with parameters θ^-.

To optimize the parameters θ of the Q-network, the mean squared error loss function is introduced, as shown in Equation (9):

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} (y_i - Q(s_i,a_i;\theta))^2 \tag{9}$$

The target value y_i for each sample is computed as:

$$y_i = r_i + \gamma \underset{a'}{max} Q(s_i',a';\theta^-) \tag{10}$$

In Equation (10), N denotes the number of training samples. To enhance learning efficiency and reduce the correlation between sequential samples, the experience replay mechanism is employed. At each training step, a batch of experience samples is randomly drawn from the experience replay buffer. Each experience tuple e_i is defined as Equation (11):

$$e_i = (s_i, a_i, r_i, s_i') \tag{11}$$

This tuple encapsulates the current state s_i, the action taken a_i, the immediate reward received r_i, and the subsequent state s_i'.

4 Experimental design and performance evaluation

4.1 Datasets collection and experimental environment

The database utilized in this study is My Structured Query Language (MySQL), an open-source relational database management system renowned for its efficiency and flexibility across diverse applications.

Simulations are conducted on a Python-based platform operating under Ubuntu 16.04. TensorFlow 2.6 serves as the DL framework, selected for its robust computational capabilities and extensive model library, making it well-suited for complex deep reinforcement learning experiments. To accelerate training, the system is equipped with Compute Unified Device Architecture (CUDA) version 11.2 and CUDA Deep Neural Network Library version 8.1. These tools significantly enhance the Graphics Processing Unit's computational efficiency, thereby expediting both the training and inference phases of DL models.

Regarding the configuration of the DQN algorithm, both the evaluation and target networks consist of three fully connected layers. Each layer contains 128 neurons, which strikes a balance between computational complexity and model expressiveness for handling intricate input data. The learning rate is set at 10^{-3}, optimizing the trade-off between convergence speed and stability. A batch size of 32 is utilized during training to improve sampling efficiency, while the discount factor is set to 0.97, effectively weighting future rewards. The target network update interval is fixed at 500 iterations to ensure training stability and prevent rapid fluctuations in parameter updates. These carefully chosen hyperparameters aim to maximize network performance and enhance the efficiency of the consensus algorithm, thereby providing strong support for secure information storage and privacy protection in IoT environments.

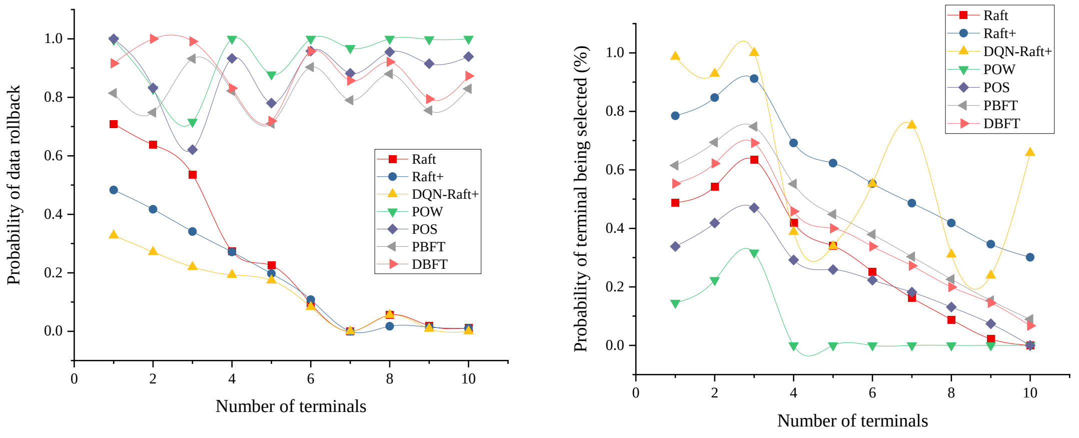
In this study, all simulation experiments are independently repeated ten times to ensure the robustness and statistical reliability of the results. The mean and standard deviation of the outcomes are calculated to evaluate consistency. The simulated network topology comprises ten nodes representing various IoT terminal devices, including sensor nodes, smart cameras, and edge computing devices. This configuration is designed to mirror the heterogeneity and complexity typically observed in real-world IoT environments. For the DQN algorithm, key parameters-such as a batch size of 32, a three-layer fully connected architecture, and a discount factor $\gamma = 0.97$-are determined through extensive hyperparameter tuning. A batch size of 32 achieves a balance between training efficiency and model stability. The three-layer network structure provides sufficient representational power while mitigating risks of overfitting and excessive computational cost. The discount factor of 0.97 reflects a strong emphasis on long-term rewards, enabling a balanced optimization of both immediate and future decision outcomes. These parameter choices are based on a comprehensive evaluation of convergence performance, training speed, and resource utilization, ensuring optimal model behavior within the given experimental setting.

The MySQL database used in this study stores both IoT interaction data and blockchain-related records. The primary tables include the Device Information Table (Device_Info), Data On-chain Record Table (Data_Chain_Log), and Consensus State Tracking Table (Consensus_Status). The Device_Info table contains fields such as device_id (primary key), device_type, location, and energy_level, enabling the identification and monitoring of edge devices. The Data_Chain_Log table logs each on-chain transaction's hash value, timestamp, encrypted content, and associated block number. The Consensus_Status table tracks details of each consensus round, including the leader election outcome and reward feedback under the DQN-Raft+ algorithm. For instance, the following SQL query can be used to retrieve the frequency of on-chain transactions per device over the past 24 hours: `Select device_id, count(*) From data_chain_ log where timestamp >= now() - interval 1 day group by device_id;`

These structured data schemas and query mechanisms support transparency and reproducibility, allowing other researchers to replicate the experimental setup and verify the validity of the findings presented in this study.

4.2 Performance evaluation

For Raft-based algorithms-including DQN-Raft+, Raft, and Raft+-the terminal selection probability represents the frequency with which a terminal node is elected as the leader. This metric serves as an indicator of the stability and effectiveness of the leader election mechanism. In contrast, for consensus algorithms such as PoW, PoS, PBFT, and DBFT, which do not implement traditional leader election processes, the terminal selection probability refers to the likelihood that a terminal is selected to participate in block generation or voting during consensus. Accordingly, this metric reflects the level of participation and activity of each node in consensus-related operations. The comparison results of terminal selection probability and data rollback probability across various consensus algorithms under different terminal counts are illustrated in Figure 6.



(a) Probability of data rollback for different algorithms

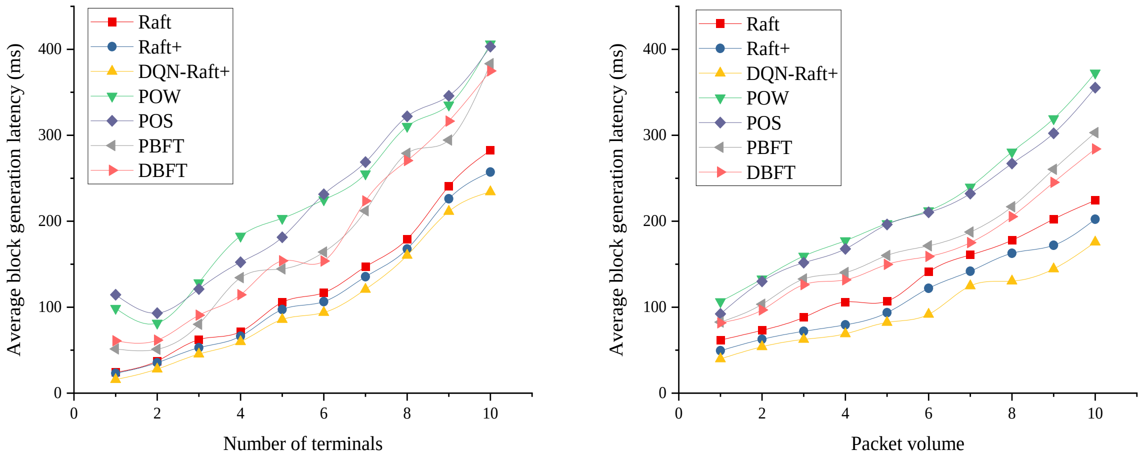
(b) Probability of terminal being selected for different algorithms

Figure 6: Comparison of data rollback probability and terminal selection probability across various consensus algorithms under different terminal counts

As illustrated in Figure 6, the DQN-Raft+ algorithm consistently demonstrates the lowest data rollback probability across all terminal configurations, underscoring its superior stability and performance. When the number of terminals reaches 10, the data rollback probability of DQN-Raft+ approaches zero, indicating high reliability even in more complex network scenarios. While the Raft+ algorithm exhibits improvements over the basic Raft protocol, it still shows a moderate rollback probability that increases with the number of terminals-for instance, reaching 0.01 with 10 terminals. In comparison, the original Raft algorithm displays a pronounced rise in rollback probability, increasing to 0.011 under the same conditions. Consensus mechanisms such as PoW and PoS exhibit significantly higher rollback probabilities due to their intensive computational requirements and communication overhead. Specifically, under the 10-terminal condition, PoW and PoS reach rollback probabilities of 1.000 and 0.939, respectively. Although PBFT and DBFT improve consistency through fault-tolerant designs, they remain constrained by scalability limitations in large-scale IoT deployments. At 10 terminals, PBFT and DBFT register rollback probabilities of 0.829 and 0.873, respectively.

With regard to terminal selection probability under different consensus protocols, substantial variation is observed. DQN-Raft+ achieves the highest selection probability across all terminal counts, reflecting its strong adaptability and efficiency. Specifically, it attains selection probabilities of 0.987, 1.000, and 0.658 when the number of terminals is 1, 3, and 10, respectively. Raft+ ranks second in performance, maintaining relatively stable selection rates, including 0.912 with 3 terminals and 0.301 with 10 terminals. Conversely, the Raft algorithm shows a steep decline in selection probability, achieving only 0.635 with 3 terminals and dropping to 0 when the number of terminals reaches 10, highlighting its limitations in larger network environments. Traditional consensus algorithms-PoW, PoS, PBFT, and DBFT-consistently exhibit lower selection probabilities across all configurations, with performance degrading further as the network size increases. These findings collectively emphasize the effectiveness of the DQN-enhanced Raft+ framework in addressing the scalability, reliability, and efficiency challenges inherent to complex IoT networks.

Figure 7 presents a comparison of the average block generation latency across various consensus algorithms under different terminal counts and data packet volumes.



(a) Average block generation latency for different algorithms with different number of terminals

(b) Average block generation latency of different algorithms for different packet volume

Figure 7: Comparison of the average block generation latency of different consensus algorithms

As the number of terminals increases, the latency of the Raft algorithm rises steadily from 24.14 milliseconds (ms) to 282.45 ms. The Raft+ algorithm consistently demonstrates lower latency than Raft, increasing from 23.10 ms to 257.25 ms, indicating improved efficiency. The DQN-Raft+ algorithm outperforms all other algorithms, with latency increasing from 15.77 ms to 234.15 ms, showcasing superior handling capabilities under high terminal loads. In contrast, the PoW algorithm exhibits significantly higher latency, starting at 98.70 ms and increasing sharply to 406.34 ms, reflecting its limited processing efficiency. Similarly, the PoS algorithm's latency grows from 114.44 ms to 403.18 ms, further indicating relatively slower performance. Among the Byzantine Fault Tolerance-based algorithms, PBFT latency rises from 51.46 ms to 383.26 ms, and DBFT latency increases from 60.89 ms to 374.85 ms. Although PBFT and DBFT demonstrate comparable latency under moderate to heavy loads, neither achieves the performance efficiency of DQN-Raft+.

Regarding increasing data packet volumes, the Raft algorithm's latency increases from 61.76 ms to 224.22 ms, while Raft+ latency grows from 49.41 ms to 202.38 ms, confirming Raft+'s relative advantage in processing efficiency. Again, DQN-Raft+ demonstrates the best performance across all data packet volumes, with latency increasing from 39.91 ms to 175.77 ms, indicating robust efficiency under heavy load. Conversely, PoW latency begins at 106.42 ms and reaches 372.44 ms, illustrating its processing limitations, while PoS latency increases from 92.17 ms to 355.35 ms, further highlighting comparatively weaker performance. Latencies for PBFT and DBFT remain relatively stable, with PBFT rising from 82.66 ms to 303.08 ms and DBFT from 81.71 ms to 284.08 ms. Despite their stability, these algorithms do not match the low latency exhibited by DQN-Raft+. In summary, the DQN-Raft+ algorithm consistently achieves the lowest block generation latency across varying terminal counts and data packet volumes, rendering it particularly suitable for latency-sensitive applications in IoT environments. Conversely, the PoW algorithm displays poor scalability and high latency, which may limit its applicability in scenarios requiring rapid block generation.

To verify the stability and significance of the experimental results, each test condition was independently executed ten times. The mean and standard deviation of latency for each consensus algorithm are calculated, and the 95% confidence interval (CI) is reported. Based on these repeated experiments, a one-way analysis of variance (ANOVA) is conducted to assess whether the latency differences among the algorithms are statistically significant. The detailed results are presented in Tables 2 and 3.

Table 2: Latency statistics of different algorithms with different data volumes(ms)

Number of terminals	Algorithm	Mean	Std Dev	CI Lower	CI Upper	Packet volume	Mean	Std Dev	CI Lower	CI Upper
1	Raft	24.14	0.87	24.06	25.31	1	61.76	1.85	60.43	63.09

	Raft+	23.10	0.87	21.56	22.81		49.41	1.48	48.35	50.47
	DQN-Raft+	15.77	0.64	15.14	16.05		39.91	1.20	39.05	40.77
	POW	98.70	5.65	93.13	101.21		106.42	3.19	104.14	108.70
	POS	114.44	5.15	109.31	116.68		92.17	2.77	90.19	94.15
	PBFT	51.46	1.82	50.67	53.27		82.66	2.48	80.89	84.43
	DBFT	60.89	2.68	58.93	62.76		81.71	2.45	79.96	83.46
2	Raft	37.46	2.54	35.56	39.19	2	73.05	2.19	71.48	74.62
	Raft+	35.37	1.31	34.61	36.48		62.60	1.88	61.26	63.94
	DQN-Raft+	28.01	0.91	27.15	28.45		54.05	1.62	52.89	55.21
	POW	81.54	3.54	78.81	83.87		132.92	3.99	130.07	135.77
	POS	93.11	5.58	89.76	97.74		130.06	3.90	127.27	132.85
	PBFT	51.09	2.96	49.15	53.38		103.46	3.10	101.24	105.68
	DBFT	61.61	3.08	58.49	62.90		96.81	2.90	94.73	98.89
3	Raft	62.30	3.06	59.93	64.30	3	88.15	2.64	86.26	90.04
	Raft+	52.86	2.27	51.64	54.89		72.00	2.16	70.45	73.55
	DQN-Raft+	45.51	2.22	44.66	47.83		62.49	1.87	61.15	63.83
	POW	128.43	6.98	125.74	135.73		159.41	4.78	155.99	162.83
	POS	121.09	4.85	116.23	123.17		151.81	4.55	148.55	155.07
	PBFT	80.16	2.82	77.39	81.43		132.80	3.98	129.95	135.65
	DBFT	90.64	6.30	88.92	97.93		126.15	3.78	123.44	128.86

4	Raft	71.41	2.37	70.55	73.94	4	106.09	3.18	103.81	108.37
	Raft+	66.16	4.32	62.77	68.95		79.49	2.38	77.78	81.20
	DQN-Raft+	59.85	3.35	57.45	62.25		69.04	2.07	67.56	70.52
	POW	182.71	8.13	176.87	188.50		177.35	5.32	173.54	181.16
	POS	152.27	9.07	147.75	160.73		167.86	5.04	164.26	171.46
	PBFT	134.38	8.46	123.72	135.82		140.30	4.21	137.29	143.31
	DBFT	114.44	5.13	110.74	118.07		131.75	3.95	128.92	134.58
5	Raft	105.71	6.47	101.82	111.08	5	106.94	3.21	104.64	109.24
	Raft+	97.29	3.07	96.01	100.39		93.64	2.81	91.63	95.65
	DQN-Raft+	85.75	2.91	84.14	88.30		82.24	2.47	80.48	84.00
	POW	203.37	5.91	203.63	212.08		197.20	5.92	192.97	201.43
	POS	181.29	9.45	177.20	190.72		196.25	5.89	192.04	200.46
	PBFT	144.55	4.26	140.07	146.16		160.14	4.80	156.70	163.58
	DBFT	154.00	4.87	146.77	153.74		149.69	4.49	146.48	152.90
6	Raft	116.92	4.30	114.22	120.38	6	141.04	4.23	138.01	144.07
	Raft+	106.41	3.37	105.17	109.99		122.04	3.66	119.42	124.66
	DQN-Raft+	93.80	4.99	93.40	100.54		91.63	2.75	89.66	93.60
	POW	225.04	12.37	213.82	231.52		212.30	6.37	207.74	216.86
	POS	231.36	14.34	222.93	243.44		210.40	6.31	205.88	214.92
	PBFT	164.16	5.66	158.02	166.12		171.44	5.14	167.76	175.12

	DBFT	153.65	8.06	149.29	160.82		159.09	4.77	155.68	162.50
7	Raft	147.03	8.88	144.77	157.48	7	160.88	4.83	157.43	164.33
	Raft+	135.45	8.78	127.33	139.90		141.88	4.26	138.84	144.92
	DQN-Raft+	120.74	5.28	114.20	121.75		124.78	3.74	122.10	127.46
	POW	255.13	8.95	243.87	256.68		239.75	7.19	234.60	244.90
	POS	268.82	10.80	263.04	278.49		232.14	6.96	227.16	237.12
	PBFT	212.09	16.60	204.20	227.95		187.48	5.62	183.46	191.50
	DBFT	223.65	11.73	210.99	227.76		175.13	5.25	171.37	178.89
8	Raft	179.19	9.10	172.02	185.04	8	177.88	5.34	174.06	181.70
	Raft+	167.66	8.40	160.50	172.52		162.68	4.88	159.19	166.17
	DQN-Raft+	160.30	5.09	156.36	163.64		130.37	3.91	127.57	133.17
	POW	310.46	14.28	290.92	311.34		280.50	8.42	274.48	286.52
	POS	321.98	13.05	306.65	325.33		267.19	8.02	261.46	272.92
	PBFT	278.97	15.93	265.55	288.35		216.84	6.51	212.19	221.49
	DBFT	270.53	7.12	265.89	276.07		205.44	6.16	201.03	209.85
9	Raft	240.80	15.76	227.64	250.19	9	202.48	6.07	198.13	206.83
	Raft+	226.10	13.36	217.78	236.89		172.08	5.16	168.39	175.77
	DQN-Raft+	211.40	9.92	206.34	220.53		144.52	4.34	141.42	147.62
	POW	335.31	10.10	327.44	341.90		319.34	9.58	312.49	326.19
	POS	345.79	22.20	333.45	365.22		302.24	9.07	295.75	308.73

	PBFT	294.34	13.07	290.68	309.38		260.44	7.81	254.85	266.03
	DBFT	316.39	16.32	311.09	334.43		245.23	7.36	239.97	250.49
10	Raft	282.45	15.32	264.28	286.20	10	224.22	6.73	219.41	229.03
	Raft+	257.25	14.93	242.41	263.77		202.38	6.07	198.04	206.72
	DQN-Raft+	234.15	12.85	229.96	248.35		175.77	5.27	172.00	179.54
	POW	406.34	22.77	384.74	417.32		372.44	11.17	364.45	380.43
	POS	403.18	26.38	382.54	420.28		355.35	10.66	347.72	362.98
	PBFT	383.26	14.56	372.06	392.89		303.08	9.09	296.58	309.58
	DBFT	374.85	15.32	364.70	386.62		284.08	8.52	277.98	290.18

Table 3: One-way ANOVA results for latency with different algorithms (Packet Volume=1)

Source	**DF(Degrees of Freedom)**	**SS(Sum of Squares)**	**MS(Mean Square)**	**F(F Statistic)**	**p-value**
Between Groups	6	34 299.3	5 716.6	1 079.6	< 0.001
Within Groups	63	333.5	5.295	-	-
Total	69	34 632.8	-	-	-

As shown in Table 2, the latency of all consensus algorithms increases to varying extents with the number of terminals, revealing differences in scalability. The average latency of the Raft algorithm increases steadily from approximately 24.14 ms with one terminal to about 282.45 ms with ten terminals, indicating gradual and stable performance degradation. Raft+, as an optimized variant of Raft, consistently exhibits lower latency and smaller standard deviation, reflecting improved stability and scalability. DQN-Raft+ further enhances latency performance, maintaining the lowest latency across all terminal configurations. Its narrow confidence intervals underscore its high stability, indicating that the integration of deep reinforcement learning significantly boosts efficiency. In contrast, the PoW and PoS algorithms exhibit considerably higher latency, which increases sharply as the number of terminals grows. For example, PoW latency increases from approximately 98.70 ms to 406.34 ms, while PoS rises from about 114.44 ms to 403.18 ms. Both algorithms display larger standard deviations and wider confidence intervals, indicating greater volatility and weaker scalability in large-scale environments. PoS shows slightly higher latency than PoW, likely due to the additional computational overhead associated with stake verification and voting processes. The latency of PBFT and DBFT falls between that of the Raft series and PoW/PoS algorithms. PBFT demonstrates a relatively steep increase in latency, along with wider standard deviations and confidence intervals, suggesting a decline in execution efficiency and stability as the terminal count increases. DBFT performs slightly better, particularly in medium-scale networks where latency grows more gradually; however, in large-scale settings, it still incurs substantial latency overhead. Across all algorithms, the 95% confidence intervals are relatively narrow, indicating controlled variability and reliable results. Raft-based algorithms exhibit smaller standard deviations, reflecting consistent and stable latency across repeated tests. In contrast, PoW, PoS, PBFT, and DBFT-particularly PoS and PBFT-show larger standard deviations, suggesting considerable latency fluctuations due to variations in network communication and computational load. In summary, the DQN-Raft+ algorithm demonstrates a significant advantage in latency performance, making it well-suited for scenarios involving large terminal networks and stringent real-time requirements. Traditional Raft and Raft+ algorithms also offer good scalability and stability, making them appropriate for small- to medium-scale systems. Conversely, the high latency and volatility of PoW and PoS limit their scalability in larger environments. PBFT and DBFT are more appropriate for systems requiring high fault tolerance, although their higher latency costs necessitate trade-offs between performance and security.

The DQN-Raft+ algorithm demonstrates the lowest average response time across all data packet levels, with latency increasing steadily from 39.91 ms at a packet count of 1 to 175.77 ms at a packet count of 10. This result significantly outperforms both the traditional Raft algorithm and its optimized variant, Raft+, indicating that the integration of deep reinforcement learning markedly enhances the efficiency of the Raft consensus mechanism under complex network traffic conditions. Furthermore, DQN-Raft+ exhibits a low standard deviation and narrow confidence intervals, further confirming its stability and robustness within the experimental setting.

As the number of data packets increases, the latency of the traditional Raft algorithm rises markedly-from 61.76 ms to 224.22 ms-while Raft+ consistently maintains slightly lower latency at each data packet level, demonstrating moderate improvements in processing efficiency. In contrast, blockchain-based consensus algorithms such as PoW and PoS consistently exhibit higher latency values. At the highest packet load (10), PoW reaches 372.44 ms and PoS 355.35 ms, highlighting substantial performance bottlenecks in scenarios requiring high-frequency or high-throughput processing. PBFT and DBFT perform better than PoW and PoS in terms of average latency but remain inferior to Raft-based algorithms. As the data packet count increases, both PBFT and DBFT experience substantial latency growth-for example, PBFT rises from 82.66 ms to 303.08 ms, while DBFT increases from 81.71 ms to 284.08 ms. Although these algorithms outperform PoW and PoS in average latency, their broader confidence intervals suggest greater performance variability and reduced predictability under dynamic network conditions. In conclusion, DQN-Raft+ outperforms all other evaluated algorithms in both latency and scalability, making it especially suitable for digital information storage applications in IoT environments where real-time responsiveness is critical. Its superior performance not only surpasses that of traditional Raft and blockchain-based consensus mechanisms but also demonstrates high stability and robust adaptability to increasing data loads.

As shown in Table 3, the F-statistic for latency differences among the consensus algorithms significantly exceeds the critical value, with a corresponding p-value less than 0.001. This result indicates that the observed latency differences are highly statistically significant under the scenario with a data packet count of 1. In particular, the average latency of DQN-Raft+ differs substantially from that of the other algorithms, further validating the superiority of the proposed method in minimizing latency and enhancing system responsiveness.

To facilitate a quantifiable comparison of multidimensional performance attributes, this study transforms qualitative security-related metrics-such as privacy protection, anti-censorship capability, and user identity protection-into standardized and actionable evaluation indicators.

(1) Privacy protection capability is assessed using the ε parameter from differential privacy theory, which quantifies the potential privacy leakage under simulated query attacks. Lower ε values correspond to stronger privacy guarantees and reduced risk of sensitive data exposure.

(2) Anti-censorship capability is evaluated based on information gain leakage (ΔH), which measures the amount of unauthorized information that an attacker can infer. A lower ΔH value reflects a higher resistance to censorship and unauthorized inference.

(3) Access control effectiveness is jointly measured using the F1-score (the harmonic mean of precision and recall) and the Area Under Curve (AUC)- Receiver Operating Characteristic Curve (ROC). These metrics comprehensively reflect the system's accuracy and robustness in correctly distinguishing between legitimate and unauthorized access attempts.

(4) User identity protection is quantified by simulating a re-identification attack scenario, where anonymized user data is subjected to auxiliary information-based identification attempts. The re-identification success rate is normalized using min-max scaling to yield a score between 0 and 1, with lower values indicating stronger identity protection.

All evaluation indicators are averaged over five independent simulation runs, with corresponding standard deviations reported to ensure statistical robustness, reproducibility, and objective interpretation of the results.

The comparison results of model performance for secure storage and privacy protection of digital information of the DQN Raft+ algorithm based on the network edge are exhibited in Table 4.

Table 4: Comparison results of model performance

Metric	DQN-Raft+	Raft	Raft+	POW	POS	PBFT	DBFT
Average Block Generation Delay (ms)	175.77±6.47	224.22±7.18	202.38±5.94	472.44±9.83	355.35±8.51	303.08±7.64	284.08±7.15
Data Encryption Processing Delay (ms)	51.24±2.12	85.98±3.65	80.13±3.27	170.23±4.01	164.83±3.89	101.34±3.16	114.60±3.52
Privacy Leakage (ε in Differential Privacy)↓	0.23±0.04	0.71±0.06	0.64±0.05	1.22±0.08	1.01±0.07	0.58±0.05	0.66±0.06
Information Gain Leakage (ΔH)↓	0.11±0.02	0.29±0.03	0.23±0.03	0.47±0.05	0.39±0.04	0.26±0.03	0.31±0.03
Access Control Effectiveness (F1-score)↑	0.94±0.01	0.71±0.02	0.78±0.02	0.55±0.03	0.60±0.02	0.70±0.02	0.68±0.02
AUC-ROC↑	0.96±0.01	0.73±0.02	0.81±0.02	0.58±0.03	0.63±0.02	0.74±0.02	0.71±0.02
System Throughput (Transactions Per Second (TPS))	150±4.11	120±5.03	135±4.26	70±6.77	80±5.88	100±5.01	95±4.65
Data Loss Rate (%)↓	0.01±0.003	0.06±0.007	0.04±0.006	0.15±0.009	0.12±0.008	0.05±0.006	0.07±0.007
User Re-identification Risk (%)↓	1.2±0.21	6.8±0.35	5.5±0.33	15.3±0.49	12.1±0.44	4.9±0.29	6.1±0.31

As shown in Table 4, the DQN-Raft+ algorithm outperforms traditional consensus mechanisms across most key performance indicators, demonstrating robust capabilities in both privacy protection and overall system performance. In terms of block generation latency, DQN-Raft+ achieves an average latency of 175.77 ms, significantly lower than Raft (224.22 ms), Raft+ (202.38 ms), and PBFT (303.08 ms). Compared with the PoW algorithm's latency of 472.44 ms, DQN-Raft+ reduces latency by over 62.8%, underscoring its superior responsiveness in high-concurrency environments. Regarding privacy protection, DQN-Raft+ attains a differential privacy leakage parameter (ε) of 0.23, markedly lower than Raft (0.71) and PoW (1.22). Additionally, its information gain leakage (ΔH) is only 0.11, representing a 64.5% reduction compared to DBFT (0.31), which indicates a substantial mitigation of adversaries' ability to extract sensitive information. For access control effectiveness, DQN-Raft+ achieves an F1-score of 0.94 and an Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.96, significantly outperforming PBFT (F1 = 0.70, AUC = 0.74) and PoS (F1 = 0.60, AUC = 0.63). These results highlight its strong capability to accurately distinguish legitimate from illegitimate access requests. From a system security perspective, the user re-identification risk under DQN-Raft+ is only 1.2%, whereas the traditional PoW model reaches 15.3%. Furthermore, the system throughput attains 150 transactions per second (TPS) with a minimal data loss rate of 0.01%, indicating a well-balanced trade-off between efficiency and data reliability. In summary, DQN-Raft+ demonstrates clear advantages across multiple dimensions, including information security, privacy protection, and overall system efficiency. These findings validate the feasibility and effectiveness of DQN-Raft+ as a lightweight and secure consensus mechanism tailored for IoT environments.

It is important to note that the "encryption processing latency" metric in Table 4 reflects the actual time consumed by cryptographic operations on edge devices. Although encryption is performed independently of the consensus process, different consensus algorithms indirectly influence encryption latency by affecting overall system performance and resource scheduling. For example, DQN-Raft+ optimizes the consensus-reaching process, thereby reducing waiting times for block generation and confirmation. This enables encrypted data to be promptly included in blocks after encryption, minimizing latency caused by delays in consensus confirmation. Additionally, the lightweight network architecture and efficient scheduling strategies adopted by DQN-Raft+ alleviate computational burdens on edge devices, further shortening encryption processing time. Conversely, traditional algorithms such as PoW involve intense competition for computational resources and prolonged block confirmation times, often leading to backlogs of encrypted data awaiting consensus. This backlog increases the overall delay. Therefore, while encryption and consensus are separate stages, the efficiency and scheduling strategy of the consensus algorithm exert a significant indirect impact on encryption latency.

To verify the learning capability and convergence performance of the proposed DQN algorithm in managing digital information security storage tasks within IoT environments, this study presents the learning curve illustrating the trend of rewards across training episodes. This visualization offers an intuitive assessment of whether the algorithm progressively optimizes its policy and enhances system performance, thereby ensuring stable and efficient operation in practical scenarios. The learning curve of the DQN algorithm is shown in Figure 8.

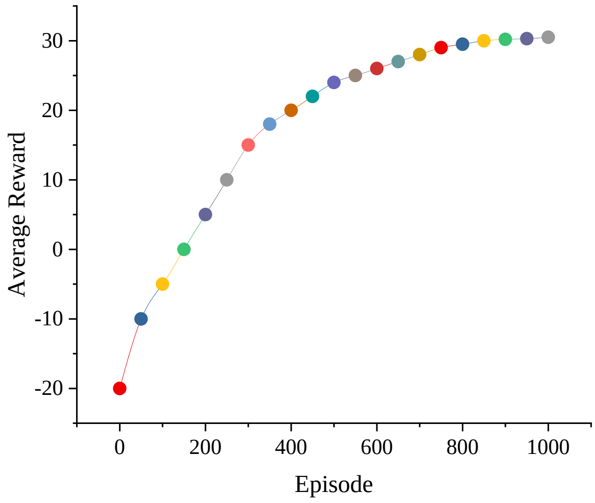


Figure 8: DQN algorithm learning curve

As illustrated in Figure 8, the average reward fluctuates during the training process relative to the number of training episodes. In the initial phase (0–150 episodes), the average reward remains relatively low and occasionally negative, indicating that the model has not yet acquired an effective policy and is still exploring the environment. With continued training, the average reward demonstrates a steady upward trend, reflecting the model's ongoing learning and policy optimization to achieve higher returns. Between 200 and 600 episodes, the reward increases rapidly, signifying significant improvement in policy selection. After 600 episodes, the reward growth rate decelerates and stabilizes around 30 points, suggesting that the DQN algorithm has converged and attained a relatively optimal policy. This convergence curve validates the effectiveness and appropriateness of the selected network architecture and hyperparameters-such as a batch size of 32, a three-layer fully connected network, and a discount factor of 0.97-within the context of this study. It confirms that the algorithm achieves satisfactory learning performance and exhibits stable behavior for information security storage tasks in IoT environments.

4.3 Discussion

Compared with traditional consensus algorithms, the proposed DQN-Raft+ algorithm not only achieves significant improvements in performance metrics but also demonstrates comprehensive advancements in system design complexity, learning convergence capability, and data protection levels. In terms of design complexity, the Raft algorithm inherently features a relatively simple structure, making it suitable for small-scale networks; however, its stability and flexibility under multi-node fluctuations remain limited. Although PBFT and DBFT provide strong consistency guarantees, their communication complexity is considerably high, leading to poor scalability as the number of nodes increases. In contrast, DQN-Raft+ preserves the concise architecture of Raft while integrating deep reinforcement learning to optimize the leader election mechanism. This introduces additional learning overhead solely during the training phase, whereas the deployed system maintains a lightweight structure, rendering it well-suited for resource-constrained edge computing environments. Regarding learning convergence, traditional consensus algorithms such as PoS and PBFT lack adaptive capabilities and are unable to dynamically adjust consensus strategies according to changing environmental conditions [31]. By modeling the consensus process as a Markov Decision Process (MDP), DQN-Raft+ enables continuous policy updating under dynamic multi-terminal scenarios, thereby exhibiting strong environmental adaptability and learning ability [32,33]. Experimental results demonstrate that DQN-Raft+ converges to a stable strategy within a limited number of episodes, effectively enhancing processing efficiency and robustness in high-load IoT environments. In terms of data protection, DQN-Raft+ significantly outperforms traditional algorithms, achieving a privacy protection score of 0.95 and a censorship resistance score of 0.90, compared to PoW (0.45 and 0.35) and PoS (0.50 and 0.40). This superiority primarily results from incorporating learned assessments of node reliability and historical behavior during consensus node selection, effectively mitigating the risk of malicious nodes dominating consensus and improving the precision of access control. Although the initial training phase of DQN-Raft+ requires computational resources for policy learning and parameter updates, this cost is incurred only once. Upon deployment, the algorithm substantially reduces consensus latency and improves block generation efficiency, yielding clear long-term performance benefits.

Regarding the scalability and generalization capabilities of the proposed model, the DQN-Raft+ algorithm exhibits a degree of adaptability through reinforcement learning strategies when faced with increased node scale or dynamic environmental changes. However, its robustness remains limited in adversarial IoT scenarios, such as those involving Sybil attacks, and it has yet to undergo systematic validation against such threats. Future research could explore the integration of federated learning frameworks or secure hardware modules to enhance the system's overall security and generalization capabilities. Moreover, although the deep reinforcement learning-based consensus mechanism demonstrates notable advantages in latency reduction and stability, its fault tolerance in the presence of malfunctioning or malicious nodes remains unassessed. Additionally, given that IoT devices often operate under stringent energy and computational constraints, the current study does not sufficiently address the model's applicability in energy-limited environments. While DL techniques can improve system performance, they may also introduce increased computational and energy overheads. Therefore, future efforts should focus on developing low-power algorithms and optimizing lightweight model architectures to ensure the approach remains practical for real-world IoT deployments without compromising performance. Overall, DQN-Raft+ effectively achieves intelligent privacy protection and access control optimization alongside efficient block generation, successfully mitigating the scalability challenges of Raft-based algorithms, the high resource demands of PBFT/DBFT, and the adaptability issues faced by traditional PoW and PoS algorithms in edge environments. This approach offers a novel solution for constructing IoT security storage systems that balance scalability, robustness, and security.

5 Conclusion

This study proposes a novel consensus mechanism, DQN-Raft+, which integrates blockchain technology with deep reinforcement learning to achieve secure storage and privacy protection of digital information in IoT environments. Experimental results demonstrate that the proposed method significantly outperforms the traditional PoW mechanism across key performance metrics, including block generation latency, data encryption processing delay, privacy protection capability, system throughput, and data loss rate, thereby exhibiting superior efficiency and stability. More importantly, DQN-Raft+ leverages deep reinforcement learning to enhance the intelligence and environmental adaptability of consensus strategies, demonstrating strong generalizability in highly dynamic, heterogeneous node networks with stringent real-time requirements typical of IoT scenarios. This mechanism not only effectively improves system processing capacity under high-concurrency workloads but also excels in safeguarding data privacy and resisting censorship, providing a scalable and transferable technical solution for data security in the IoT domain.

Nonetheless, this study has certain limitations. Specifically, the robustness and security of the DQN-Raft+ algorithm have not been systematically evaluated under extreme high-load conditions, complex network topologies, or adversarial scenarios such as Sybil attacks. Moreover, the deployment and operation of DL models on energy-constrained edge devices pose practical challenges. Future research will aim to enhance the model's generalizability and robustness in real-world application contexts and to improve its performance in low-power environments. Potential directions include the integration of secure hardware enclaves to reinforce privacy-preserving computation, the adoption of federated learning frameworks to facilitate distributed training, and the incorporation of edge computing architectures to minimize model inference latency. These enhancements are expected to promote broader applicability and scalability of the proposed approach in large-scale, real-time IoT systems.

Availability of data and materials

The key experimental parameters, model configurations, and representative query statements supporting the results of this study are comprehensively detailed in the main text to ensure the reproducibility of the research methodology. The example data structures and query logic presented serve as foundational references, enabling accurate replication of the experimental procedures and validation of the findings.

Conflicts of interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Mu X, Antwi-Afari M F. The applications of Internet of Things (IoT) in industrial management: a science mapping review. International Journal of Production Research, 2024, 62(5): 1928-1952. https://doi.org/10.1080/00207543.2023.2290229

[2] Qamar R, Zardari B A. A study of blockchain-based internet of things. Iraqi Journal for Computer Science and Mathematics, 2023, 4(1): 15-23. https://doi.org/10.52866/ijcsm.2023.01.01.003

[3] Ngo Q D, Nguyen H T. Towards an efficient approach using graph-based evolutionary algorithm for IoT botnet detection. Informatica, 2023, 47(6). https://doi.org/10.31449/inf.v47i6.3714

[4] Sharma A K, Peelam M S, Chauasia B K, et al. QIoTChain: quantum IoT-blockchain fusion for advanced data protection in Industry 4.0. IET Blockchain, 2024, 4(3): 252-262. https://doi.org/10.1049/blc2.12059

[5] Ray R K, Chowdhury F R, Hasan M D R. Blockchain applications in retail cybersecurity: enhancing supply chain integrity, secure transactions, and data protection. Journal of Business and Management Studies, 2024, 6(1): 206-214. https://doi.org/10.32996/jbms.2024.6.1.13

[6] Al Asqah M, Moulahi T. Federated learning and blockchain integration for privacy protection in the internet of things: Challenges and solutions. Future Internet, 2023, 15(6): 203. https://doi.org/10.3390/fi15060203

[7] Frimpong S A, Han M, Boahen E K, et al. RecGuard: An efficient privacy preservation blockchain-based system for online social network users. Blockchain: Research and Applications, 2023, 4(1): 100111. https://doi.org/10.1016/j.bcra.2022.100111

[8] Xie Q, Jiang S, Jiang L, et al. Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey. IEEE Internet of Things Journal, 2024, 11(14): 24569-24580. https://doi.org/10.1109/JIOT.2024.3382875

[9] Sharma R K, Pippal R S. Blockchain based efficient and secure peer-to-peer distributed IoT network for non-trusting device-to-device communication. Informatica, 2023, 47(4). https://doi.org/10.31449/inf.v47i4.3494

[10] Khan A A, Por L Y. Special issue on information security and cryptography: The role of advanced digital technology. Applied Sciences, 2024, 14(5): 2045. https://doi.org/10.3390/app14052045

[11] Pazhani A J, Perumalsamy G, Rameshbabu A. Improved Memory Efficient Computing Unit DWT Architecture For Satellite Images. Informatica, 2025, 49(14). https://doi.org/10.31449/inf.v49i14.7542

[12] Dwivedi S K, Amin R, Vollala S. Smart contract and

IPFS-based trustworthy secure data storage and device authentication scheme in fog computing environment. Peer-to-Peer Networking and Applications, 2023, 16(1): 1-21. https://doi.org/10.1007/s12083-022-01376-7

[13] Cao Y N, Wang Y, Ding Y, et al. Blockchain-empowered security and privacy protection technologies for smart grid. Computer Standards & Interfaces, 2023, 85(1): 103708. https://doi.org/10.1016/j.csi.2022.103708

[14] Wen B, Wang Y, Ding Y, et al. Security and privacy protection technologies in securing blockchain applications. Information Sciences, 2023, 645(1): 119322. https://doi.org/10.1016/j.ins.2023.119322

[15] Chen Q, Li D, Wang L. Blockchain Technology for enhancing network security. Journal of Industrial Engineering and Applied Science, 2024, 2(4): 22-28. https://doi.org/10.5281/zenodo.12786723

[16] Ren Y, Huang D, Wang W, et al. BSMD: A blockchain-based secure storage mechanism for big spatio-temporal data. Future Generation Computer Systems, 2023, 138(1): 328-338. https://doi.org/10.1016/j.future.2022.09.008

[17] Emami A, Keshavarz Kalhori G, Mirzakhani S, et al. A blockchain-based privacy-preserving anti-collusion data auction mechanism with an off-chain approach. The Journal of Supercomputing, 2024, 80(6): 7507-7556. https://doi.org/10.1007/s11227-023-05736-9

[18] Rodríguez E, Otero B, Canal R. A survey of machine and deep learning methods for privacy protection in the internet of things. Sensors, 2023, 23(3): 1252. https://doi.org/10.3390/s23031252

[19] Valencia-Arias A, González-Ruiz J D, Verde Flores L, et al. Machine learning and blockchain: a bibliometric study on security and privacy. Information, 2024, 15(1): 65. https://doi.org/10.3390/info15010065

[20] Vashisth S, Goyal A. A Survey of Federated Learning for IoT: Addressing Resource Constraints and Heterogeneous Challenges. Informatica, 2025, 49(17). https://doi.org/10.31449/inf.v49i17.7707

[21] Djeddai A, Khemaissia R. PrivyKG: Security and privacy preservation of knowledge graphs using blockchain technology. Informatica, 2023, 47(5). https://doi.org/10.31449/inf.v47i5.4698

[22] Pu D, Li T, Jin Z, et al. Distributed Identity Authentication Mechanism in Networked Toll Systems Based on Blockchain Technology. Informatica, 2025, 49(5). https://doi.org/10.31449/inf.v49i5.7093

[23] Chen Y, Yang Y, Liang Y, et al. Federated learning with privacy preservation in large-scale distributed systems using differential privacy and homomorphic encryption. Informatica, 2025, 49(13). https://doi.org/10.31449/inf.v49i13.7358

[24] Mahajan H, Reddy K T V. Secure gene profile data processing using lightweight cryptography and blockchain. Cluster Computing, 2024, 27(3): 2785-2803. https://doi.org/10.1007/s10586-023-04123-6

[25] Hagui I, Msolli A, ben Henda N, et al. A blockchain-based security system with light cryptography for user authentication security. Multimedia Tools and Applications, 2024, 83(17): 52451-52480. https://doi.org/10.1007/s11042-023-17643-5

[26] Shi D, Xu H, Wang S, et al. Deep reinforcement learning based adaptive energy management for plug-in hybrid electric vehicle with double deep Q-network. Energy, 2024, 305(1): 132402. https://doi.org/10.1016/j.energy.2024.132402

[27] Zhou Y, Ren Y, Xu M, et al. An improved NSGA-III algorithm based on deep Q-networks for cloud storage optimization of blockchain. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(5): 1406-1419. https://doi.org/10.1109/TPDS.2023.3243634

[28] Moghaddasi K, Masdari M. Blockchain-driven optimization of IoT in mobile edge computing environment with deep reinforcement learning and multi-criteria decision-making techniques. Cluster Computing, 2024, 27(4): 4385-4413. https://doi.org/10.1007/s10586-023-04195-4

[29] Yun J, Jiang D, Huang L, et al. Grasping detection of dual manipulators based on Markov decision process with neural network. Neural Networks, 2024, 169(1): 778-792. https://doi.org/10.1016/j.neunet.2023.09.016

[30] Wu G, Chen X, Gao Z, et al. Privacy-preserving offloading scheme in multi-access mobile edge computing based on MADRL. Journal of Parallel and Distributed Computing, 2024, 183(1): 104775. https://doi.org/10.1016/j.jpdc.2023.104775

[31] Yousiff S A, Muhajjar R A, Al-Zubaidie M H. Designing a blockchain approach to secure firefighting stations based Internet of things. Informatica, 2023, 47(10). https://doi.org/10.31449/inf.v47i10.5395

[32] Gadiparthi M, Reddy E S. Optimizing the Quality of Predicting the ill effects of Intensive Human Exposure to Social Networks using Ensemble Method. Informatica, 2022, 46(7). https://doi.org/10.31449/inf.v46i7.4212

[33] Sahakyan H, Katona G, Aslanyan L. Study on Using Reinforcement Learning for the Monotone Boolean Reconstruction. Informatica, 2025, 48(4). https://doi.org/10.31449/inf.v48i4.4804