A Dynamic Adaptive Framework for Practical Byzantine Fault Tolerance Consensus Protocol in the Internet of Things

Chunpei Li, Wangjie Qiu, Xianxian Li, Chen Liu, and Zhiming Zheng

Abstract—**The Practical Byzantine Fault Tolerance (PBFT) protocol-supported blockchain can provide decentralized security and trust mechanisms for the Internet of Things (IoT). However, the PBFT protocol is not specifically designed for IoT applications. Consequently, adapting PBFT to the dynamic changes of an IoT environment with incomplete information represents a challenge that urgently needs to be addressed. To this end, we introduce DA-PBFT, a PBFT dynamic adaptive framework based on a multi-agent architecture. DA-PBFT divides the dynamic adaptive process into two sub-processes: optimality-seeking and optimization decision-making. During the optimality-seeking process, a PBFT optimization model is constructed based on deep reinforcement learning. This model is designed to generate PBFT optimization strategies for consensus nodes. In the optimization decision-making process, a PBFT optimization decision consensus mechanism is constructed based on the Borda count method. This mechanism ensures consistency in PBFT optimization decisions within an environment characterized by incomplete information. Furthermore, we designed a dynamic adaptive incentive mechanism to explore the Nash equilibrium conditions and security aspects of DA-PBFT. The experimental results demonstrate that DA-PBFT is capable of achieving consistency in PBFT optimization decisions within an environment of incomplete information, thereby offering robust and efficient transaction throughput for IoT applications.**

Manuscript received 29 July 2023; revised 7 December 2023; accepted 9 March 2024. Date of publication 18 March 2024; date of current version 11 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20474, in part by Guangxi Science and Technology Major Project under Grant AA22068070, in part by Guangxi Natural Science Foundation under Grant 2020GXNSFAA297075, in part by Guangxi Science and Technology Project under Grant 2021AA11006 and Grant GuikeAD21220114, in part by Guangxi "Bagui Scholar" Teams for Innovation and Research Project, in part by Guangxi Talent Highland Project of Big Data Intelligence and Application, in part by the National Natural Science Foundation-Funded Project of China under Grant 62166004 and Grant 62062016, and in part by Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201. Recommended for acceptance by D. He. *(Corresponding authors: Wangjie Qiu; Xianxian Li.)*

Chunpei Li and Xianxian Li are with the College of Computer Science and Engineering, Guangxi Normal University, Guangxi 541001, China, and also with the Key Laboratory of Education Blockchain and Intelligent Technology, Ministry of Education, Guilin 541000, China (e-mail: licp@gxnu.edu.cn; lixx@gxnu.edu.cn).

Wangjie Qiu and Zhiming Zheng are with the Institute of Artificial Intelligence, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China, and also with Zhongguancun Laboratory, Beijing, Haidian 100086, China (e-mail: wangjieqiu@buaa.edu.cn; zzheng@pku.edu.cn).

Chen Liu is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with Zhongguancun Laboratory, Beijing, Haidian 100086, China (e-mail: liuchen@act.buaa.edu.cn).

Digital Object Identifier 10.1109/TC.2024.3377921

Index Terms—**Internet of Things, blockchain, practical byzantine fault tolerance, deep reinforcement learning, dynamic adaptive.**

I. Introduction

The Internet of Things (IoT) utilizes internet technologies to connect various information-sensing devices to the network, creating a fully interconnected ecosystem [1]. Today, IoT technologies have been extensively applied to enhance operational efficiency and several quality in domains such as smart healthcare, smart cities, and intelligent manufacturing [2]. Conventional IoT technologies mainly rely on centralized methods for data management and device monitoring, which laid the groundwork for the early development of IoT. However, with the geometric increase in device numbers, a surge in demands for cross-domain collaboration, the rise of edge computing, and the growing complexity of systems, the traditional centralized IoT architecture faces challenges such as insufficient scalability, single-point failures, high costs of establishing trust mechanisms, and data security vulnerabilities.

As a novel decentralized database technology, blockchain features immutability, multi-party collaboration, and global consistency [3], [4]. In recent years, blockchains have become broadly adopted by both the academic and industrial sectors to address the challenges faced by centralized IoT architectures [5], [6]. The distributed ledger technology of a blockchain can augment the scalability of IoT technologies and alleviate the risks associated with single-point failures. Furthermore, the security and integrity of data can be ensured through blockchain consensus protocols and encryption technologies [7], which provide a foundation for establishing a transparent and tamper-proof trust system for IoT. Considering the distinct value and profound implications that it presents in the IoT field, blockchain technology has been widely employed to bolster edge computing and data sharing in IoT environments [8].

However, because blockchain technology was not initially designed specifically for the IoT, its integration into IoT environments still presents numerous challenges. Among these challenges, the compatibility of the blockchain's consensus protocols with the IoT environment is particularly critical [9], [10], [11], [12]. Blockchain consensus protocols are primarily categorized as competitive (e.g., Proof of Work [13]), stake-based (e.g., Proof of Stake [14]), or voting-based (e.g., the

Practical Byzantine Fault Tolerance (PBFT) consensus protocol [15], [16]). Competitive consensus protocols demand significant amounts of resources and exhibit low transaction throughput, rendering them unsuitable for IoT applications that are resource-constrained and require high transaction rates. Stake-based consensus protocols necessitate the creation of an independent virtual cryptographic token system. This not only requires additional resources to maintain the token system but also poses potential legal risks. Although voting-based consensus protocols offer certain advantages in terms of resource consumption and higher transactional throughput, they are best suited for small-scale networks. These protocols are sensitive to the dynamic joining and departure of nodes [11] and are ill-equipped to handle IoT environments characterized by large-scale and pronounced dynamics [17].

In light of the predicaments and shortcomings that traditional blockchain consensus protocols encounter in the IoT domain, several research efforts have turned to deep reinforcement learning (DRL) techniques to optimize the PBFT consensus protocol and its supporting sharded blockchain systems [9], [17], [18], [19], [20], [21], [22], adapting them to the dynamic characteristics of IoT systems. Although recent studies have made breakthroughs in employing DRL to optimize the PBFT consensus protocol and its sharded system configurations, achieving consistent optimization decisions fairly in a decentralized environment with incomplete information remains an inadequately addressed issue. Given the complexity, dynamics, and information asymmetry inherent to IoT environments, it is exceedingly challenging to construct a decision-making model that provides all participants with identical and complete information. Such scenarios may lead to participants proposing vastly different optimization suggestions, making it difficult to reach a consensus. Blockchain, being a decentralized system with multiple participants, mandates that any changes in its configuration obtain agreement and recognition from the majority of the consensus nodes. Thus, without a broad-based consensus, even the most advanced optimization proposals will fail to be successfully implemented, let alone achieve the anticipated results.

The process of achieving the dynamic adaptability of consensus protocols through DRL can be divided into two sub-processes: optimality-seeking and optimization decision-making. The optimality-seeking process entails the generation of optimization schemes (including various parameters and configurations) via DRL based on the current IoT state, whereas the decision-making process entails the participation of nodes in blockchain consensus for the optimization scheme. By analyzing existing studies, we found that the collective optimization decision-making process for PBFT is complex, multi-dimensional, and multi-optional. The dimensionality of decisions encompasses consensus committee elections, block size adjustments, block interval modifications, and other factors, with each dimension containing a variety of potential options [9], [17], [18], [19]. For instance, the PBFT consensus committee may have numerous combinations of consensus nodes (there are $\binom{100}{21}$ possible combinations for selecting 21 consensus nodes from a pool of 100 nodes), and the block size can be any integer between 1 and 8.

Faced with the uncertain state information of IoT environments and the intrinsic complexity of decision-making processes, traditional consensus methods have encountered difficulties in ensuring consistency and fairness. To achieve consensus in optimization decisions, Skychain [17] attempts to set error thresholds. However, this method is contingent on the completeness of the information. If the error threshold is improperly set or the required information is insufficient, the entire optimization decision process can stall. Similarly, a study by Zou et al. [22] on decision-making consistency in multi-agent systems demonstrated a significant reliance on the completeness of state information. Because this approach fails to fully anticipate and address cases of information sparsity, the entire optimization process may come to a standstill. Furthermore, prior studies on consensus optimization have not thoroughly considered the general preferences of the majority of nodes, resulting in a lack of fairness in the decision-making process.

To bridge the aforementioned research gap, we employed Markov game theory to model the PBFT consensus committee as a multi-agent system and designed a dynamic adaptive framework for PBFT consensus (DA-PBFT), aimed at supporting the dynamic adaptability of PBFT to the evolving landscape of IoT. DA-PBFT divides the PBFT dynamic adaptive process into two sub-stages: optimality seeking and optimization decision-making. In the former stage, a PBFT optimization model is constructed based on deep reinforcement learning and deployed across consensus nodes to generate optimization plans. In the latter stage, decision consistency is achieved through the PBFT consensus mechanism. Unlike existing methods, DA-PBFT not only finds the optimal configuration of consensus protocols through DRL but also ensures fair consensus on multi-dimensional, multi-option optimization decisions in decentralized environments with incomplete information. The primary contributions of DA-PBFT can be summarized as follows:

1) To fully account for the scalability and large-scale dynamics of IoT environments, we modeled the optimization problem of the PBFT consensus protocol in a dynamic IoT environment as a Markov decision process. Subsequently, we employed the Double Dueling Deep Q-Network (DQN) algorithm to develop a PBFT optimization model. This model enables consensus nodes to continuously propose rational PBFT optimization schemes in response to the dynamic changes in the IoT environment.
2) To achieve fair consistency within decentralized environments with incomplete information, we designed a PBFT optimization decision consensus mechanism based on the Borda count method from social choice theory. This mechanism ensures that PBFT optimization decisions are effectively aggregated from the individual level to the collective level. Moreover, we implemented a hash-based commitment scheme to mitigate strategic manipulations in multi-option decisions, in accordance with the Gibbard–Satterthwaite theorem [23].
3) To enhance the security and sustainability of DA-PBFT during its dynamic adaptation process, we designed a dynamic adaptive incentive mechanism. Within this framework, we delved into the Nash equilibrium

conditions of DA-PBFT to comprehensively analyze security during dynamic adaptation. The experimental results indicate that DA-PBFT is capable of effectively achieving consistency in PBFT optimization decisions in environments with incomplete information. This characteristic enables PBFT to dynamically adapt to the evolving landscape of IoT, thereby providing robust and efficient transaction processing capabilities.

The rest of this paper is organized as follows: Section II discusses related studies. Section III presents a high-level overview of the system model along with a workflow for DA-PBFT. Section IV describes the construction of the PBFT optimization model, and Section V presents the design of the PBFT consensus mechanism. Section VI introduces a dynamic adaptive incentive mechanism for PBFT and examines the Nash equilibrium conditions and security based on this mechanism. Section VII assesses the performance and consistency probability of DA-PBFT. Finally, Section VIII concludes the paper.

II. Related Work

A. Application of Blockchain Technology in the IoT

Blockchain technology has recently been employed to address issues faced by centralized IoT systems, such as limited scalability, single points of failure, and inadequate data security protection [5], [6]. For instance, Novo et al. [24] designed a decentralized blockchain access control system to address the centralized shortcomings of IoT environments, whereas Sushil et al. [25] proposed the BlockIoTIntelligence security architecture, which integrates blockchain with artificial intelligence. Liu et al. [8] proposed a blockchain data sharing scheme in the zero-trust environment of IoT to ensure privacy throughout the sharing process.

However, blockchain technology was not originally designed for IoT applications, and its consensus protocols face several challenges when implemented in IoT systems. Lao et al. [11] designed a PBFT consensus protocol based on crypto-spatial coordinates with the objective of providing a more robust resource pool of IoT nodes to enhance consensus performance. Li et al. [10] developed a multi-layered PBFT consensus mechanism to optimize communication efficiency in the PBFT protocol for IoT applications. Zhao et al. [12] proposed the PoEM consensus protocol, which reduces the energy consumption of blockchains in IoT by training machine learning models to improve consensus efficiency. Ai et al. [26] designed a consensus protocol that improves the fairness, security, and performance of IoT blockchains using pre-built block proofs to validate new blocks. Anagnostakis et al. [27] proposed a micro blockchain framework suitable for IoT applications based on Gödel's incompleteness theorem and the free energy principle. This framework utilizes Proof of Existence (PoE) verification to meet the low requirements of IoT devices. Nevertheless, existing methods still fall short of addressing the continuous changes inherent to IoT networks.

B. Application of DRL in Dynamic Adaptive Consensus Protocols

To address the challenges encountered by blockchain consensus protocols in adapting to the multi-dimensional dynamics of the IoT, some studies have leveraged deep reinforcement learning technologies to facilitate dynamic adaptability. Liu et al. [9] proposed a blockchain performance optimization framework utilizing DRL technology. The framework dynamically optimizes block producers, consensus mechanisms, and block sizes and intervals while considering scalability, decentralization, latency, and security. Liu et al. [19] introduced the RAFT+ scheme, employing DQN to address consistency issues among IoT devices, supporting consensus across multiple device types to maintain the strong consistency of the blockchain. Zou et al. [22] investigated the use of a consensus mechanism based on Proof of Communication (PoC) and multi-agent reinforcement learning (MARL) within the IoT context, proposing an algorithm to optimize the efficiency and fairness of blockchain consensus.

For blockchain sharding systems, Zhang et al. [17] proposed a sharding method based on multi-agent DRL to address the dynamic nature of blockchain networks. They aimed to establish a framework that evaluates blockchain sharding systems using performance and security as assessment metrics. Yun et al. [18] introduced a blockchain IoT PBFT sharding scheme based on the DQN (DQNSB), aiming to dynamically optimize blockchain sharding configurations using DRL techniques and thereby enhancing the throughput of the consensus protocol. Addressing the issues of cross-shard communication and collaborative computation in sharded blockchains for IoT applications, Yang et al. [20] proposed a DRL-based method to optimize the number of blockchain consensus shards and parameters, aiming to enhance the performance and scalability of sharded blockchains. Considering the dynamic assembly characteristics of IoT environments, Xi et al. [21] introduced a new dynamic blockchain sharding scheme called HMMDShard. By integrating a hidden Markov model, they enabled HMMDShard to adaptively update blockchain sharding, consequently reducing cross-shard transactions and enhancing blockchain performance.

For other application scenarios, Li et al. [28] proposed the Athena system, which utilizes Policy-Based Multi-Agent Deep Deterministic Policy Gradient (PB-MADDPG) technology to conduct heterogeneous parameter optimization and enhance the performance of Fabric. Wu et al. [29] designed the AdaChain framework, a dynamic and adaptive permissioned blockchain structure that employs reinforcement learning to adjust to workload changes and can safely and correctly switch to a more optimal architecture in real time. Gadiraju et al. [30] proposed the DRLPB scheme, which applies DRL to dynamically optimize the Prism blockchain parameters, aiming to enhance consensus performance. Goh et al. [31] introduced a secure trust-based delegated consensus blockchain approach utilizing D3P with prioritized experience replay, aimed at simultaneously addressing scalability and security issues within consensus protocols.

Although numerous studies have been conducted on the dynamic adaptation of blockchains supported by DRL, most have not accounted for consistency in optimization decisions within a decentralized environment. Only a few studies, such as [17], [22], have addressed the issue of decision consistency. The consensus decision-making method of Skychain [17] does not account for the influence of individual preferences on final

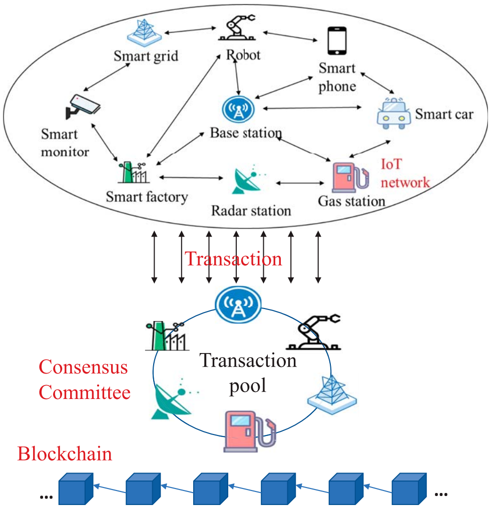


Fig. 1. IoT network supported by blockchain-DA-PBFT.

decisions. Moreover, it is sensitive to discrepancies in state information and may stall in cases of incomplete information. The approach proposed by Zou et al. [22] relies on the complete state information of the entire network to achieve consistency in multi-agent optimization decisions. In situations with incomplete information, this can lead to ongoing leader competitions or even result in a stalemate where a new round of leaders cannot be formed. In summary, existing methods cannot ensure the consistency of optimization decisions in fully informed, decentralized environments. This limitation hinders the effective deployment and execution of DRL models in blockchain-IoT applications.

III. System Overview

In the blockchain-IoT system, under the support of DA-PBFT, a vast array of heterogeneous IoT nodes is involved. These nodes encompass resource-constrained monitoring devices, smartphones, and mobile vehicles, as well as resource-rich infrastructures such as base stations, fueling stations, and factories, as depicted in Fig. 1. DA-PBFT prioritizes the selection of resource-abundant IoT nodes as consensus nodes, thereby offering a more stable and efficient overall consensus service.

DA-PBFT can be regarded as a series of PBFT instances connected in a temporal sequence. These instances can dynamically adapt and adjust various consensus parameters, including block size, block interval, and the composition of the consensus committee. To realize dynamic adaptability in PBFT, we adopted Markov game theory to model the consensus committee as a multi-agent system, wherein consensus nodes act as agent roles responsible for executing the blockchain consensus protocol and dynamically optimizing the PBFT consensus protocol based on the IoT environment. From a reinforcement learning perspective, DA-PBFT represents a multi-agent DRL framework that employs centralized training with decentralized decision-making.

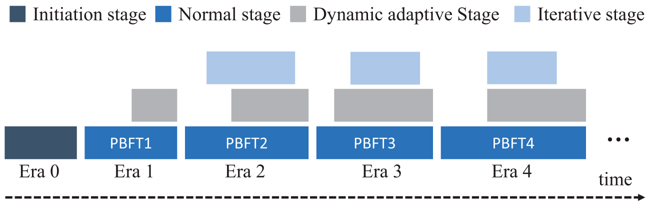


Fig. 2. DA-PBFT workflow.

A. Consensus Committee

Within the DA-PBFT framework, the consensus committee is a dynamic assembly, i.e., the consensus committee comprises different consensus nodes in different epochs. To fit the context, nodes that enter the consensus committee are referred to as intelligent consensus nodes (ICNs). Nodes that aim to join the consensus committee in the next epoch during the dynamic adaptation process are termed ICN candidate nodes. DA-PBFT models the PBFT consensus committee as a multi-agent system through Markov games [32]:

$$(n, S, A_1, A_2, ..., A_n, T, \mu, R_1, R_2, ..., R_n). \tag{1}$$

Here, n denotes the number of ICNs; S represents the state of the IoT environment, which includes block size, block interval, consensus committee, ICN candidate list, node failure rate, and other information; $A_1, A_2, ..., A_n$ are the actions of the ICNs; T is the state transition function; μ is the future reward discount factor; and R_i represents the reward acquired by the i^{th} ICN after executing a joint action.

B. DA-PBFT Workflow

As illustrated in Fig. 2, the workflow of DA-PBFT can be segmented into four phases: the initialization stage, the normal stage, the optimization stage, and the iteration stage. Fig. 2 showcases four PBFT instances with varied configurations: PBFT1, PBFT2, PBFT3, and PBFT4, each of which possesses different consensus committees, block sizes, and block intervals.

(1) Initialization Stage: The primary task of this stage is for the PBFT primary node to centrally train a PBFT optimization model using the DRL algorithm. Subsequently, the primary node shares this model with other replica nodes.

(2) Normal Stage: Following the initialization stage, all nodes that aim to join the consensus committee in the next epoch periodically send statements to the blockchain network regarding their geographical location, resource availability, and other relevant details.

(3) Dynamic Adaptive Stage: When the consensus configuration of the normal stage fails to adapt to the dynamic changes of the IoT environment, the dynamic adaptive stage is initiated. This stage is divided into two sub-stages: the optimality-seeking stage and the optimization decision-making stage. Specifically, the ICN of the current era takes the environmental state as input to generate an optimization plan through the PBFT optimization model. Finally, DA-PBFT ensures consistent decision-making through the PBFT optimization decision consensus mechanism.

(4) Iterative Stage: Following the previous stage, the primary node iteratively trains the PBFT optimization model based on the latest data and feedback. It then synchronizes the improved version with other replica nodes, continuously enhancing the accuracy and reliability of the optimization model.

C. Security Assumptions

DA-PBFT operates under the following security assumptions:

(1) Communication and Computation Limitations: Within DA-PBFT, malicious nodes are constrained by the following conditions: the delays they cause must not exceed the set timeout threshold; their computational power is limited, preventing them from cracking or manipulating consensus protocols and security measures that rely on cryptography.

(2) Determinism and Consistency Requirements: In DA-PBFT, all honest consensus nodes must operate under deterministic conditions. This means that, given the same state inputs and parametric conditions, the results of operations executed by all honest consensus nodes must be consistent. This also represents one of the important assumptions of the conventional PBFT protocol [15].

IV. Construction of PBFT Optimization Model

Initially, the PBFT optimization problem is transformed into a Markov decision process to construct the state space, action space, and reward function for PBFT optimization. Then, the primary node of PBFT employs the Double Dueling DQN algorithm [33], [34] to construct the PBFT optimization model. The model aims to produce a viable PBFT optimization scheme for the ICN in response to dynamic changes in the IoT environment. Unlike [9], the Markov decision process of DA-PBFT not only focuses on the scalability of the consensus committee but also establishes security constraints through the fault probability determination (FPD) model. This model is more suitable for dynamically changing large-scale IoT systems and applications [10].

A. State Space

The state space $\mathcal{S}$ of the PBFT optimization model consists of block size B^S, block interval T^I, ICN candidate node list L^C, and Byzantine fault tolerance rate P. This state space can be expressed as

$$\mathcal{S}^t = \left[B^S, T^I, L^C, P\right]^t. \tag{2}$$

where t denotes the epoch, and $|L^C| = N$. Throughout this study, we use $\mathcal{N}$ to indicate the number of nodes in the IoT network, N to represent the number of ICN candidate nodes, and n to denote the number of nodes in the consensus committee.

B. Action Space

The action space $\mathcal{A}$ of the PBFT optimization model consists of the PBFT consensus committee CC, block size B^S, and block interval T^I, which are critical factors influencing the transaction throughput of the PBFT consensus protocol. This action space is defined as

$$\mathcal{A} = \begin{pmatrix} CC \\ B^S \\ T^I \end{pmatrix}, \quad \text{where} \;\; CC = \begin{pmatrix} ICN_1 \\ ICN_2 \\ \vdots \\ ICN_n \end{pmatrix}. \tag{3}$$

Given a consensus committee $|CC| = n$ with $n \leq N$, the size of the committee is constrained to satisfy $n_{\min} \leq n \leq n_{\max}$, with $n_{\min}$ and $n_{\max}$ representing the minimum and maximum thresholds, respectively. The block size B^S falls within the set $\{1, 2, ..., \dot{B}^S\}$, and the block interval T^I falls within $\{0.5, 1, ..., \dot{T}^I\}$, where $\dot{B}^S$ and $\dot{T}^I$ denote the maximum values for the block size and interval, respectively.

C. Reward Function

The optimization objective of the PBFT optimization model is to maximize transactional throughput. However, this optimization process must satisfy constraints relating to decentralization, latency, and security. Therefore, the reward function of the PBFT optimization model can be expressed as

$$R^t(S, A) = \begin{cases} \dfrac{\left(\frac{B^S}{T^S} - \left(\frac{B^S}{T^S} \mod 1\right)\right)}{T^I}, & \\ \quad \text{s.t. } G(\boldsymbol{x}) \leq \beta, & \\ \quad T^I + T^D + T^V \leq \xi \cdot T^I, & \\ \quad 3N \cdot P + 1 \leq n. & \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where T^S denotes the size of a single transaction. If these constraints are not met, the reward (transaction throughput) is set to zero. The specific meaning of each constraint is described in the following paragraphs.

$G(\boldsymbol{x}) \leq \beta$ relates to the degree of decentralization in the geographical distribution of ICNs. To quantify the decentralization level of ICNs, we adopted the Gini coefficient for evaluation. The Gini coefficient is a widely used inequality measure in social sciences and economics, typically employed to evaluate the inequality of income or resource distribution [35]. Its value ranges within $[0, 1]$, with values closer to 0 indicating a more equal distribution. Assuming that ICNs are randomly distributed in a two-dimensional region $\mathbb{R}^2$, the Gini coefficient for the DA-PBFT consensus committee is defined as

$$G(\mathbf{x}) = \frac{1}{2n^2\bar{\mathbf{x}}} \sum_{i=1}^{n} \sum_{j=1}^{n} |\mathbf{x}_i - \mathbf{x}_j|. \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^2$ represents the longitude and latitude of the ICN, and n is the number of ICNs.

$T^I + T^D + T^V \leq \xi \cdot T^I$ states that a block must be validated within multiple consecutive block intervals. Here, T^V and T^D represent the validation and transmission times of a single block during the consensus process, respectively. These times relate to the number of nodes n and block size B^S. The specific expression is as follows:

$$T^V = \frac{1}{Y} \max_{\substack{1\le i\le n-1\\ 1\le p\le n}} \left\{ \begin{matrix} \frac{Y\gamma+[2Y+4(n-1)]\eta}{CR_{c_p}}, \\ \frac{Y\gamma+[Y+4(n-1)]\eta}{CR_{c_i}} \end{matrix} \right\}. \tag{6}$$

where Y is the batch size of PBFT (i.e., the number of blocks), γ represents the CPU cycles used to verify digital signatures, and η represents the CPU cycles used to generate and verify message authentication codes (MACs). CR_{c_p} and CR_{c_i} are the computational resources possessed by the primary and replica nodes, respectively.

$$\begin{aligned} T^D &= \frac{1}{Y}(t_1+t_2+t_3) \\ &= \frac{1}{Y}\left(\min\left\{ \max_{\substack{1\le i\le n-1\\ i\ne p, 1\le p\le n}} \frac{Y\cdot BS}{R_{c_p,c_i}}, TO \right\} \right. \\ &\quad + \min\left\{ \max_{\substack{1\le j\le n-1\\ j\ne i, 1\le i\le n\\ c_j\ne c_p}} \frac{Y\cdot BS}{R_{c_j,c_i}}, TO \right\} \\ &\quad \left. + \min\left\{ \max_{\substack{1\le i,j\le n\\ i\ne j}} \frac{Y\cdot BS}{R_{c_i,c_j}}, TO \right\} \right). \end{aligned} \tag{7}$$

Here, t_1, t_2, and t_3 represent the transmission times of a block in the pre-preparation, preparation, and commit phases of the PBFT protocol, respectively. TO represents the maximum delay tolerated by PBFT, and R_{c_i,c_j} indicates the data transfer rate between nodes c_i and c_j. T^V and T^D can be obtained by analyzing the native PBFT protocol. For the sake of brevity, we will not delve into the specific analytical process.

$3N\cdot P+1\le n$ pertains to the PBFT security constraint. This constraint is built upon the FPD model, making it more suitable for dynamic large-scale IoT systems and applications [10]. Specifically, this constraint assumes that the proportion of Byzantine faulty nodes among the ICN candidates is $N\cdot P$, and that any of these faulty nodes can become ICN. To ensure that the PBFT protocol can safely reach consensus in such an environment, it must satisfy $3N\cdot P+1\le n$.

D. Training Algorithm for PBFT Optimization Model

Following the construction of the state space, action space, and reward function, the PBFT primary node trains the objective function of the PBFT optimization model, which is an action-value function $Q_m(s_t, a_t)$, through the simulated dynamic changes of the IoT environment. This function can be expressed as

$$\begin{aligned} Q_m(s_t,a_t) = Q_m(s_t,a_t) + \varphi * (R_{t+1} + \mu Q_t(s_{t+1}, a_m) \\ - Q_m(s_t,a_t)). \end{aligned} \tag{8}$$

where $a_m = \underset{a}{\mathrm{argmax}} Q_m(s_{t+1}, a)$. Specifically, the PBFT primary node employs the Double Dueling DQN algorithm to train the objective function $Q_m(s_t,a_t)$ of the optimization model. A detailed training process of the PBFT optimization model is presented in Algorithm 1.

Algorithm 1 DA-PBFT Training Algorithm

```
Require:
    DO-PBFT policy $\pi$
    Evaluation network update frequency $\mathcal{F}_\theta$
    Target network update frequency $\mathcal{F}_{\theta^-}$
Initialize evaluation network of $\pi$ with $\theta$
Initialize target network of $\pi$ with $\theta^-$
Initialize experience buffer $buf \Leftarrow \emptyset$
Initialize timestep $n \Leftarrow 0$
for all around $r$ in $\{r_1, r_2, \cdots\}$ do
    $n \Leftarrow n+1$
    Observe state $s$ according to $\mathcal{S}^t = [B^S, T^I, L^C, P]^t$
    if $s^- \ne$ null then
        $buf \Leftarrow buf \cup \{(s^-, a^-, r^-, s)\}$
    end if
    with probability $1-\epsilon$ select
        $a \Leftarrow \arg\max_{a\in\mathcal{A}} Q^\pi(s,a)$
    otherwise
        Select random action $a$ from $\mathcal{A}$
    Execute action $a$ in environment
    Obtain reward $r$ according to $R^t(s,a)$
    if $n \mod \mathcal{F}_{\theta^-} = 0$ then
        Update $\theta^-$ by copying $\theta$
    end if
    if $n \mod \mathcal{F}_\theta = 0$ then
        Sample a mini-batch from $buf$
        Minimize $loss$ to update $\theta$
    end if
    $s^-, a^-, r^- \Leftarrow s, a, r$
end for
```

E. DA-PBFT Multi-Agent Architecture

Fig. 3 illustrates the centralized training and decentralized decision-making architecture of DA-PBFT multi-agent DRL. This architecture is designed to balance global optimization and flexibility in actual execution. During the centralized training phase, the PBFT primary node trains the parameters of the PBFT optimization model's objective function and subsequently validates the model in the simulated blockchain-IoT environment. Upon completion of training, the primary node shares the optimization model with all ICNs through blockchain transactions. Specifically, the primary node publishes the hash address of the PBFT optimization model on-chain, allowing other ICNs to download the corresponding model from a distributed storage system (such as IPFS) using the hash address. This mechanism ensures that all honest ICNs possess the same PBFT optimization model, thereby enabling the generation of relatively consistent PBFT optimization proposals when dynamic changes occur in the blockchain-IoT environment.

F. ICN Candidate Node Election Mechanism

Typically, nodes with fixed geographical locations have more abundant resources and are less likely to become malicious [11]. To this end, we designed an ICN candidate node election mechanism (NEM) based on crypto-spatial coordinates (CSCs)

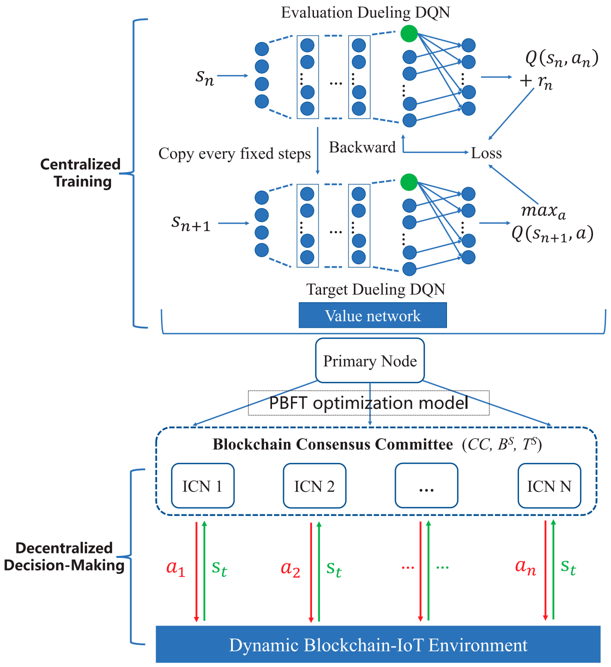


Fig. 3. DA-PBFT's centralized training and decentralized decision-making multi-agent architecture.

and Proof of Work (PoW), ensuring that only nodes with fixed geographic locations and abundant computational resources can become ICN candidates. CSCs [11] encourage nodes to honestly declare their geographic locations. Projects offering similar services include Foam Space[1] and the XYO Network[2]. Notably, although many consensus algorithms also adopt similar node election mechanisms, our proposed election mechanism is designed to not only screen for healthier ICN candidate nodes but also provide a list of ICN candidates for the PBFT optimization model and assist in evaluating the decentralization of the consensus committee. Additionally, this paper's ICN candidate node election mechanism records the list of candidate nodes through the blockchain's global ledger, which helps to alleviate the dilemma of incomplete information that PBFT faces in IoT environments.

Specifically, the ICN candidate node election mechanism requires IoT nodes that aim to join the DA-PBFT consensus committee to periodically declare their geographical locations via CSC. This declaration proceeds in the form of the following blockchain transaction:

$$Tr_{declare} = (CSC, CR, R, Sig, Timestamp). \tag{9}$$

where CR represents the computational resources of an ICN candidate node and R denotes the corresponding data transfer rate. To prevent malicious nodes from tampering with their own location information by mimicking the geographic positions of other nodes, we implemented cryptographic commitment technology to maintain the confidentiality of geographic information. The actual location of each node is revealed only when the dynamic adaptation process is initiated.

Furthermore, when the consensus protocol optimization trigger is activated, the ICN candidate node election mechanism also requires IoT nodes to complete the following PoW within a specified time:

$$\text{Proof} = H(t \,\|\, \text{IP} \,\|\, \text{CSC} \,\|\, \text{Nonce}) < D. \tag{10}$$

where t denotes the epoch, IP represents the node's IP address, Nonce is a random number, and D is the difficulty threshold for the PoW. The ICN candidate node election mechanism further elevates the entry threshold for ICN candidate nodes through PoW, thereby increasing the probability of healthier IoT nodes joining the consensus committee.

V. Detailed Design of PBFT Optimized Decision Consensus Mechanism

The DA-PBFT optimization decision process is complex, multi-dimensional, and multi-optional. The primary challenge associated with this process is to ensure a fair consensus within a decentralized environment with incomplete information. To address this challenge, we constructed an anti-strategic manipulation PBFT optimization decision consensus mechanism based on the Borda count method [36] from social choice theory, ensuring that every submitted optimization proposal is processed fairly and securely. The mechanism fully accounts for the strategic manipulation risks revealed by the Gibbard–Satterthwaite theorem [23], integrating a hash-based commitment scheme into the decision-making process, which prevents malicious ICNs from manipulating final decision outcomes based on the proposals submitted by other ICNs.

Unlike existing decision consensus methods [17], [22], the DA-PBFT consensus mechanism ensures consistent optimization decision-making through deterministic mathematical statistical calculations, delving deeper into the optimization preferences of the majority of consensus nodes. This not only realizes consensus decision consistency under complete information states but also achieves the same effect under incomplete information states. Although the consensus mechanism may occasionally lead to suboptimal decision outcomes, it effectively prevents deadlock situations that arise when optimization is needed (e.g., when the number of consensus nodes is significantly below the safety threshold). Such deadlocks can potentially compromise the system's availability and security.

The PBFT consensus mechanism mainly consists of two phases: the optimization decision proposal phase and the optimization decision consensus phase.

A. Optimization Decision Proposal Phase

When changes in environmental state trigger the conditions for dynamic adaptation, the current epoch's ICN i takes the state information of the blockchain global ledger as input for the PBFT optimization model, generating the corresponding

[1] https://foam.space/
[2] https://xyo.network/

optimization proposal a_i^{t+1}:

$$\begin{aligned} a_i^{t+1} &= (\mathrm{cc}, b^s, t^i)_i^{t+1} = \mathrm{PO}M_{\mathrm{No-\ Sto}}(S^t) \\ &= \mathrm{PO}M_{\mathrm{No-Sto}}\left(\left[b^s, t^i, l^c, p\right]^t\right). \end{aligned} \tag{11}$$

Here, t indicates the current epoch and $t+1$ denotes the subsequent epoch. $\mathrm{PO}M_{\mathrm{No-Sto}}$ represents the PBFT optimization model under a non-stochastic policy. This model selects actions based on the highest Q value in the current state S^t. The non-random nature of this selection can reduce uncertainty by providing consistent and predictable outputs, which is beneficial for achieving uniformity in optimization decisions.

After ICN i calculates and obtains the optimization proposal a_i^{t+1}, it must first prevent other ICNs from prematurely discerning said proposal by computing and pre-submitting a corresponding commitment value. Specifically, ICN i generates a random number r_i and concatenates it with a_i^{t+1}:

$$a_i^{t+1,r} = (\mathrm{cc}, b^s, t^i, r_i)_i^{t+1}. \tag{12}$$

Subsequently, ICN i utilizes a secure hash function $H()$ (such as SHA-256) to compute the encrypted commitment Commit_i:

$$\mathrm{Commit}_i = H\left(a_i^{t+1,r_i}, h_s, h_m\right). \tag{13}$$

h_s and h_m respectively represent the current state S^t used by ICN node i in generating the optimization scheme a_i^{t+1}, and the block height of the PBFT optimization model $\mathrm{PO}M_{\mathrm{No-Sto}}$. These will be used in the next section for quality assessment of the optimization scheme and reward distribution. Upon completing the encrypted commitment, ICN i submits the commitment value Commit_i to the blockchain's smart contract in the form of a transaction.

Once the smart contract receives proposal commitments from more than 2/3 of the ICNs, it will require all participating ICNs to disclose their commitments. In detail, this process involves each ICN i transmitting $(a_i^{t+1}, r_i, h_s, h_m)$ to the blockchain network. Subsequently, the smart contract executes the following commitment verification computations:

$$a_i^{t+1,r} = (cc, b^s, t^i, r)_i^{t+1} \tag{14}$$

$$Commit_i' = H\left(a_i^{t+1,r}, h_s, h_m\right). \tag{15}$$

Next, the smart contract compares the commitment value Commit_i' with the value Commit_i already stored in the blockchain ledger:

$$H'\left(a_i^{t+1,r}, h_s, h_m\right) \overset{?}{=} H\left(a_i^{t+1,r}, h_s, h_m\right). \tag{16}$$

If the two commitment values are equal, the optimization proposal of ICN i is considered valid. This hash-based commitment scheme ensures that malicious ICNs cannot preemptively discern the optimization proposals of other ICNs, thereby reducing the risk of manipulation. Moreover, to maintain the integrity and fairness of the PBFT optimization process, DA-PBFT encourages every ICN to participate in the proposal phase.

B. Optimization Decision Consensus Phase

Once the smart contract receives a sufficient number of optimization proposals, it assigns scores to the consensus nodes, block sizes, and block interval dimensions using the Borda count model. We note that the Borda count model requires each ICN to express a clear preference for all possible actions A, which may prevent the formation of a final optimization proposal. To address this issue, we adopted a partial ordering Borda count model that allows ICNs to rank subsets of options. The specifics are as follows:

(1) ICN Scoring: Assuming that $S_{\mathrm{IC}N_j^{t+1}}$ represents the score for the consensus candidate node $\mathrm{IC}N_j^{t+1}$ in epoch $t+1$, this score is defined as

$$S_{\mathrm{IC}N_j^{t+1}} = \sum_{i=1}^{N} p_{\mathrm{ij}}. \tag{17}$$

Here, p_{ij} is the score ICN i allocates to $\mathrm{IC}N_j^{t+1}$. If $\mathrm{IC}N_j^{t+1}$ ranks at position m, then $p_{\mathrm{ij}} = n - m$.

(2) Block Size Scoring: Similarly, the score $S_{B^S}^{t+1}$ for B^S can be defined as

$$S_{B^S}^{t+1} = \sum_{i=1}^{N} q_{i,B^S}. \tag{18}$$

Here, q_{i,B^S} is the score ICN i allocates to the block size B^S. If B^S ranks at position m, then $q_{i,B^S} = \dot{B}^S - m$.

(3) Block Interval Scoring: Similarly, the score $S_{T^I}^{t+1}$ for T^I can be defined as

$$S_{T^I}^{t+1} = \sum_{i=1}^{N} r_{i,T^I}. \tag{19}$$

Here, r_{i,T^I} is the score ICN i allocates to the block interval T^I. If T^I ranks at position m, then $r_{i,T^I} = \dot{T}^I - m$.

Integrating the aforementioned three dimensions, DA-PBFT defines a social welfare function W to represent the global optimality of the PBFT optimization decision consensus mechanism:

$$W = \omega_1 \times \max_{j\in[n]} S_{\mathrm{IC}N_j}^{t+1} + \omega_2 \times \max_{B^S} S_{B^S}^{t+1} + \omega_3 \times \max_{T^I} S_{T^I}^{t+1}. \tag{20}$$

Here, ω_1, ω_2, and ω_3 are weighting factors used to balance the significance of the three dimensions to satisfy $\omega_1 + \omega_2 + \omega_3 = 1$.

Finally, the Borda count smart contract model generates the optimized consensus committee for the current round based on the social welfare function W, as well as the block size and block interval configurations that the consensus committee will carry. Specifically, the Borda count model selects the top n candidates with the highest $S_{\mathrm{IC}N_j^{t+1}}$ as the consensus committee for the next era. Similarly, it selects the block size and interval with the highest $S_{B^S}^{t+1}$ and $S_{T^I}^{t+1}$, respectively, as the block size and interval for the next era. It is worth noting that, given the need for complex computations in the optimization decision-making process, it is advisable to consider the use of off-chain smart contracts [37] or off-chain computing mechanisms supported by oracles [38], in order to effectively avoid potential issues.

VI. Design of Incentive Mechanism and System Analysis

We designed a dynamic adaptive incentive mechanism, aiming to encourage ICNs to participate in the PBFT optimization decision process by honestly submitting high-quality optimization proposals. Subsequently, we explored the Nash equilibrium conditions and security of DA-PBFT.

A. Dynamic Adaptive Incentive Mechanism

In the proposal of optimization decisions, ICN i receives a base reward R_{base} when it submits the commitment value of its proposal and honestly reveals its optimization scheme as stipulated. If the optimization scheme a_i^{t+1} of ICN i aligns closely or exactly with the optimal solution derived from the social welfare function W, the node will receive an additional consistency reward R_{c}. Furthermore, if multiple ICNs propose optimization schemes identical to those of ICN i, ICN i will be granted a consensus reward based on R_{s}. On the other hand, if an ICN's proposal is deemed to be low-quality — e.g., it significantly deviates from the optimal solution or can potentially jeopardize the security of the decision-making process — it will incur a penalty, represented in the utility function by P_{q}. We note that although both R_{c} and R_{s} encourage ICNs to make consistent decisions beneficial for PBFT optimization, they do so from different perspectives.

Based on the aforementioned definitions, the utility function $U_i(a_i^{t+1}, a_{-i}^{t+1})$ for ICN i can be formulated as:

$$\begin{aligned} U_i(a_i^{t+1}, a_{-i}^{t+1}) = R_{\text{base}} &+ \alpha \times R_{\text{c}}(a_i^{t+1}, W) \\ &+ \theta \times R_{\text{s}}(a_i^{t+1}, A^{t+1}) - \lambda \times P_{\text{q}}(a_i^{t+1}). \end{aligned} \tag{21}$$

where a_{-i}^{t+1} represents the set of optimization proposals submitted by all ICNs except for ICN i, and A^{t+1} denotes the collection of optimization proposals submitted by all ICNs. α, θ, and λ are weight factors instrumental in influencing the behaviors of ICNs. $R_{\text{s}} = \frac{|j:a_j^{t+1}=a_i^{t+1}, j\neq i|}{n-1}$ represents the proportion of ICNs that proposed the same scheme as ICN i to the total number of ICNs. $P_{\text{q}} = \delta$ is a predefined positive number if a_i^{t+1} is a low-quality scheme, and zero otherwise.

In practical implementation, the dynamic adaptive incentive mechanism can verify the quality of the action a_i^{t+1} submitted by ICN i through the block heights h_s and h_m submitted during the optimization decision proposal stage. Here, h_s represents the block height of the current state S^t used by ICN i in generating the action a_i^{t+1}, and h_m indicates the block height of the PBFT optimization model PO$M_{\text{No-Sto}}$. More specifically, assuming the state corresponding to h_s is S^{h_s} and the PBFT optimization model corresponding to h_m is PO$M_{\text{No-Sto}}^{h^m}$, the quality of a_i^{t+1} as a low-quality optimization scheme can be judged by the following computation:

$$\begin{aligned} a_{h_s}^{h_m} = (\text{cc}, b^s, t^i)_{h_s}^{h_m} &= \text{PO}M_{\text{No-Sto}}^{h_m}(S^{h_s}) \\ &= \text{PO}M_{\text{No-Sto}}^{h_m}\left(\left[b^s, t^i, l^c, p\right]^{h_s}\right) \end{aligned} \tag{22}$$

$$d(a_i^{t+1}, a_{h_s}^{h_m}) = \sqrt{w_{cc} \cdot \Delta_{cc}^2 + w_{b^s} \cdot \Delta_{b^s}^2 + w_{t^i} \cdot \Delta_{t^i}^2} \tag{23}$$

$$\Delta_{cc} = |cc_{h_s}^{h_m} \triangle cc_i^{t+1}| \tag{24}$$

$$\Delta_{b^s} = (\{b^s\}_{h_s}^{h_m} - \{b^s\}_i^{t+1}) \tag{25}$$

$$\Delta_{t^i} = (\{t^i\}_{h_s}^{h_m} - \{t^i\}_i^{t+1}) \tag{26}$$

Where w_{cc}, w_{b^s}, w_{t^i} represent the weights of the consensus committee, block size, and block interval respectively in determining the quality of the optimization scheme. $|cc_{h_s}^{h_m} \triangle cc_i^{t+1}|$ denotes the number of elements in the symmetric difference of the two consensus committee member lists. A larger value of $d(a_i^{t+1}, a_{h_s}^{h_m})$ indicates a lower quality of the optimization scheme, and vice versa. Additionally, the dynamic incentive mechanism sets a consistency reward R_c and consensus reward R_s, reinforcing ICN i to generate optimization schemes through the latest state and PBFT optimization model.

B. Nash Equilibrium Analysis

Given the definition of the utility function $U_i(a_i^{t+1}, a_{-i}^{t+1})$, to ensure the game of PBFT optimization decision-making is in Nash equilibrium, the following inequality must hold for each ICN:

$$U_i(a_i^{t+1}, a_{-i}^{t+1}) \geq U_i(a_i'^{t+1}, a_{-i}^{t+1}) \quad \forall a_i'^{t+1} \in \{0,1\}, \forall i. \tag{27}$$

For the sake of discussion, we assume the globally optimal solution is $a^* = 1$. Thus, $U_i(1, a_{-i})$ and $U_i(0, a_{-i})$ can be represented as

$$\begin{aligned} U_i(1, a_{-i}^{t+1}) = R_{\text{base}} &+ \alpha \times R_{\text{c}}(1, W) \\ &+ \theta \times R_{\text{s}}(1, A^{t+1}) - \lambda \times P_{\text{q}}(1). \end{aligned} \tag{28}$$

$$\begin{aligned} U_i(0, a_{-i}^{t+1}) = R_{\text{base}} &+ \alpha \times R_{\text{c}}(0, W) \\ &+ \theta \times R_{\text{s}}(0, A^{t+1}) - \lambda \times P_{\text{q}}(0). \end{aligned} \tag{29}$$

To ensure $U_i(1, a_{-i}) > U_i(0, a_{-i})$, the following inequality must be satisfied:

$$\begin{aligned} \alpha \times R_{\text{c}}\,(1, W) &> \lambda \times P_{\text{q}}(0) \\ &- \theta \times (R_{\text{s}}(1, A^{t+1}) - R_{\text{s}}(0, A^{t+1})). \end{aligned} \tag{30}$$

By appropriately selecting α, θ, λ, and δ, we can ensure that the above inequality holds, thereby ensuring that the PBFT optimization decision-making process is in a Nash equilibrium state.

Consequently, the selection of appropriate parameters α, θ, λ, and δ is crucial and must be determined situationally. A potential Nash equilibrium state exists when all four parameters are positive, ensuring the effectiveness of the incentive mechanism. This is further constrained by the following inequality:

$$\alpha > \lambda \times \delta - \theta \times \left(\frac{|\{j : a_j = 1, j \neq i\}|}{n-1} - \frac{|\{j : a_j = 0, j \neq i\}|}{n-1}\right). \tag{31}$$

This inequality can be further simplified to

$$\alpha > \lambda \times \delta - \theta \times \left(\frac{n_1 - n_0}{n-1}\right). \tag{32}$$

Where $n_1 = |\{j : a_j = 1, j \neq i\}|$ and $n_0 = |\{j : a_j = 0, j \neq i\}|$.

To realize the above Nash equilibrium, we present more specific recommendations for the selection of weight parameters α, θ, λ, and penalty coefficient δ:

1) Determining λ and δ: These parameters directly relate to the quality of the proposal. Generally, we want $\lambda \times \delta$ to be sufficiently large to effectively penalize nodes that submit low-quality proposals.
2) Choosing θ: This parameter relates to the degree of consensus. In design, it is prudent to sufficiently reduce θ to prevent the system from excessively relying on group consensus, thereby avoiding potential "herd behavior."
3) Adjusting α: This is the most crucial parameter and directly relates to the consistency reward. In theory, it suffices that $\alpha > \lambda \times \delta$. However, in practical applications, α must be set to a significantly high value to ensure that the Nash equilibrium conditions are met, even when there are notable differences between n_1 and n_0.

C. Security Analysis

We analyzed the security of DA-PBFT based upon the Nash equilibrium.

(1) Consistency

Under Nash equilibrium conditions, ICNs have a strong incentive to maintain consistency in their optimization decisions. Given the existence of a potentially malicious ICN, the utility function for its dishonest behavior can be expressed as

$$U_i(a_{\text{dishonest}}, a_{-i}^*) = R_{\text{base}} - \lambda \times P_{\text{q}}(a_{\text{dishonest}}). \tag{33}$$

where $P_{\text{q}}(a_{\text{dishonest}}) = \delta$ signifies that the malicious behavior will invoke a quality penalty δ, resulting in a significant reduction in utility. This mechanism ensures that system consistency is maintained even in the face of dishonest actions, as the incentive for any single ICN to deviate from the consensus becomes negligible due to the potential significant loss.

(2) Robustness

Robustness refers to the ability to maintain normal operations when subjected to external attacks or internal failures. Under Nash equilibrium conditions, even if certain nodes are compromised or fail (denoted as a_{attack}), other nodes will maintain their optimal strategy. In other words, anomalous states cannot propagate through the network. Specifically, the number of ICNs under attack or failure f and the number of normally operating ICNs $n - f$ satisfy the following condition:

$$U_i(a_i^*, a_{-i}^*) \geq U_i(a_i^*, a_{-i}^* \cup a_{\text{attack}}), \text{s.t.} \quad 3N \cdot P + 1 \leq n. \tag{34}$$

Thus, even if a proportion $N \cdot P$ of the nodes are Byzantine, the system can still guarantee consistency and proper functioning as long as the number of normally operating ICNs n is at least $3N \cdot P + 1$.

(3) Predictability

In the state where DA-PBFT reaches Nash equilibrium, all participating ICNs jointly adopt an optimal strategy a^*. This consistent decision-making process results in high predictability of system behavior:

$$P(a_i^* = a_j^*) = 1, \quad \forall i, j. \tag{35}$$

Predictability enables more effective planning and response to the dynamic adaptive challenges of DA-PBFT, ensuring the stable operation and security of the network.

(4) Self-Organization

The self-organizing property of DA-PBFT in its Nash equilibrium state means that the decision-making process does not rely on a centralized coordination mechanism to maintain its security. That is, there exists no centralized strategy a_{central} that would allow the utility of any ICN to exceed that under the Nash equilibrium:

$$U_i(a_{\text{central}}, a_{-i}) \leq U_i(a_i^*, a_{-i}^*). \tag{36}$$

This unique advantage not only reflects the decentralized nature of the decision-making process, but also enhances its resilience and security when faced with potential attacks.

(5) Resistance to Strategic Manipulation

According to the Gibbard-Satterthwaite theorem, for any domain with two or more outcomes, any social choice function that satisfies the Pareto condition is either manipulable or dictatorial. The Borda count ensures that the optimization process of DA-PBFT is non-dictatorial; however, it may be susceptible to strategic manipulation. To counteract this risk, DA-PBFT requires each ICN i to produce an optimization proposal a_i^{t+1} during the decision-making process and subsequently generate a hash commitment $\text{Commit}_i = H(a_i^{t+1,r})$, where $H()$ is a cryptographically secure hash function and r is a random number. This commitment scheme possesses the following properties:

(i) Unforgeability: Due to the use of a cryptographically secure hash function, a malicious ICN cannot produce another commitment Commit_i' that matches Commit_i without knowing $a_i^{t+1,r}$.
(ii) Irreversibility and Concealment: The irreversibility and concealment properties of the hash function ensure that malicious ICNs cannot deduce the optimization proposals a_i^{t+1} of other ICNs through Commit_i.

Suppose there exists an adversary model $\mathcal{M}$ that controls a subset of malicious ICNs. In the cryptographic commitment scheme, $\mathcal{M}$ cannot access the a_i^{t+1} of other honest ICNs unless these ICNs open their commitments. This reduces the capability of $\mathcal{M}$ to manipulate the decision process, as $\mathcal{M}$ cannot know the optimization proposals of other ICNs in advance. Consider the following game $\mathcal{G}$:

- All ICNs (including those controlled by $\mathcal{M}$) submit their encrypted commitments.
- All ICNs decrypt and verify their commitments.
- Compute and determine the social welfare function W.

In this game, the goal of $\mathcal{M}$ may be to maximize the weight of the ICNs it controls in W. However, due to the hash-encrypted commitment scheme, $\mathcal{M}$ cannot effectively manipulate the submission contents of other ICNs in step 1. This implies that even if $\mathcal{M}$ attempts manipulation in step 3, its influence will be limited. In summary, DA-PBFT is capable of effectively mitigating strategic manipulation attacks.

TABLE I
SIMULATION PARAMETERS

Parameters	Symbol	Value
Number of IoT Nodes	K	100-1000
IoT Nodes Geographic Coverage Areas	$\mathbb{R}^2$	100km×100km
Number of ICN candidates	N	4-100
Consensus committee size (Number of ICNs)	n	4-100
Byzantine Failure Probability of ICN Nodes	P	$0, 0.05, 0.1, 0.15, 0.2$
Transaction size	T^S	200B
Block interval	T^I	0.5,1,1.5,...,10s
Blockchain size	B^S	0.2,0.4,...,8MB
ICN candidates computing resources for NEM	CR	10-30 GHz
ICN candidates computing resources for without NEM	CR	1-30 GHz
ICN candidates data transmission rate for NEM	R	10-100Mbps
ICN candidates data transmission rate for without NEM	R	1-100Mbps
Computing cost for verifying signature	α	2 MHz
Computing cost to generate and verify MAC	β	1 MHz
Batch size	Y	3
The number of intervals that a new block should be validated	ξ	6

VII. SIMULATION RESULTS AND DISCUSSION

We evaluated the transaction throughput of DA-PBFT along with its probability of achieving decision consistency under incomplete information.

A. Experimental Setup

In the throughput evaluation of DA-PBFT, we simulated a dynamic IoT environment where the number of IoT nodes varied between 100 and 1000. The size of the consensus committee was restricted to between 4 and 100 consensus nodes. Subsequently, we implemented the PBFT optimization model using PyTorch. Table I summarizes the primary performance parameters used in the evaluation process.

To fully demonstrate the dynamic adaptability of DA-PBFT, we established four baseline schemes:

1) DA-PBFT1: This scheme activates the ICN candidate selection mechanism, allowing the block size B^S to be dynamically adjusted. However, the block interval T^I is fixed at 2 s.
2) DA-PBFT2: This scheme activates the ICN candidate selection mechanism, allowing the block interval T^I to be dynamically adjusted. However, the block size B^S is fixed at 4 MB.
3) DA-PBFT3: This scheme does not activate the ICN candidate selection mechanism, but the block size B^S and block interval T^I can both be dynamically adjusted.
4) DA-PBFT4: This scheme activates the ICN candidate selection mechanism, allowing both the block size B^S and block interval T^I to be dynamically adjusted.

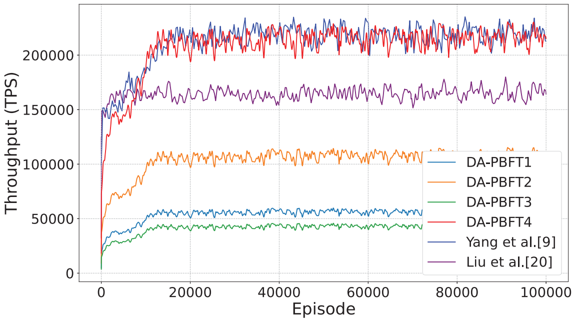


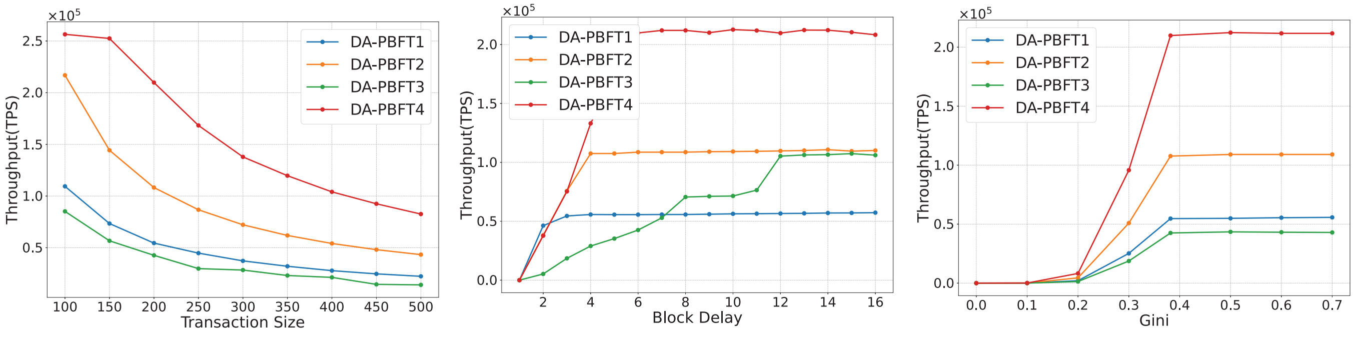
Fig. 4. Throughput performance analysis.

B. Performance Evaluation

Fig. 4 illustrates the throughput performance and convergence trend of the four benchmark schemes of DA-PBFT. In terms of throughput performance, DA-PBFT4 notably exhibits superior results. Because DA-PBFT1 and DA-PBFT2 have fixed block intervals or sizes, they cannot fully adapt to the dynamic changes in the IoT network. Although DA-PBFT3 can dynamically adjust both block size and block interval, it does not activate the ICN candidate node election mechanism. This may lead to a significant number of resource-limited IoT nodes in the consensus committee, triggering the Cannikin Law of PBFT[3]. In contrast, DA-PBFT4 activates the ICN candidate node election mechanism as well as the dynamic adjustment of block sizes and intervals, resulting in a higher blockchain transaction throughput.

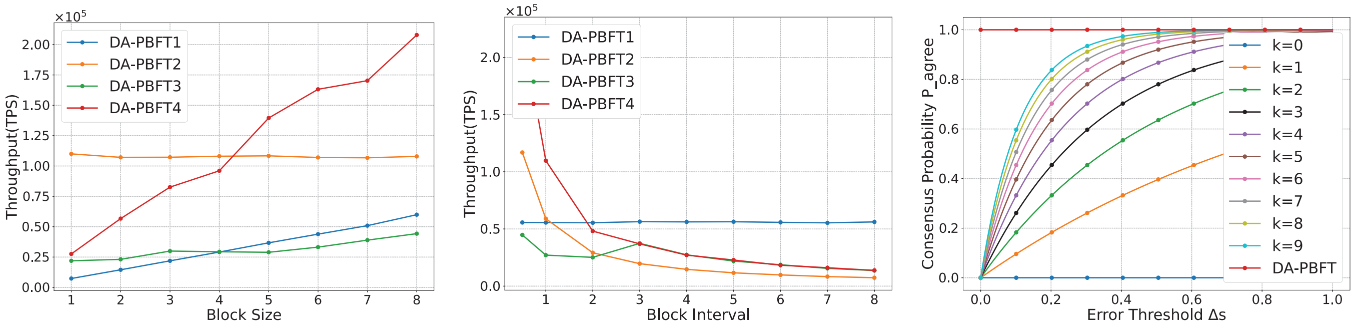
Compared to the dynamic adaptive PBFT scheme proposed by Liu et al. [9] (namely their non-switching consensus protocol scheme), the DA-PBFT scheme exhibits a more pronounced advantage in dynamic heterogeneous IoT environments. Firstly, DA-PBFT activates the ICN candidate node election mechanism, offering the consensus committee more resource-healthy nodes. More importantly, DA-PBFT flexibly adjusts the size of the blockchain consensus committee during the dynamic adaptation process, effectively enhancing the blockchain's adaptability in fluctuating environments. Thus, DA-PBFT not only responds to the changing demands of IoT environments but also augments the system's overall flexibility and efficiency. In the dynamic adaptive PBFT scheme proposed by Yang et al. [20] (specifically referring to their non-sharded method), the blockchain consensus network is reconstructed during the dynamic adaptation process using the K-means clustering method, thereby enhancing the throughput of blockchain transactions. Compared to their approach, DA-PBFT can achieve either a higher or equivalent blockchain transaction throughput. DA-PBFT maintains efficient transaction processing capabilities during the dynamic adaptation process by adjusting the size of the consensus committee without resorting to sharding. Additionally, it ensures a decentralized geographical distribution of members in the consensus committee, further bolstering the fairness and security of the blockchain system.

[3] The Cannikin Law of PBFT suggests that the throughput of transactions depends on resource-constrained IoT nodes, as there is a need to wait for these nodes' responses.



(a) (b) (c)

Fig. 5. Effects of different parameters on throughput.



(a) (b) (c)

Fig. 6. Effects of different parameters on throughput and decision consistency.

We conducted additional performance evaluation experiments to examine how dynamic changes in transaction size, block latency, degree of decentralization, block size, and block interval affect blockchain transaction throughput, centered around the four baseline schemes of DA-PBFT. Fig. 5(a) illustrates the variations in transaction throughput for the four baseline schemes as the size of an individual transaction changes. The experiments reveal that as the transaction size increases, the transaction throughput of the four baseline schemes gradually decreases. This is because altering the size of a single transaction impacts the number of transactions a block can accommodate, thereby affecting the overall transaction throughput of the system.

Fig. 5(b) depicts the variations in transaction throughput for the four baseline schemes of DA-PBFT as the block latency requirement ξ changes. The horizontal axis of Fig. 5(b) represents the magnitude of ξ, where a smaller ξ indicates a lower anticipated latency. As evident from Fig. 5(b), as ξ increases, the transaction throughput gradually grows, but an upper limit exists for each scheme. The trend in transaction throughput for DA-PBFT3 can be attributed to the absence of a consensus node selection mechanism, leading to an imbalanced resource distribution within its consensus committee. Consequently, in scenarios demanding low latency, DA-PBFT3 is more likely to trigger Cannikin's Law, thereby limiting transaction throughput. As ξ increases, the probability of DA-PBFT3 triggering Cannikin's Law diminishes, resulting in improved system performance. Fig. 5(c) presents variations in transaction throughput for the four baseline schemes under different Gini coefficients, where a smaller Gini coefficient indicates a higher geographical degree of decentralization among the DA-PBFT consensus committee nodes. In the unpredictable IoT environment with variable node behavior, we observe that all baseline schemes achieved the optimal and most stable performance for $gini \geq 0.382$. According to international standards, a Gini coefficient of 0.382 is typically used as a benchmark threshold for equality. Accordingly, we recommend setting the Gini coefficient to $gini \leq 0.382$ when deploying DA-PBFT.

Fig. 6(a) illustrates the variations in transaction throughput for the four baseline schemes with respect to block size. Because DA-PBFT2 utilizes a fixed block size, its throughput remains unaffected by block size variations. Notably, the throughputs of DA-PBFT3 and DA-PBFT1 intersect when $B^S = 4$, primarily because DA-PBFT3 incorporates 50% of resource-constrained nodes. As the block size escalates to 4M, DA-PBFT3 manifests deficiencies in validating and transmitting blocks, thereby triggering Cannikin's Law. Fig. 6(b) showcases the trends in transaction throughput for the four baseline schemes with respect to block interval. Because DA-PBFT1 is characterized by a fixed block interval, its throughput remains unaffected by block interval variations. Notably, as the block interval increases, DA-PBFT3 surpasses DA-PBFT2 and gradually approaches the throughput curve of DA-PBFT4. This is because with the extension of the block interval, the consensus process becomes more tolerant to higher latencies, reducing the impact of Cannikin's Law on DA-PBFT3. When $T^I > 3$, the variation curves of DA-PBFT3 and DA-PBFT4 nearly overlap, indicating that DA-PBFT3 is no longer influenced by Cannikin's Law.

C. Consistency Probability Evaluation

In the final experimental phase, we demonstrated the superiority of DA-PBFT in achieving consistent optimization decisions by assuming the following scenario: all nodes participating in the optimization proposal are honest. However, due to the complexity, dynamics, and information asymmetry of the IoT, these nodes may struggle to construct entirely identical states, making it challenging to produce consistent optimization schemes.

Existing optimization decision schemes either demand the construction of a global state with complete information (e.g., [22] and blockchain platforms such as Hyperledger Fabric) or achieve decision consistency by setting an error threshold Δ_s (e.g., [17]). Specifically, this error threshold Δ_s defines the extent of possible discrepancies between nodes regarding the system state. Consequently, the probability of consensus methods achieving consistency is influenced by the error threshold Δ_s, which can be modeled by the following mathematical function:

$$P_{agree}(\Delta_s) = 1 - e^{-k\cdot\Delta_s}. \tag{37}$$

where k represents the sensitivity of decision consistency to state discrepancies. The above function indicates that as Δ_s increases, the probability of achieving optimization decision consistency also increases, but the rate of increase diminishes with the increment of Δ_s (i.e., there exists an upper limit). Based on the aforementioned formula, existing optimization decision methods [17], [22] may incur stalemates due to information inconsistencies or incompleteness, especially in scenarios with significant opinion divergence, as they demand widespread consensus.

Fig. 6(c) shows the evaluation of the probability of existing optimization decision methods achieving consistency by setting different values for k. Unlike existing optimization decision methods, DA-PBFT fully accounts for the optimization preferences of the majority of ICNs. It fairly realizes optimization decision consistency through the Borda count method of social choice theory, which aggregates individual decisions into collective decisions through explicit calculations. Therefore, even under conditions of highly incomplete information, DA-PBFT can provide a 100% probability of optimization decision consistency. Although DA-PBFT may produce sub-optimal outcomes during the decision optimization process in scenarios where consensus protocols must undergo optimization (e.g., when the number of members in the consensus committee falls below the safety threshold), even a sub-optimal solution suffices to guarantee the security and performance of the blockchain. Conversely, if the consensus protocol becomes deadlocked due to inconsistent information when urgent optimization is needed, this poses a threat to the security and stability of both the entire blockchain and IoT system. It is worth mentioning that the DA-PBFT method can also effectively execute optimization decisions in a complete information state, demonstrating its adaptability and robustness across different application scenarios. Thus, the optimization decision consensus method of DA-PBFT represents a more universal approach applicable for optimization decisions in both complete and incomplete information contexts.

VIII. Conclusion

To facilitate the adaptation of PBFT to dynamic changes in the IoT environment, we propose a dynamic adaptive framework for the PBFT protocol, namely DA-PBFT. First, DA-PBFT employs Markov game theory to represent the PBFT consensus committee as a multi-agent system and constructs a PBFT optimization model based on the Double Dueling DQN algorithm. This model is designed to generate optimization strategies for consensus nodes based on the IoT state. Next, we designed a PBFT optimization decision consensus mechanism that is resistant to strategic manipulation. This mechanism ensures consistent optimization decisions in environments with incomplete information. Furthermore, we developed an incentive mechanism to enhance the security of the dynamic adaptive process. The evaluation results indicate that DA-PBFT not only provides stable and efficient transaction throughput in dynamic IoT environments but also achieves consistency in PBFT optimization decisions in environments with incomplete information. However, DA-PBFT only supports the PBFT consensus protocol at present. In the future, we plan to extend DA-PBFT to accommodate other consensus protocols and sharded blockchain systems.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[2] Y. Cui, F. Liu, X. Jing, and J. Mu, "Integrating sensing and communications for ubiquitous IoT: Applications, trends, and challenges," *IEEE Netw.*, vol. 35, no. 5, pp. 158–167, Sep./Oct. 2021.

[3] D. Liu, C. Huang, J. Ni, X. Lin, and X. S. Shen, "Blockchain-cloud transparent data marketing: Consortium management and fairness," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3322–3335, Dec. 2022.

[4] C. Liu et al., "Extending on-chain trust to off-chain–trustworthy blockchain data collection using trusted execution environment (TEE)," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3268–3280, Dec. 2022.

[5] L. Da Xu, Y. Lu, and L. Li, "Embedding blockchain technology into IoT for security: A survey," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10452–10473, Jul. 2021.

[6] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based federated learning for securing Internet of Things: A comprehensive survey," *ACM Comput. Surveys*, vol. 55, no. 9, pp. 1–43, 2023.

[7] M. Xu, Z. Zou, Y. Cheng, Q. Hu, D. Yu, and X. Cheng, "SPDL: A blockchain-enabled secure and privacy-preserving decentralized learning system," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 548–558, Feb. 2023.

[8] Y. Liu et al., "A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust Internet-of-Things," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 501–512, Feb. 2023.

[9] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial internet of things (IIoT) systems: A deep reinforcement learning approach," *IEEE Trans. Ind. Inform.*, vol. 15, no. 6, pp. 3559–3570, Jun. 2019.

[10] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1146–1160, 2020.

[11] L. Lao, X. Dai, B. Xiao, and S. Guo, "G-PBFT: A location-based and scalable consensus protocol for IoT-blockchain applications," in *Proc.*

IEEE Int. Parallel Distrib. Process. Symp. (IPDPS), Piscataway, NJ, USA: IEEE Press, 2020, pp. 664–673.

[12] Y. Zhao, Y. Qu, Y. Xiang, Y. Zhang, and L. Gao, "A lightweight model-based evolutionary consensus protocol in blockchain as a service for IoT," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2343–2358, Jul./Aug. 2023.

[13] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 3–16.

[14] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," Self-Published Paper, vol. 19, no. 1, 2012.

[15] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 7th Symp. Oper. Syst. Des. Implementations (OSDI)*, 1999, vol. 99, no. 1999, pp. 173–186.

[16] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable Byzantine consensus via hardware-assisted secret sharing," *IEEE Trans. Comput.*, vol. 68, no. 1, pp. 139–151, Jul. 2018.

[17] J. Zhang, Z. Hong, X. Qiu, Y. Zhan, S. Guo, and W. Chen, "Skychain: A deep reinforcement learning-empowered dynamic blockchain sharding system," in *Proc. 49th Int. Conf. Parallel Process.*, 2020, pp. 1–11.

[18] J. Yun, Y. Goh, and J.-M. Chung, "DQN-based optimization framework for secure sharded blockchain systems," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 708–722, Jan. 2021.

[19] Z. Liu, L. Hou, K. Zheng, Q. Zhou, and S. Mao, "A DQN-based consensus mechanism for blockchain in IoT networks," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11962–11973, Jul. 2022.

[20] Z. Yang, R. Yang, F. R. Yu, M. Li, Y. Zhang, and Y. Teng, "Sharded blockchain for collaborative computing in the Internet of Things: Combined of dynamic clustering and deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16494–16509, Sep. 2022.

[21] J. Xi et al., "A blockchain dynamic sharding scheme based on hidden Markov model in collaborative IoT," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14896–14907, Aug. 2023.

[22] Y. Zou, Z. Jin, Y. Zheng, D. Yu, and T. Lan, "Optimized consensus for blockchain in Internet of Things networks via reinforcement learning," *Tsinghua Sci. Technol.*, vol. 28, no. 6, pp. 1009–1022, 2023.

[23] A. Gibbard, "Manipulation of voting schemes: A general result," *Econometrica J. Econometric Soc.*, vol. 41, no. 4, pp. 587–601, 1973.

[24] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[25] S. K. Singh, S. Rathore, and J. H. Park, "BlockIoTIntelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence," *Future Gener. Comput. Syst.*, vol. 110, pp. 721–743, 2020.

[26] Z. Ai and W. Cui, "A proof-of-transactions blockchain consensus protocol for large-scale IoT," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 7931–7943, Jun. 2022.

[27] A. G. Anagnostakis, C. Naxakis, N. Giannakeas, M. G. Tsipouras, A. T. Tzallas, and E. Glavas, "Scalable consensus over finite capacities in multiagent IoT ecosystems," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6673–6688, Apr. 2023.

[28] M. Li et al., "Auto-tuning with reinforcement learning for permissioned blockchain systems," *Proc. VLDB Endowment*, vol. 16, no. 5, pp. 1000–1012, 2023.

[29] C. Wu, B. Mehta, M. J. Amiri, R. Marcus, and B. T. Loo, "AdaChain: A learned adaptive blockchain," 2022, *arXiv:2211.01580*.

[30] D. S. Gadiraju, V. Lalitha, and V. Aggarwal, "An optimization framework based on deep reinforcement learning approaches for prism blockchain," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2451–2461, Jul. 2023.

[31] Y. Goh, J. Yun, D. Jung, and J.-M. Chung, "Secure trust-based delegated consensus for blockchain frameworks using deep reinforcement learning," *IEEE Access*, vol. 10, pp. 118498–118511, 2022.

[32] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. Mach. Learn.,* New York, NY, USA: Elsevier, 1994, pp. 157–163.

[33] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2016, pp. 1995–2003.

[34] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1.

[35] C. Gini, "Measurement of inequality of incomes," *Econ. J.*, vol. 31, no. 121, pp. 124–125, 1921.

[36] I. McLean, "The Borda and Condorcet principles: Three medieval applications," *Social Choice Welfare*, vol. 7, no. 2, pp. 99–108, 1990.

[37] C. Li, B. Palanisamy, and R. Xu, "Scalable and privacy-preserving design of on/off-chain smart contracts," in *Proc. IEEE 35th Int. Conf. Data Eng. Workshops (ICDEW)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 7–12.

[38] Y. Lin et al., "A novel architecture combining oracle with decentralized learning for IIoT," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3774–3785, Mar. 2023.

Chunpei Li received the master's degree from Guangxi Normal University, in 2020. He is currently working toward the Ph.D. degree with Guangxi Normal University. He works as a Project Manager with Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. He is an open-source contributor to the blockchain platform-ChainMaker. His research interests include blockchain, the Internet of Things, and privacy computing.

Wangjie Qiu received the Ph.D. degree from Beihang University, China, in 2012. He is currently an Associate Research Fellow with Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. He is an essential open-source contributor to the blockchain platform-ChainMaker. His research interests include cryptography, blockchain, and privacy computing.

Xianxian Li received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2002. He worked as a Professor with Beihang University, from 2003 to 2010. He is currently a Professor with the School of Computer Science and Information Technology, Guangxi Normal University, Guilin, China. His research interest includes information security, blockchain, and privacy computing.

Chen Liu is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Beihang University, China. Her research interests include machine learning, cyber security, and data mining.

Zhiming Zheng received the Ph.D. degree in mathematics from the School of Mathematical Sciences, Peking University, Beijing, China, in 1987. He is currently a Professor with the Institute of Artificial Intelligence, Beihang University, Beijing, China. His research interests include refined intelligence, blockchain, and privacy computing. He is one of the initiators of Blockchain-ChainMaker. He is a Member of Chinese Academy of Sciences.