Received: 21 August 2022 | Revised: 22 May 2023 | Accepted: 27 May 2024

DOI: 10.1002/ett.5009

RESEARCH ARTICLE

WILEY

Cognitive blockchain and its application to optimize performance in blockchain systems

Reyhaneh Ameri | **Mohammad Reza Meybodi**

Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

Correspondence
Reyhaneh Ameri, Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.
Email: r.ameri@aut.ac.ir

Abstract

A distributed ledger called a blockchain is used for logging authenticated cryptographic transactions. The global ledger is updated with transactions using consensus techniques. Consensus algorithms are developed for networks with untrusted nodes to achieve reliability. Academics are paying attention to this technology because it has essential features like decentralization, stability, anonymity, and transparency. Even though the blockchain has some unique features, it has to deal with many challenges and restrictions, such as scalability, security, hidden centrality, and high cost. Artificial intelligence and blockchain are technologies that have been much discussed in the last decade and are developing rapidly. Combining the two to meet the existing challenges can have fascinating results. In this paper, we introduce the novel idea of cognitive blockchain by incorporating intelligent thought into the blockchain. With cognitive capabilities, blockchain technology can perceive the state of the network, analyze the data it has collected, make good decisions, and take appropriate action to improve network performance. We provide an operational framework for cognitive blockchain, which primarily refers to the connections between fundamental cognitive processes, including the perception-action cycle, data analytics, knowledge discovery, intelligent decision-making, and service provision. Then, using learning automata, we provide methods for creating cognitive engines for performance optimization by intelligently adjusting the block size, time interval, and validators in blockchain systems with BFT-based consensus algorithms. Several experiments have been conducted to assess the suggested approaches' effectiveness. The findings demonstrate that the proposed performance optimization approach improves blockchain performance criteria.

1 | INTRODUCTION

In the age of technology, data, and information play a leading role, and enormous amounts of data and information are produced daily. However, the proper storage, recording, and use of this information lead human beings to progress and comfort; emerging blockchain technology plays a significant role in helping them. Blockchain technology[1] promises to alter fundamentally the way people and businesses transact in digital assets and track ownership of such assets securely and independently of a centralized authority. Blockchain is a distributed, peer-to-peer ledger that is decentralized and preserves a permanent record of interactions and transactions among users who have access to the decentralized blockchain

 © 2024 John Wiley & Sons Ltd.

network.[2,3] The blockchain core is a distributed ledger consistent with high probability under specific conditions. Remarkably, the distributed collection of participants maintains consistency despite arbitrary behavior by possibly malicious players, often known as Byzantine failures.[4] Distributed ledgers include a variety of forms in addition to a chain of blocks, such as a directed acyclic graph (DAG).[5] Before adding a block to the chain, consensus techniques are used to verify each block. Blockchain offers novel opportunities for coordinating many dishonest parties and enabling decentralized governance in existing networks. The fundamental characteristics of blockchain are decentralization, transparency, immutability, security, auditability, anonymity, and autonomy.[6,7]

While blockchain technology offers much potential for building new Internet infrastructure,[7] it has several technical issues that limit its use, including scalability, throughput, environmental cost, security, and privacy leakage. Scalability is the main obstacle preventing blockchain from being used as a universal platform for many services and applications.[8] The current structural limitations of the blockchain network, such as the complexity of the consensus algorithm and the limited block size, make it difficult to process many transactions in seconds, which lowers throughput.[8] Bitcoin, the first blockchain-based cryptocurrency that introduced the Proof of Work as a consensus algorithm, has a maximum transaction rate of seven transactions per second (TPS).[9] In the meantime, Ethereum's throughput increases to around 14 TPS,[10] which still does not handle high-frequency transaction situations like IoT. Byzantine Fault Tolerant (BFT) algorithms have been employed to build consensus instead of PoW to satisfy the actual business needs, which improves throughput and reduces transaction confirmation delay. To gain consensus on a block using BFT-based techniques, a consensus committee must first be established and challenging to specify this committee in a public blockchain when the trustworthiness of nodes is unknown. Moreover, efficiency issues prevent using a network's entire node structure as the consensus committee. Due to limited nodes for consensus count and high communication complexity, this category of consensus algorithms still faces scalability challenges.[11] Also, most blockchains on the market, especially those that employ the proof-of-work consensus algorithm, use a lot of energy, which can be a problem for sustainability and the environment. While blockchain is considered secure due to its distributed nature, consensus, and cryptography, the technology is not immune to hacking or other cyber threats such as the 51% attack, double spending attack, and Sybil attack. Blockchain's public and private keys can help to protect some privacy issues.[9] Users conduct transactions using their public and private keys without revealing their true identities. However, it is demonstrated in References 12,13 that blockchain cannot support transactional privacy because all transaction values and balances for each public key are made available to the general public.

The approaches that have been proposed recently to improve performance and solve the blockchain scalability problem can be divided into on-chain and off-chain scaling methods.[14] Some solutions have been proposed in on-chain scaling categories, such as presenting novel consensus algorithms,[15,16] adjusting the interval and block size,[17] and applying the Sharding technique.[18–21] Any innovation outside of a blockchain that reduces the redundancy on the main blockchain, such as sidechains,[22] Lightning network,[23] and so forth, is known as off-chain scaling. However, these proposed solutions continue to face considerable difficulties when used in business applications and often prioritize transactional throughput while lowering other crucial performance indicators like decentralization, security, or latency. A blockchain system can only have two of the following three properties: decentralization, scalability, and security that this problem is known as the "Trilemma."

In recent years, Artificial Intelligence (AI)-based blockchain solutions have been proposed in several areas, including optimizing scalability and performance,[8,14,24–26] improving security,[11,27–33] and providing a prediction model on blockchain data.[34–38] AI can help in the optimal design of the blockchain mechanism by efficiently analyzing data and predicting system behaviors. Blockchain may be automated and optimized with AI to increase performance and improve governance.[34] The number of research studies that directly used artificial intelligence in the blockchain for optimal performance is limited. Also, most of the papers proposed in this field discussed optimization and scalability in a specific use case, with the solutions that the blockchain distributed environment's execution details are undefined. To solve the issues described, we provide an operational framework for blockchain that employs artificial intelligence to improve performance and solve its challenges. We integrate intelligent cognition with blockchain and propose the cognitive blockchain concept in this article. A cognitive blockchain combines cognitive and cooperative systems to improve performance and reach intelligence. We also presented the cognitive blockchain framework. The cognitive blockchain can sense current network conditions, process the knowledge it has gathered, effectively make decisions, and take adaptive steps to improve network performance. Multi-domain collaboration can enhance network capacity in the cognitive process, and machine learning can improve intelligence in the future.

Learning automata might be a suitable model for cognitive blockchain learning because of the unavailability of the correct answer to the problem, the unfamiliarity of the environment, and the dynamics of the network. A learning automaton

(LA) is an adaptive decision-making tool that learns the best action from a set of permissible actions through interaction with a random environment. In order to demonstrate the potential of the cognitive blockchain, we also provide methods based on learning automata for constructing cognitive engine performance optimization in blockchain systems using BFT-based consensus algorithms. Our proposed approaches are applicable to consortium blockchains and is not restricted to a single application. Furthermore, all the details for executing the proposed algorithms are described in the template of a cognitive blockchain architecture. These approaches select the block size, time interval, and validators to improve transactional throughput for blockchain with a BFT-based consensus protocol while maintaining the decentralization, security, and latency of the system. By carefully choosing a group for consensus based on learning about their reputation on the blockchain, it will be possible to increase the number of network nodes in this consensus protocol while decreasing the risk of centralization. The following is a list of the most significant contributions that this paper proposes:

- To the best of our knowledge, we are the first to introduce the cognitive blockchain concept and develop an operational cognitive blockchain framework.
- We design a new LA-based approach for developing cognitive engines to adjust the block size and time interval in the cognitive blockchain with BFT-based consensus. Also, we propose novel approaches based on GG, CGG, and LA to develop cognitive engines for select validators intelligently. The designed cognitive engines aim to enhance transactional throughput while preserving the blockchain's decentralization, security, and latency.
- We perform several blockchain simulator experiments to evaluate the effects of suggested cognitive engines on blockchain performance and demonstrate that employing the proposed cognitive engines improves blockchain performance criteria. Also, we give the node fairness measure as a novel metric for evaluating the fairness of blockchain systems.

The rest of this article is structured as follows. We provide some preliminary materials used in this paper and review the relevant literature in Section 2. We present a framework for cognitive blockchain in Section 3. Then, we propose approaches for designing cognitive engines in blockchain for performance optimization based on learning automata in Section 4. The simulation results used to assess the suggested approach are shown and discussed in Section 5. The paper is concluded in Section 6.

2 | BACKGROUND AND RELATED WORK

In this section, to give context for the remainder of the paper, we briefly overview Learning Automata, Goore Game, the Cellular Goore Game, BFT-Based consensus protocols, and performance measurements for blockchain systems. Then we describe some related works on performance optimization for blockchain systems.

2.1 | Learning automata

An adaptive decision-making system known as a learning automaton[39] can enhance its performance by learning how to select the best action from a list of possible actions through repeated interactions with the random environment. A learning automaton has a limited number of actions, and each action has an undetermined probability of being rewarded by its environment. Through numerous interactions with the system, the goal is to learn to select the action that has the most significant chance of rewarding oneself. The best action to take can be determined by the iterative process of interacting with the environment if the learning algorithm is appropriately selected.

Fixed and variable structure LAs are the two categories into which LAs can be divided.[39] Sextuple $\langle \beta \cdot \phi \cdot \alpha \cdot P \cdot G \cdot T \rangle$, is used in this paper to represent the variable structure LAs. β is a set of input actions (called a response or reinforcement signal), ϕ is a set of internal states, α is a set of outputs, P denotes the state probability vector, G is the output mapping, and T is learning algorithm. The probability vector is changed using the learning process. Based on the characteristics of β, environments might be divided into the P-, Q-, and S-models. The learning algorithm is used to modify the probability vector. A P-model environment's output consists of two components: success and failure. While β lies in the interval [0, 1] in S-model environments, it can take a finite number of values in Q-model environments.

$$p_i(t+1) = p_i(t) + a(1 - p_i(t))$$
$$p_j(t+1) = p_j(t) - ap_j(t) \cdot \forall j \neq i, \tag{1}$$

$$p_i(t+1) = (1-b)p_i(t)$$
$$p_j(t+1) = \frac{b}{r-1} + (1-b)p_j(t) \cdot \forall j \neq i. \tag{2}$$

Suppose α_i is the action selected at step t as an example realization from distribution $P(t)$. The equation for updating the probability vector $P(t)$ in a linear learning algorithm is defined by (1) for a positive response ($\beta = 1$) and (2) for a negative response ($\beta = 0$). Reward and penalty parameters are given by the two parameters a and b, respectively. The parameter a (b) controls how much the action probabilities increase (decrease). R represents the total number of possible actions the LA may do. The learning process described above is known as the linear reward penalty (L_{RP}) if $a = b$, the linear reward penalty (L_{RP}) if $a \gg b$, and the linear reward inaction (L_{RI}) method if $b = 0$. The linear reward-penalty ($SL_{R\text{-}P}$) scheme, the linear reward e-penalty (SL_{RP}) scheme, and the linear reward-inaction (SL_{RI}) scheme, respectively, are the linear reinforcement schemes for the S-three model's possible values of a: $a = b$, $a > b$, and $b = 0$.[40]

2.2 | The Goore Game

In this part, we will explain the Goore Game and the Goore Game with LA. Tsetlin[41] first presented it, and Narendra and Thathachar[42] and Thathachar and Arvind[43] also provided in-depth analyses. The Goore Game has fascinating characters that can be solved entirely distributedly without any player-to-player communication. It is a cooperative game where participants can choose between yes or no options. The referee assesses the players' votes. A unimodal performance evaluation function G is present in the referee. Each time, each player chooses one of their options, and the referee calculates the ratio λ, which is the proportion of players that voted "yes" to all players. The referee then distributes a dollar with probability $G(\lambda)$ and independently assigns a dollar with probability $1 - G(\lambda)$ to each player. The players then separately determine how to vote for the next iteration based on their profits and losses. The maximum of $G(\lambda)$ is associated with the number of players who will answer yes after enough iterations. In most circumstances, a player might not even be aware of the existence of other players or the total number of participants. Only the results of each player's choice of action need to be known.

Cooperative, mobile robotics,[44] and Quality of Service (QoS) management in wireless sensor networks[45] are two areas where the GG has found critical applications. Tsetlin used his so-called Tsetlin automaton to solve the Goore Game when it was initially studied. Later, more research was conducted in the LA region, and many families there were successful at solving the Goore Game. For instance, Thathachar and Arvind[43] suggested using Variable Structure Learning Automata (VSLA)[46] to solve the Goore Game. A Learning Automata (LA) that represents each player performs actions based on the player's strategies. Each LA updates its action probability using the LR-I learning algorithm.[47,48] displays the pseudo-code for using GG with learning automata.

2.3 | The Cellular Goore Game

The Cellular Goore Game (CGG), described in Reference 48, can be used as a model for systems with many simple, identical parts that interact locally. Every node (cell) in a CGG network can act as a referee while playing a Goore Game with its neighboring nodes. In CGG, each cell can act as a player and a referee simultaneously. CGG maximizes the objective functions in the referee nodes. Similar to GG, each player separately chooses their most appropriate action from two options based on the gains and losses they received from the nearby referees. In CGG, participants have no idea how other players behave or even how or why they are rewarded or penalized. The neighborhood structure required for specific applications is present in GGG, as GGG and CGG's differences. Additionally, it is possible to simultaneously maximize multiple criteria by including multiple referees at once in the suggested model.

In the proposed CGG, each player is represented by a learning automaton with two actions (YES and NO), which updates their action probability vector using the LR-I learning algorithm. A CGG with learning automata has been defined as a structure of the form (N, P, LA, R, G), where P is a subset of V acting as the players, and $N = (V, E)$ is an undirected network that specifies the structure of the CGG. Additionally, R is a subset of V acting as referees. Additionally, $G = \{G_1, G_2, \ldots G_n\}$is a collection of unimodal performance criteria for the referees. $LA = \{LA_1, LA_2, \ldots, LA_n\}$ is a set of learning

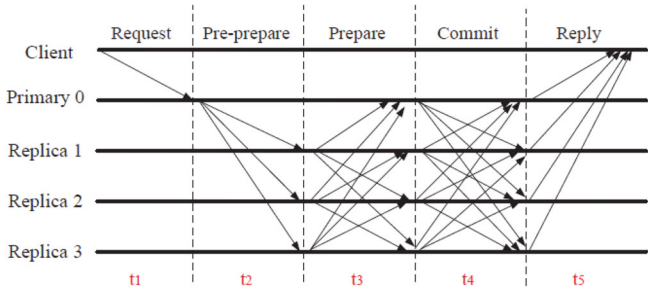


FIGURE 1 The workflow of PBFT consensus protocol.[14]

automata given to cells. For Quality-of-Service (QoS) control in WSNs when clusters overlap, one for clustered WSNs and one for WSNs with multiple sinks,[49] has employed the CGG model.

2.4 | BFT-based consensus protocols

Lamport et al.[50] introduced the BFT algorithm that declared an allegory for the problems of achieving consensus in a decentralized system. BFT-based consensus algorithms are the set of consensus algorithms designed to solve the problem of Byzantine fault-tolerant and reach consensus on data in an environment with the possibility of malicious nodes. A distributed network's ability to reach consensus even when some nodes fail to react or provide inaccurate information is known as Byzantine Fault Tolerance (BFT). A BFT mechanism's goal is to prevent system failures by using collective decision-making (both from correct and faulty nodes), which attempts to minimize the impact of the faulty nodes. PBFT, Tendermint, Quorum consensus algorithms are among these algorithms. These protocols also belong to the category of based voting-based consensus algorithms. One of the features of this type of algorithm is the fast verification of transactions. It is worth noting that one of their problems is to reduce operational efficiency and scalability by increasing the number of network nodes.

The Practical Byzantine Fault Tolerance (PBFT) system, which Barbara Liskov and Miguel Castro introduced in the late 1990s,[51] is the best implementation of the Byzantine Fault Tolerant system for asynchronous protocols. Its objective was to address a number of issues with byzantine fault tolerance systems that were already in use. The PBFT is used by Hyperledger Fabric[52] as its consensus algorithm since it can handle up to 1/3 malicious byzantine replicas. During a round, a new block is decided. A primary will be decided upon in each round by specific rules. The transaction must be ordered by it.[49] As indicated in Figure 1, the entire process may be broken down into three phases: pre-prepared, prepared, and commit. In each step, if a node has earned votes from more than 2/3 of all nodes, it will move to the next phase.[9] Therefore, for PBFT, every node must be known to the network.

2.5 | The performance measurements for evaluation of blockchain systems

We express quantitative metrics for the performance of blockchain systems in this section based on the four parts listed below:

- Scalability: Each block in a blockchain is made up of several transactions. The scalability of blockchain systems is assessed by transactional throughput, the number of transactions the system can handle in a given amount of time. The block size and interval directly impact the transactional throughput (measured in transactions per second, or TPS).
- Decentralization: Decentralization is the fragmentation of overall system control, which enables the system to achieve additional objectives, including immunity to specific attacks, open participation, and the removal of single points of failure. We measure the decentralized blockchain networks generated by equation using the Gini coefficient,[53] a metric for wealth or income inequality that has been extensively investigated (3).

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} | X_i - X_j |}{2n^2\overline{X}}. \tag{3}$$

X is a value that has been observed, n is the number of values that have been observed, and x bar is the mean value.

- Latency: Time to finality (TTF), which is the duration until a transaction is irreversibly confirmed and finalized, is used to evaluate the latency of blockchain systems.[24]
- Security: Blockchains should protect the ledger's immutability, demonstrated by its general resistance to attacks like the 51% attack, Sybil attack, distributed denial of service (DDoS), and so forth. Only a high likelihood of security can be provided by the PoW and PoS consensus algorithms used in first-generation blockchains. Theoretically, someone may utilize sufficient (>51%) mining power/stake to mine/mint a different, "longer" blockchain that returns to genesis.[35,39] However, under all network conditions, unambiguous finality can be achieved for BFT-type protocols as long as a portion of participants is truthful.[42] Therefore, for BFT-type consensus algorithms, the loyalty of the validators is crucial.

2.6 | The performance optimization for blockchain systems

The scalability issue, which prevents blockchain technology from being used in practical commercial applications, is one of the main problems with the technology.[54] Scalability's main objective is to enhance throughput while reducing block validation time.[8] The first investigations increase the block size[55] to address this problem. Although it can be somewhat beneficial, increasing block size also decreases the circulation of blocks in the network. The scalability of Bitcoin is examined by Croman et al.,[56] who calculate the ratio of block size to propagation time needed to reach a specific proportion of network nodes. Also, they apply the concept of "sharding" to blockchain networks in this paper, inspired by distributed databases and cloud infrastructures. Subsequent studies,[18–21,57] suggested using side chains or individual chains that work with blockchains; however, using different chains causes security problems.[57] A scalable Bitcoin NG protocol was suggested by Eyal et al.[21] The primary concept of this article is to decouple a block into two parts: the fundamental block for leader election and the macroblock to hold transactions, thereby incentivizing miners to compete for the position of leading the process of macroblock creation.

The vast scale and distributed nature of the Internet of Things (IoT) networks make security and privacy issues for IoT networks a persistent problem.[58–60] Blockchain can help in addressing major security requirements in IoT.[7,61] Traditional blockchain solutions, however, face difficulties in meeting the high transactional throughput requirements of the IoT. Liu et al.'s proposal[24] for deep reinforcement learning-based performance optimization in blockchain-enabled Internet of Vehicles maximizes transactional throughput while maintaining the underlying blockchain system's decentralization, latency, and security. This study also measures blockchain systems' scalability, decentralization, latency, and security performance. Additionally, Liu et al.[14] present a method for increasing the transactional throughput of the Industrial Internet of Things powered by blockchain by choosing the block producers and consensus algorithm (from the Quorum,[62] Zyzzyva,[63] and PBFT[64] consensus algorithms) and modifying the block size and block interval using the DRL technique. This strategy tries to increase the underlying blockchain's scalability without compromising the system's decentralization, latency, or security.

Lunardi et al.[65] introduce an appendable-block blockchain architecture that supports several consensus algorithms to make it suited for the IoT environment and shorten the time to reach consensus. It is suggested in Reference 66 to use a Proof-of-Quality-Factor (PoQF) consensus to shorten the time it takes to obtain consensus in vehicular ad hoc networks (VANETs). Liao and Cheng[67] present a Reputation and Voting based Consensus mechanism (RVC) that integrates a reputation calculation algorithm, assisted by edge servers. In RVC, a filtering algorithm is developed that can identify and remove nodes exhibiting malicious activities in order to ensure the security of the proposer selection procedure.

Because BFT-based methods provide the fastest way to consensus and do not require over-processing, blockchain transactions are valid as soon as they are added to the chain. Therefore, using BFT-based consensus algorithms can be a solution to the problem of blockchain scalability. Due to this type of consensus algorithm structure, all network members must be known. Operational problems emerge when the system's node count reaches a certain threshold value. This problem must be solved to use this type of consensus algorithm in a public blockchain. Most studies have proposed the formation of a subcommittee of all members to solve these problems.[68–70] A decision-theoretic online learning-based methodology for choosing a subset of nodes to take part in the consensus process in BFT-based consensus methods is proposed by Bogdi et al.[8] The suggested model successfully chooses the consensus committee's nodes with a higher reputation. The reputation value is determined for the nodes that want to participate in the consensus committee to decrease the probability that the nodes in the consensus committee will be damaged. Nodes with high reputation values are chosen

for the consensus committee. Response time, stock quantity, and response type have been used as features to measure the validity of nodes. For fully-connected wireless-networked systems, Jiang et al.[71] provide the Sybil-proof wireless network coordinate (WNC)-based Byzantine consensus (SENATE) protocol as a real-time distributed BFT protocol in which only chosen nodes to take part in the final consensus reaching process.

The Credit Reinforcement Byzantine Fault Tolerance Consensus (CRBFT) algorithm proposed in Reference 72 assigns the credit attribute to the node by utilizing the reinforcement learning that adaptively alters the node credit. The suggested method can automatically identify malicious nodes in the consensus network, enhancing consensus network security, decreasing consensus delay, and maximizing energy savings. Tang et al.[25] improve pbft algorithm by a trust equity scoring mechanism to dynamically adjust consortium chain consensus nodes, which is appropriate for high-frequency trading scenarios.[26] introduces consortium blockchain consensus technique, Weighted Byzantine Fault Tolerance (WBFT), enhances system performance and consensus delay. A dynamic weighting approach for consensus nodes decreases the influence of malicious nodes and increases blockchain system security.

Table 1 outlines the literature review on blockchain performance optimization. As seen in Table 1, most of the articles presented in this context focused on optimization and scalability in a specific use case, with the solution provided depending on the application. Furthermore, most proposed solutions have only focused on improving scalability or operational efficiency, and their approach may weaken other aspects of blockchain, such as decentralization and security. Also, proposed solutions to optimize blockchain performance have not addressed the details of how to implement their algorithm in a distributed blockchain environment. For example, in studies that select a committee of consensus nodes, it is not defined by whom the suggested algorithm will be executed. To address the mentioned problems, we introduce the cognitive blockchain concept and its operational architecture, which combines intelligent cognition with blockchain for the performance optimization of blockchain, in this paper. We design cognitive engines to maximize blockchain performance while maintaining decentralization, security, and latency in blockchains with BFT-based consensus algorithms by intelligently determining the modifiable parameters in the blockchain. Our approach applies to consortium blockchains and is not limited to a specific application. Also, all the details of how to execute it in a distributed environment are mentioned in the cognitive blockchain framework, described in the next sections.

3 | A FRAMEWORK FOR COGNITIVE BLOCKCHAIN

In recent years, the term "cognitive" has been used to describe a variety of networking and communications systems, including cognitive radio networks,[73–76] cognitive sensor networks,[77] cognitive wireless mesh networks,[78] cognitive mobile ad-hoc networks,[79] and cognitive peer-to-peer networks.[80] In this definition of cognition, Mitola[73] goes beyond the cognitive scientists' conception of simply adaptive behavior. The concept of a feedback loop, which describes how past interactions with the environment influence current and future interactions, is a recurring theme in these more complex conceptions of cognition.

To our knowledge, this article is the first to propose a new cognitive blockchain concept. A cognitive blockchain uses past decisions to create better decisions in the future, similar to a cognitive network.[81] Blockchains with cognitive processes that can learn from their actions are called cognitive blockchains. A cognitive blockchain is a blockchain in which a cognitive process is embedded to optimize performance, making intelligent decisions based on the state of the network and using the consequences of its actions to improve future decisions. The main goal of the cognitive blockchain is to improve the network's performance metrics. It can perceive the current network condition, assess network behavior and traits using sensed knowledge, and take appropriate actions in response to observations. The primary motivation behind developing the cognitive blockchain is to minimize human intervention by incorporating the cognition process.

Due to the distributed nature of the blockchain network, in the cognitive blockchain presented in this article, a cognitive engine set is embedded in the network nodes, which helps them make a series of intelligent decisions. The use of this cognitive engine by nodes can be optional. Therefore, according to Figure 2, the cognitive blockchain consists of several cognitive and simple nodes. A cognitive node is a node with the ability to optimize blockchain performance based on the situation at hand. Furthermore, compared to the cognitive node, a simple node lacks intelligence. Several approaches for implementing the cognitive blockchain regarding the distributed nature of blockchain are proposed below.

- Centralized approach: The cognitive engine set is embedded in one of the blockchain nodes in this approach. In other words, all the blockchain network nodes are simple nodes except one, and there is a cognitive node in the cognitive blockchain. Private blockchains can only use this approach.

TABLE 1 An overview of the literature on the performance optimization for blockchain systems.

Reference	Year	Use case	Purposes	Contributions	Limitations
56	2016	Bitcoin system	Investigating blockchain scalability scientifically, comprehending scalability bottlenecks, and developing more scalable blockchains	Measuring resource costs and performance of the Bitcoin network and studying reparametrization, providing a vast design space for scalable blockchains, and presenting open challenges for the development of more scalable blockchains	Lack of comprehensively assessing the influence of the ideas suggested on other features of the blockchain, such as decentralization; not fully implementing some provided proposals to increase the scalability of the Bitcoin network; and the potential security risks associated with the proposed methods
21	2016	Bitcoin system	Proposing a variation of the Bitcoin consensus method designed to improve scalability, throughput, and latency	Presenting the Bitcoin-NG scalable blockchain with the Bitcoin trust assumptions and greater throughput than Bitcoin, proposing quantitative metrics for assessing the Nakamoto consensus method, and evaluating the robustness and scalability of Bitcoin-NG	A major deviation from Bitcoin's working, security implications such as asset double-spending, the probability of consensus finality, and possibility of fork in the proposed blockchain protocol
24	2019	Internet of Vehicles	Presenting a method to evaluate the blockchain system from the aspects of scalability, decentralization, delay, and security and proposing a performance optimization framework based on deep reinforcement learning for the Internet of Vehicles equipped with blockchain	Quantifying the scalability, decentralization, latency, and security aspects of blockchain and providing a deep reinforcement learning method to select block producers, adjust block size and interval to adapt to the dynamics of Internet of Vehicle scenarios, and improve system performance	Lack of study of the suggested method's overhead on block confirmation time, lack of details about the proposed framework's execution mechanism in the blockchain distributed network (e.g., who executes the proposed approach), slow rate of proposed method convergence
14	2019	Industrial Internet of Things	Improve blockchain-enabled IIoT system scalability without negatively affecting system decentralization, latency, and security	Providing a framework for optimizing the performance of DRL-based industrial Internet of Things systems with the capability of dynamically selecting block producers, consensus algorithm, block size, and block interval	Not specifying the details of the executing the procedure of the proposed framework in the blockchain distributed network (e.g., who executes the framework?), lack of analysis of the overhead of the proposed method on block confirmation time
65	2019	Internet of Things	Providing a modular lightweight blockchain framework that can quickly insert new information and meet different IoT device consensus approaches	Introducing a formalization of a gateway-based IoT architecture-compatible modular lightweight blockchain, discussing the main security issues in blockchain and their effects on the proposed blockchain in an IoT scenario, and assessing the SpeedyChain using two different consensus algorithms in the IoT context	Not considering variables prevalent in actual network environments (such as node communication latency) and the mobility of nodes, and the vulnerability of the proposed approach against Eclipse attacks

(Continues)

TABLE 1 (Continued)

Reference	Year	Use case	Purposes	Contributions	Limitations
8	2019	Blockchain network with BFT-based consensus algorithm	Solving the fundamental challenge of selecting nodes to take part in the consensus process of BFT-based algorithms in public blockchains	Dynamic selection of a subset of nodes with a higher reputation to participate in the consensus process in BFT-based consensus methods using online learning	Vulnerable to security attacks such as Sybil attacks, not guaranteeing maximum throughput, and not considering the malicious node's incorrect response as a byzantine behavior of the nodes in the proposed method
66	2020	Vehicular Edge Computing network	Developing a consensus algorithm for vehicular networks that efficiently performs message validation and Quality Factor in multi-hop relaying on mobile edge nodes in a decentralized manner	Proposing a PoQF consensus, where mobile edge nodes serve as mining nodes, calculating message validation failure, latency, and block generation throughput of PoQF, and introducing an approach for incentive distribution to reward honest nodes and punish malicious mining nodes	Decreasing in throughput with the increasing of miner nodes and potential of the forks in the proposed consensus algorithm
71	2020	Wireless Networks	Developing a Sybil attack-resistant distributed BFT method for fully connected wireless networked systems without identity authentication	Introducing a solution for Sybil-proof BFT consensus with low delay and high throughput in permissionless systems for large-scale dense wireless networks	Vulnerability of the proposed method against consensus-related attacks such as eclipse attacks and lack of evaluation of the suggested method against security threats except for Sybil attacks
72	2021	Internet of Things	Proposing a BFT consensus algorithm based on reinforcement learning that adjusts node credit and automatically identifies malicious nodes	Presenting a Credit Reinforcement Byzantine Fault Tolerance (CRBFT) consensus algorithm based on the RL algorithm with improved consensus network security and reduced consensus delay	Lack of comprehensive evaluation of the proposed consensus algorithm in terms of security and decentralization, being susceptible to DoS attacks as a result of the proposed protocol's synchronicity, and considering only the authenticity of the response of the nodes in the reinforcement learning algorithm
25	2022	Consortium Blockchain	Developing a PBFT-based optimal consensus method for scaled alliance blockchain nodes and high-frequency transaction scenarios	Improving Byzantine fault tolerance algorithm by proposed trust equity scoring mechanism for high-frequency trading scenarios	Insufficient evaluation of the impact of the proposed pbft based algorithm on other aspects of the blockchain, such as decentralization and security, and inadequate information regarding the influencing factors used to determine credit scores
26	2022	Consortium Blockchain	Designing a consortium blockchain consensus algorithm that increases system throughput and reduces consensus delay	Introducing a consortium blockchain consensus algorithm with a weighting mechanism that weakens malicious nodes' influence and reduces malicious behavior	Weak study of the suggested algorithm's effects on blockchain aspects like decentralization and unable to satisfy the high concurrent processing needs of large-scale nodes in the community governance scenario

(Continues)

TABLE 1 (Continued)

Reference	Year	Use case	Purposes	Contributions	Limitations
67	2023	Edge Computing-enabled IoT systems	Providing a reputation- and voting-based consensus mechanism to improve consensus efficiency and security as nodes increase	Presenting a lightweight and safe blockchain consensus method with a malicious node filtering algorithm and a thorough reputation calculation algorithm that involves both the behavior of nodes in edge computing and the blockchain consensus process	Data storage limitations in the suggested consensus method as the number of nodes increases and not considering decentralization in selecting block proposers

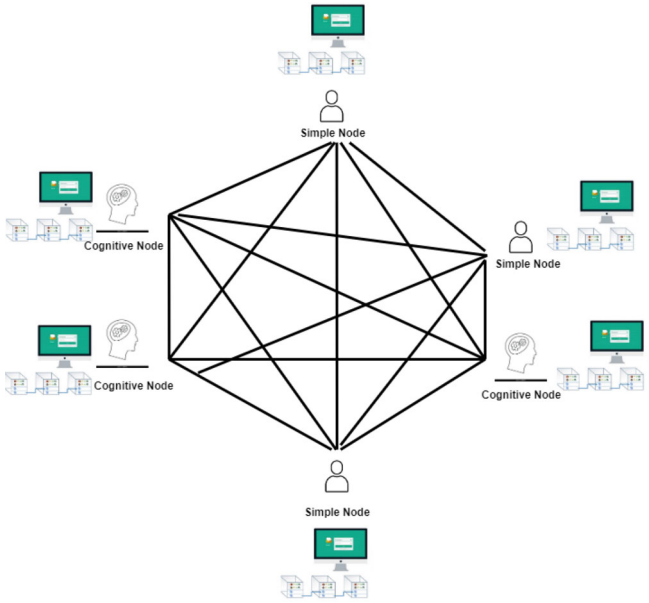


FIGURE 2 Cognitive blockchain network.

- Semi-centralized approach: The cognitive engine set is embedded in several blockchain nodes, and there are some cognitive nodes in the network in this approach.
- Fully-distributed approach: The cognitive engine set is embedded in all blockchain nodes, and all network nodes are cognitive node types in this approach. This approach is suitable for public blockchains.

The three layers of the Cognitive Blockchain framework—the Sensing Control Layer, Cognitive Process Layer, and Service Evaluation Layer—are depicted in Figure 3 and are further explained below. Note that some concepts from the cognitive Internet of Things framework introduced by References 82,83 inspire a series of concepts used in the proposed framework.

3.1 | Sensing control layer

The sensing control layer is a component of the blockchain network layer that is positioned in the data layer and directly interacts with the physical environment. In order to see its surroundings, each cognitive node uses sensors to collect local data about related information in the blockchain network. After providing the cognitive process layer with the necessary data, this layer performs operations on the system's manageable elements. Functions needed for cognitive node sensors and modifiable factors must be defined based on the goals of the cognitive blockchain in the design of the sensing control layer. For example, suppose the purpose of cognitive blockchain is to optimize transactional throughput and latency. Cognitive node sensors must be able to execute the proper functions to find local and global information

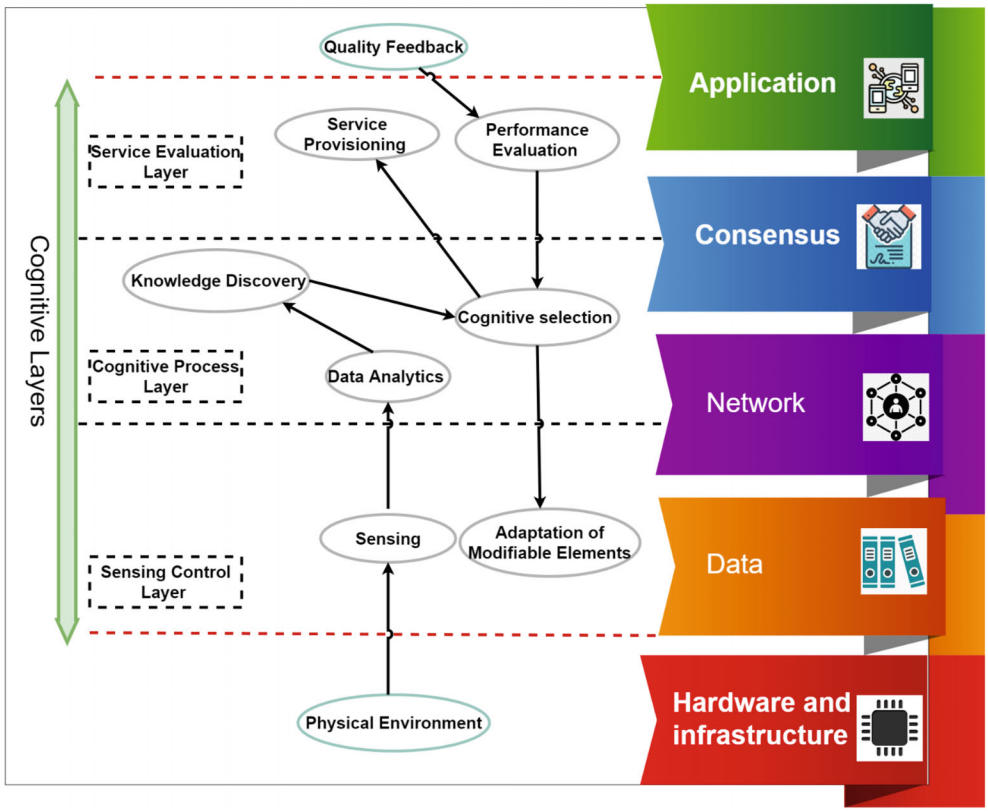


FIGURE 3 Framework of cognitive blockchain.

on the delays and transactional throughput in that situation. Based on the allowed blockchain adjustable elements that impact the cognitive blockchain's objectives, such as block size in various types of blockchains, the required functions for the modifiable elements are developed.

3.2 | Cognitive process layer

A cognitive engine set that resides in the cognitive nodes implements the cognitive process layer. This layer is located in the part of the blockchain network and consensus layers. It analyzes the sensing control layer data to discover valuable knowledge. Then, by employing this knowledge, this layer enables multiple interactive cognitive engines set to reason, plan, and select the proper action cognitively for service provisioning or adaptation of modifiable elements.

Data is typically noisy, distorted, heterogeneous, high-dimensional, and nonlinearly separable in blockchains where the number of nodes or the frequency of transactions to be confirmed is high. To fully utilize the value of the enormous data, it is necessary to create practical algorithms for extensive data analytics. Outlier analysis and clustering are valuable techniques in descriptive data mining that can be utilized as knowledge discovery techniques in this layer. Several decisions should be made during this layer's implementation, such as the appropriate methods for data analytics, knowledge discovery, and modifiable elements used in the cognitive blockchain's processes to fulfill the cognitive blockchain's goals. Additionally, it is essential to define the goal of the cognitive blockchain system, which determines which objective function should be improved.

3.3 | Service evaluation layer

The service evaluation layer provides services according to cognitive selection and is located in the part of the blockchain consensus and application layers. The services provided in this layer are specified based on the cognitive blockchain goal

defined in the cognitive process layer—for example, determining abnormal and suspicious transactions based on improving the blockchain network's security. Performance measurements are developed at this layer to assess the services provided and provide feedback on the results to the layers below. Assessing the blockchain services' performance is challenging since numerous considerations, such as the distribution and dynamics of the environment, are involved. These performance criteria differ according to the type of blockchain and the service the cognitive blockchain is supposed to provide.

4 | APPROACHES BASED ON LEARNING AUTOMATA FOR DESIGNING COGNITIVE ENGINES IN BLOCKCHAIN SYSTEMS WITH BFT-BASED CONSENSUS ALGORITHMS

BFT-based methods provide the fastest way to consensus and do not require over-processing. Also, unlike consensus algorithms such as PoW, the finalization of the transaction is definite. Due to the BFT structure, performance problems will occur when the number of system nodes surpasses a certain threshold. There is also a high probability of risk of centralization in these consensus methods. In this paper, learning-based automata algorithms are proposed for designing cognitive engines in the blockchain with BFT-based consensus to select the block size, time interval, and validators to optimize blockchain scalability and performance. Smartly selecting a group for consensus based on their credibility on the blockchain will reduce the risk of centralization in this consensus protocol. The proposed cognitive blockchain attempts to increase throughput while preserving the decentralization, finality, and security of the blockchain system. The proposed methods are suitable for consortium blockchains because our blockchain approaches give options for validator selection in blockchains with BFT-based consensus algorithms, and all consensus nodes of the network must be known.

The primary, or leader, node embedded the cognitive engine to select the block size, time interval, and validators intelligently. However, to prevent possible abuses and increase decentralization, the cognitive engine is also embedded in other network nodes, allowing others to evaluate the correctness of the primary's decisions. If the execution of this algorithm contradicts what the leader has done in practice, they can broadcast the change primary request to switch to a different primary node on the network. If the change of leader is confirmed by at least a specific number of network nodes, the faulty leader changes. So, the proposed cognitive blockchain uses a semi-centralized approach. Figure 4 demonstrates the PBFT protocol's communication pattern as the best example of the BFT-based protocols in the cognitive blockchain.

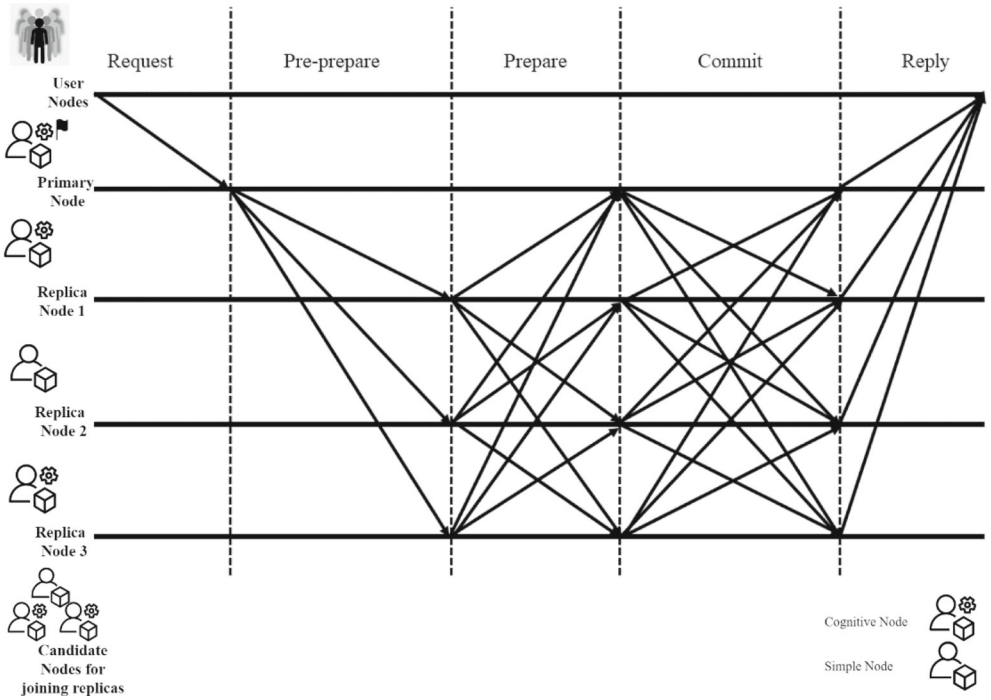


FIGURE 4 Protocol communication pattern of PBFT in cognitive blockchain.

The low overhead and simplicity of the learning automata models are the main reasons for choosing this learning model for knowledge discovery in the cognitive process layer. Also, due to the dynamic and uncertainty nature of the network, the use of learning automata is appropriate. Only the learning automata will need to be re-learned with the network change. After generating each block, the cognitive engine algorithm is not used to set the modified elements. Still, after producing several blocks, this process is performed because, by creating only one block, the impact of chosen actions on network performance cannot be appropriately evaluated. The time interval of algorithm execution can be varied over time. In the initial rounds, this interval is shorter. However, after a certain number of rounds, this interval can be increased in the case of environmental conditions with slight changes and convergence of learning models.

This section presents a new LA-based approach for developing cognitive engines to adjust the block size and time interval, named LA-SI-CB-BFT, in the cognitive blockchain with BFT-based consensus. Then the novel approaches based on GG, CGG, and LA are proposed to develop cognitive engines for select validators: GG-V-CB-BFT, CGG-V-CB-BFT, and LA-V-CB-BFT, respectively.

4.1 | An LA-based approach to adjusting the block size and time interval in the cognitive blockchain with BFT-based consensus (LA-SI-CB-BFT)

In this section, we provide an approach based on LA for developing the cognitive engine to adjust the block size and time interval in a blockchain with BFT-based consensus. The proposed cognitive engine's modifiable elements of cognitive processes are block size and time interval. Cognitive node sensors can gather information about the transactional throughput, the blockchain system's finality, and security, as described in Section 2.5.

A learning automaton to determine the ideal block size and a learning automaton to determine the block time interval are both included in the cognitive engine learning model. The reward function in this learning automata is also defined using the number of transactions processed per unit time and the security, and delay criteria. As the action spaces of these learning automata, block size s^B and block interval t^I should be predefined based on previous experiences and network constraints like node delay.

The block size and interval should be adapted to the dynamic environment to maximize the throughput. Equation (4) illustrates how the reward function is defined to enhance transactional throughput while ensuring the security and finality of the blockchain system. Be mindful that the blockchain system performs poorly in terms of latency or security if any condition is not met. In order to prevent an incorrect circumstance, we set the reward in this case to be 0.

$$\begin{cases} C_1 : \ T^F \leq w \times t^I \\ C_2 : \ f \leq \left\lfloor \frac{k-1}{3} \right\rfloor \end{cases}$$
$$A^t = \left[s^B, t^I\right]^t \tag{4}$$
$$R^t\left(A^t\right) = \begin{cases} \frac{\mathrm{TPS}(t)}{\max(\mathrm{TPS})} & \text{if } C_1 - C_2 \text{ are satisfied} \\ 0 & \text{otherwise.} \end{cases}$$

W is the threshold constant of latency, f is the number of malicious validators, k is the number of replicas, TPS(t) denotes the number of transactions confirmed per second, and max(TPS) is the maximum number of transactions verified per second in the blockchain. The general block diagram of the LA-SI-CB-BFT algorithm for BFT-based consensus with learning automata is given in Figure 5.

4.2 | Approaches for select validators in the cognitive blockchain with BFT-based consensus

Cognitive engines for select validators in the cognitive blockchain with BFT-based consensus are designated in this section. This cognitive engine provides a list of nodes selected as a replicas as a service resulting from the cognitive selection in the cognitive process layer. There is the ability to gather information about the transactional throughput, the blockchain system's decentralization, finality, and security, as described in Section 2.5, by cognitive node sensors. We introduce the fairness measure as a novel metric for assessing the fairness of blockchain systems, as defined in

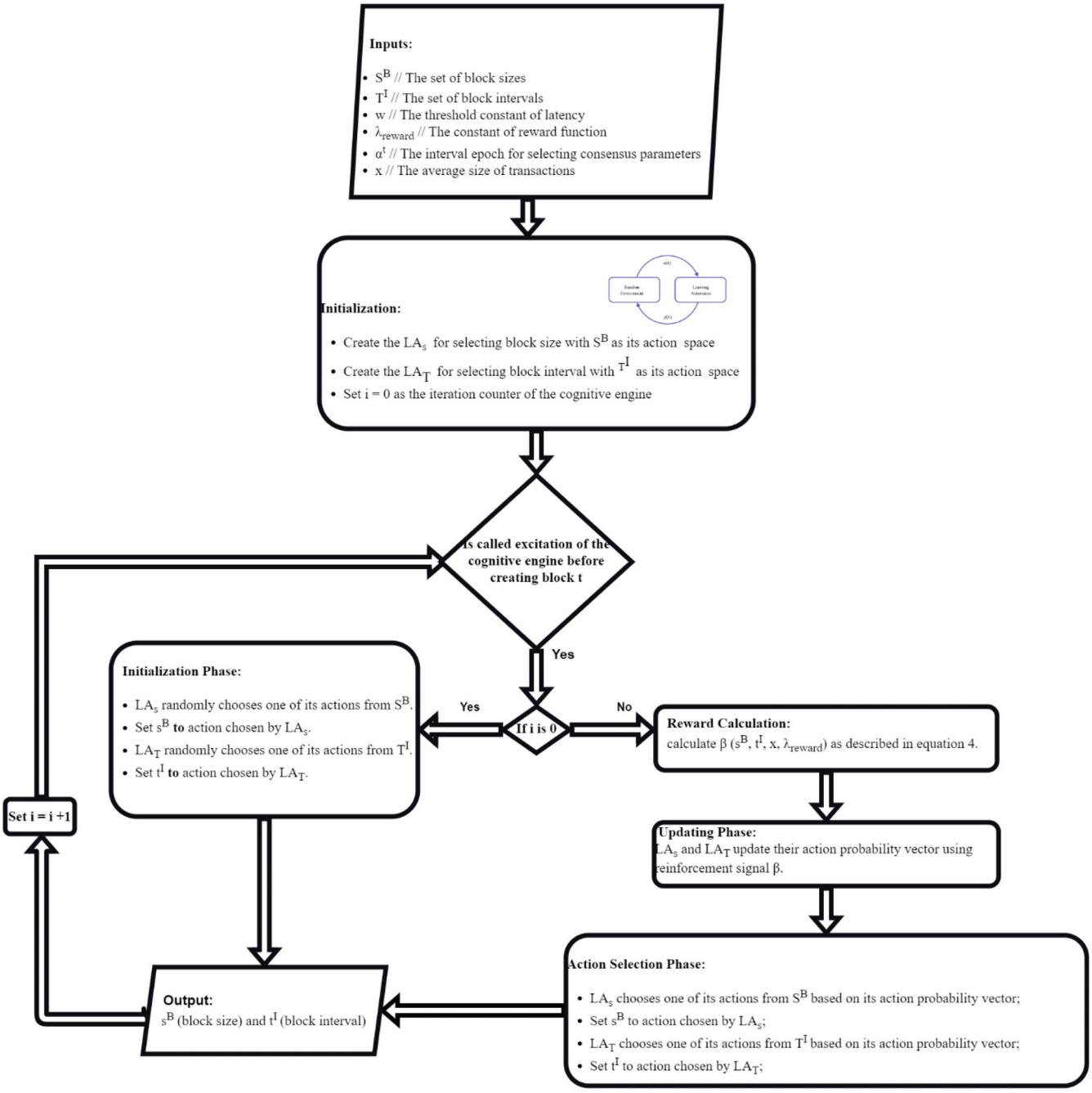


FIGURE 5 General block diagram of the LA-SI-CB-BFT algorithm.

Equation (5). The blockchain's fairness was studied in terms of the rewards mechanism and the selection mechanism.[84] The reward system outlines how rewards are split among nodes, while the selection mechanism selects which nodes will participate in the consensus on the next block to be added to the blockchain. We provide a formal definition of selection mechanism fairness. There should be a method for selecting validator members where the size of the candidate nodes for joining validators is less than the number of processes in the blockchain system. The system can be centralized and unfair by continually selecting the same nodes as validators.

$$\text{Fairness}\,(N,R)=\frac{\sqrt{\frac{\sum_{i=1}^{|N|}\left(\left(\sum_{t=1}^{T}I\{N_i\in R_t\}\right)-\frac{T\times|R|}{|N|}\right)^2}{|N|^2}}}{\frac{T\times|R|}{|N|}}. \tag{5}$$

R is the replicas set, N is the set of candidate nodes for joining replicas, T is the number of episodes, and I is an indicator function that shows the selection of the node as a validator in round t. Notably, the values of the fairness metric are greater than or equal to 0, with 0 denoting the most fairness in node selection; that is, each node has been selected $\frac{T\times|R|}{|N|}$ times as a validator in the T episode. In other words, if the fairness criterion is 0, all candidate consensus nodes have an equal opportunity to participate in the consensus process. As the proposed fairness metric increases, the chances of candidate nodes participating in the consensus process become increasingly unfair and unequal. We present the detailed descriptions of GG-V-CB-BFT, CGG-V-CB-BFT, and LA-V-CB-BFT as approaches for select validators in the following subsections.

4.2.1 | A GG-based approach to select validators in the cognitive blockchain (GG-V-CB-BFT)

The Goore Game model with LA is used as a learning model in this proposed cognitive engine that is embedded in every cognitive node for select validators. In the cognitive blockchain, each network candidate node for joining replicas indicates one of the player roles in the GG, with the ability to independently determine whether to vote yes or no in each trial. Candidate nodes can be in one of two states: selected or unselected as a replica node. Converge to "yes" means the candidate node has been selected, and "no" means the node has not been chosen. A referee assigns a reward probability based on the proportion of players who vote yes throughout each trial. The referee has a particular uni-modal performance criterion that is maximized at $K/|N|$ (the ratio of the number of replicas to total candidate nodes for joining replicas). Following is a general definition of a uni-modal performance criterion:

$$G(k_t) = c + ae^{-\left(\frac{(k^*-k_t)}{b}\right)^2}. \tag{6}$$

c, a, and b are learning parameters; k_t is the number of players with "selected" action in round t; and k^* is the ratio of the number of replicas to total candidate nodes/players. It is necessary to run the described GG model until it converges to the optimal value. Nodes will be selected as replicas whose equivalent player has converged to the "select" action. The reward function must be defined to achieve the purpose of cognitive blockchain and reflect the impact of the learning model's chosen nodes on the number of transactions processed per unit of time and the security, decentralization, and latency criteria. The received reward resulting from the execution of the cognitive engine affects the initial probability of choosing the action of GG players in the future. Equation (7) describes the reward function to increase transactional throughput while maintaining the security, decentralization, and finality of the blockchain system. If any criterion is not met, and the blockchain system has poor decentralization, latency, or security performance, we set the reward to 0.

$$\begin{cases} C_1 : G(\gamma) \le \eta_s, G(\lambda) \le \eta_l \\ C_2 : T^F \le w \times t^I \\ C_3 : f \le \left\lfloor \frac{k-1}{3} \right\rfloor \\ C_4 : \text{fairness}(N,R) \le \eta_f \end{cases} \tag{7}$$

$$A^t = [R]^t$$

$$R^t\left(A^t\right) = \begin{cases} \frac{\text{TPS}(t)}{\max(\text{TPS})} & \text{if } C_1 - C_4 \text{ are satisfied} \\ 0 & \text{otherwise.} \end{cases}$$

$G(\gamma)$, and$G(\lambda)$ are the block producers' stakes and locales' Gini coefficients, respectively. The decentralization thresholds with respect to stakes distribution and geographic locations are η_s, $\eta_l \in [0, 1]$. W is the threshold constant of latency, f is the number of malicious validators, k is the number of replicas, R is the set of replicas, TPS(t) denotes the number of transactions confirmed per second, and max (TPS) is the maximum number of transactions verified per second in the blockchain. N is the set of candidate nodes for joining replicas and η_f is the threshold of fairness.

Remember that the proposed GG model must reach its optimal value for each execution of this cognitive engine; if there are many candidate nodes, the proposed method's execution time will lengthen, and the blockchain's performance will suffer. The proposed algorithm is executed in the background of blockchain processes and according to the previous rewards received to solve this challenge. For example, if the cognitive engine is to be executed after every 10 block generation. After the tenth-block is generated, the cognitive engine execution starts to be used in the 20th block. Thus,

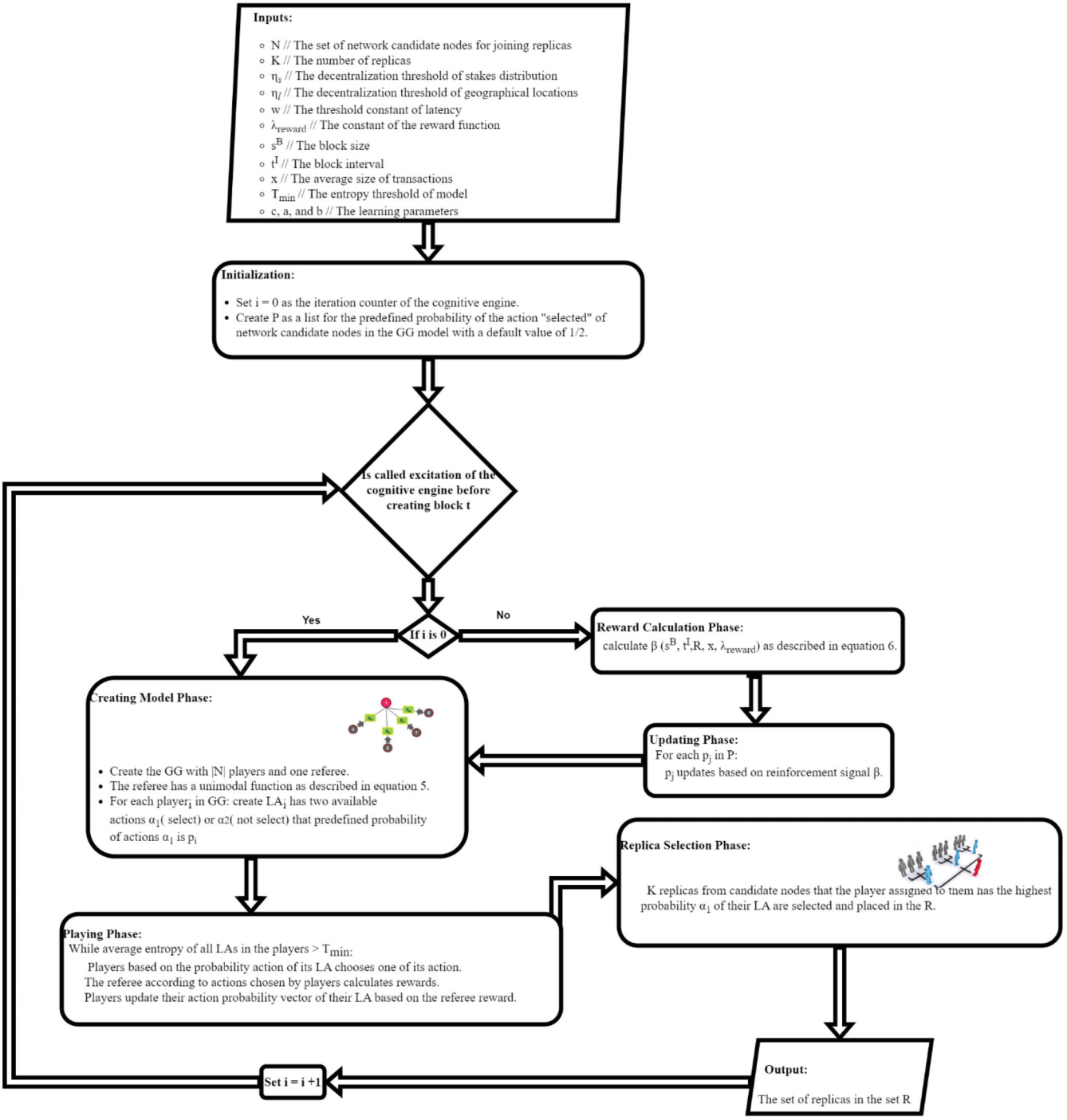


FIGURE 6 General block diagram of the GG-V-CB-BFT algorithm.

at the beginning of the creation of the 20th block, the proposed algorithm has converged, and we will not have a few seconds of delays to run this algorithm while producing the 20th block. The only problem with this solution is that the reward received based on which the trained model is not quite up to date. Figure 6 shows the general block diagram of GG-V-CB-BFT.

4.2.2 | A CGG-based approach to select validators in the cognitive blockchain (CGG-V-CB BFT)

This proposed cognitive engine employed a Cellular Goore Game as a learning model for select validators. This approach grouped candidate nodes for joining replicas to maintain decentralization based on geographical location and stock

distribution. We assume that CN denotes n candidate nodes $=\{cn_1, cn_2, \ldots, cn_n\}$ clustered into m groups, denoted by $C=\{c_1, c_2, \ldots, c_m\}$, based on geographical location and stock distribution in the cognitive blockchain. We denote the ith cluster head by CH_i where $i \in \{1, 2, \ldots, m\}$. The overlap of clusters is permitted. A graph $N=(V, E)$ can represent the CGG network, where V is the set of nodes, which includes candidate nodes and CH. In addition, the set of edges is specified by E below. Each edge belongs to a node in this cluster. In the CGG-based algorithm, each CH acts as the referee and each node as the player. Equation (8) illustrates how each CH rewards its nodes using a uni-modal performance criterion.

$$G_i(k_t) = c + ae^{-\left(\frac{\left(k_i^* - k_t\right)}{b}\right)^2}, \tag{8}$$

where, c, a, and b are learning parameters; k_t is the number of nodes with "select" action for ith cluster in round t; and k_i^* is the ratio of the optimum number of replicas to total players in the ith cluster. Each CH describes the goal value and rewards its neighboring nodes by the performance criterion function to achieve this. The sum of the desired value for clusters in CGG that all its clusters are disjoint should equal the number of replicas to the total candidate nodes for joining replicas. Depending on the various desired values, the performance criterion for referees may have a different function. In this case, the proposed CGG is inhomogeneous.

This defined CGG model should be run until it converges to the optimal value, and then nodes will be chosen as replicas whose equivalent player has converged to the "select" action. In this proposed approach to realize the goal of cognitive blockchain, we define a reward function that shows the influence of the selected nodes in the learning model on processing rates in terms of transactions per unit time and the security, decentralization, and delay criteria. Similar to GG-V-CB-BFT, Equation (7), is used as the formula for calculating the reward in this section. Calculated reward resulting from executing the cognitive engine modifies the initial probability of choosing the action of CGG players in the future. Whenever there are many candidate nodes, this approach's execution time will increase and will reduce the performance of the blockchain. The CGG models can be executed according to the previous rewards in the background of blockchain processes before generating a block that needs to choose a validator to solve this problem. Figure 7 illustrates the general block diagram of CGG-V-CB-BFT.

4.2.3 | An LA-based approach to select validators in the cognitive blockchain (LA-V-CB-BFT)

In this proposed approach for cognitive engine, the learning model contains a set of learning automata where LA_i is the resident learning automaton in $node_i$ to select or not select that $node_i$ as a validator. The reward function in this learning automata is also defined as described in Equation (7). Figure 8 gives the general block diagram of LA-V-CB-BFT. Network nodes are also grouped in the proposed method to maintain decentralization in selecting validators based on geographical location and stock distribution. Since only k nodes should be selected as validators in this consensus protocol, a method for selecting only k validators from the candidate nodes is also provided in the pseudocode in Figure 9.

5 | EXPERIMENTS

Several experiments are performed in this section to assess the proposed approaches. We utilize the Talaria simulator[85,86] for blockchain simulation to accurately examine the performance of the proposed methods. Talaria is the first blockchain simulator based on the open-source blockchain simulator BlockSim for simulating private blockchain models.[87] However, changes have been made to the simulator code to implement the proposed protocol, such as dynamizing a series of parameters such as block size, block time interval, and selecting validators from candidate nodes. General simulation parameters are given in Table 2. All of the results are averages obtained over 30 runs to minimize statistical bias. Experiments are conducted in Python using the Talaria simulator on a PC, which has a single CPU of Intel(R) Corei7-960 3.2 GHz and 16 GB of memory. Experiments reported in this section for a more detailed evaluation are conducted on nine baseline schemes in the PBFT consensus protocol, as given in Table 3. The learning automata employed SL_{RI} as their learning algorithm for all experiments. In the rest of this section, we first give the evaluation metric employed in our experiments and then describe a series of experiments.

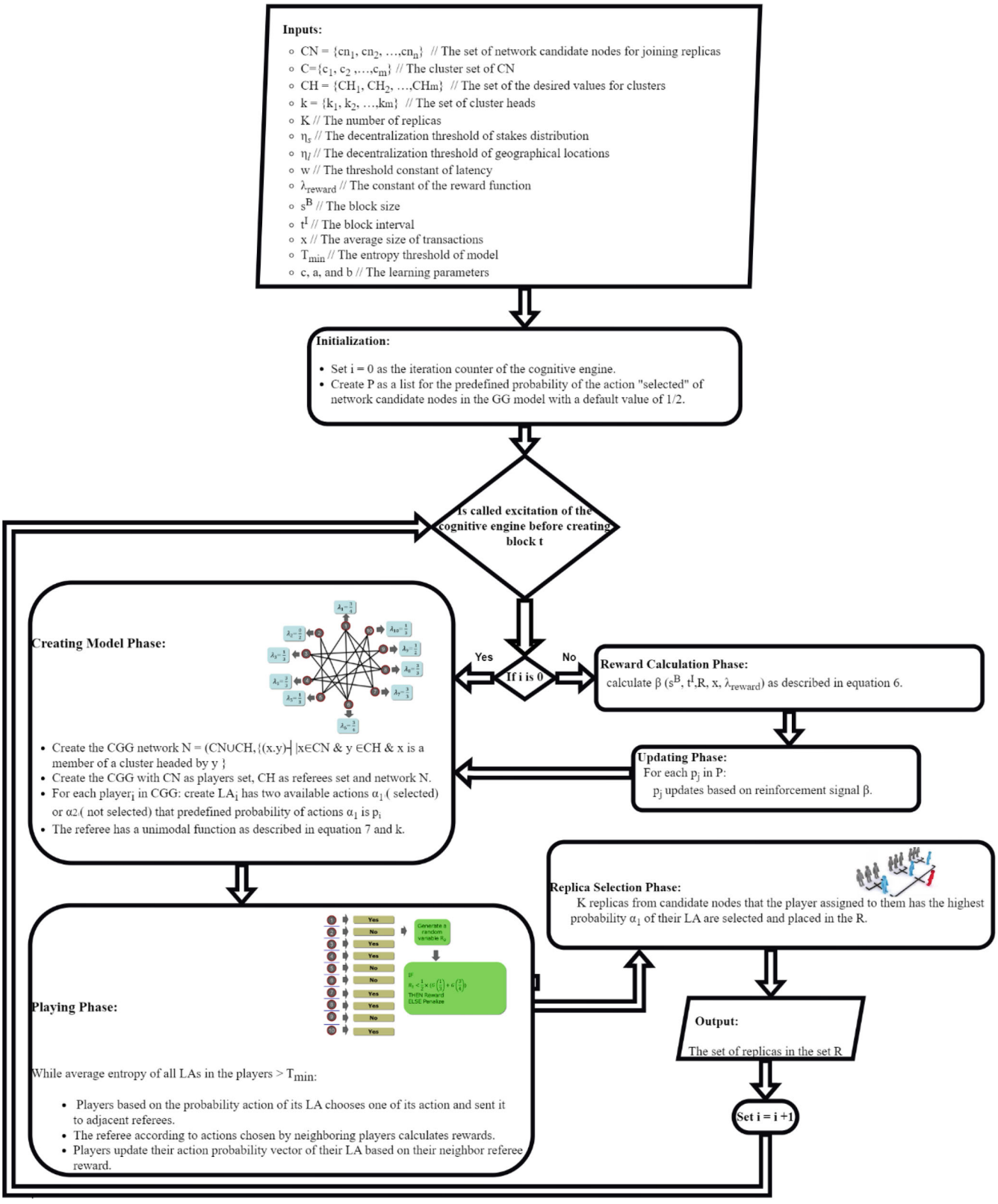


FIGURE 7 General block diagram of the CGG-V-CB-BFT algorithm.

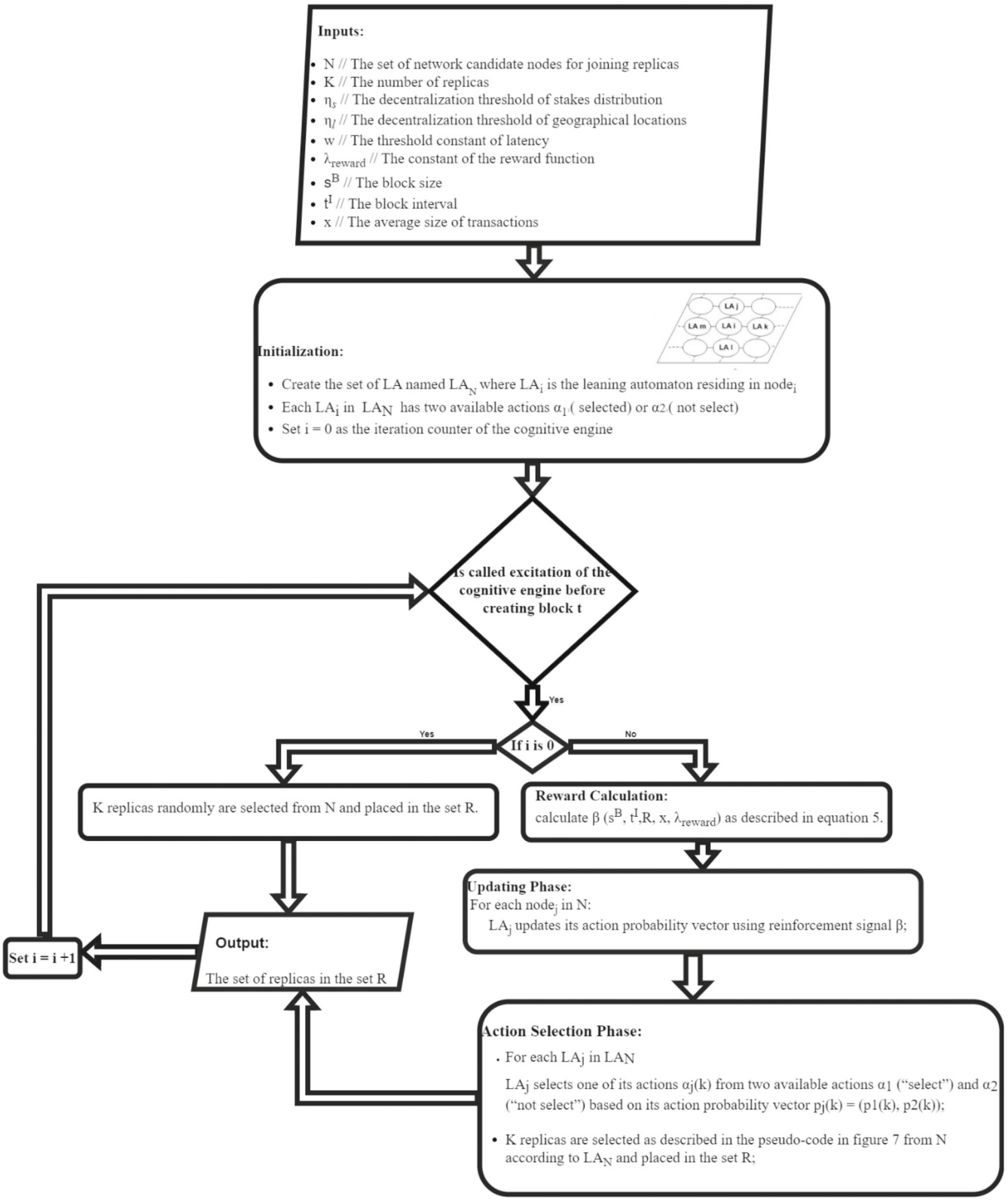


FIGURE 8 General block diagram of the LA-V-CB-BFT algorithm.

```
Inputs
N // The set of network candidate nodes for joining to replicas
K // The number of replicas
G_N // Grouping of nodes according to the number of stocks and their geographical location
Notations:
R // The set of replicas
LA_N // The set of learning automata where LA_i is the is the resident learning automaton in node_i
k_g // The limit number of each group
Begin
  For each G_j in G_N do
    Selected_Nodes = list of nodes that belong to this group and the last action of this
    node is actions α1 ("Select") order by action probability asc;
    Not_Selected_Nodes = list of nodes that belong to this group and the last action of
    this node is actions α2 ("Not select") order by action probability desc;
    Diff = The size of Selected_Nodes - k_g;
    If (Diff < 0) Then
      For each node_k in Not_Selected_Nodes do
        LA_k = LA that related to node_k
        If (Diff is 0 ) Then
          break;
        EndIf
        Set last action of LA_k to α1 ("Select");
        Get reinforcement signal 1 as reward to LA_k;
        LA_k update their action probability vector using reinforcement signal 1;
        Diff = Diff +1;
      EndFor
    EndIf
    If (Diff > 0) Then
      For each node_k in Selected_Nodes do
        LA_k = LA that related to node_k
        If (Diff is 0 ) Then
          break;
        EndIf
        Set last action of LA_k to α2 ("Not select");
        Get reinforcement signal 1 as reward to LA_k;
        LA_k update their action probability vector using reinforcement signal 1;
        Diff = Diff -1;
    EndIf
  EndFor
  K replicas are selected from N that its LA select action α1 ("Select") as the last action
  and placed in the set R;
End
```

FIGURE 9 Pseudocode of the selection *K* replicas from network candidate nodes.

5.1 | Evaluation metrics

To investigate the performance of the proposed cognitive engines, the average entropy, the average performance function, the faulty node selection rate, fairness in node selection, fairness in non-faulty node selection, time overhead, and blockchain performance metrics: throughput, average latency, and average block creation time are utilized, as formulated in the following paragraphs. The change in the learning automata's states can be investigated using entropy. The equation below defines the LA-based model's average entropy at iteration t:

$$H(t) = -1/n \sum_{i=1}^{n} \sum_{j=1}^{r_i} p_j^i(t) \cdot \ln\left(p_j^i\right). \tag{9}$$

TABLE 2 General simulation parameters.

Simulation parameter	Value
The number of nodes, N	100
The number of replicas, K	21
Average transaction size, x	200 B
The number of intervals that a new block should be validated, w	6
The decentralization thresholds in regard to stakes distribution and geographic locations, η_s, η_l	0.2, 0.3
The thresholds of fairness, η_f	0.75
The set of block sizes, T^I	{0.5, 1, 1.5 s}
The set of block intervals, S^B	{0.5, 1, 1.5 MB}
The percentage of faulty nodes	40%
The maximum number of transactions verified per second, max (TPS)	350
The number of clusters in CGG models	5
The limit number of each group, k_g	5
The learning rate (a, b) in learning automata	(0.02, 0)

TABLE 3 Baseline schemes.

Baseline schemes	Descriptions
LA&GG-SIV-CB	The schema presented in this paper with dynamic selective replicas, block size and block interval intelligently with LA-SI-CB-BFT and GG-V-CB-BFT approaches
LA&CGG-SIV-CB	The schema presented in this paper with dynamic selective replicas, block size and block interval intelligently with LA-SI-CB-BFT and CGG-V-CB-BFT approaches
LA-SIV-CB	The schema presented in this paper with dynamic selective replicas, block size and block interval intelligently with LA-SI-CB-BFT and LA-V-CB-BFT approaches
GG-V-CB	The schema presented in this paper with dynamic selective replicas intelligently with GG-V-CB-BFT approaches
CGG-V-CB	The schema presented in this paper with dynamic selective replicas intelligently with CGG-V-CB-BFT approaches
LA-V-CB	The schema presented in this paper with dynamic selective replicas intelligently with LA-V-CB-BFT approaches
LA-SI-CB	The schema presented in this paper with dynamic selective block size and block interval intelligently with LA-SI-CB-BFT approach
Random-V	The schema with randomly selective replicas
Static scheme	The schema that the blocks generated by fixed pre-specified block producers with the same size (4 MB) every 1 s

The number of learning automata in the model is given by n, and each learning automaton's actions are given by r_i. The probability that the ith learning automaton will select action α_j at iteration t is $p_j^i(t)$. If all learning automata in the model stabilize their selected action, $H(t)$ value will be 0. Higher $H(t)$ values indicate higher rates of change in the actions chosen by the model's learning automata. The behavior of the CGG model can be analyzed in terms of the scaled average performance function (G_{avg}) defined in Reference 48. The behavior quality of the GG model can also be evaluated by the value of the referee's performance criterion function in each round.

The faulty node selection rate is one of the most meaningful criteria for evaluating the proposed protocol's correctness. The lower this criterion of the proposed approach is, the more intelligent it will be and the fewer malicious nodes it will choose. The average ratio of faulty nodes chosen as validators to all validators is known as the faulty node selection rate. It can be expressed as follows, where V indicates the validator set, and T is the number of episodes:

$$\text{Faulty Node Selection Rate}(V) = \frac{\sum_{t=1}^{T} \frac{\text{The Number of Faulty Nodes in } V_t}{|V_t|}}{T}. \tag{10}$$

Fairness in node selection and fairness in non-faulty node selection are the measures for assessing the proposed method in terms of fairness presented for the first time in this paper. Fairness in non-faulty node selection is also calculated similarly to the fairness measure, with the difference that N is the set of non-faulty candidate nodes for joining validators. It should be emphasized that the faulty node selection rate and fairness in non-faulty node selection measures are not applicable in the actual world because it is usually unclear which node is malicious. Time overhead is the time required to perform cognitive engine processes to select parameters or provide services.

Throughput and latency are critical metrics to evaluate the performance of consensus methods in distributed systems such as blockchain.[88] As mentioned earlier, the throughput of a blockchain is the number of transactions processed per second (TPS). It can be calculated based on how many transactions are processed over a specific test duration and then calculated for a second. The average time between the creation of a transaction and its confirmation is used to compute the average latency, which is the average amount of time it takes for a transaction to be confirmed.

5.2 | Experiment 1

This experiment investigates the convergence performance of our proposed cognitive engine schemas in terms of entropy and the value of the performance function. Figure 10 depicts the proposed schema's entropy versus the cognitive engine iteration number for the learning automaton that selected the block size (LAS) and the block interval (LAT) in LA-SI-CB. Furthermore, Figure 11 shows the average entropy versus the cognitive engine iteration number for the set of network node automata in LA-V-CB. The average entropy versus iteration number in GG-V-CB and CGG-V-CB are represented in Figure 12. Entropy values are high in the initial rounds and steadily decline, indicating a decreasing rate of change, as shown in Figures 10–12. Finally, the actions chosen by the learning automata residing in the suggested models experience a change rate of zero. Considering the results of this experiment, we can conclude that LAS and LAT reach a stable state after around 203 and 291 iterations, respectively. The set of network node automata also converges after 139 iterations.

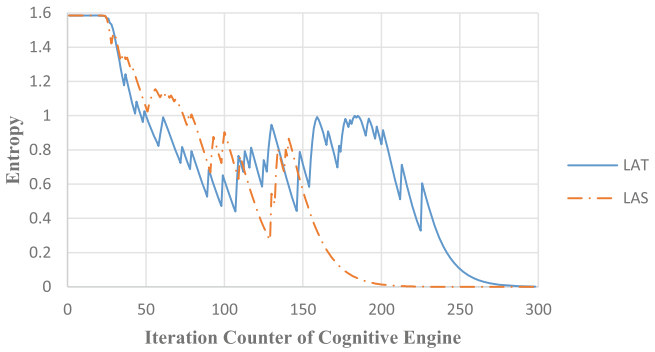


FIGURE 10 The entropy versus iteration counter of the cognitive engine for LAT and LAS.

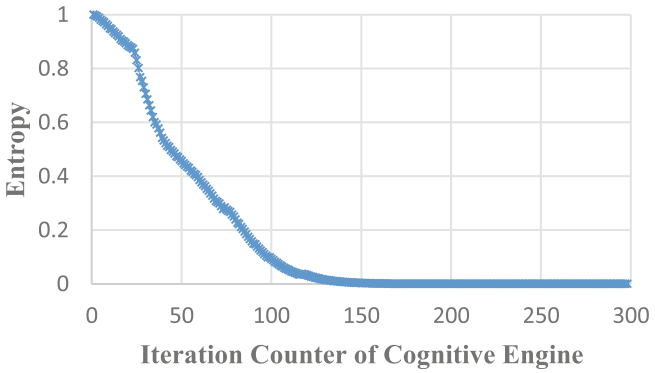


FIGURE 11 The average entropy versus iteration counter for the set of network node automata in LA-V-CB.

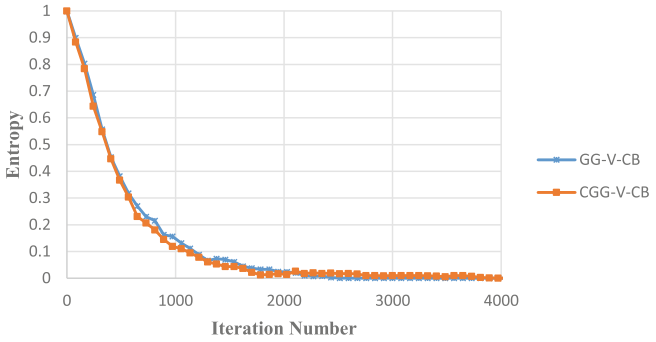
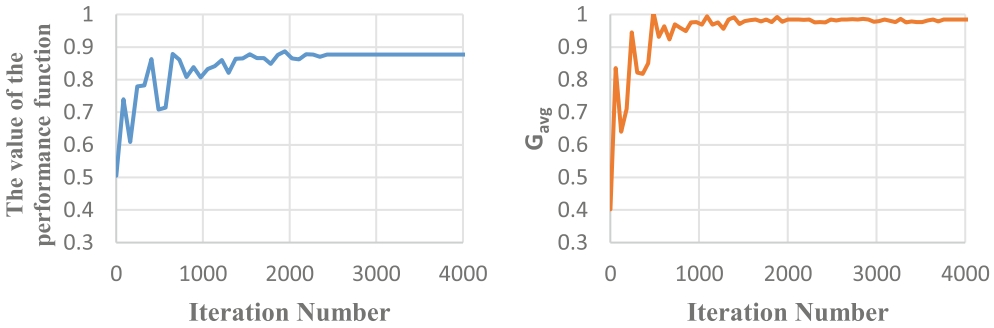


FIGURE 12 The average entropy versus iteration number in GG-V-CB and CGG-V-CB.



(A) GG-V-CB. (B) CGG-V-CB.

FIGURE 13 G_{avg} and the performance function of the proposed models in GG-V-CB and CGG-V-CB.

We plot G_{avg} and the performance functions of the proposed models in GG-V-CB and CGG-V-CB. Figure 13 depicts the results of this experiment. The findings of this experiment show that G_{avg} of CGG finally converges to the highest value and that the referee's performance criterion function in the GG model for GG-V-CB converges to the maximum value.

5.3 | Experiment 2

This experiment analyzes and compares the performance of our proposed cognitive engines. The results of this experiment are given in Figure 14 and Table 4. Figure 14 outlines throughput (in transaction per second) versus episode number of different proposed cognitive engine schemas. The following conclusions can be drawn from the experiment's results based on evaluation measures.

- Figure 14 illustrates that while transactional throughput is low early in the learning process, it increases as the number of episodes increases. Therefore, enhancing network throughput can demonstrate that the suggested cognitive engine adjusts the appropriate parameters.
- According to our findings, the LA&GG-SIV-CB and LA&GG-SIV-CB can receive higher throughput than the other baselines since the suggested approach enables the adaptive selection of block producers, block size, and block interval.
- The time overhead of the LA&GG-SIV-CB and LA&GG-SIV-CB is higher than the others, which is the biggest weakness of these methods. The proposed algorithms are executed in the background of blockchain processes and are based on the previous rewards to solve this issue in this experiment. So this more time overhead has almost no effect on the throughput of these schemas. However, it has caused the average consensus time for these methods to be slightly higher.
- As shown in Table 4, the average selection of faulty nodes in LA&CGG-SIV-CB is lower than in others, indicating that the number of selected faulty nodes is less than in other schemas.

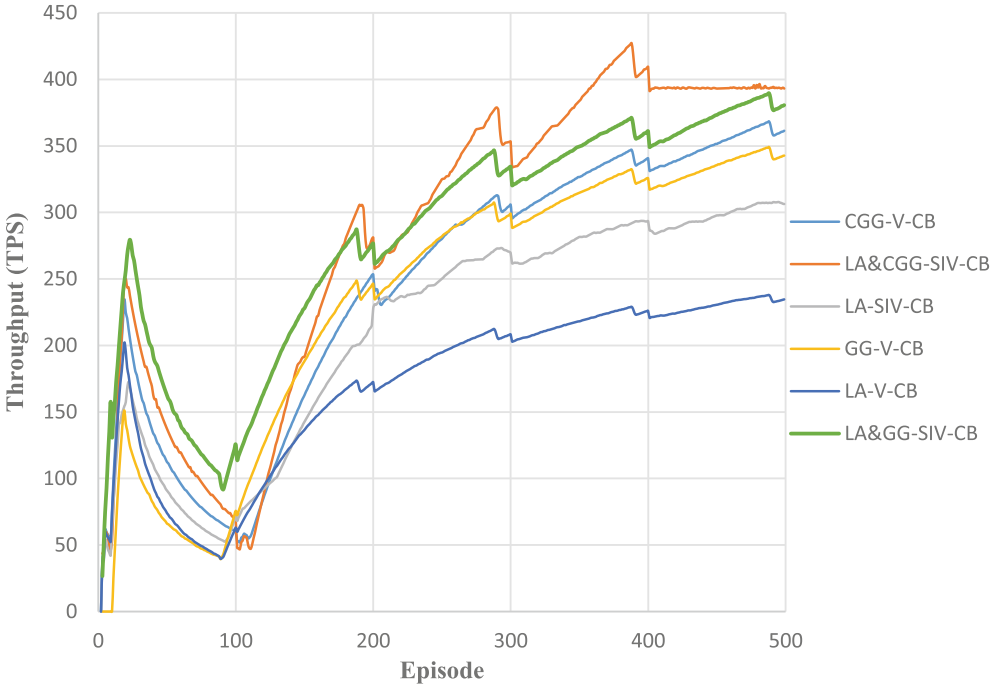


FIGURE 14 Throughput versus episode number of different schemes in experiment 2.

TABLE 4 Comparative analysis of different proposed schemes in experiment 2.

Baseline scheme	Average latency (ms)	Average block creation time (ms)	Faulty node selection rate (%)	Fairness in node selection	Fairness in non-faulty node selection	Time overhead (ms)
LA&GG-SIV-CB	3512	1252	19	0.55	0.51	2153
LA&CGG-SIV-CB	3294	1289	14	0.43	0.41	3571
LA-SIV-CB	3711	1118	25	0.52	0.49	156
GG-V-CB	3798	1244	19	0.54	0.5	2190
CGG-V-CB	3695	1269	14	0.45	0.42	3394
LA-V-CB	4013	1091	27	0.52	0.50	79

- Table 4 shows that the fairness in node selection and fairness in non-faulty node selection metrics of LA&CGG-SIV-CB have decreased compared to others. Therefore, LA&CGG-SIV-CB is better than others in fairness measures. Since the LA&CGG-SIV-CB schema clusters the nodes and a certain number of these clusters are selected, there is a greater chance of selecting all the nodes. LA&GG-SIV-CB, in the fairness measures, has performed worse than others.

5.4 | Experiment 3

In this experiment, we compare the performance of the proposed cognitive engine schemas with the existing static and Random-V schemes to demonstrate a comparative analysis. Considering that in this experiment, we intended to compare our proposed schemas with the static scheme, if the number of network nodes exceeds a specific limit, the efficiency of the network will be very low in the static schema. Therefore, we reduced the number of network nodes in this experiment. In this simulation experiment, the number of network candidate nodes is 30, and the number of replicas is 20 in the proposed schemas. All network nodes act as replica in the existing static schema as PBFT protocols. The percentage of faulty nodes

TABLE 5 Comparative analysis of different schemes in experiment 3.

Baseline scheme	Throughput (TPS)	Average latency (ms)	Average block creation time (ms)	Faulty node selection rate (%)	Fairness in node selection	Fairness in non-faulty node selection	Time overhead (ms)
LA&GG-SIV-CB	381	3623	1452	12	0.54	0.44	942
LA&CGG-SIV-CB	421	3312	1473	9	0.41	0.39	1414
LA-SIV-CB	352	3821	1329	15	0.51	0.49	85
Random-V	302	4215	1211	22	0.32	0.37	9
Static scheme	153	7031	6329	29	-	-	0

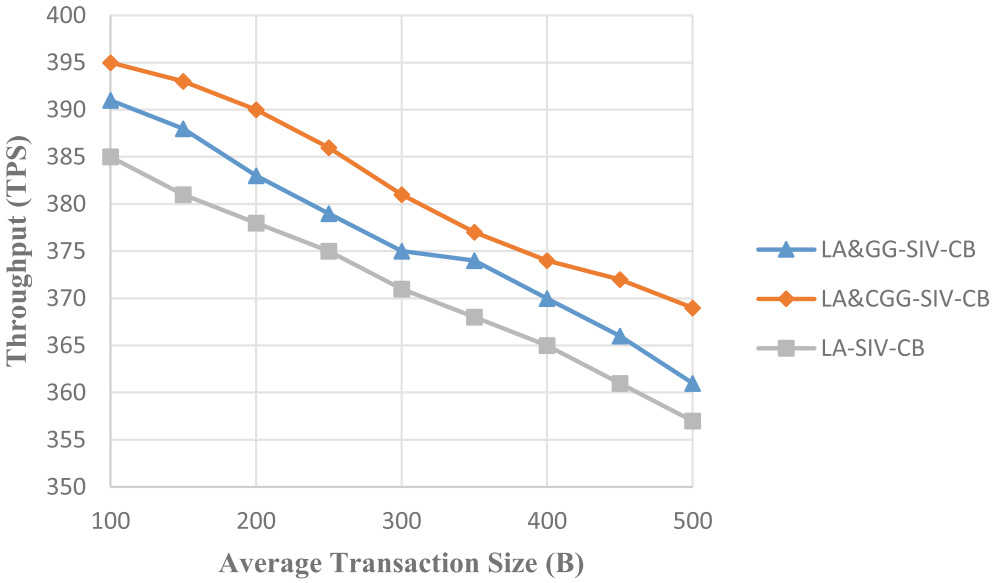


FIGURE 15 Throughput versus average transaction size in experiment 4.

is also 29%. The reported results shown in Table 5 result from creating 1000 blocks in the blockchain. According to the results obtained in this experiment, we may conclude the following:

- We can conclude that the proposed schemas, in terms of the average latency, and throughput perform much higher than the existing static scheme from the results of this experiment. Since the proposed schemas select some nodes as validators, the number of nodes participating in the consensus process decreases, and the throughput improves.
- The average consensus time in the existing static schema is much higher than in others. The reason for that is the more significant number of nodes that must exchange messages to reach a consensus.
- The faulty node selection rate is a meaningful parameter for the performance evaluation of the selection mechanism of the proposed schema. The faulty node selection rate in Random-V is higher than in proposed schemas, indicating that the number of selected faulty nodes is higher, considering that the selection of validators is entirely random.
- Note that the fairness in node selection and the fairness in non-faulty node selection of Random-V have decreased compared to others. Therefore, Random-V is better than others at fairness. This result is justified because of the utterly random selection of validators in this schema. Of course, according to the results, the LA&CGG-SIV-CB schema has performed well regarding fairness criteria.
- Similar to the previous experiment, the time overhead of the LA&GG-SIV-CB and LA&GG-SIV-CB is greater than the others.

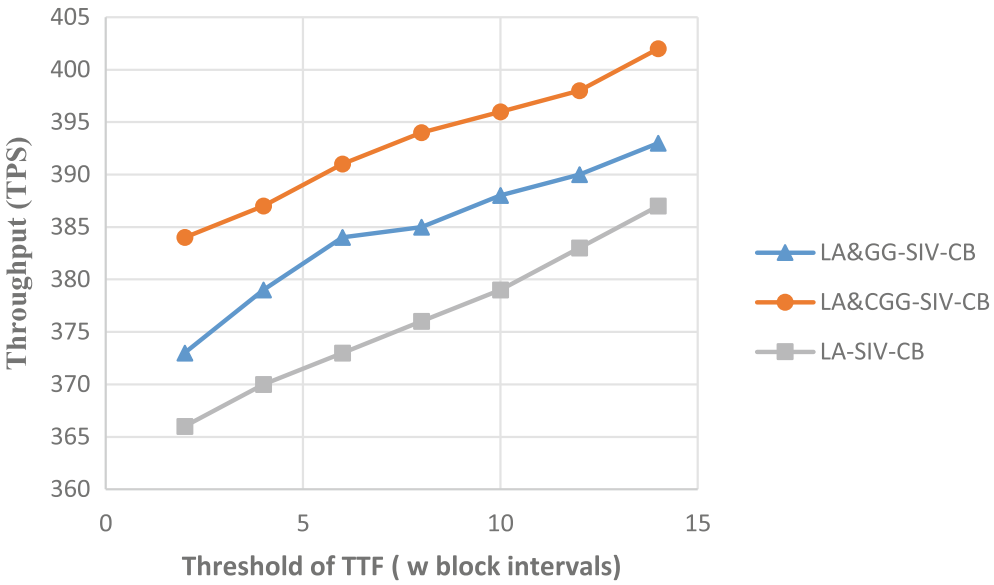


FIGURE 16 Throughput versus threshold of TTF in experiment 4.

5.5 | Experiment 4

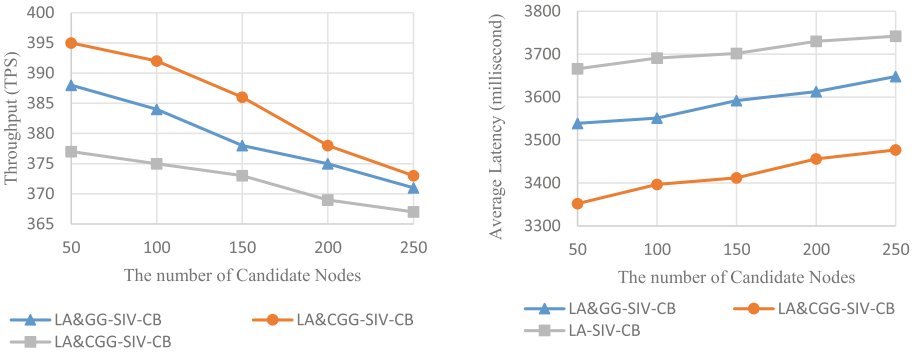
This experiment is conducted to study the effects of different parameters, such as average transaction size and threshold of time to finality (TTF), on the performance of the blockchain network. According to the results of this experiment in 1000 episodes of block creation, which are shown in Figures 15 and 16, we can conclude the following:

- Figure 15 presents the transactional throughput versus average transaction size for different proposed cognitive engine schemas. Figure 15 shows that the throughput of the blockchain reduces as transaction sizes rise for all the schemes. This is so that fewer transactions of larger sizes can fit in a single block. In addition, LA&CGG-SIV-CB outperforms the competition in throughput despite variations in average transaction size.
- Figure 16 plots the throughput versus threshold of TTF for different proposed cognitive engine schemas to explore the transactional throughput of the system as a function of the TTF threshold. It is evident from Figure 16 that increasing the threshold of TTF causes an increase in throughput for all the schemes since the validators can handle more transactions in one block with a higher threshold of TTF.

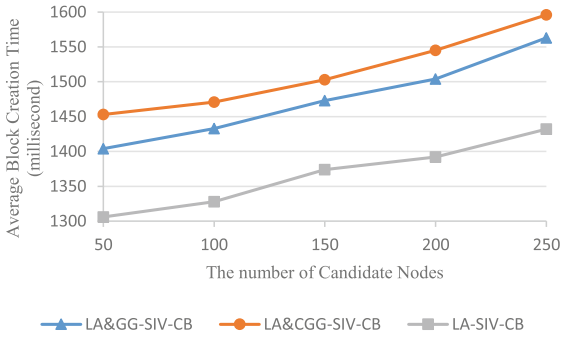
5.6 | Experiment 5

In this experiment, the performance of the schemas is assessed by the number of candidate nodes and replicas. Figure 17 shows throughput, average latency, and average block creation time results for the proposed cognitive engine schema with different numbers of candidate nodes and 1000 episodes. By increasing the number of candidate nodes, we can see that the proposed schemes obtain a slightly lower transactional throughput and a slightly higher average latency and block creation time. However, the throughput decreases with increasing the number of candidate nodes with a slight slope. As the number of nodes increases, there are more learning automata in the proposed models and the need for more coordination between nodes, so the efficiency of the proposed method decreases slightly. However, this reduction in performance is negligible compared to when BFT-based consensus protocols face an increase in nodes.

Figure 18 demonstrates the throughput versus the number of replicas. As can be seen, the throughput decreases as the number of replicas or nodes participating in the consensus mechanism increases because the proposed schemas procedures require more processing and coordination.



(A) Throughput versus the Number of Candidate Nodes. (B) Average Latency versus the number of candidate nodes.



(C) Average Block Creation Time versus the Number of Candidate Nodes.

FIGURE 17 The effect of the number of candidate nodes on schemas' performance in experiment 5.

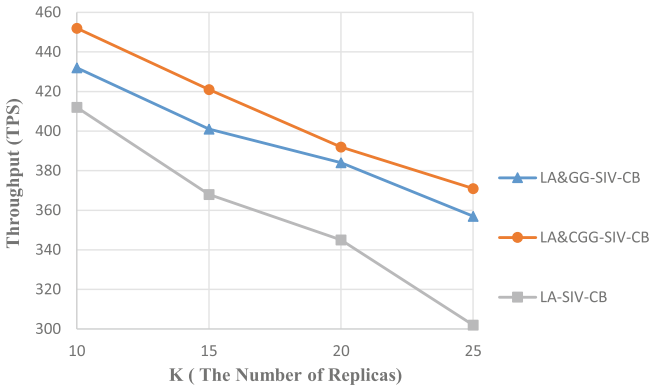


FIGURE 18 Throughput versus the number of replicas in experiment 5.

5.7 | Experiment 6

In a simulation experiment, we evaluated the effect of the percentage of faulty nodes on the proposed schemas' performance. Figures 19 and 20 depict the simulation results in 1000 episodes. As shown in Figure 19, the throughput decreases as the percentage of faulty nodes increases. The faulty node selection rate increases with the increase of the percentage of faulty nodes, as seen in Figure 20. Even if 50% of the nodes are faulty, the faulty node selection rates of LA&GG-SIV-CB and LA&CGG-SIV-CB are below 30% and have acceptable throughputs. However, the throughput of LA-SIV-CB significantly decreases when the percentage of defective nodes is more than 40%, because this schema cannot select a sufficient number of non-faulty nodes as replicas in this condition.

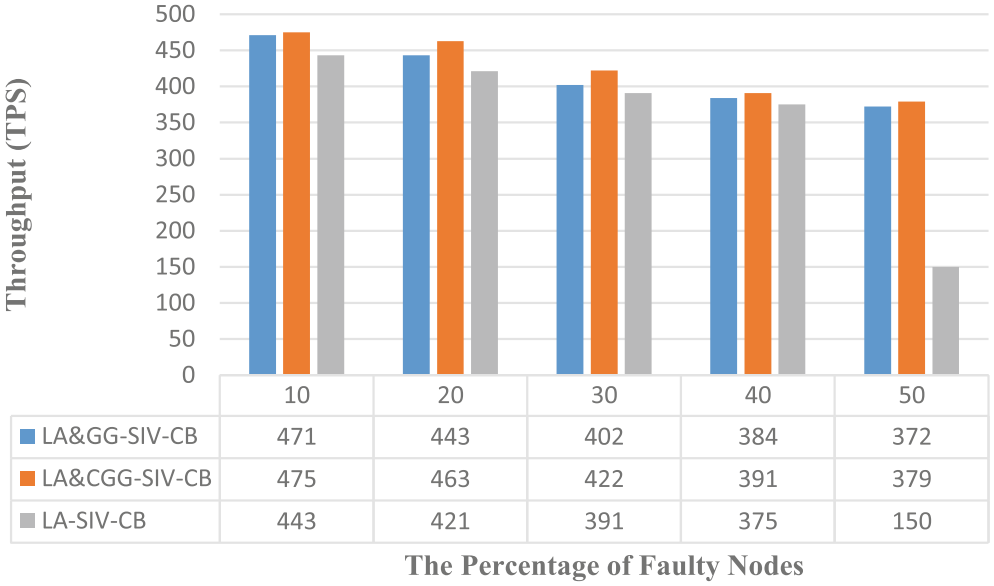


FIGURE 19 Throughput versus the percentage of faulty nodes in experiment 6.

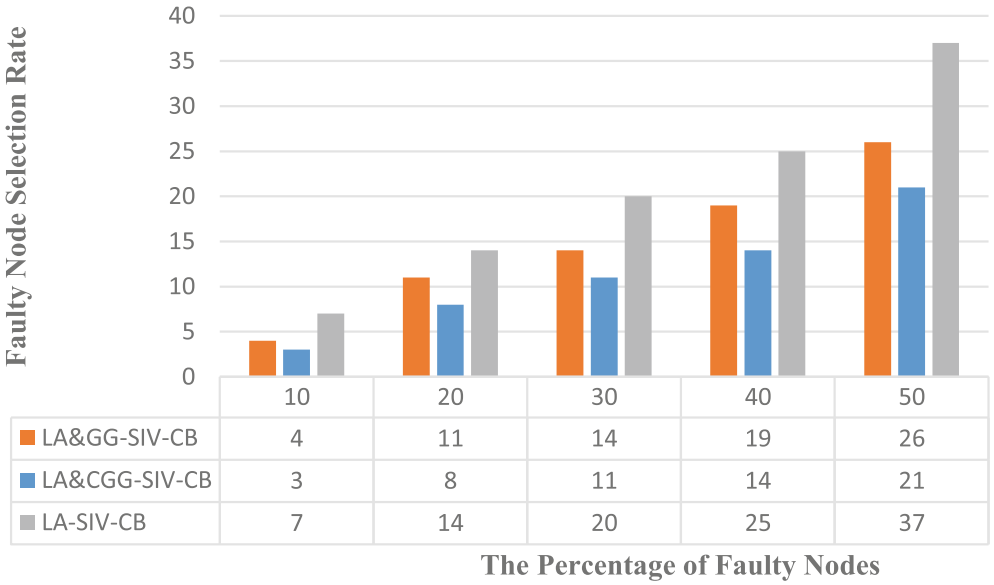


FIGURE 20 The faulty node selection rate versus the percentage of faulty nodes in experiment 6.

6 | CONCLUSIONS

We introduced cognitive blockchain in this paper, which can perceive network conditions, analyze gained knowledge, effectively decide, and adapt to improve network performance. We additionally provided a framework for cognitive blockchain that illustrated the fundamental cognitive tasks. We also proposed approaches based on learning automata for developing cognitive engines for performance optimization in blockchain systems with BFT-based consensus in order to show the potential of the cognitive blockchain. These proposed cognitive engines intelligently select the block size, time interval, and validators to optimize blockchain scalability and performance. We clarified all the details for executing the proposed algorithms in the template of a cognitive blockchain architecture. Our suggested approaches can be employed on consortium blockchains and are not limited to a specific application. We conducted several blockchain simulator experiments in the Talaria simulator to assess the effects of the proposed cognitive engines on blockchain performance and demonstrated that utilizing the proposed cognitive engines enhances blockchain performance metrics. Furthermore, we present the node fairness measure as a novel metric for assessing the fairness of blockchain systems. However, one of the

disadvantages of our proposed methods is that according to the learning model used, we must consider a smaller number of actions for learning automata to increase the speed of convergence. In future work, we will define more measures to quantify blockchain performance and develop cognitive engines in other consensus algorithms to optimize blockchain performance. Also, we will intend to apply other artificial intelligence methods, such as outlier analysis and clustering, as knowledge discovery techniques in the proposed cognitive engines of cognitive blockchain.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Reyhaneh Ameri https://orcid.org/0000-0002-0335-0052

REFERENCES

1. Nakamoto S, Bitcoin A. A peer-to-peer electronic cash system. *Bitcoin*. 2008;4(2):15. https://bitcoin.org/bitcoin.pdf
2. Hu X, Song X, Cheng G, Wu H, Gong J. Efficient sharing of privacy-preserving sensing data on consortium blockchain via group key agreement. *Comput Commun*. 2022;194:44-54.
3. Wang X, Ni W, Zha X, et al. Capacity analysis of public blockchain. *Comput Commun*. 2021;177:112-124.
4. Lamport L. The weak byzantine generals problem. *J ACM*. 1983;30(3):668-676.
5. Benčić FM, Žarko IP. Distributed ledger tecchnology: blockchain compared to directed acyclic graph. *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE; 2018:1569-1570.
6. Liu Y, Yu FR, Li X, Ji H, Leung VCM. Blockchain and machine learning for communications and networking systems. *IEEE Commun Surv Tutor*. 2020;22(2):1392-1431. doi:10.1109/COMST.2020.2975911
7. Dhar S, Khare A, Singh R. Advanced security model for multimedia data sharing in internet of things. *Trans Emerg Telecommun Technol*. 2022;34:e4621.
8. Bugday A, Ozsoy A, Öztaner SM, Sever H. Creating consensus group using online learning based reputation in blockchain networks. *Pervasive Mob Comput*. 2019;59:101056. doi:10.1016/j.pmcj.2019.101056
9. Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: architecture, consensus, and future trends. *2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE; 2017:557-564.
10. Ethereum. *Home | ethereum.org. Welcome to Ethereum*; 2021. Accessed November 22, 2021. https://ethereum.org/en/
11. Yang J, Jia Z, Su R, Wu X, Qin J. Improved fault-tolerant consensus based on the PBFT algorithm. *IEEE Access*. 2022;10:30274-30283.
12. Kosba A, Miller A, Shi E, Wen Z, Papamanthou C. Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE; 2016:839-858.
13. Meiklejohn S, Pomarole M, Jordan G, et al. A fistful of bitcoins: characterizing payments among men with no names. *Proceedings of the 2013 Conference on Internet Measurement Conference*. ACM; 2013:127-140.
14. Liu M, Yu FR, Teng Y, Leung VCM, Song M. Performance optimization for blockchain-enabled industrial internet of things (iiot) systems: a deep reinforcement learning approach. *IEEE Trans Ind Inform*. 2019;15(6):3559-3570. doi:10.1109/TII.2019.2897805
15. Kiayias A, Russell A, David B, Oliynykov R. Ouroboros: A provably secure proof-of-stake blockchain protocol. In: *Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part I*; 2017:357-388.
16. Grigg I. *Eos—An Introduction. White Paper*; 2017. https://whitepaperdatabase.com/eos-whitepaper
17. Cash B. *Bitcoin Cash-Peer-to-Peer Electronic Cash*; 2017. Accessed October 5, 2017. https://www.bitcoincash.org
18. Poon J, Dryja T. *The Bitcoin Lightning Network: Scalable off-Chain Instant Payments*; 2016.
19. Decker C, Wattenhofer R. A fast and scalable payment network with bitcoin duplex micropayment channels. *Lect Notes Comput Sci*. 2015;9212:3-18. doi:10.1007/978-3-319-21741-3_1
20. Green M, Miers I. Bolt: anonymous payment channels for decentralized currencies. *Proceedings of the ACM Conference on Computer and Communications Security*. ACM; 2017:473-489. doi:10.1145/3133956.3134093
21. Eyal I, Gencer AE, Sirer EG, Van Renesse R. Bitcoin-NG: a scalable blockchain protocol. In: *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016*; 2016:45-59.
22. Poon J, Buterin V. *Plasma: Scalable Autonomous Smart Contracts. White Paper*; 2017:1-47.
23. Joseph Poon TD. *The Bitcoin Lightning Network: Scalable Off-Chain Payments*; 2016:1-59. http://lightning.network/lightning-network-paper.pdf
24. Liu M, Teng Y, Yu FR, Leung VCM, Song M. Deep reinforcement learning based performance optimization in blockchain-enabled internet of vehicle. *IEEE International Conference on Communications*. Vol 2019. IEEE; 2019:1-6. doi:10.1109/ICC.2019.8761206
25. Tang S, Wang Z, Jiang J, Ge S, Tan G. Improved PBFT algorithm for high-frequency trading scenarios of alliance blockchain. *Sci Rep*. 2022;12(1):1-12.
26. Qin H, Cheng Y, Ma X, Li F, Abawajy J. Weighted byzantine fault tolerance consensus algorithm for enhancing consortium blockchain efficiency and security. *J King Saud Univ Inf Sci*. 2022;34(10):8370-8379.
27. Pham T, Lee S. Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv Prepr. arXiv1611.03941*; 2016. http://arxiv.org/abs/1611.03941

28. Monamo PM, Marivate V, Twala B. A multifaceted approach to bitcoin fraud detection: global and local outliers. *Proceedings – 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016*. IEEE; 2017:188-194. doi:10.1109/ICMLA.2016.19
29. Monamo P, Marivate V, Twala B. Unsupervised learning for robust bitcoin fraud detection. In: *2016 Information Security for South Africa – Proceedings of the 2016 ISSA Conference*; 2016:129-134. doi:10.1109/ISSA.2016.7802939
30. Sayadi S, Ben Rejeb S, Choukair Z. Anomaly detection model over blockchain electronic transactions. In: *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*; 2019:895-900. doi:10.1109/IWCMC.2019.8766765
31. Morishima S. Scalable anomaly detection method for blockchain transactions using GPU. In: *Proceedings – 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2019*; 2019:60-165. doi:10.1109/PDCAT46702.2019.00039
32. Salimitari M, Joneidi M, Chatterjee M. AI-enabled blockchain: an outlier-aware consensus protocol for blockchain-based iot networks. *2019 IEEE Global Communications Conference, GLOBECOM 2019 – Proceedings*. IEEE; 2019:1-6. doi:10.1109/GLOBECOM38437.2019.9013824
33. Podgorelec B, Turkanović M, Karakatič S. A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. *Sensors (Switzerland)*. 2020;20(1):147. doi:10.3390/s20010147
34. Jang H, Lee J. An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information. *IEEE Access*. 2017;6:5427-5437. doi:10.1109/ACCESS.2017.2779181
35. Demir A, Akilotu BN, Kadiroglu Z, Sengur A. Bitcoin price prediction using machine learning methods. In: *1st International Informatics and Software Engineering Conference: Innovative Technologies for Digital Transformation, IISEC 2019 – Proceedings*; 2019:144-147. doi:10.1109/UBMYK48245.2019.8965445
36. McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. In: *Proceedings – 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*; 2018:339-343. doi:10.1109/PDP2018.2018.00060
37. Li L, Arab A, Liu J, Liu J, Han Z. Bitcoin options pricing using LSTM-based prediction model and blockchain statistics. *Proceedings – 2019 2nd IEEE International Conference on Blockchain, Blockchain*. Vol 2019. IEEE; 2019:67-74. doi:10.1109/Blockchain.2019.00018
38. Saad M, Choi J, Nyang D, Kim J, Mohaisen A. Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions. *IEEE Syst J*. 2020;14(1):321-332. doi:10.1109/JSYST.2019.2927707
39. Thathachar MAL, Sastry PS. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Springer Science & Business Media; 2004.
40. Lee BH, Lee KY. Application of S-model learning automata for multi-objective optimal operation of power systems. *IEE Proc Gen Transm Distrib*. 2005;152(2):295-300.
41. Tsetlin ML. *Automaton Theory and Modeling of Biological Systems*. Vol 102. Academic Press; 1973. http://orton.catie.ac.cr/cgi-bin/wxis.exe/?IsisScript=UACHBC.xis&method=post&formato=2&cantidad=1&expresion=mfn=059666
42. Narendra KS, Thathachar MAL. Learning automata: an introduction. Englewood Cliffs, NJ: Prentice Hall; 1989:1-476.
43. Thathachar MAL, Arvind MT. Solution of Goore game using modules of stochastic learning automata. *J Indian Inst Sci*. 1997;77(1):47-61.
44. Cao YU, Kahng AB, Fukunaga AS. Cooperative mobile robotics: antecedents and directions. *Robot Colonies*. Springer; 1997:7-27.
45. Chen D, Varshney PK. QoS support in wireless sensor networks: a survey. *International Conference on Wireless Networks*. Vol 233. IEEE; 2004:1-7.
46. Rezvanian A, Saghiri AM, Vahidipour SM, Esnaashari M, Meybodi MR. Recent advances in learning automata. *Stud Comput Intell*. 2018;754:1-458. doi:10.1007/978-3-319-72428-7
47. Norman MF. On the linear model with two absorbing barriers. *J Math Psychol*. 1968;5(2):225-241. doi:10.1016/0022-2496(68)90073-4
48. Ameri R, Meybodi MR, Daliri Khomami MM. Cellular Goore game and its application to quality-of-service control in wireless sensor networks. *J Supercomput*. 2022;78:1-48.
49. Zheng Z, Xie S, Dai H-N, Chen X, Wang H. Blockchain challenges and opportunities: a survey. *Int J Web Grid Serv*. 2018;14(4):352-375.
50. Lamport L, Shostak R, Pease M. The byzantine generals problem. In: *Concurrency: The Works of Leslie Lamport*. New York: ACM; 2019:203-226.
51. Castro M, Liskov B. Practical byzantine fault tolerance. *OSDI*. 1999;99(1999):173-186.
52. Hyperledger. *Hyperledger—Open Source Blockchain Technologies*. Hyperledger; 2019. Accessed November 22, 2021. https://www.hyperledger.org/
53. Gini C. Measurement of inequality of incomes. *Econ J*. 1921;31(121):124-126.
54. Sanka AI, Cheung RCC. A systematic review of blockchain scalability: issues, solutions, analysis and future research. *J Netw Comput Appl*. 2021;195:103232.
55. Gobel J, Krzesinski AE. Increased block size and bitcoin blockchain dynamics. In: *2017 27th International Telecommunication Networks and Applications Conference, ITNAC 2017*; 2017:1-6. doi:10.1109/ATNAC.2017.8215367
56. Croman K, Decker C, Eyal I, et al. On scaling decentralized blockchains. *International Conference on Financial Cryptography and Data Security*. Springer; 2016:106-125.
57. Back A, Corallo M, Dashjr L. *Enabling Blockchain Innovations with Pegged Sidechains*. Vol 72; 2014:1-25. http://newspaper23.com/ripped/2014/11/blockstream-com-sidechains.pdf
58. Kaur J, Rani R, Kalra N. A blockchain-based framework for privacy preservation of electronic health records (EHRs). *Trans Emerg Telecommun Technol*. 2022;33(9):e4507.
59. Singh R, Dwivedi AD, Mukkamala RR, Alnumay WS. Privacy-preserving ledger for blockchain and internet of things-enabled cyber-physical systems. *Comput Electr Eng*. 2022;103:108290.

60. Dwivedi AD, Srivastava G, Dhar S, Singh R. A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors*. 2019;19(2):326.
61. Saxena S, Bhushan B, Ahad MA. Blockchain based solutions to secure IoT: background, integration trends and a way forward. *J Netw Comput Appl*. 2021;181:103050.
62. Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery. *ACM Trans Comput Syst*. 2002;20(4):398-461. doi:10.1145/571637.571640
63. Kotla R, Alvisi L, Dahlin M, Clement A, Wong E. Zyzzyva: speculative byzantine fault tolerance. *ACM Trans Comput Syst*. 2009;27(4):45-58. doi:10.1145/1658357.1658358
64. Guerraoui R, Knežević N, Quéma V, Vukolić M. The next 700 BFT protocols. In: *Proceedings of the 5th European Conference on Computer Systems*; 2010:363-376.
65. Lunardi RC, Michelin RA, Neu CV, Nunes HC, Zorzo AF, Kanhere SS. Impact of consensus on appendable-block blockchain for IoT. In: *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*; 2019:228-237.
66. Ayaz F, Sheng Z, Tian D, Guan YL. A proof-of-quality-factor (PoQF)-based blockchain and edge computing for vehicular message dissemination. *IEEE Internet Things J*. 2020;8(4):2468-2482.
67. Liao Z, Cheng S. Rvc: an reputation and voting based blockchain consensus mechanism for edge computing-enabled Iot systems. *J Netw Comput Appl*. 2023;209:103510.
68. Abraham I, Malkhi D, Nayak K, Ren L, Spiegelman A. Solidus: an incentive-compatible cryptocurrency based on permissionless byzantine consensus. *Leibniz Int Proc Informatics, LIPIcs*. 2016;95:1-15.
69. Kogias EK, Jovanovic P, Gailly N, Khoffi I, Gasser L, Ford B. Enhancing bitcoin security and performance with strong consistency via collective signing. In: *25th Usenix Security Symposium (Usenix Security 16)*; 2016:279-296.
70. Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S, Saxena P. A secure sharding protocol for open blockchains. *Proceedings of the ACM Conference on Computer and Communications Security*. Vol 24-28. ACM; 2016:17-30. doi:10.1145/2976749.2978389
71. Jiang Z, Cao Z, Krishnamachari B, Zhou S, Niu Z. SENATE: a permissionless byzantine consensus protocol in wireless networks for real-time internet-of-things applications. *IEEE Internet Things J*. 2020;7(7):6576-6588.
72. Chen P, Han D, Weng T-H, Li K-C, Castiglione A. A novel byzantine fault tolerance consensus for Green IoT with intelligence based on reinforcement. *J Inf Secur Appl*. 2021;59:102821.
73. Mitola JI. *Cognitive Radio. An Integrated Agent Architecture for Software Defined Radio*; 2002.
74. Mitola J, Maguire GQ. Cognitive radio: making software radios more personal. *IEEE Pers Commun*. 1999;6(4):13-18.
75. Haykin S. Cognitive radio: brain-empowered wireless communications. *IEEE J Sel Areas Commun*. 2005;23(2):201-220.
76. Raychaudhuri D, Mandayam NB, Evans JB, Ewy BJ, Seshan S, Steenkiste P. Cognet: an architectural foundation for experimental cognitive radio networks within the future internet. *Proceedings of First ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*. IEEE; 2006:11-16.
77. He J, Peng J, Jiang F, Qin G, Liu W. A distributed Q learning spectrum decision scheme for cognitive radio sensor network. *Int J Distrib Sens Networks*. 2015;11(5):301317.
78. Song Y, Zhang C, Fang Y. Stochastic traffic engineering in multihop cognitive wireless mesh networks. *IEEE Trans Mob Comput*. 2009;9(3):305-316.
79. Sakellari G. The cognitive packet network: a survey. *Comput J*. 2010;53(3):268-279.
80. Saghiri AM, Meybodi MR. An approach for designing cognitive engines in cognitive peer-to-peer networks. *J Netw Comput Appl*. 2016;70:17-40.
81. Thomas RW, Friend DH, DaSilva LA, MacKenzie AB. Cognitive networks. *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*. Springer; 2007:17-41.
82. Wu Q, Ding G, Xu Y, et al. Cognitive internet of things: a new paradigm beyond connection. *IEEE Internet Things J*. 2014;1(2):129-143.
83. Zhang M, Zhao H, Zheng R, Wu Q, Wei W. Cognitive internet of things: concepts and application example. *Int J Comput Sci Issues*. 2012;9(6):151.
84. Amoussou-Guenou Y, Del Pozzo A, Potop-Butucaru M, Tucci-Piergiovanni S. On fairness in committee-based blockchains. *arXiv Prepr. arXiv1910.09786*; 2019.
85. Xing J, Fischer D, Labh N, et al. Talaria: a framework for simulation of permissioned blockchains for logistics and beyond. *arXiv Prepr. arXiv2103.02260*; 2021.
86. *GitHub - Jiali-Xing/Talaria*. Accessed November 23, 2021. https://github.com/Jiali-Xing/Talaria
87. Alharby M, van Moorsel A. Blocksim: a simulation framework for blockchain systems. *ACM Sigmetrics Perform Eval Rev*. 2019;46(3):135-138.
88. Mohsenzadeh A, Bidgoly AJ, Farjami Y. A novel reputation-based consensus framework (RCF) in distributed ledger technology. *Comput Commun*. 2022;190:126-144.

How to cite this article: Ameri R, Meybodi MR. Cognitive blockchain and its application to optimize performance in blockchain systems. *Trans Emerging Tel Tech*. 2024;35(7):e5009. doi: 10.1002/ett.5009