A Framework for Cognitive Defense in Blockchain: A Case Study on AI-Based Protection Against Selfish Mining Attacks

Ali Nikhalat-Jahromi, Ali Mohammad Saghiri, and Mohammad Reza Meybodi

Department of Computer Engineering,
Amirkabir University of Technology, Tehran, Iran
{ali.nikhalat,a_m_saghiri,mmeybodi}@aut.ac.ir

Abstract. Blockchain is a revolutionary protocol that enables transactions to be both anonymous and secure through a tamper-proof public ledger. Despite its significant potential, blockchain faces unresolved security challenges. Blockchain systems are highly dynamic and large-scale. Therefore, many management problems, such as defense mechanisms against a wide range of attacks, cannot be handled by humans because human reaction time is insufficient for many management tasks in blockchain systems. In other words, human reasoning is required in many situations, but we cannot use real humans as managers or system admins. Meanwhile, cognitive systems, designed to mimic human thinking processes through digitalized models, have seen widespread adoption. The core component, the cognitive engine, is responsible for implementing these functionalities. This paper proposes a novel framework that integrates cognitive systems into blockchain to defend against attacks. To our knowledge, no existing framework leverages cognitive systems for blockchain security. We specifically design a reinforcement learning (RL)-based defense mechanism to counter selfish mining attacks based on the cognitive defense framework. Simulation results demonstrate that our proposed cognitive framework significantly enhances blockchain security.

Keywords: Blockchain, Cognitive System, Defense Mechanism, Reinforcement Learning, Selfish Mining

1 Introduction

Blockchain [1–5] has emerged as a groundbreaking protocol that ensures both anonymity and security in transactions through a decentralized and tamper-proof public ledger. Its application spans various domains [6], from cryptocurrencies to supply chain management [7] and health care systems [8], promising to revolutionize how transactions are recorded and verified. However, despite the enormous potential and opportunities it offers, blockchain technology is not without its challenges. One of the most pressing issues is the vulnerability of blockchain systems to various types of attacks that exploit their security weaknesses [9–11]. These vulnerabilities, if not addressed, can undermine the integrity

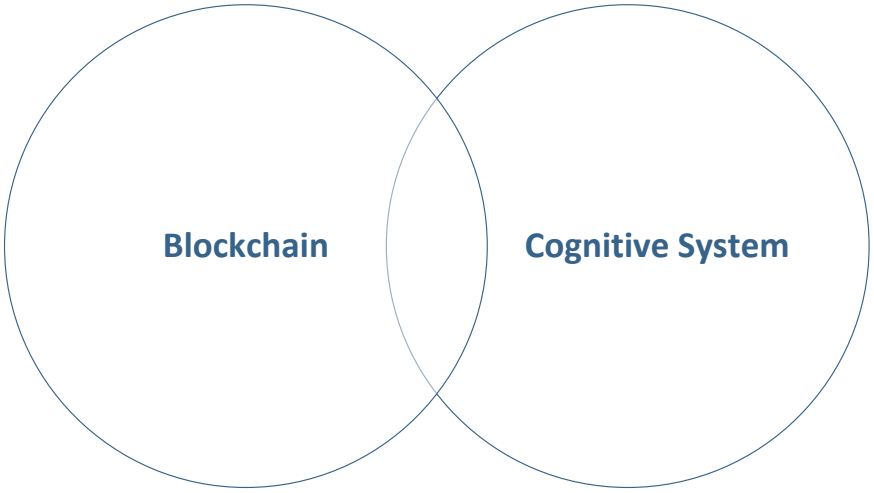


Fig. 1: A combination of Blockchain and Cognitive System

and trust that are the bedrocks of blockchain systems. It should be noted that because of the large scale and dynamic nature of blockchain systems, many management operations, such as defense mechanisms and other vital network management algorithms, should be implemented in a fully autonomous manner without any human intervention. In other words, since many management operations, like defense mechanisms against a wide range of attacks, cannot be handled by humans because human reaction time is insufficient for many management tasks, we need to design intelligent systems that act like humans in critical situations.

Cognitive systems [12–14], on the other hand, represent a significant advancement in artificial intelligence, aimed at creating digital models capable of emulating human cognitive processes. These systems are designed to think, learn, and adapt in ways similar to the human brain, providing intelligent solutions to complex problems. The cognitive engine, which forms the core of these systems, implements the functionalities that enable such advanced capabilities. Recently, these systems have received much attention due to the emergence of low-cost generative artificial intelligence services that can provide human-like analyses for a wide range of tasks.

In this context, integrating cognitive systems with blockchain technology (Figure. 1) presents a promising avenue for enhancing blockchain security[15]. This paper proposes a novel framework that leverages cognitive systems to defend against blockchain attacks, with a specific focus on selfish mining—a prevalent and particularly challenging attack. Selfish mining undermines the blockchain's integrity by allowing malicious miners to gain disproportionate rewards, thereby threatening the fairness and stability of the network.

To demonstrate the applicability of our cognitive framework, we have developed a reinforcement learning (RL)-based [16, 17] defense mechanism that is firmly rooted in our proposed cognitive framework. Reinforcement learning, a branch of machine learning, is well-suited for this purpose due to its ability to learn optimal strategies through interaction with an environment. By employing an RL-based approach, our framework can dynamically adapt and respond to selfish mining [18, 19] attempts, significantly bolstering blockchain security.

To our knowledge, this is the first framework that applies cognitive systems to blockchain security. Our simulation results demonstrate the efficacy of this approach, showing significant improvements in defending against selfish mining attacks. This paper aims to contribute to the ongoing efforts to secure blockchain technology, providing a robust and adaptive defense mechanism inspired by cognitive processes.

The structure of the paper is organized as follows: Section 2 provides the necessary preliminaries and background information. Section 3 presents the details of the proposed framework. A case study of the proposed framework is discussed in Section 4. In Section 5, we discuss the paper's limitations, strengths, and challenges. Finally, Section 6 highlights the key findings and contributions of the paper.

2 Preliminaries

In this section, we present the essential background information on the proposed framework and its application in designing a defense against selfish mining attacks. This includes a detailed overview of key aspects of blockchain technology, cognitive systems, and Q-Learning.

2.1 Blockchain

Blockchain [1, 20] is a decentralized and distributed ledger system that securely records transactions across a network of computers. It was originally conceptualized as the backbone of Bitcoin[1, 21], the first cryptocurrency, by an unknown person or group of people using the pseudonym Satoshi Nakamoto[1]. The core innovation of blockchain lies in its ability to provide a secure, transparent, and immutable record of transactions without the need for a central authority. Each transaction is grouped into a "block," which is then cryptographically linked to the previous block, forming a "chain" of blocks. This chaining process ensures that once a transaction is recorded, it cannot be altered or deleted without changing all subsequent blocks, making the blockchain tamper-proof and highly secure[22–24].

The decentralized nature of blockchain technology also enhances its resilience and trustworthiness. Instead of relying on a single centralized server, blockchain networks are maintained by a distributed network of nodes, each of which holds a copy of the entire ledger. This decentralization prevents any single point of failure and makes the system more robust against attacks and malfunctions. Consensus mechanisms, such as Proof-of-Work (PoW) [25–28] and Proof-of-Stake

(PoS)[29–31], are employed to validate transactions and ensure agreement among the nodes. These mechanisms not only secure the network but also incentivize participants to act honestly. As a result, blockchain technology has found applications beyond cryptocurrencies, including supply chain management, healthcare, finance, and more, revolutionizing how data integrity and security are managed in various industries.

Since our focus is on PoW-based blockchains, we go deeper into understanding the consensus mechanisms and potential attacks associated with this type of system. By expanding our knowledge in this area, we aim to enhance the security measures specifically tailored for PoW-based blockchains.

Mining Process The state of the blockchain is modified through the execution of transactions, which are then grouped into blocks and appended to the blockchain. A typical block in the blockchain consists of two main components: the header and the body[32]. The block header contains critical information, including the hash of the previous block, the hash of the current block, the Merkle root representing all transactions within the block, and a nonce value. In contrast, the block body contains the transactions selected by the miner to be included in the block [33, 24].

To validate a block, miners must solve a cryptographic puzzle[18, 34]. This involves finding the correct nonce value to be placed in the block header, resulting in a block hash that is lower than the predefined difficulty target. The block difficulty is dynamically adjusted to maintain an average block generation rate of approximately one block every ten minutes. Upon successfully solving the puzzle and adding the mined block to the longest chain, the miner is rewarded with newly created Bitcoins and the transaction fees from the included transactions.

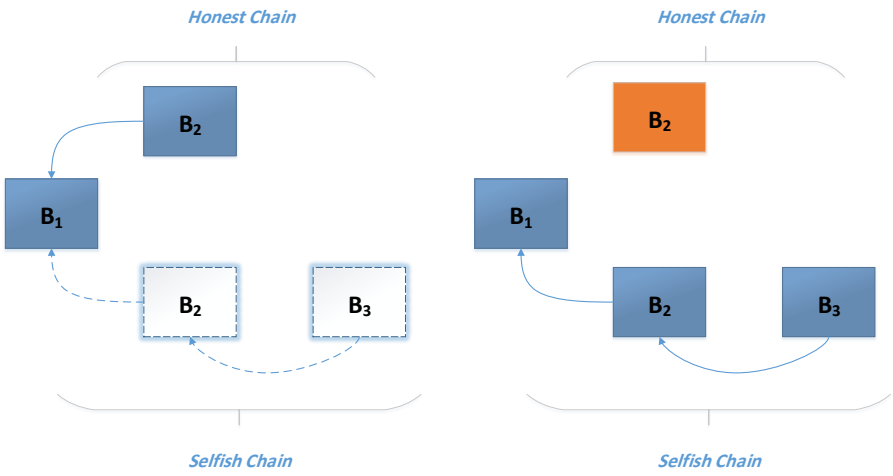
The likelihood of mining a new block is directly proportional to the computational resources employed in solving the puzzle[18, 34]. However, the mining process is inherently unpredictable, leading to significant variance in the time intervals between mining events for individual miners. Consequently, many miners join mining pools, where they collaborate to collectively mine each block and share the rewards whenever a block is successfully mined. Although participating in a mining pool does not change a miner's expected revenue, it reduces revenue variance and provides a more predictable monthly income[18, 34].

Threats Blockchain faces a variety of security threats [35] that can be categorized into several key areas. ***Double Spending*** threats involve the risk of the same digital currency being spent more than once, undermining the integrity of transactions [36, 37]. ***Mining and Pool*** threats include attacks such as selfish mining (thoroughly discussed in the next subsection) and 51% attacks, where a single entity or group of miners gains control over the majority of the network's hash rate, potentially allowing them to manipulate the blockchain[38, 39]. ***Wallet*** threats pertain to the security of digital wallets, which can be compromised through malware, phishing attacks, or weak security practices, leading to the theft of private keys and the loss of funds[40, 41]. ***Network*** threats encompass a

range of vulnerabilities, including Distributed Denial of Service (DDoS) attacks, which can disrupt the network's availability, and Eclipse attacks, where an attacker isolates and manipulates a node's view of the blockchain[42, 43]. Finally, ***Smart Contract*** threats arise from vulnerabilities within the code of smart contracts themselves, which can be exploited to trigger unintended behaviors or unauthorized transactions, as seen in high-profile incidents like the decentralized autonomous organization(DAO) hack. Addressing these threats requires a comprehensive approach that includes robust security practices, continuous monitoring, and the development of advanced defensive mechanisms[44, 45].

Selfish Mining Bitcoin's documentation [1] offers a detailed explanation of the block release process following successful mining. When a miner mines a new valid block, it is expected to share it with the network immediately. However, Eyal and Sirer introduced the concept of "selfish mining" [18], where miners deviate from Bitcoin's standard mining rules to unfairly boost their revenue. This strategy, known as "SM1," is the first and most widely recognized form of selfish mining.

In the SM1 strategy[18], miners engaged in selfish mining withhold their newly mined blocks instead of broadcasting them immediately. This creates a blockchain fork, with one fork being visible to all network participants and the other remaining hidden, known only to the selfish miners. Honest miners, unaware of the hidden chain, continue to work on the public chain. When the selfish miners reveal their hidden chain, the honest chain gets discarded, resulting in an



(a) Development of private chain using selfish miners(white blocks)

(b) After publishing private chain and adoption of private chain

Fig. 2: A scenario of the selfish mining attack [64]

increased relative revenue for the selfish miners and incentivizing other miners to adopt similar selfish tactics.

To illustrate the SM1 strategy, consider a simplified example (depicted in Figure. 2) where selfish miners keep their mining activities secret. If these miners are two blocks ahead of the honest miners (Figure. 2a), they will reveal their hidden chain when the honest miners discover a new block. This results in the honest miners' efforts being wasted, as the network adopts the selfish miners' previously hidden blocks (Figure. 2b).

Beyond the SM1 strategy, new selfish mining strategies have been explored. Sapirshtein et al. [34] used a Markov Decision Process (MDP) to study the minimum resource fraction required for a profitable selfish mining attack, known as the profit threshold. They established a bound to ensure system security against such attacks and modified the protocol to assess its vulnerability to selfish mining by calculating the optimal attack under various conditions. Their findings indicated scenarios where selfish miners could maintain control of a private chain even when the public chain appears longer.

Recent research has further explored selfish mining, particularly through the use of machine learning techniques[32, 46]. Many studies have employed reinforcement learning to optimize the effectiveness of selfish mining attacks[34]. For example, [47, 48] developed an advanced MDP-based solution using reinforcement learning algorithms to maximize mining revenue. They proposed a deep reinforcement learning framework to analyze the behavior of rational miners under different conditions and established an upper bound for the security threshold of proof-of-work-based blockchains.

2.2 Cognitive System

Cognitive systems represent an advanced area of artificial intelligence designed to simulate human-like thinking and decision-making processes. These systems are engineered to emulate human cognitive functions such as learning, reasoning, problem-solving, and adapting to new situations. Unlike traditional AI models, which often rely on predefined algorithms and static rules, cognitive systems utilize dynamic learning mechanisms to continuously improve their performance based on new data and experiences. This ability to adapt and evolve allows cognitive systems to handle complex, unstructured tasks and make decisions that closely mirror human thought processes [49–51].

At the core of cognitive systems is the cognitive engine, which integrates various AI technologies, including machine learning, natural language processing, and neural networks, to create a unified model of cognition. These systems are capable of interpreting and responding to inputs in a manner akin to human reasoning, enabling them to perform tasks such as understanding context, recognizing patterns, and generating insights[52–54]. By leveraging advanced algorithms and computational techniques, cognitive systems can analyze vast amounts of data, identify underlying patterns, and make predictions or recommendations[55]. This capacity for sophisticated analysis and adaptive learning makes cognitive systems highly valuable in applications ranging from customer

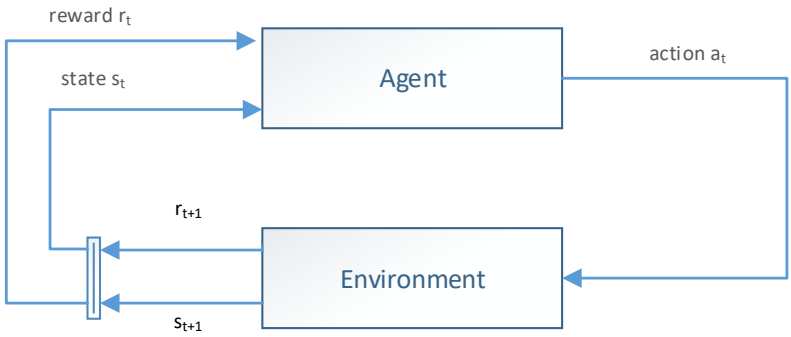


Fig. 3: Interaction between Q-Learning agent and environment [64]

service and healthcare [56, 57] to financial forecasting and autonomous systems [58].

2.3 Q-Learning

Q-Learning [16, 59, 60] is a prominent reinforcement learning algorithm that has garnered substantial interest in the machine learning community. This model-free algorithm allows an agent to determine the best actions to take in a given environment through a process of trial and error.

Fundamentally, Q-Learning operates by exploring and exploiting a state-action space. The agent engages with the environment by executing actions and receiving corresponding rewards or penalties (Figure. 3). The primary objective is to optimize the cumulative reward accumulated over time.

The algorithm uses a Q-table, which is a matrix that records the expected cumulative rewards for each state-action pair. Initially, the Q-values in the table are either set to arbitrary values or zero. As the agent interacts with the environment, it updates these Q-values based on the rewards received and the new information learned. The updating rule for Q-learning is given by the following equations:

$$Q(s,a) \leftarrow (1-\alpha_q)\, Q(s,a) + \alpha_q\, (r + \gamma_q\, Q(s',a')) \tag{1}$$

Where:

- $Q(s,a)$ denotes the Q-value associated with state s and action a.
- α_q represents the learning rate, which dictates how much new information affects the Q-values.
- r is the immediate reward obtained after performing action a in state s.

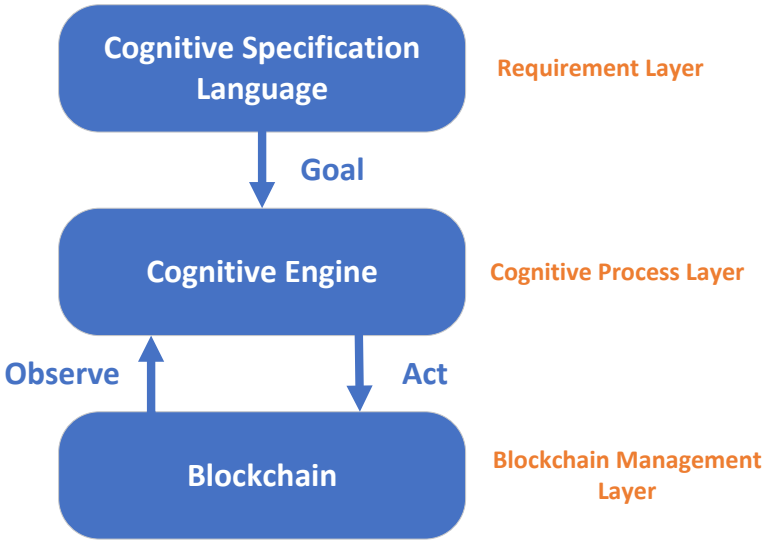


Fig. 4: Architecture of the Proposed Framework

- γ_q is the discount factor, which determines the value of future rewards.
- s' refers to the state resulting from taking action a in state s.
- a' is the action that yields the highest Q-value in state s'.

This updating rule adjusts the Q-value by combining the previous Q-value with new information derived from the received reward and the maximum Q-value of the subsequent state. The learning rate regulates the influence of new information relative to the existing knowledge.

By continuously applying this updating rule throughout the learning process, the Q-values gradually converge to their optimal values. This enables the agent to make more informed decisions and effectively maximize its cumulative reward over time.

3 The Proposed Framework

In this section, we introduce a defense framework for blockchain systems based on cognitive systems. The structure of the proposed framework is illustrated in Figure 4. This framework comprises three layers: the *Requirement Layer*, the *Cognitive Process Layer*, and the *Blockchain Management Layer*. The requirement layer outlines the primary objectives of the system and communicates them to the cognitive process layer. The cognitive process layer is responsible for the continuous management of the system, addressing the challenges posed by the

distributed, dynamic, and large-scale nature of blockchain, which are not adequately addressed by traditional or non-intelligent solutions. The details of each layer are provided in the remainder of this section. It is worth noting that some concepts in this framework are adapted from the cognitive networks framework introduced by [61–63].

3.1 Requirement Layer

In the Requirement Layer, the goals and behaviors of the nodes in the blockchain are defined using a *Cognitive Specification Language (CSL)*. This language is employed to populate a configuration file, which is subsequently shared among nodes in accordance with the structure of the blockchain network. It is important to note that altering the goals in the requirement layer results in changes to the optimization functions within the cognitive process layer. The system's goals, particularly those related to security, are established through commands received via voice, command line, or any other direct interaction between users and the system. The specific features of these goals can be determined by the following elements:

- Blockchain Type (Permissioned or Permissionless)
- Consensus Type
- Security Flaw
- Attack Type

Additionally, various approaches have been suggested to account for the distributed nature of blockchain in sharing configuration files:

- **Centralized:** In this approach, the latest version of the configuration files is stored on a single, well-known server.
- **Semi-centralized:** This approach involves multiple servers that are responsible for managing the configuration files.
- **Fully-distributed:** Here, each node in the blockchain network periodically downloads the latest version of the configuration files from its neighboring nodes.

3.2 Blockchain Management Layer

This layer supplies the cognitive process layer with necessary information by observing various units and then acting upon the manageable elements of the system. By continuously monitoring and assessing these units, this layer ensures that any potential security threats or anomalies are quickly identified and addressed. This proactive approach is crucial for maintaining the integrity and security of the blockchain. The structure of this layer is depicted in Fig. 5.

This layer consists of four manageable units which are defined as below:

Fig. 5: Blockchain management layer architecture

1. **Blockchain Unit:** This unit is responsible for monitoring and controlling various parameters of the blockchain based on the selected consensus algorithm, as specified in the configuration file. The primary parameters managed by this unit include blocks and their validation within the main chain. Blocks are examined starting from their headers, which typically contain information such as the timestamp, version, and hash of the previous block, and extending to the body, which includes the transactions. Ensuring the integrity and security of these parameters is paramount, as any unauthorized alterations or inconsistencies could compromise the entire blockchain system. Any change in these parameters triggers an action from this unit, prompting it to call the cognitive process layer for the necessary commands to execute the most effective response.
2. **Attack Detector Unit:** The attack detector unit in the blockchain plays a crucial role in maintaining the blockchain security by continuously monitoring for potential threats and vulnerabilities. This unit is designed to detect malicious activities such as double-spending, sybil attacks, and selfish mining, leveraging advanced algorithms and real-time data analysis. By analyzing patterns in transaction data, network behavior, and block generation, the attack detector can identify anomalies that suggest security breaches. Upon detecting an attack, it triggers an alert to the cognitive process layer, which then evaluates the threat and implements appropriate countermeasures. This proactive approach not only enhances the resilience of the blockchain against various types of attacks but also ensures the integrity and trustworthiness of the entire system.
3. **Fork Manager Unit:** The fork manager unit in the blockchain is essential for handling and mitigating the security implications of chain splits, commonly known as forks. This unit's primary responsibility is to monitor the blockchain for any occurrences of forks, whether they arise from network latency, intentional attacks, or protocol updates. When a fork is detected, the fork manager unit assesses the competing chains by evaluating parameters such as the length of each branch and the weight of the blocks within those branches. By leveraging cognitive algorithms, the fork manager determines the optimal chain to continue, ensuring the security and stability of the blockchain. The fork manager unit works in close conjunction with the cognitive process layer, which provides the intelligence and learning capabilities required for optimal decision-making.

4. **P2P Communication Unit:** This unit is tasked with ensuring the security of communication and data exchange among the nodes within the blockchain. It continuously monitors and verifies the integrity, confidentiality, and authenticity of all interactions, safeguarding against potential security threats and vulnerabilities.
5. **Wallet Unit:** The wallet unit in a blockchain system is pivotal for managing the security of user transactions and data storage. It securely generates, stores, and manages cryptographic keys, ensuring private keys remain confidential and public keys facilitate secure transaction addresses. This unit verifies digital signatures and safeguards transaction integrity, employing encryption to protect sensitive data exchanges between nodes. It also implements robust authentication mechanisms, such as multi-factor authentication and biometric verification, to prevent unauthorized access. By continuously monitoring for suspicious activities and potential breaches, the wallet unit ensures the legitimacy of transactions and protects users' assets against fraud and cyber-attacks, thereby reinforcing the blockchain's overall security framework.
6. **Smart Contract Unit:** The smart contract Unit in a blockchain system is essential for ensuring the secure execution and management of self-executing contracts. This unit oversees the deployment and execution of smart contracts, verifying their integrity and correctness before they are added to the blockchain. It employs stringent validation mechanisms to ensure that contract code is free from vulnerabilities and adheres to predefined security standards. By doing so, it prevents malicious code and logic flaws that could be exploited by attackers. The smart contract unit also implements encryption and access control measures to protect sensitive data within contracts, ensuring that only authorized parties can interact with the contract. Additionally, it continuously monitors the execution of smart contracts to detect and respond to any abnormal or suspicious activities, thereby maintaining the integrity and security of automated transactions within the blockchain ecosystem.

3.3 Cognitive Process Layer

This layer functions as the brain of the system. Initially, it acquires the system's goals from the configuration files in the requirement layer. It continuously observes security-related changes through parameters from the blockchain management layer. Utilizing cognitive algorithms, which can include various AI algorithms such as large language models (LLMs) [65–67], it makes critical decisions to maintain the security of the blockchain. These decisions are then conveyed to the blockchain management layer.

Based on the received goals from the requirement layer, it can design various types of cognitive engines, such as:

- An engine for monitoring the type of consensus mechanism.
- An engine for monitoring predefined and newly designed security attacks based on the consensus mechanism.

- An engine for monitoring the process of mining blocks.
- An engine dedicated to detecting and monitoring attacks on the blockchain.
- An engine for tracking and managing the creation of forks.
- An engine for monitoring P2P communications.
- An engine for monitoring the security of wallets.
- An engine for monitoring the security of smart contracts.

After designing the required engines, these engines should make decisions to enhance the security of the blockchain, such as:

- Decisions about choosing the correct fork among the created forks.
- Decisions about the validity of blocks in the fork.
- Decisions about the validity of transactions in the block.
- Decisions about how to respond to detected attacks.
- Decisions about the quality and security of P2P communications.
- Decisions about enhancing wallet security.
- Decisions about strengthening smart contract security.

4 Case Study

In this section, we introduce Q-Defense[64], a cognitive defense mechanism designed within the proposed framework to counteract selfish mining attacks. We start by outlining the system model and providing essential definitions to create a comprehensive foundation for our proposed algorithm. Following this, we detail the defense algorithm in a step-by-step format, emphasizing its critical components and operational mechanisms.

4.1 System Model

A thorough understanding of the target environment is critical for the effective proposition of any algorithm. Hence, it is imperative to establish a detailed model of our blockchain system, particularly from the perspective of honest miners, to analyze the attack scenario accurately.

First, we configure the requirement layer with the following settings:

- The blockchain type is permissionless.
- The consensus protocol is proof-of-work.
- The attack under consideration is selfish mining.

Consider a network composed of two groups of miners. The first group consists of selfish miners who deviate from the standard mining rules, denoted by α percent of the total mining power. The second group comprises rational miners who strictly follow the mining rules and hold $1 - \alpha$ percent of the total mining power.

For our analysis, we disregard the impact of network propagation delay, assuming that miners strive to propagate blocks swiftly to minimize disruptions in the subsequent mining process.

The blocks in the network are organized in a tree structure to form a chain. When two blocks share the same previous block hash, a fork is created. However, our analysis focuses solely on the type of fork resulting from selfish mining. This fork leads to two chains: the selfish chain and the honest chain.

When a selfish miner learns that an honest miner has discovered a new block, it will attempt to replace its private block. We introduce the parameter γ as the advertisement factor, representing the proportion of computing power required for nodes to accept the selfish miner's block over the honest miner's block. In terms of block height, when the Bitcoin network reaches height h, the selfish miner's block at height h has a probability of $\gamma(1-\alpha)$ of being accepted.

4.2 Required Definitions

This subsection provides essential prerequisite definitions to facilitate a comprehensive understanding of the proposed algorithm. These definitions pertain to the characteristics of each branch of the fork resulting from selfish mining. The following definitions are provided:

- **Fork Branch**: A fork branch is one of the competing chains in the network, created as a result of selfish mining. It comprises a sequence of blocks starting from the common ancestor block up to the current block in that branch.
- **Branch Length (L)**: The length of a branch, denoted by L, indicates the number of blocks it contains. This parameter is crucial for assessing the size and extent of competing chains resulting from selfish mining.
- **Branch Weight (W)**: The weight of a branch, denoted by W, is determined by comparing blocks within that branch to blocks of the same height in other branches. The branch with the most recent creation time increases its weight by one, encompassing all blocks from the first to the last in that branch.
- **Fail-Safe Parameter (K)**: The fail-safe parameter K assists the miner in selecting a branch based on either its length (L) or its weight (W). If the length of a branch in the fork exceeds the others by at least K, that branch is chosen. Otherwise, the weight parameter W is used to determine the preferred branch.
- **Decision-Making Time (τ)**: The decision-making time, denoted by τ, represents the time taken by a miner to check for the presence of forks. If a fork is detected, the miner must select one of the branches, considering the parameter K.
- **Time Windows Parameter (θ)**: The time window parameter θ is used to determine the next value for K using the Q-learning agent. It is composed of multiple τ values.

4.3 Algorithm

With the necessary definitions established, we can now outline our algorithm within the cognitive process layer to defend against selfish mining attacks. The proposed algorithm, which operates by observing parameters from the blockchain layer, consists of the following steps:

1. Calculate the length L of each branch.
2. Calculate the weight W of each branch.
3. Evaluate branch lengths: If the difference between the length of the longest branch and the length of the second-longest branch exceeds K, select the longest branch. Otherwise, choose the heaviest branch based on the calculated W parameter.
4. Q-Learning adjustment: When the decision-making time τ elapses, the Q-Learning agent determines the next value for K. Typically, K oscillates between $K_{\min}$ and $K_{\max}$. The Q-Learning agent operates with one state and three actions: 1-**Grow**, which increases K by one; 2-**Stop**, which keeps K unchanged; and 3-**Shrink**, which decreases K by one.
5. Feedback and reward calculation: When the time window parameter θ elapses, the Q-Learning agent receives feedback from the environment. A virtual environment is designed for the Q-Learning agent to provide information about its decisions. The reward is calculated by dividing the number of weight-based decisions by the total number of decisions, which includes both height-based decisions and weight-based decisions. The equation below illustrates the calculation of the reward parameter r for the Q-Learning agent:

$$r = \frac{Number; of; Weight; Decisions}{Total; Number; of; Decisions} \tag{2}$$

By following these steps, the algorithm leverages cognitive processes and Q-Learning to dynamically adjust and optimize the defense against selfish mining, ensuring a robust and adaptive blockchain security mechanism [64].

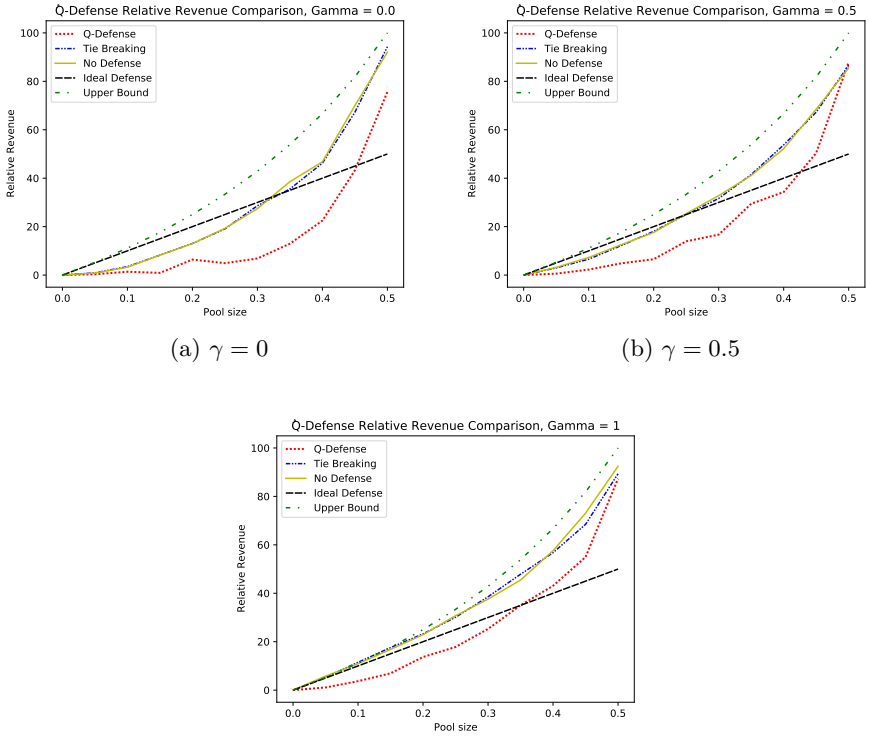
4.4 Evaluation

To evaluate the proposed algorithm, we conducted a series of experiments to assess its performance from multiple perspectives. In these experiments, we modeled a network with two types of miners: selfish miners and rational miners. The mining process in our simulations followed a Poisson process, where a new block is mined with a probability of α by the selfish miners and a probability of $1 - \alpha$ by the rational miners.

We further enhanced the selfish miner's behavior to function within a learning-based mining environment, allowing us to observe the dynamics of the proposed algorithm in a more realistic setting.

To meet our objectives, we ran simulations involving a total of 10,000 blocks. The fail-safe parameter K was allowed to vary within the range of $K_{\min} = 1$ and $K_{\max} = 3$. This range was selected to enable Q-Learning agents to reach consensus promptly, given the time-intensive nature of selecting K. The Q-Learning agents were configured with a learning rate of 0.1 and a discount factor of 0.75, optimizing their learning and decision-making processes within the proposed algorithm.

We compared the effectiveness of the proposed defense against a tie-breaking defense mechanism. In our experiments, τ was set to the mining time of five



(a) $\gamma = 0$ (b) $\gamma = 0.5$

(c) $\gamma = 1$

Fig. 6: Comparing Q-defense with tie-breaking using various values of α and γ [64]

blocks, and θ was composed of ten τ periods. These parameters were chosen to allow Q-Learning agents sufficient time to accurately explore and exploit their environment. The results are depicted in Figure 6.

The experimental results highlight several key findings. Firstly, they demonstrate the superiority of the proposed defense over the tie-breaking mechanism in terms of reducing the relative revenue of selfish miners. This suggests that the proposed defense effectively mitigates the gains of selfish miners. Notably, at $\gamma = 1$, where selfish miners have maximum power, the proposed defense excels in decreasing the relative revenue. These outcomes provide compelling evidence of the efficacy of Q-Learning in complex environments like blockchain.

Furthermore, we examined another critical metric: the lower bound threshold for initiating an attack. The results reveal that the proposed defense, utilizing Q-Learning, significantly raises this threshold from 0.25 to approximately 0.4. This indicates that the proposed defense enhances the security and stability of the

blockchain network by making it more challenging for selfish miners to execute successful attacks [64].

5 Discussion

The integration of cognitive systems with blockchain technology presents both substantial challenges and promising opportunities for enhancing security. One primary challenge lies in the inherent complexity of cognitive systems, which require advanced algorithms and substantial computational resources to mimic human-like decision-making processes. Implementing these systems within the blockchain framework necessitates seamless interaction between cognitive engines and blockchain protocols, ensuring real-time monitoring and response to security threats. Additionally, the decentralized nature of blockchain poses a significant hurdle, as cognitive systems must operate effectively across distributed networks without compromising the speed and efficiency of transaction processing. Despite these challenges, the potential benefits of integrating cognitive systems into blockchain security are immense, providing a robust and adaptive defense mechanism against evolving cyber threats.

The opportunities presented by this integration are manifold. Cognitive systems, through their ability to learn and adapt, offer a dynamic approach to blockchain security that traditional methods lack. By incorporating reinforcement learning-based mechanisms, as demonstrated in our proposed framework, cognitive systems can continuously improve their defense strategies against specific attacks such as selfish mining. This adaptability ensures that the blockchain network remains resilient against new and sophisticated attacks that might emerge over time. Furthermore, the cognitive framework can enhance decision-making processes within the blockchain, optimizing parameters like consensus protocols and transaction validation in response to observed threats. This proactive and intelligent approach to security not only fortifies the blockchain against existing vulnerabilities but also anticipates and mitigates future risks.

Moreover, the integration of cognitive systems into blockchain technology opens up new avenues for research and development. The cross-disciplinary nature of this endeavor, combining elements of artificial intelligence, machine learning, and blockchain technology, invites innovative solutions and collaborative efforts from diverse fields. Future research could explore the application of different AI models, such as large language models, in enhancing the cognitive capabilities of blockchain systems. Additionally, this integration could lead to the development of more advanced and scalable cognitive engines, capable of handling the increased complexity and scale of modern blockchain networks. By addressing the current limitations and exploring these opportunities, the integration of cognitive systems and blockchain technology promises to revolutionize the security landscape, creating a more secure and resilient digital ecosystem.

6 Conclusion and Future Research

In this paper, we introduced a novel framework for enhancing the security of blockchain systems using cognitive systems. Our framework leverages the cognitive engine within the cognitive process layer to monitor and analyze the status of the blockchain layer. Acting as a sophisticated decision maker, the cognitive engine aids in defending the blockchain layer against various attacks that can compromise elements such as the blockchain itself, P2P networks, wallets, or smart contracts. The framework's versatility allows it to be applied across diverse blockchain systems, regardless of their consensus mechanisms or network topologies.

To demonstrate the effectiveness of our proposed framework, we focused on the selfish mining attack as a representative threat. Utilizing the cognitive framework, we developed a reinforcement learning-based engine to counteract this specific attack, highlighting the framework's adaptability and robustness. Our results indicate that the cognitive approach significantly enhances the blockchain's resilience to such adversarial actions.

Future work can expand on this foundation by incorporating more detailed mathematical analyses, exploring defenses against a wider array of attacks, and integrating large language models as decision makers within cognitive engines. These advancements could further solidify the role of cognitive systems in fortifying blockchain security, paving the way for more intelligent and adaptive defense mechanisms.

References

1. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. (2008)
2. Zheng, Z., Xie, S., Dai, H., Chen, X. & Wang, H. Blockchain challenges and opportunities: A survey. *International Journal Of Web And Grid Services.* **14**, 352-375 (2018)
3. Ressi, D., Romanello, R., Piazza, C. & Rossi, S. AI-enhanced blockchain technology: A review of advancements and opportunities. *Journal Of Network And Computer Applications.* pp. 103858 (2024)
4. Revolution, B. How the Technology Behind Bitcoin is Changing Money. *Business And The World, Page.* **324** (2016)
5. Bennet, D., Maria, L., Sanjaya, Y. & Zahra, A. Blockchain technology: Revolutionizing transactions in the digital age. *ADI Journal On Recent Innovation.* **5**, 192-199 (2024)
6. Rani, P., Sharma, P. & Gupta, I. Toward a greener future: A survey on sustainable blockchain applications and impact. *Journal Of Environmental Management.* **354** pp. 120273 (2024)
7. Ray, R., Chowdhury, F. & Hasan, M. Blockchain Applications in Retail Cybersecurity: Enhancing Supply Chain Integrity, Secure Transactions, and Data Protection. *Journal Of Business And Management Studies.* **6**, 206-214 (2024)
8. Al-Nbhany, W., Zahary, A. & Al-Shargabi, A. Blockchain-IoT healthcare applications and trends: a review. *IEEE Access.* (2024)

9. Dwivedi, K., Agrawal, A., Bhatia, A. & Tiwari, K. A Novel Classification of Attacks on Blockchain Layers: Vulnerabilities, Attacks, Mitigations, and Research Directions. *ArXiv Preprint ArXiv:2404.18090*. (2024)
10. Santhosh, A. & Subramanian, N. Classify Attacks Based on Blockchain Components. *2024 12th International Symposium On Digital Forensics And Security (ISDFS)*. pp. 1-6 (2024)
11. Guru, A., Mohanta, B., Mohapatra, H., Al-Turjman, F., Altrjman, C. & Yadav, A. A survey on consensus protocols and attacks on blockchain technology. *Applied Sciences*. **13**, 2604 (2023)
12. Giuliano, A., Hilal, W., Alsadi, N., Gadsden, S. & Yawney, J. A Review of Cognitive Dynamic Systems and Cognitive IoT. *2022 IEEE International IOT, Electronics And Mechatronics Conference (IEMTRONICS)*. pp. 1-7 (2022)
13. M., S., Murugappan, A. & T., M. Cognitive computing technological trends and future research directions in healthcare — A systematic literature review. *Artificial Intelligence In Medicine*. **138** pp. 102513 (2023), https://www.sciencedirect.com/science/article/pii/S0933365723000271
14. Khasawneh, M., Azab, A., Alrabaee, S., Sakkal, H. & Bakhit, H. Convergence of IoT and cognitive radio networks: A survey of applications, techniques, and challenges. *IEEE Access*. **11** pp. 71097-71112 (2023)
15. Lin, R., Li, F., Wang, J., Hu, J., Zhang, Z. & Wu, L. A blockchain-based method to defend against massive SSDF attacks in cognitive Internet of Vehicles. *IEEE Transactions On Vehicular Technology*. (2023)
16. Andrew, B. & Richard S, S. Reinforcement Learning: An Introduction. (The MIT Press,2018)
17. Shakya, A., Pillai, G. & Chakrabarty, S. Reinforcement learning algorithms: A brief survey. *Expert Systems With Applications*. **231** pp. 120495 (2023)
18. Eyal, I. & Sirer, E. Majority is not enough: Bitcoin mining is vulnerable. *Communications Of The ACM*. **61**, 95-102 (2018)
19. Bai, Q., Xu, Y., Liu, N. & Wang, X. Blockchain mining with multiple selfish miners. *IEEE Transactions On Information Forensics And Security*. **18** pp. 3116-3131 (2023)
20. Bhutta, M., Khwaja, A., Nadeem, A., Ahmad, H., Khan, M., Hanif, M., Song, H., Alshamari, M. & Cao, Y. A survey on blockchain technology: Evolution, architecture and security. *Ieee Access*. **9** pp. 61048-61073 (2021)
21. Zaghloul, E., Li, T., Mutka, M. & Ren, J. Bitcoin and blockchain: Security and privacy. *IEEE Internet Of Things Journal*. **7**, 10288-10313 (2020)
22. Badertscher, C., Maurer, U., Tschudi, D. & Zikas, V. Bitcoin as a transaction ledger: A composable treatment. *Journal Of Cryptology*. **37**, 18 (2024)
23. Shahsavari, Y., Zhang, K. & Talhi, C. A theoretical model for block propagation analysis in bitcoin network. *IEEE Transactions On Engineering Management*. **69**, 1459-1476 (2020)
24. Antonopoulos, A. & Harding, D. Mastering bitcoin. (" O'Reilly Media, Inc.",2023)
25. Garay, J., Kiayias, A. & Shen, Y. Proof-of-work-based consensus in expected-constant time. *Annual International Conference On The Theory And Applications Of Cryptographic Techniques*. pp. 96-125 (2024)
26. Gervais, A., Karame, G., Wüst, K., Glykantzis, V., Ritzdorf, H. & Capkun, S. On the security and performance of proof of work blockchains. *Proceedings Of The 2016 ACM SIGSAC Conference On Computer And Communications Security*. pp. 3-16 (2016)
27. Porat, A., Pratap, A., Shah, P. & Adkar, V. Blockchain Consensus: An analysis of Proof-of-Work and its applications. (Stanford University: Stanford, CA, USA,2017)

28. Garay, J., Kiayias, A. & Panagiotakos, G. Proofs of Work for Blockchain Protocols.. *IACR Cryptol. EPrint Arch.*. **2017** pp. 775 (2017)
29. Sriman, B., Ganesh Kumar, S. & Shamili, P. Blockchain technology: Consensus protocol proof of work and proof of stake. *Intelligent Computing And Applications: Proceedings Of ICICA 2019*. pp. 395-406 (2021)
30. Nguyen, C., Hoang, D., Nguyen, D., Niyato, D., Nguyen, H. & Dutkiewicz, E. Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. *IEEE Access*. **7** pp. 85727-85745 (2019)
31. Spasovski, J. & Eklund, P. Proof of stake blockchain: performance and scalability for groupware communications. *Proceedings Of The 9th International Conference On Management Of Digital EcoSystems*. pp. 251-258 (2017)
32. Wang, T., Liew, S. & Zhang, S. When blockchain meets AI: Optimal mining strategy achieved by machine learning. *International Journal Of Intelligent Systems*. **36**, 2183-2207 (2021)
33. Narayanan, A., Bonneau, J., Felten, E., Miller, A. & Goldfeder, S. Bitcoin and cryptocurrency technologies: a comprehensive introduction. (Princeton University Press,2016)
34. Sapirshtein, A., Sompolinsky, Y. & Zohar, A. Optimal selfish mining strategies in bitcoin. *Financial Cryptography And Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016, Revised Selected Papers 20*. pp. 515-532 (2017)
35. Mosakheil, J. Security threats classification in blockchains. (2018)
36. Nicolas, K., Wang, Y., Giakos, G., Wei, B. & Shen, H. Blockchain system defensive overview for double-spend and selfish mining attacks: A systematic approach. *IEEE Access*. **9** pp. 3838-3857 (2020)
37. Begum, A., Tareq, A., Sultana, M., Sohel, M., Rahman, T. & Sarwar, A. Blockchain attacks analysis and a model to solve double spending attack. *International Journal Of Machine Learning And Computing*. **10**, 352-357 (2020)
38. Chen, Y., Chen, H., Han, M., Liu, B., Chen, Q. & Ren, T. A novel computing power allocation algorithm for blockchain system in multiple mining pools under withholding attack. *IEEE Access*. **8** pp. 155630-155644 (2020)
39. Li, W., Cao, M., Wang, Y., Tang, C. & Lin, F. Mining pool game model and nash equilibrium analysis for pow-based blockchain networks. *IEEE Access*. **8** pp. 101049-101060 (2020)
40. Aggarwal, S. & Kumar, N. Attacks on blockchain. *Advances In Computers*. **121** pp. 399-410 (2021)
41. Prashar, D. & Others Analysis on blockchain vulnerabilities & attacks on wallet. *2021 3rd International Conference On Advances In Computing, Communication Control And Networking (ICAC3N)*. pp. 1515-1521 (2021)
42. Singh, S., Hosen, A. & Yoon, B. Blockchain security attacks, challenges, and solutions for the future distributed iot network. *Ieee Access*. **9** pp. 13938-13959 (2021)
43. Dai, Q., Zhang, B. & Dong, S. A DDoS-Attack Detection Method Oriented to the Blockchain Network Layer. *Security And Communication Networks*. **2022**, 5692820 (2022)
44. Sayeed, S., Marco-Gisbert, H. & Caira, T. Smart contract: Attacks and protections. *Ieee Access*. **8** pp. 24416-24427 (2020)
45. Bhardwaj, A., Shah, S., Shankar, A., Alazab, M., Kumar, M. & Gadekallu, T. Penetration testing framework for smart contract blockchain. *Peer-to-Peer Networking And Applications*. **14** pp. 2635-2650 (2021)

46. Hou, C., Zhou, M., Ji, Y., Daian, P., Tramer, F., Fanti, G. & Juels, A. SquirRL: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning. *ArXiv Preprint ArXiv:1912.01798*. (2019)
47. Bar-Zur, R., Abu-Hanna, A., Eyal, I. & Tamar, A. WeRLman: to tackle whale (transactions), go deep (RL). *Proceedings Of The 15th ACM International Conference On Systems And Storage*. pp. 148-148 (2022)
48. Bar-Zur, R., Dori, D., Vardi, S., Eyal, I. & Tamar, A. Deep bribe: Predicting the rise of bribery in blockchain mining with deep RL. *2023 IEEE Security And Privacy Workshops (SPW)*. pp. 29-37 (2023)
49. Vernon, D. Artificial cognitive systems: A primer. (MIT Press,2014)
50. De Brigard, F. Cognitive systems and the changing brain. *Philosophical Explorations*. **20**, 224-241 (2017)
51. Barfuss, W. Dynamical systems as a level of cognitive analysis of multi-agent learning: Algorithmic foundations of temporal-difference learning dynamics. *Neural Computing And Applications*. **34**, 1653-1671 (2022)
52. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G. & Others Human-level control through deep reinforcement learning. *Nature*. **518**, 529-533 (2015)
53. Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. & Others Mastering the game of Go with deep neural networks and tree search. *Nature*. **529**, 484-489 (2016)
54. Ghalavand, M., Hatami, J., Setarehdan, S., Nosrati, F., Ghalavand, H. & Nikhalat-Jahromi, A. Comparison of the Effects of Interaction with Intentional Agent and Artificial Intelligence using fNIRS. *ArXiv Preprint ArXiv:2402.17650*. (2024)
55. Dellermann, D., Ebel, P., Söllner, M. & Leimeister, J. Hybrid intelligence. *Business & Information Systems Engineering*. **61**, 637-643 (2019)
56. Vahdati, M., Gholizadeh HamlAbadi, K. & Saghiri, A. IoT-Based healthcare monitoring using blockchain. *Applications Of Blockchain In Healthcare*. pp. 141-170 (2021)
57. Topol, E. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*. **25**, 44-56 (2019)
58. Saghiri, A., Vahdati, M., Gholizadeh, K., Meybodi, M., Dehghan, M. & Rashidi, H. A framework for cognitive Internet of Things based on blockchain. *2018 4th International Conference On Web Research (ICWR)*. pp. 138-143 (2018)
59. Watkins, C. & Dayan, P. Q-learning. *Machine Learning*. **8** pp. 279-292 (1992)
60. Clifton, J. & Laber, E. Q-learning: Theory and applications. *Annual Review Of Statistics And Its Application*. **7**, 279-301 (2020)
61. Thomas, R., Friend, D., DaSilva, L. & MacKenzie, A. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*. **44**, 51-57 (2006)
62. Khan, A., Rehmani, M. & Rachedi, A. Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions. *IEEE Wireless Communications*. **24**, 17-25 (2017)
63. Arslan, H. & Mitola, J. Cognitive radio, software defined radio, and adaptive wireless systems. (Springer,2007)
64. Nikhalat-Jahromi, A., Saghiri, A. & Meybodi, M. Q-Defense: When Q-Learning Comes to Help Proof-of-Work Against the Selfish Mining Attack. *Proceedings Of The 16th International Conference On Agents And Artificial Intelligence, ICAART 2024, Volume 1, Rome, Italy, February 24-26, 2024*. pp. 37-46 (2024), https://doi.org/10.5220/0012378600003636

65. Zhao, W., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z. & Others A survey of large language models. *ArXiv Preprint ArXiv:2303.18223*. (2023)
66. Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D. & Du, M. Explainability for large language models: A survey. *ACM Transactions On Intelligent Systems And Technology*. **15**, 1-38 (2024)
67. Schaeffer, R., Miranda, B. & Koyejo, S. Are emergent abilities of large language models a mirage?. *Advances In Neural Information Processing Systems*. **36** (2024)