Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)
Survey Track

Survey on Strategic Mining in Blockchain: A Reinforcement Learning Approach

Jichen Li[1,2], **Lijia Xie**[1], **Hanting Huang**[1,4], **Bo Zhou**[1], **Binfeng Song**[1,4], **Wanying Zeng**[1,3], **Xiaotie Deng**[1,2] and **Xiao Zhang**[1,3,5]
[1]Zhongguancun Laboratory
[2]Center on Frontiers of Computing Studies, Computer Science Department, Peking University
[3]LMIB and School of Mathematical Sciences, Beihang University
[4]LMIB and Institute of Artificial Intelligence, Beihang University
[5]Hangzhou International Innovation Institute of Beihang University
{limo923, xiaotie}@pku.edu.cn, {xielj, zhoubo}@zgclab.edu.cn,
{hantinghuang, zb2342110, zengzeng, xiao.zh}@buaa.edu.cn

Abstract

Strategic mining attacks, such as selfish mining, exploit blockchain consensus protocols by deviating from honest behavior to maximize rewards. Markov Decision Process (MDP) analysis faces scalability challenges in modern digital economics, including blockchain. To address these limitations, reinforcement learning (RL) provides a scalable alternative, enabling adaptive strategy optimization in complex dynamic environments.

In this survey, we examine RL's role in strategic mining analysis, comparing it to MDP-based approaches. We begin by reviewing foundational MDP models and their limitations, before exploring RL frameworks that can learn near-optimal strategies across various protocols. Building on this analysis, we compare RL techniques and their effectiveness in deriving security thresholds, such as the minimum attacker power required for profitable attacks. Expanding the discussion further, we classify consensus protocols and propose open challenges, such as multi-agent dynamics and real-world validation.

This survey highlights the potential of reinforcement learning to address the challenges of selfish mining, including protocol design, threat detection, and security analysis, while offering a strategic roadmap for researchers in decentralized systems and AI-driven analytics.

1 Introduction

In recent years, blockchain technology has been widely applied to solve problems in various domains, developing innovative solutions previously considered impossible. It all began with an inventive ledger design to record all transactions generated in a decentralized system called Bitcoin, invented by [Nakamoto, 2008]. This revolutionary ledger design maintains a sequentially growing list of blocks, each containing several transactions and linked to the preceding block using cryptographic techniques. The process of adding new blocks is governed by a consensus mechanism called Proof of Work (PoW), in which participants, called miners, use their computational power to calculate a hash function. The miner who finds a valid solution for this hash function has the right to produce a new block and earn rewards for generating it. In the design of the Bitcoin protocol, each miner is expected to broadcast the generated block to the network immediately. As long as all participants behave honestly, the expected revenue will be proportional to their computational power.

However, in practice, miners are economically rational and profit-seeking. They may adopt strategic behaviors, implying that honestly broadcasting a block is not necessarily the most rewarding strategy. This is indeed the case, as evidenced by the study in [Eyal and Sirer, 2014], which introduced the selfish mining strategy - arguably the most notorious mining attack in blockchain. In this attack, a miner strategically delays broadcasting the blocks they mine, causing other miners to generate blocks at invalid positions and inducing honest miners to waste their mining power. As a result, the strategic miner can earn more (expected) revenue than their fair share. Pushing this approach to the extreme, [Sapirshtein *et al.*, 2016] expanded the action space of selfish mining, modeling it as a Markov Decision Process (MDP), and analyzed the optimal mining strategy for a miner when faced with other honest miners. Several works have since been initiated to study mining strategies using this approach, such as [Feng and Niu, 2019; Grunspan and Pérez-Marco, 2020; Li *et al.*, 2021; Ferreira *et al.*, 2022].

However, with the continuous updates of blockchain consensus and the introduction of new protocols, directly computing a miner's strategy using MDP faces computational difficulties. To address this issue, [Hou *et al.*, 2019] proposed a generalizable framework for using reinforcement learning (RL) to analyze blockchain incentive mechanisms. Using this approach, researchers only need to model the states and strategies from the miner's perspective in an MDP and then use machine learning methods to learn an approximate optimal strategy. This method provides a framework for the detailed analysis of various blockchain protocols and attack patterns, including [Bar-Zur *et al.*, 2022; Bar-Zur *et al.*, 2023;

Sarenche *et al.*, 2024].

In this survey, we provide a comprehensive overview of blockchain strategic mining analysis. We first summarize the MDP modeling approaches to analyze miners' strategic mining behavior and review the resulting security thresholds for attackers in different types of consensus protocols. Next, we focus on summarizing existing findings on miners' strategic behavior using RL methods and compare the learning techniques employed as well as the resulting security thresholds. Finally, we introduce the MDP modeling paradigms for other consensus protocols in blockchain, such as voting-based and parallel confirmation protocols. We also propose several open problems and discuss the potential of using reinforcement learning to analyze these protocols.

The differences between our survey and others. Past surveys on strategic mining have focused mainly on how to prevent mining attacks. [Madhushanie *et al.*, 2024] focuses solely on the harms of selfish mining attacks and analyzes existing detection and mitigation methods, while [Nicolas *et al.*, 2020] focuses on defending against double-spending attacks and selfish mining attacks, proposing various defense strategies. However, no existing work has systematically surveyed analytical methods for strategic mining. Our paper fills this important gap.

Roadmap. In Section 2, we introduce the MDP modeling method for consensus strategies and the definition and results of the security threshold. Then, in Section 3, we present the results of using RL methods to analyze the strategy for mining blocks. After that, in Section 4, we provide an overview of the classification of blockchain consensus mechanisms and how analysis methods are applied within each category. Finally, we summarize this survey in Section 5.

2 Strategic Mining Analysis via Markov Decision Processes

In this section, we review the MDP modeling approach for consensus strategies, as well as the definition and analysis of security thresholds. We begin by examining previous work on the theoretical foundations of MDPs, highlighting key components such as states, actions, transitions, and rewards, which serve as the basis for decision-making in uncertain environments. We then explore the application of MDP models to strategic mining, demonstrating how they can optimize mining strategies in consensus protocols. Finally, we summarize the security threshold analysis through MDPs, focusing on how these models can be utilized to evaluate and to enhance security measures. This section offers both theoretical insights and practical applications of MDPs in addressing strategic mining problems. The results for the security threshold of strategic mining are summarized in Table 1.

2.1 Theoretical Foundations of MDP Modeling

We present the definition of MDP and its fundamental concepts as follows.

Definition 1 (Markov Decision Process (MDP))**.** *An MDP is formally defined as a quintuple*

$$MDP = (S, A, P, R, \gamma), \tag{1}$$

where:

- S*: Finite or countable state space.*
- A*: Finite action space available from each state.*
- $P(s' \mid s, a) : S \times S \times A \to [0, 1]$*, the transition probability matrix, denoting the probability of transitioning from state* s *to state* s' *with action* a*.*
- $R(s, a) : S \times A \to \mathbb{R}$*, the reward function providing the immediate reward obtained from state* s *with action* a*.*
- $\gamma \in [0, 1]$*: The discount factor, controlling the importance of future rewards.*

A policy $\pi(a|s)$ is a mapping that defines the probability actions should take in each state. To calculate the total reward under a given policy π in MDP, let G_t be the cumulative reward starting from time t with start state s_t, denoted by

$$G_t = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_\pi[R\big(s_{t+k}, \pi(s_{t+k})\big)],$$

where s_{t+k} is the state reached at time step $t+k$. For any state $s \in S$, the value function $V(s)$ is the expected cumulative reward calculated according to the current policy π. Given the state s_t, the agent selects the action a_t with probability $\pi(a_t|s_t)$. Therefore, the form of the value function is as follows:

$$V^\pi(s) = \mathbb{E}_\pi \left[G_t \mid s_t = s \right].$$

Furthermore, under a given policy π, the expected cumulative reward after performing action a in state s is defined as the state-action value function $Q^\pi(s, a)$, given by the form as

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[R(s_t, a_t) + \gamma G_{t+1} \mid s_t = s, a_t = a \right].$$

The MDP model exhibits the following core properties:

- *Markov Property:* MDP has the *memoryless* property, indicating that the transition from the current state depends only on the present state and not on past historical states. Therefore, for states s_t and s_{t+1}, it holds that:

$$P(s_{t+1} \mid s_t, a_t) = P(s_{t+1} \mid s_0, a_0, \ldots, s_t, a_t).$$

- *Bellman Recursive:* For a given MDP with fixed policy π, $V^\pi(s)$ and $Q^\pi(s, a)$ satisfies the Bellman equation:

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} P(s'|s, a) \left[R(s, a) + \gamma V^\pi(s') \right],$$

$$Q^\pi(s, a) = \sum_{s'} P(s'|s, a)[R(s, a) + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s', a')].$$

- *Stationary Distribution:* If the state space is finite and the transition matrix is ergodic, then there exists a stationary distribution $\phi(s)$ under a deterministic policy $a = \pi(s)$ that satisfies

$$\phi(s') = \sum_s \phi(s) P(s' \mid s, \pi(s)),$$

where $\phi(s)$ describes the long-run probability of each state and is an eigenvector of the transition matrix.

Literature	Security Threshold	Method	Blockchain Consensus	Description
[Eyal and Sirer, 2014]	0.25	Markov Reward Process	Bitcoin PoW	Introduce original selfish mining strategy.
[Sapirshtein *et al.*, 2016]	0.232	MDP	Bitcoin PoW	Compute the optimal strategy for selfish mining.
[Marmolejo-Cossío *et al.*, 2019]	0.26297	MDP + Game Theory	Bitcoin PoW	Consider scenario of multiple non-colluding semi-selfish miners.
[Feng and Niu, 2019]	0.26	Two-dimensional MDP	Ethereum PoW	Propose a two-dimensional MDP to model selfish mining in Ethereum.
[Zur *et al.*, 2020]	0.2468	Average Reward Ratio MDP	Ethereum PoW	Propose the Average Reward Ratio (ARR) MDPs, reducing the complexity of computing optimal strategy.

Table 1: Studies on security threshold of blockchain consensus using Markov Decision Process

2.2 MDP Model for Strategic Mining

The seminal work on selfish mining attack [Eyal and Sirer, 2014] considers a system with two types of miners: selfish miners and honest miners. Let α denote the fraction of mining power possessed by the selfish miner. The block generation process is modeled as a random process, where a new block is generated each time slot. In this scenario, selfish miners maintain a private chain after mining a new block and selectively reveal it when the public chain approaches the length of the private chain, making sure that the honest miners waste their computational power on their mined blocks. Therefore, selfish miners can earn excessive rewards, which the following MDP can represent:

- The state space (l_a, l_h) is used to record the current state of the blockchain, where l_a is the length of the private chain, and l_h is the length of the public chain.
- The action space $\{$*adopt, override, match, wait*$\}$ represents the possible actions of the selfish miner in each state regarding mining and broadcasting blocks. Specifically, *adopt* indicates abandoning the private chain and switching to mining on the longest chain, while *override* refers to broadcasting two blocks from the private chain. Action *match* represents broadcasting one block from the private chain, and *wait* means not broadcasting and continuing mining.
- The transition probability of the system is determined by parameter α. For example, when a selfish miner chooses the action *wait* in state (l_a, l_h), the next state will be $(l_a + 1, l_h)$ with probability α (the selfish miner mines a new block) or $(l_a, l_h + 1)$ with probability $1 - \alpha$ (the honest miner mines a new block).
- The reward of the selfish (honest) miner is the number of blocks accepted by all parties that the selfish (honest) miner mined. The goal of the selfish miner is to maximize its proportion of the total reward.

2.3 Security Threshold Analysis by MDP

The security threshold analysis through the Markov Decision Process provides critical insights into blockchain protocol vulnerabilities by quantifying the minimum resource requirements for attackers to profit from strategic mining. This methodology systematically models state transitions, reward mechanisms, and strategic interactions, enabling rigorous evaluation of blockchain security boundaries.

Foundational Model on Selfish Mining. The seminal work [Eyal and Sirer, 2014] established the theoretical framework for strategic mining by formalizing the *selfish mining* strategy as a Markov Reward Process. Their analysis revealed Bitcoin's non-incentive compatibility: attackers with over 25% hashing power could gain disproportionate rewards by selectively withholding blocks. Subsequent research expanded the strategic mining design space through novel attack vectors. [Nayak *et al.*, 2016] proposed the *stubborn mining* strategy, demonstrating a quarter profit increase over traditional selfish mining by persisting on private chains despite public chain dominance. By considering this strategy space, the 25% threshold was further refined by [Sapirshtein *et al.*, 2016], who introduced an ϵ-optimal algorithm to demonstrate that attackers could exploit vulnerabilities with only 23.2% computational power, thereby lowering Bitcoin's security boundary.

Multi-Attacker Scenarios and Equilibrium Analysis. The emergence of multi-miner competition introduced new dimensions to security threshold analysis. [Liu *et al.*, 2018] developed the *publish-n strategy*, reducing stale block rates by 26.3% compared to selfish mining through deterministic block release patterns. Building on this, [Marmolejo-Cossío *et al.*, 2019] introduced semi-selfish Mining, which imposed a two-block limit on private chains to lower the security threshold to 26.297%. Their simulations also revealed an inverse relationship between attacker count and security thresholds. Complementing these studies, although solving the MDP game has been proven to be computationally complex [Deng *et al.*, 2023], [Zhang *et al.*, 2022] designed a mining game model and proved that honest mining remains an equilibrium strategy when attackers' computational power stays below 33%.

Ethereum Analysis and Methodological Advances. Ethereum's unique reward mechanism necessitated tailored MDP frameworks. [Feng and Niu, 2019] constructed a two-dimensional MDP incorporating uncle/nephew block rewards, identifying a 26% security threshold. To address Ethereum's nonlinear reward structure, [Zur *et al.*, 2020] proposed the Probability Termination Optimization (PTO) method, converting complex MDPs into solvable forms. This innovation reduced Ethereum's security threshold to 24.68%, demonstrating the critical role of reward function design in protocol robustness.

3 Reinforcement Learning Framework for Analyzing Strategic Mining

Through the reinforcement learning (RL) technique, an agent learns optimal strategies via its interactions with the environment by selecting actions based on the current state and adjusting its behavior based on rewards or penalties. Similar to many important applications [Arulkumaran *et al.*, 2017; Nosratabadi *et al.*, 2020; Zhao *et al.*, 2022], incentive mechanisms ensure the security of blockchain protocols by rewarding miners who follow the protocol. The reinforcement learning method helps users to simplify the complex dynamics of blockchain protocols for better security analysis.

3.1 Theoretical Foundations of RL

The learning process in RL can be approached through two fundamental methods:

- **Value-based Methods**: These methods indirectly select the optimal policy by calculating the value of each state. For example, Q-learning [Kröse, 1995] is a value-based method that learns Q-values (state-action value functions) to evaluate the quality of actions.
- **Policy-based Methods**: These methods directly optimize the policy itself, rather than first learning a value function and then selecting a policy. Policy gradient methods are a common form of this approach.

The core objective of reinforcement learning is to maximize cumulative rewards, and this process is typically modeled as a Markov Decision Process [Puterman, 2014].

Value-based methods evaluate the long-term return under a given state by a value function. The optimal policy is then derived by maximizing the value function:

$$\pi^* = \arg\max_{\pi} V_\pi(s).$$

Policy-based methods directly learn the policy $\Pi(a|s)$, which usually maps states to actions with the policy parameters θ. The goal of policy optimization is to maximize the expected return:

$$J(\theta) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_t \right].$$

Many classical algorithms in reinforcement learning, such as Q-learning, Monte Carlo methods, and Temporal Difference (TD) learning, can be viewed as numerical estimation methods for the value function in an MDP.

- **Q-learning**: It is a model-free method to find the optimal policy by updating Q values (state-action values). Although it does not require a model of the environment, it is similar to MDP's state value function because both aim to optimize the expected future return.
- **Monte Carlo Methods**: Monte Carlo methods estimate the value function by sampling the environment multiple times. Their computation process closely mirrors the value function calculation in an MDP, with the key difference being that they use actual returns rather than expected returns.
- **Temporal Difference (TD) Learning**: TD learning updates the estimate of the current state by incorporating the immediate reward and the value function of the next state at each step. It combines concepts from dynamic programming and the Bellman equation in an MDP.

3.2 Strategic Mining Analysis Through RL

Building on the foundational concepts of RL in modeling strategic mining behaviors, this subsection delves into the application of RL-based frameworks to analyze and optimize mining strategies in blockchain protocols. Specifically, we explore how RL has been employed to develop analytical frameworks, estimate advanced security thresholds, and devise novel attack strategies and countermeasures. These studies highlight the versatility of RL in capturing the complex dynamics of strategic mining, offering insights into both the vulnerabilities and potential mitigations within blockchain systems. The key findings are summarized in Table 2.

RL-Based Analytical Frameworks. Several studies have leveraged reinforcement learning to analyze and optimize strategic mining behaviors. SquirRL is one of these pioneering works. This analytical framework proposed in [Hou *et al.*, 2019] utilizes RL to analyze blockchain's incentive mechanisms. It defines agent capabilities and action spaces, creates a simulation environment, and incorporates elements such as agent extraction, RL algorithm selection, and reward function design. In turn, it successfully identified the optimal selfish mining attack in Bitcoin and discovered a novel attack on Ethereum's Casper FFG protocol. Another work [Wang *et al.*, 2021] explored the feasibility of dynamically learn optimal strategic mining approaches. Unlike conventional analytical models that require explicit parameter extraction, their approach employs RL to observe the blockchain network and consensus protocol, adapting mining strategies to time-varying network conditions without relying on prior knowledge of MDP parameters.

Advanced Security Threshold Estimation. Building on the application of RL, [Bar-Zur *et al.*, 2022] introduced WeRLman, an RL framework that incorporates "whales" (high-value transactions) and variance reduction techniques to more accurately estimate security thresholds. By using variance reduction to mitigate high sampling noise and optimizing strategies with Monte Carlo Tree Search, the framework determined Bitcoin's security threshold to be approximately 25% (which decreases over time) and Ethereum's to be around 17%. Expanding on this work, [Bar-Zur *et al.*,

Literature	Security Threshold	Threshold Condition	Method	Consensus Type	Description
[Hou *et al.*, 2019]	0.25	\	Deep Q Network	Bitcoin & Ethereum PoW	Recover optimal selfish mining strategy in Bitcoin and surpasses the existing selfish mining strategies in Ethereum.
[Bar-Zur *et al.*, 2022]	0.20 0.17 0.12	3 minting rate halving 4 minting rate halving 5 minting rate halving	Monte Carlo Tree Search & DQN	Bitcoin PoW	Shows that transaction fee volatility will reduce blockchain security.
[Bar-Zur *et al.*, 2023]	0.21 0.19	0.5 petty compliant 0.75 petty compliant	Monte Carlo Tree Search & DQN	Bitcoin PoW	Shows that selfish miners can boost profits by bribing partially compliant miners.
[Sarenche *et al.*, 2024]	0.24198	\	DQN	Longest-Chain Proof of Stake	Uses DQN in LC-PoS protocols and shows that the security threshold for selfish proposing attacks is lower than that of selfish mining.

Table 2: Research on security threshold of blockchain consensus protocols by reinforcement learning methods

2023] extended WeRLman to explore the impact of small miners on blockchain security. They assumed a mix of compliant small miners and honest miners, and found that selfish miners could exploit weakened attack defenses, increasing their profits by over 10%. In the Bitcoin scenario, when half of the miners were small and compliant, the security threshold dropped from 25% to 21%.

Alternative RL-Based Attacks and Countermeasures. Reinforcement learning has enabled novel attack strategies beyond selfish mining. [Yang *et al.*, 2020] introduced an intelligent bribery-based selfish mining attack, using RL to optimize strategies and outperform traditional models in profitability and equity thresholds. By modeling the environment as an MDP and leveraging RL-based decision-making, their approach surpassed traditional selfish mining models in terms of equity threshold and profitability. Similarly, [Jeyasheela Rakkini and Geetha, 2024] applied machine learning to predict miner rewards with high accuracy (MSE: 0.0032) and used Q-learning to simulate selfish mining behaviors. Their ϵ-greedy value iteration approach improved attacker profitability while offering insights into countermeasures. These studies highlight RL's dual role in advancing attacks and defenses in blockchain systems.

RL in Proof-of-Stake Blockchains. While strategic mining attacks have traditionally been associated with Proof-of-Work (PoW) blockchains, the longest chain rule is also employed in some Proof-of-Stake (PoS) protocols, where proposer elections replace mining. This shift introduces a new attack vector, known as selfish proposing, which is analogous to selfish mining. [Sarenche *et al.*, 2024] investigated selfish proposing attacks in longest chain PoS (LC-PoS) blockchains, analyzing how attackers exploit the "nothing-at-stake" problem and proposer predictability. The study found that the "nothing-at-stake" phenomenon slightly increases the proportion of blocks proposed by attackers, while the predictability of proposers significantly increases the proportion of attack blocks. To analyze selfish proposing attacks in more complex scenarios, they also used deep Q-learning tools to approximate the optimal attack strategies under different stake shares.

4 Consensus Protocol Classification and Open Problems

Consensus protocols are fundamental to blockchain systems, ensuring that all participants agree on transaction validity and the blockchain's state, thereby preventing unauthorized modifications and preserving the network's integrity. These protocols can be classified on the basis of their chain selection rules, which include chain-based, vote-based, and DAG-based (parallel confirmation) approaches. When analyzing strategic mining across different consensus protocols, a key step is constructing the environment, tailored to the specific incentive mechanism and underlying consensus algorithm. In this section, we will explore the similarities and differences in constructing environments for various consensus protocols and propose meaningful open questions for future research.

4.1 Consensus Protocol Overview

Blockchain consensus mechanisms can be classified based on their chain selection rules, including chain-based rules, vote-based rules, and DAG-based rules.

Chain-based Consensus Rules. Chain-based consensus rules rely on a linear blockchain structure, where blocks are linked in a chain, and the main chain is determined by the accumulated work or weight. The Longest Chain Rule, used in Proof-of-Work (PoW) systems like Bitcoin [Nakamoto, 2008] and Ethereum 1.0 [Wood and others, 2014], selects the chain with the most computational work. FruitChain [Pass and Shi, 2017] extends the Longest Chain Rule by incorporating a dual-reward system, where miners earn rewards not only for block creation but also for broadcasting "fruit" structures, thus enhancing system security. Similarly, the Heaviest Chain Rule selects the chain with the greatest accumulated weight, typically based on stake, as seen in systems like Ouroboros [Kiayias *et al.*, 2017]. Additionally, Proof

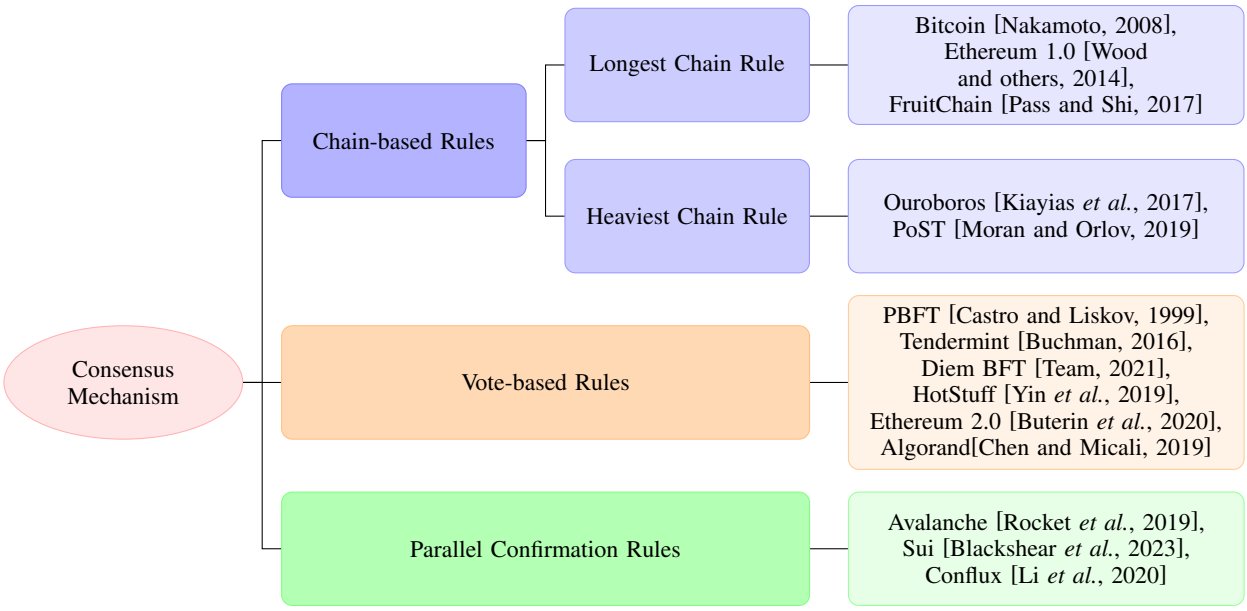

Figure 1: Blockchain Consensus Protocol Classification

of Space and Time (PoST) [Moran and Orlov, 2019] relies on the amount of storage and time invested to secure the chain, using a similar structure where the "heaviest" chain is the one that accumulates the most storage and time.

Vote-based Consensus Rules. These consensus protocols involve nodes casting votes on the validity of transactions or blocks. In vote-based consensus systems, such as Practical Byzantine Fault Tolerance (PBFT) [Castro and Liskov, 1999], Tendermint [Buchman, 2016], Algorand [Chen and Micali, 2019] and HotStuff [Yin *et al.*, 2019], a multi-phase voting process ensures that once a block is committed, it cannot be reverted, thus preventing forks. These protocols offer strong finality guarantees, with blocks being immediately finalized once they receive a sufficient number of votes.

Parallel Confirmation Rules. Parallel confirmation rules deviate from traditional chain-based structures by allowing multiple branches to exist in parallel. In protocols like Avalanche [Rocket *et al.*, 2019], Conflux [Li *et al.*, 2020], and Sui [Blackshear *et al.*, 2023], consensus is achieved probabilistically, with transactions validated concurrently across different branches. Since there is no main chain, consensus is reached without relying on a single-chain structure.

It is also important to note that some consensus mechanisms may exhibit both primary and secondary attributes, blending features from different categories. For example, Ethereum 2.0 [Buterin *et al.*, 2020] primarily relies on vote-based consensus (LMD-GHOST), while also incorporating the heaviest chain rule (via stake-weighted mechanisms) as a secondary feature.

4.2 Design Components Across Different Consensus Protocols

In this section, we outline the similarities and differences in environment construction for various consensus protocols, highlighting key design components such as *state space*, *action space*, and *reward design*.

State Space. The state space must encode three critical components: (1) Action availability: features representing permissible actions in the current state, such as the fork status in Bitcoin (to track competing chains) [Sapirshtein *et al.*, 2016] or the match flag indicating active participation in protocols like LC-PoS [Sarenche *et al.*, 2024]. (2) Reward computation: features enabling reward calculation based on the canonical chain or subgraph. For example, in Bitcoin, this involves tracking the lengths of competing chains (l_a, l_h) to resolve forks [Nakamoto, 2008]. For protocols with non-linear reward mechanisms (e.g., FruitChain's "fruits" [Pass and Shi, 2017; Zhang and Preneel, 2019] or Ethereum 2.0's attestations [Zhang *et al.*, 2024]), additional metrics are required to compute relative rewards. (3) State transition: The system state evolves through block generation, a discrete-time process with a probability determined by mining power in PoW or stake in PoS. Transitions follow the consensus protocol's stochastic rules and network assumptions, including idealized instant block propagation. During ties, honest nodes adopt adversarial blocks with a probability (rushing factor), modeling latency exploitation.

Action Space. The design of the action space is closely tied to the adversarial model. For each consensus mechanism, different types of adversaries can be defined. For example, in the Bitcoin protocol, the action space is defined as *adopt, override, wait, match*. Other attack strategies may involve actions outside this predefined space, such as the consideration of petty compliant miners in [Bar-Zur *et al.*, 2023]. In the selfish proposing attack targeting the LC-PoS protocol [Sarenche *et al.*, 2024], the action space includes sub-actions to capture the 'jump' strategy, allowing a selfish proposer to alter the parent block due to the "nothing-at-stake" property. In

Ethereum 2.0, the action space may exclude the 'match' action, as the rule for determining the canonical chain is based on the heaviest chain rather than the longest chain. This distinction removes the need to consider network propagation when competing chains have equal block lengths.

Reward Design. In reward design, previous approaches have primarily focused on relative rewards, defined as the attacker's rewards as a fraction of the total network rewards. These rewards are typically established on a per-unit basis, such as for blocks, as seen in earlier works. However, rewards can also be more intricately characterized, including transaction rewards [Bar-Zur *et al.*, 2022] and attestation rewards in Ethereum 2.0 [Zhang *et al.*, 2024]. The goal of the analysis is to identify a strategy $\pi : S \to \Delta(A)$ that maximizes the expected reward $R(s, a)$.

4.3 Open Problems

The evolution of blockchain technology and the digital economy has led to a significant shift from traditional longest-chain consensus mechanisms to alternative models. However, the economic security of these systems hinges on resolving critical open problems that remain inadequately addressed. Moreover, the growing sophistication of miners in employing strategic mining techniques has introduced the risk of multi-agent strategic behaviors. These behaviors could destabilize the ecosystem, leading to economic inefficiencies or even systemic failures. This section identifies important open problems in using RL for blockchain security analysis. Each problem highlights the need for advanced modeling techniques to address the complexities of modern blockchain systems.

Open problem 1: How can extend strategic mining attack analysis to non-longest-chain consensus protocols? Existing research on strategic mining attacks has predominantly centered on longest-chain consensus protocols [Eyal and Sirer, 2014; Sapirshtein *et al.*, 2016; Sarenche *et al.*, 2024]. To generalize this framework to alternative consensus mechanisms, three directions emerge:

- **Weight-Based Protocols:** The state space must explicitly model weight accumulation dynamics. This requires incorporating weight-related parameters into the strategy space while preserving backward compatibility with existing analysis methods for longest-chain systems.
- **Parallel Proof-of-Work Protocols:** The state space can be generalized to include additional features, such as topological order and uncle block rewards. These additions introduce multi-dimensional optimization challenges absent in linear chain protocols.
- **Vote-Based Consensus:** Participants attempt to minimize the costs associated with validating blocks and sending votes, which can result in coordination failures that undermine the validity of consensus protocols [Amoussou-Guenou, 2020]. Furthermore, attackers can execute censorship attacks [Srivastava and Gujar, 2024] by strategically excluding specific information from being incorporated into the final consensus.

Multiple tricks such as imposing artificial limits within the adversarial model [Sapirshtein *et al.*, 2016; Zur *et al.*, 2020; Hou *et al.*, 2019] can help maintain a manageable state space size. The application of the analysis framework to adversarial models targeting the aforementioned attacks still requires further exploration.

Open problem 2: How to develop more realistic MDP models for blockchain security? Current RL models for blockchain security rely on simplified assumptions, such as synchronized networks and fixed miner strategies [Zhang and Preneel, 2019; Sarenche *et al.*, 2024], to reduce complexity. However, in real-world environments, network conditions, miner strategies, and blockchain dynamics are highly unpredictable. For example, attackers may exploit network latency to conduct undetectable attacks, limiting the applicability of existing models [Bahrani and Weinberg, 2023].

Future research should focus on relaxing these assumptions by incorporating dynamic, time-varying environments. RL models can account for unpredictable network delays, adaptive miner strategies, and changing blockchain dynamics. These uncertain conditions can be handle by using advanced RL algorithms, while ensuring robust security analysis under evolving threats.

Open problem 3: How can strategic mining be analyzed in multi-agent environments using RL? A key challenge is applying RL in multi-agent environments, where multiple miners or validators interact strategically to maximize rewards. In these environments, agents may compete, complicating the analysis of selfish mining attacks.

[Marmolejo-Cossío *et al.*, 2019] use a Markov chain model to analyze multi-agent mining dynamics by simplifying the state space of selfish miners. This analysis restrict to semi-selfish mining, where miners only maintain private chains of length at most two. The Partially Observed Markov Game (POMG) extends MDP to multi-agent environments with partial information, allowing it to model strategic behaviors like selfish mining, such as SquirRL [Hou *et al.*, 2019]. However, POMG has limitations, including assumptions about partial observability and challenges with scalability as the number of miners increases. Future research should focus on improving its scalability and refining agent interaction models to better capture the complexities and unpredictability of real-world blockchain dynamics.

5 Conclusion

This survey examines strategic mining in blockchain systems, with a focus on reinforcement learning as a tool for optimizing mining strategies. Traditional Markov Decision Process (MDP) approaches are useful for analyzing behaviors like selfish mining but face scalability challenges. Reinforcement learning (RL) provides adaptability in complex environments, enabling the identification of optimal strategies and security thresholds. This survey reviews previous studies that use MDPs and RL to analyze PoW and PoS consensus, and discusses the potential of these methods for analyzing other vote-based and parallel confirmation blockchains. Future research should focus on refining RL algorithms to improve blockchain security and efficiency, ultimately advancing decentralized systems.

Acknowledgments

Supported by Zhongguancun Laboratory, NSFC/RGC Joint Research Scheme (Project No. 62261160391 and No. N_PolyU529/22), the National Science and Technology Major Project (2022ZD0116401), the National Natural Science Foundation of China (62141605), and Fundamental Research Funds for the Central Universities.

Contribution Statement

Jichen Li and Lijia Xie are co-first authors. Xiaotie Deng and Xiao Zhang are the corresponding authors.

References

[Amoussou-Guenou, 2020] Yackolley Amoussou-Guenou. *Governing the commons in blockchains*. PhD thesis, Sorbonne Université, 2020.

[Arulkumaran *et al.*, 2017] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[Bahrani and Weinberg, 2023] Maryam Bahrani and S Matthew Weinberg. Undetectable selfish mining. *arXiv preprint arXiv:2309.06847*, 2023.

[Bar-Zur *et al.*, 2022] Roi Bar-Zur, Ameer Abu-Hanna, Ittay Eyal, and Aviv Tamar. Werlman: to tackle whale (transactions), go deep (rl). In *Proceedings of the 15th ACM International Conference on Systems and Storage*, pages 148–148, 2022.

[Bar-Zur *et al.*, 2023] Roi Bar-Zur, Danielle Dori, Sharon Vardi, Ittay Eyal, and Aviv Tamar. Deep bribe: Predicting the rise of bribery in blockchain mining with deep rl. In *2023 IEEE Security and Privacy Workshops (SPW)*, pages 29–37. IEEE, 2023.

[Blackshear *et al.*, 2023] Same Blackshear, Andrey Chursin, George Danezis, Anastasios Kichidis, Lefteris Kokoris-Kogias, Xun Li, Mark Logan, Ashok Menon, Todd Nowacki, Alberto Sonnino, et al. Sui lutris: A blockchain combining broadcast and consensus. *arXiv preprint arXiv:2310.18042*, 2023.

[Buchman, 2016] Ethan Buchman. *Tendermint: Byzantine fault tolerance in the age of blockchains*. PhD thesis, University of Guelph, 2016.

[Buterin *et al.*, 2020] Vitalik Buterin, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X Zhang. Combining ghost and casper. *arXiv preprint arXiv:2003.03052*, 2020.

[Castro and Liskov, 1999] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999*, pages 173–186. USENIX Association, 1999.

[Chen and Micali, 2019] Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science*, 777:155–183, 2019.

[Deng *et al.*, 2023] Xiaotie Deng, Ningyuan Li, David Mguni, Jun Wang, and Yaodong Yang. On the complexity of computing markov perfect equilibrium in general-sum stochastic games. *National Science Review*, 10(1):nwac256, 2023.

[Eyal and Sirer, 2014] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014*, volume 8437, pages 436–454. Springer, 2014.

[Feng and Niu, 2019] Chen Feng and Jianyu Niu. Selfish mining in ethereum. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1306–1316. IEEE, 2019.

[Ferreira *et al.*, 2022] Matheus VX Ferreira, Ye Lin Sally Hahn, S Matthew Weinberg, and Catherine Yu. Optimal strategic mining against cryptographic self-selection in proof-of-stake. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 89–114, 2022.

[Grunspan and Pérez-Marco, 2020] Cyril Grunspan and Ricardo Pérez-Marco. Selfish mining in ethereum. In *Mathematical Research for Blockchain Economy: 2nd International Conference MARBLE 2020, Vilamoura, Portugal*, pages 65–90. Springer, 2020.

[Hou *et al.*, 2019] Charlie Hou, Mingxun Zhou, Yan Ji, Phil Daian, Florian Tramer, Giulia Fanti, and Ari Juels. Squirrl: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning. *arXiv preprint arXiv:1912.01798*, 2019.

[Jeyasheela Rakkini and Geetha, 2024] MJ Jeyasheela Rakkini and K Geetha. Q-learning model for selfish miners with optional stopping theorem for honest miners. *International Transactions in Operational Research*, 31(6):3975–3998, 2024.

[Kiayias *et al.*, 2017] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual international cryptology conference*, pages 357–388. Springer, 2017.

[Kröse, 1995] Ben J. A. Kröse. Learning from delayed rewards. *Robotics Auton. Syst.*, 15(4):233–235, 1995.

[Li *et al.*, 2020] Chenxin Li, Peilun Li, Dong Zhou, Zhe Yang, Ming Wu, Guang Yang, Wei Xu, Fan Long, and Andrew Chi-Chih Yao. A decentralized blockchain with high throughput and fast confirmation. In *2020 USENIX Annual Technical Conference*, pages 515–528, 2020.

[Li *et al.*, 2021] Quanlin Li, Yanxia Chang, Xiaole Wu, and Guoqing Zhang. A new theoretical framework of pyramid markov processes for blockchain selfish mining. *Journal of Systems Science and Systems Engineering*, 30:667–711, 2021.

[Liu *et al.*, 2018] Hanqing Liu, Na Ruan, Rongtian Du, and Weijia Jia. On the strategy and behavior of bitcoin mining

with n-attackers. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 357–368, 2018.

[Madhushanie *et al.*, 2024] Nadisha Madhushanie, Sugandima Vidanagamachchi, and Nalin Arachchilage. Selfish mining attack in blockchain: a systematic literature review. *International Journal of Information Security*, pages 1–19, 2024.

[Marmolejo-Cossío *et al.*, 2019] Francisco J Marmolejo-Cossío, Eric Brigham, Benjamin Sela, and Jonathan Katz. Competing (semi-) selfish miners in bitcoin. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 89–109, 2019.

[Moran and Orlov, 2019] Tal Moran and Ilan Orlov. Simple proofs of space-time and rational proofs of storage. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I 39*, pages 381–409. Springer, 2019.

[Nakamoto, 2008] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[Nayak *et al.*, 2016] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 305–320. IEEE, 2016.

[Nicolas *et al.*, 2020] Kervins Nicolas, Yi Wang, George C Giakos, Bingyang Wei, and Hongda Shen. Blockchain system defensive overview for double-spend and selfish mining attacks: A systematic approach. *IEEE Access*, 9:3838–3857, 2020.

[Nosratabadi *et al.*, 2020] Saeed Nosratabadi, Amirhosein Mosavi, Puhong Duan, Pedram Ghamisi, Ferdinand Filip, Shahab S Band, Uwe Reuter, Joao Gama, and Amir H Gandomi. Data science in economics: comprehensive review of advanced machine learning and deep learning methods. *Mathematics*, 8(10):1799, 2020.

[Pass and Shi, 2017] Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM symposium on principles of distributed computing*, pages 315–324, 2017.

[Puterman, 2014] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[Rocket *et al.*, 2019] Team Rocket, Maofan Yin, Kevin Sekniqi, Robbert van Renesse, and Emin Gün Sirer. Scalable and probabilistic leaderless bft consensus through metastability. *arXiv preprint arXiv:1906.08936*, 2019.

[Sapirshtein *et al.*, 2016] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In Jens Grossklags and Bart Preneel, editors, *Financial Cryptography and Data Security - 20th International Conference, FC 2016*, volume 9603 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2016.

[Sarenche *et al.*, 2024] Roozbeh Sarenche, Svetla Nikova, and Bart Preneel. Deep selfish proposing in longest-chain proof-of-stake protocols. *Cryptology ePrint Archive*, 2024.

[Srivastava and Gujar, 2024] Varul Srivastava and Sujit Gujar. Towards Rational Consensus in Honest Majority . In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pages 1439–1441, Los Alamitos, CA, USA, July 2024. IEEE Computer Society.

[Team, 2021] Diem Team. Diembft v4: State machine replication in the diem blockchain. *Diem (Libra, Novi a Facebook Project. 2021. url: https://developers. diem. com/papers/diem-consensus-state-machine-replication-in-the-diem-blockchain/2021-08-17. pdf.(accessed: 18.11. 2022)(pages 35, 121)*, 2021.

[Wang *et al.*, 2021] Taotao Wang, Soung Chang Liew, and Shengli Zhang. When blockchain meets ai: Optimal mining strategy achieved by machine learning. *International Journal of Intelligent Systems*, 36(5):2183–2207, 2021.

[Wood and others, 2014] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[Yang *et al.*, 2020] Guoyu Yang, Yilei Wang, Zhaojie Wang, Youliang Tian, Xiaomei Yu, and Shouzhe Li. Ipbsm: An optimal bribery selfish mining in the presence of intelligent and pure attackers. *International Journal of Intelligent Systems*, 35(11):1735–1748, 2020.

[Yin *et al.*, 2019] Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 347–356, 2019.

[Zhang and Preneel, 2019] Ren Zhang and Bart Preneel. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 175–192. IEEE, 2019.

[Zhang *et al.*, 2022] Mengqian Zhang, Yuhao Li, Jichen Li, Chaozhe Kong, and Xiaotie Deng. Insightful mining equilibria. In *International Conference on Web and Internet Economics*, pages 21–37. Springer, 2022.

[Zhang *et al.*, 2024] Mingfei Zhang, Rujia Li, and Sisi Duan. Max attestation matters: Making honest parties lose their incentives in ethereum {PoS}. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6255–6272, 2024.

[Zhao *et al.*, 2022] Xusheng Zhao, Jia Wu, Hao Peng, Amin Beheshti, Jessica JM Monaghan, David McAlpine, Heivet Hernandez-Perez, Mark Dras, Qiong Dai, Yangyang Li, et al. Deep reinforcement learning guided graph neural networks for brain network analysis. *Neural Networks*, 154:56–67, 2022.

[Zur *et al.*, 2020] Roi Bar Zur, Ittay Eyal, and Aviv Tamar. Efficient mdp analysis for selfish-mining in blockchains. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 113–131, 2020.