Exploring Profitable Miner Strategies: A Comparative Analysis Beyond Honest Mining

First Author, Second Author, and Third Author
firstauthor@virginia.edu, secondauthor@vt.edu, thirdauthor@gmu.edu

***Abstract* – In recent years, there has been a surge in the popularity of cryptocurrencies because of the trust, transparency, security, and accessibility they provide in contrast to centralized financial systems. The most well-known cryptocurrency is Bitcoin, introduced in 2009 as a solution to the problems associated with centralized management of transactions that have dominated the entire cryptocurrency market. As of March 2024, there are around 8985 active cryptocurrencies, with 420 million cryptocurrency users across the globe and approximately 18000 businesses have integrated cryptocurrency payment methods, reflecting the growing acceptance of digital assets in the commercial sector [49-51] [1]. In decentralized cryptocurrency, for instance, in Bitcoin, the transactions are verified by Bitcoin users who possess the required hardware and computing power. This process is solved based on a difficult mathematical puzzle called Proof of Work (PoW). All the miners compete among themselves to solve the PoW puzzle and the miner who solves the puzzle receives a reward in the form of bitcoins. The miners follow different mining strategies with the intention of maximizing their rewards. For a long time, the most profitable mining strategy was believed to be the honest mining strategy. But it was proved later to gain more mining rewards by selfish mining deviating from honest mining by withholding the newly solved blocks without publishing them to the network. However, recent research shows that selfish mining attacks may not be optimal and proposed different versions of selfish mining allowing the attacker to earn potentially higher rewards. Moreover, Sapirshtein et al. formulated the mining problem as a Markov Decision Process (MDP) which can be solved to derive an optimal mining strategy [9]. Later, Feng et al. extended this model for other cryptocurrencies such as Ethereum [10]. In a more recent study, Wang et al. proposed a model free approach adopting the MDP model and employing Reinforcement Learning to achieve an optimal mining strategy [13]. This paper outlines the findings of our Systematic Literature Review, which delves into the latest research in the domain of adversarial mining strategies in Bitcoin like PoW based blockchains with the objective of investigating and assessing the diverse mining strategies that are more profitable than honest mining.**

***Index Terms* - Blockchain, Proof-of-Work, Adversary Strategies, Selfish Mining, PoW, Reinforcement Learning, Markov Decision Process**

1. Introduction

[2]Cryptocurrency is a digital or virtual form of currency that utilizes cryptography for security and operates on decentralized networks, meaning they are not controlled by a central governing body such as a government or financial institution. Today, the investors can choose from thousands of different cryptocurrencies. The market leader and original cryptocurrency in the cryptocurrency market is Bitcoin which makes up 48.6% of the total value of the crypto market as of 2024. As of February 2024, the global cryptocurrency market cap is $2.09 trillion whereas bitcoin's market cap of $1.02 trillion accounts for around 50% of that total. Bitcoin cryptocurrency was introduced in the 2009 global economic crisis to overcome the problems of centralized transaction management providing several benefits such as improved trust, security, transparency among member organizations by improving the traceability of data shared across a business network and delivering cost savings through new efficiencies [1], [2]. This makes cryptocurrencies more secure and less prone to fraud, tampering or general system failure than keeping them in a single centralized location [3], [4], [5].

Nearly all cryptocurrencies, including Bitcoin, Ethereum [25], Bitcoin Cash [26], and Litecoin [27], are secured by blockchain networks. Fundamentally, a blockchain is a ledger of transactions that is accessible for viewing and verification by anyone. Users are interconnected through a peer-to-peer network where they broadcast their intended transactions, and these transactions can subsequently be validated by any participant using the publicly accessible blockchain ledger. The security of Bitcoin is maintained through its decentralized network of users, known as miners, who verify transactions using specialized hardware and computational power. This process is known as *mining*. This mining process is solved based on a difficult cryptographic puzzle called proof of work (PoW). Bitcoin uses the Proof-of-Work (PoW) scheme as the underlying consensus algorithm for Bitcoin mining [3], [4], [5]. Therefore, each miner engages in solving complex mathematical puzzles, competing with one another to validate transactions and subsequently create a new block in the blockchain. This

Commented [1]: Madam, some of the strategies that I have included in the paper like 51% attack, pool hopping attack are not directly related to my original research. But other attacks like eclipse attack has been considered formulating the MDP model in the newest paper. So it is directly related to the research. However regarding the above mentioned non related attacks, Should I include those in this paper?

Commented [2]: Is this a reliable source?

Commented [3R2]: Yes. CoinGecko has built a strong reputation over the years since its inception in 2014. It is frequently cited by other reputable sources in the cryptocurrency industry, and many investors and analysts rely on its data for making informed decisions.

Commented [4R2]: Are these sources included reputed journals or conferences?

Commented [5R2]: Yes madam. I'll put some links to those papers

[1] Cryptocurrency Prices, Charts, and Crypto Market Cap | CoinGecko
Cryptocurrency Ownership Data – Triple-A

[2] Information and Communication Technology Agency of Sri Lanka (icta.lk)

process, however, requires miners to invest in substantial hardware and computing resources to tackle the computationally intensive PoW challenge. The first miner to successfully solve the puzzle is granted the right to add the new block to the blockchain, ensuring the integrity and security of the transaction ledger while also earning a reward for their efforts. This reward serves as an incentive that motivates miners to fulfill the primary objective of mining: earning the privilege to record transactions on the blockchain, thereby enabling the network to verify and confirm them. However, miners may follow various strategies to solve the PoW challenge. Miners may follow various approaches and techniques that deviate from established protocols and rules of the blockchain network to optimize their chances of successfully solving the proof-of-work puzzle and earning higher rewards. These approaches are called *mining strategies*.

Honest mining is the ethical and protocol-compliant approach to cryptocurrency mining meaning that the honest miners engage in the mining process according to the rules and protocols established by the network. It involves immediate broadcasting of discovered blocks across the network and maintaining transparency. This approach assumes that participating honestly in the network, by immediately sharing valid blocks, will yield the most profitable outcomes for miners in the long term. They contribute their computational power to validate and secure transactions, add new blocks to the blockchain, and maintain the network's integrity. Honest miners follow the consensus rules of the network, compete fairly for block rewards, and they do not engage in practices that would give them an unfair advantage or harm the network, such as selfish mining, double-spending, exploiting vulnerabilities or other forms of malicious manipulations. Therefore, a miner with control over α fraction of the network's computational resources should ideally receive only α fraction of the mining reward. However, a malicious attacker can deploy diverse tactics to unfairly acquire a larger portion of the mining reward. These tactics aim to provide a specific group of malicious miners with an unfair advantage, ensuring they receive a disproportionate share of mining rewards compared to their legitimate contribution of computational power. This unfair advantage enables them to achieve greater expected revenue from mining activities than what would be equitable based on their actual computational resources. These tactics are known as *mining attacks*.

In this paper, we have narrowed various mining attacks of blockchain systems and our main focus on the formulation and deep-dive analysis of selfish mining attack. Selfish mining is a strategy in blockchain mining where miners withhold newly solved blocks instead of immediately broadcasting them to the network [6]. Given that the blockchain continuously adjusts the mining difficulty to ensure that, on average, one valid block is added to the main chain per specified interval (e.g., one block every 10 minutes for Bitcoin, and every 10-20 seconds for Ethereum) [7], this tactic allows the selfish miner to gain a competitive advantage by secretly working on the next block while other miners continue to work on the current one. Once the selfish miner finds the next block, he releases both blocks, causing other miners' work on the current block to become wasted. This enables the selfish miner to earn a larger share of the block rewards compared to other miners, ultimately increasing their profits at the expense of the overall network integrity. Selfish mining has proved to be more profitable than honest mining, allowing the miner to collect higher rewards. The objective of a selfish miner is not to maximize its absolute cumulative rewards. Instead, the goal is to maximize the ratio of its cumulative rewards relative to the cumulative rewards of the entire network. For example, with computing power ratio $\alpha = \frac{1}{4}$, the rewards obtained by selfish mining can be up to $\frac{1}{3}$ fraction of the total rewards of the whole network [8]. Building on this observation, the authors subsequently proposed several selfish mining strategies that yield even higher rewards [9].

However different studies show that selfish mining may not be optimal under certain conditions [8] [9] [12]. Despite the many versions of selfish mining, the optimal (i.e., most profitable) mining strategy remained elusive until [9]. The authors of [9] formulated the mining problem as a general Markov Decision Process [28] (MDP) with a large state-action space (see Section 4.1). The objective of the mining MDP, however, is not a linear function of the rewards as in standard MDPs. Thus, the mining MDP cannot be solved using a standard MDP solver. To solve the problem, they have first transformed the mining MDP with the non-linear objective to a family of MDPs with linear objectives, and then employed a standard MDP solver over the family of MDPs to iteratively search for the optimal mining strategy [9].

Mining is a fundamental aspect of these systems, playing a crucial role in validating transactions and adding new blocks to the chain. Selfish mining, where miners deviate from the standard protocol to gain an unfair advantage, represents a significant deviation from honest mining practices. Despite extensive research into selfish mining, the field continues to evolve, necessitating a thorough examination of the current state of these strategies and their inherent limitations. In addition to selfish mining, various other strategies have been identified that can yield higher rewards than honest mining. In this paper, we aim to explore various mining strategies and their limitations that diverge from honest mining, with a primary focus on selfish mining. Therefore, understanding these dynamics is critical for several reasons:

- Identifying Limitations: Adversarial mining strategies have inherent limitations that impact their effectiveness. Analyzing these limitations provides insights into the constraints and vulnerabilities of such strategies, offering a comprehensive understanding of their practical implications.
- Enhancing Theoretical Models: By examining the current state of adversarial mining strategies, this research contributes to the refinement of theoretical models that describe miner behavior. These models

are essential for accurately simulating and predicting the actions of miners in real-world blockchain networks.

- Informing the Blockchain Community: Developers, researchers, and stakeholders within the blockchain community benefit from a detailed understanding of different adversarial mining strategies. This knowledge helps in assessing the current landscape and informs decisions related to mining practices and protocol design.
- Guiding Future Research: Highlighting the current state and limitations of adversarial mining strategies helps identify gaps in the literature and directs future research efforts. This focus ensures that subsequent studies build upon a solid foundation, addressing unresolved issues and exploring innovative approaches.

By examining the current state and limitations of adversarial mining strategies, this research aims to provide a comprehensive understanding of how these strategies operate in contemporary blockchain networks. This focus will contribute to the broader discourse on mining strategies and miner behavior, offering valuable insights for the blockchain community.

The rest of the paper is organized as follows. Section 2 outlines the background, including blockchain preliminaries, the operation of the consensus algorithm, and the mining process. Section 3 details the planning and execution phases of the conducted SLR. Section 4 presents our findings from the SLR, divided into two primary subsections. In Section 4, we investigate potential adversarial mining strategies in detail, evaluating and reporting the limitations of these strategies. Section 5 proposes future research directions to enhance these models. Finally, Section 6 concludes the SLR.

2. BACKGROUND

In this section, we briefly recap the blockchain preliminaries, operation of the PoW consensus algorithm, and the operation of mining pools, and Reinforcement Learning (RL).

2.1. Proof of Work and Mining

Blockchain is a decentralized, distributed ledger technology that enables secure and transparent recording of transactions across a network of computers. At its core, a blockchain is a chain of blocks, each containing a list of transactions. These blocks are linked together using cryptographic techniques, forming a chronological chain, which is immutable and resistant to tampering.

Bitcoin mining is the process of validating and adding new transactions to the Bitcoin blockchain, as well as the mechanism through which new bitcoins are created. It is a crucial component of the decentralized consensus mechanism that underpins the Bitcoin network. The structure of a block in the Bitcoin blockchain typically consists of components such as Block Header, Transactions, Block Size, Block Height, Block Hash, and Block Reward. The block header contains metadata about the block and is used to validate its integrity. It includes Version, Previous Block Hash, Merkle Root, Timestamp, Nonce, Difficulty Target. It's important to note that blocks in the blockchain are linked together using cryptographic hashing and referencing the hash of the previous block within the block header. (See Fig. 1)

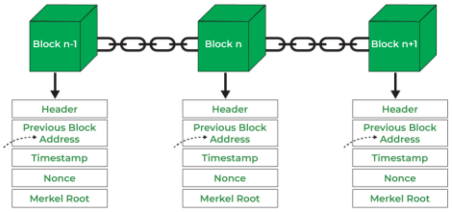


Fig. 1

The PoW algorithm is the consensus mechanism used in blockchain networks, including Bitcoin, to achieve agreement on the state of the blockchain and to validate transactions. It requires participants, known as miners, to solve complex mathematical puzzles to create new blocks and add them to the blockchain. This puzzle requires miners to find a nonce (an arbitrary number used only once) such that the hash of the block's header is less than or equal to the current target value. In PoW, miners construct a block header, including fields such as the previous block hash, Merkle root, timestamp, target difficulty, and a nonce (number used once). The nonce is a 32-bit (4-byte) field that miners can change to find a valid block hash. Once the block header has been constructed, miners then concatenate the block header fields and hash the concatenated string using a hashing algorithm (SHA-256 for Bitcoin). Then, miners select a random nonce and keep adjusting the nonce and repeatedly hash the block header until the resulting hash is less than or equal to the target difficulty. In other words, in order to discover the next block, a miner must generate a hash value that has an equal or higher number of zeros in front of it than the target difficulty. The target difficulty is a 64-digit hexadecimal code that defines how difficult it is to find a valid hash. It is adjusted periodically (e.g., every 2016 blocks in Bitcoin) to ensure blocks are mined at a consistent rate (approximately every 10 minutes for Bitcoin). This process can be denoted mathematically as follows.

$$H(n, p, m) < D$$

where n is the nonce value, p is the hash value of the previous block, m is the Merkle root of all the included transactions in the block, and D is the target. If the hash value meets the difficulty target (i.e., it is lower than the target), the miner has found a valid solution to the puzzle. An example of a target hash is given below.

0000000000000000000abcd1234567890fedcba0987654321 0abcdef123456789

The above target hash has a large number of leading zeros, indicating a high level of difficulty. A miner must find a hash value for the block header that is less than or equal to this target hash in order to successfully mine a block. The number of leading zeros determines the difficulty level; the more leading zeros, the more difficult it is to find a valid hash. This whole process is known as "proof of work" because the miner has demonstrated that they have expended computational effort (work) to find a valid hash. The miner broadcasts the new block to the network, along with the nonce and hash value. Other nodes in the network verify the validity of the block by independently hashing the block header with the nonce and comparing the resulting hash to the target. If the block is valid, it is accepted by the network and added to the blockchain. The miner who successfully mines a new block is rewarded with a certain number of newly created bitcoins, known as the block reward. Additionally, they may collect transaction fees associated with the transactions included in the block.

2.2. Blockchain Forks

What happens when two miners publish different valid blocks referencing the same preceding block simultaneously? If it happens, a temporary situation known as a blockchain *fork* occurs. This scenario can happen due to the decentralized and asynchronous nature of blockchain networks, where multiple miners may find valid solutions to the cryptographic puzzle required to create a new block simultaneously. When two miners independently create valid blocks that reference the same preceding block, the blockchain temporarily forks into two separate branches. Each branch contains a different valid block at the same height. Both blocks are broadcast to the network, and nodes in the network receive and propagate both blocks. As a result, different parts of the network may initially see different versions of the blockchain, leading to a temporary lack of consensus on which block is the "correct" one. Miners and nodes in the network will continue to build upon the block they received first, extending their respective branches of the blockchain. This leads to a race to find the next block, with miners attempting to create longer chains to establish dominance. Eventually, one of the branches will become longer than the other as miners add more blocks to it. When this happens, nodes in the network recognize the longer chain as the valid one according to the blockchain's consensus rules. The shorter branch is then discarded, and the network returns to a single, unified chain. The blocks that were part of the shorter branch (i.e., not included in the longer chain) become orphaned or *stale blocks*. Transactions included in these orphaned blocks are returned to the memory pool and can be included in future blocks.

2.3. Mining Pools

Mining pools provide miners with a higher probability of mining blocks and earning rewards, reducing the financial risks associated with solo mining by increasing their chances of earning rewards more consistently compared to solo mining thereby increasing the efficiency of the mining process [14]. Nowadays more than 90% of cryptocurrency mining is done by pooled mining [41],[52],[53]. A mining pool is a collaborative effort where multiple miners combine their resources to collectively work on discovering blocks and earning rewards. The rewards earned from successfully mining blocks are then distributed among the participants based on the amount of work each miner contributed. In a typical setup, a pool operator oversees the management of the mining, coordinating the pool's activities. The operator sets up and maintains the pool's server, monitors the pool's performance, and distributes the rewards among the participants and may charge a fee for his services. Individual miners join the pool by connecting their mining hardware to the pool's server, allowing their computational power to be combined with that of other miners in the pool. Miners contribute to the pool's computational effort by finding and submitting shares. The pool operator assigns smaller, manageable tasks to each miner. When a miner discovers a block that hashes to a value beginning with a significant number of zeros, such as 32 zeros, they submit this value to the pool manager as a share. Each hash attempt has a probability of $\frac{1}{2^{32}}$ of resulting in a share. Solving shares involves the same process as mining a block, but shares are solutions that meet a lower difficulty target set by the pool, rather than the full network difficulty. The pool manager verifies the share submitted by the miner to ensure it meets the pool's difficulty target. When a miner in the pool finds a share that meets the network's difficulty target, it is submitted as a valid block to the blockchain network. The rewards are then distributed among the miners based on the number of shares each miner has submitted, reflecting their contribution to the pool's overall work.

2.4. Reinforcement Learning

Refer [16]

3. Methodology

The study employed a systematic literature review approach to fulfill its objectives as outlined in Section 3.1. Following the updated 2020 Preferred Reporting Items for Systematic Reviews (PRISMA) guidelines [40], The selection of studies for our systematic literature review on profitable mining strategies in blockchain involves a rigorous, transparent, and methodical process to ensure the inclusion of relevant, high-quality research that contributes meaningfully to understanding the strategies. The selection process will consist of the following steps:

3.1. Formulation of research questions

To guide our investigation into mining strategies that deviate from honest mining, particularly selfish mining, we have formulated the following research questions (RQs):

RQ1: What are the current adversarial strategies used in blockchain mining that deviate from honest mining practices?

Commented [6]: I'm not clear whether this word is suitable or not

Objective1: To identify and categorize various mining strategies that miners employ to gain an advantage over the standard honest mining approach.

Commented [7]: I prefer this inclusion. However, this seems not common. Name them as Objectives 1,2,3, and 4. Try to find the consent of sir for this as it's mixing up both objectives and research questions.

RQ2: How does selfish mining operate within blockchain networks, and what variations of this strategy exist?

Objective2: To analyze the mechanics of selfish mining and its different variants to understand how these strategies are implemented and executed.

RQ3: What are the limitations and vulnerabilities associated with selfish mining and other adversarial mining strategies?

Objective3: To evaluate the weaknesses and potential pitfalls of these strategies, providing a comprehensive assessment of their effectiveness and sustainability.

RQ4: What are the current and future research directions that can address the limitations of these adversarial mining strategies?

Objective4: To identify gaps in the existing literature and propose future research avenues that could enhance the understanding and mitigation of adversarial mining strategies in blockchain networks.

By addressing these research questions, this paper aims to provide a thorough examination of adversarial mining strategies, with a particular focus on selfish mining, their operational mechanisms, limitations, and implications for blockchain networks. This investigation will contribute to the body of knowledge on blockchain mining practices and inform future research directions.

3.2. Data Sources and Search Strategies

The systematic literature review conducted in this study involved an in-depth exploration of published articles spanning the period from 2018 to 2024 across a wider range of electronic databases: Scopus, IEEE Xplore Digital Library, Springer Link and Google Scholar. These databases were selected due to their globally acknowledged impact indices, which encompass a wide array of peer-reviewed scientific and scholarly literature from various scientific domains and disciplines worldwide.

We determined search strings by identifying associated key terms within blockchain mining. This was based on our subject knowledge and previous most-cited research papers and journals.

We established two generic search terms in association with informatics and employed Boolean operators as follows to encompass all literature focusing on profitable mining strategies and to capture information on Bitcoin and Ethereum cryptocurrencies.

(blockchain OR cryptocurrency)

Drawing upon subject knowledge, we identified various types of attacks or mining strategies that deviate from standard Bitcoin and Ethereum protocols. Subsequently, we formulated search terms tailored to capture these strategies effectively as given below.

- selfish mining
- double spending
- 51% attack
- pool hopping
- stubborn mining
- block discarding
- block withholding
- honest mining
- adversarial strategies
- faw attack

Subsequently, we combined these key values into the generic query and executed iterative queries across digital libraries for each mining strategy. An example search query is given below.

("blockchain" OR "cryptocurrency") AND ("adversarial strategies" OR "selfish mining" OR "double-spending" OR OR "honest mining" AND "security")

3.3. Selection of studies

These inclusion criteria help ensure that the systematic review focuses on relevant, high-quality research that contributes meaningfully to understanding profitable mining strategies deviating from Honest mining.

- Empirical studies employing quantitative, qualitative, or mixed-methods research designs, including experimental studies, case studies, surveys, and observational studies.
- Articles specifically addressing the development, implementation, evaluation, or impact of different mining strategies on Bitcoin and Ethereum.
- Articles published within the last ten years (2014-2024) to capture recent advancements and developments in the field.
- Only the articles published in English are considered.

- Articles with a clear description of research methods, data collection techniques, analysis procedures, and consideration of potential sources of bias.

- Articles available through academic databases, institutional repositories, or other accessible sources for data extraction and analysis.

Furthermore, we defined a set of exclusion criteria for this systematic literature review to ensure that only relevant and high-quality studies were included. Studies that did not meet the following criteria were excluded from the review:

- Articles that were not peer-reviewed, such as blog posts, opinion pieces, or other informal publications, were excluded to maintain the academic rigor of the review.
- Irrelevance to Blockchain Adversarial Strategies: Studies that did not focus specifically on adversarial strategies in blockchain were excluded. This included studies primarily centered on other aspects of blockchain technology, such as performance optimization or general security measures, unless they specifically addressed adversarial tactics.
- Only peer-reviewed journal articles, conference papers, and thesis were included. Studies such as book chapters, white papers, and industry reports were excluded due to potential biases and lack of rigorous peer review.
- Studies not published in English were excluded. While this may limit the inclusion of some relevant research, it ensures that all included studies are fully understood and accurately assessed by the reviewers.
- Duplicate studies, or those with overlapping content, were identified and only the most comprehensive version was included. This was to avoid redundancy and ensure a wide range of perspectives and findings.
- Studies that lacked sufficient methodological details, making it impossible to assess the validity and reliability of the findings, were excluded. This included studies with vague descriptions of their experimental or analytical methods.

By applying these exclusion criteria, we aimed to ensure that our systematic literature review includes only the most relevant, high-quality studies. This approach enhances the reliability and validity of the review’s findings, providing a solid foundation for understanding adversarial strategies in blockchain.

3.4. Screening Process

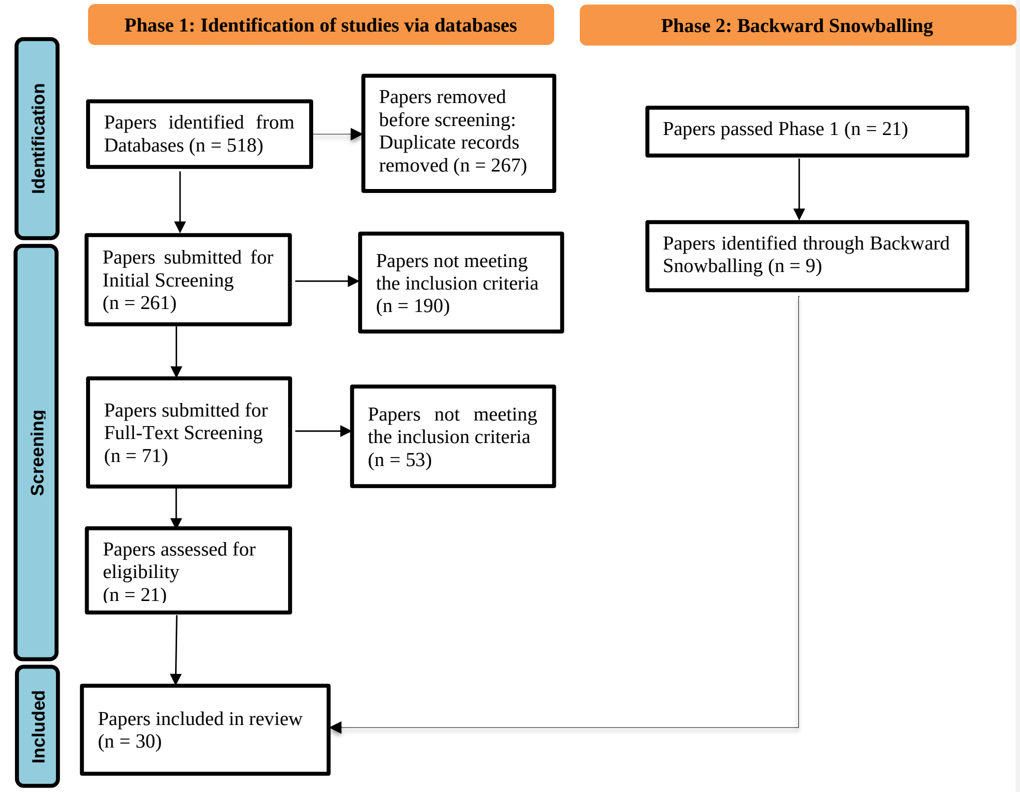
The screening process for our systematic literature review (SLR) on adversarial strategies in blockchain involves a systematic and rigorous approach to ensure the inclusion of relevant, high-quality studies. The screening process consists of two main stages: initial screening and full-text screening. This process helps in efficiently filtering out irrelevant studies and selecting those that meet the predefined inclusion criteria.

3.4.1. Initial Screening

The initial screening phase involves a preliminary review of the titles and abstracts of all studies retrieved through our search strategy. The primary goal is to quickly assess the relevance of each study to the research question and to exclude those that do not meet the basic inclusion criteria. During this phase, the titles, and abstracts of 518 publications were reviewed by two independent reviewers to assess their relevance. The reviewers then applied predefined inclusion and exclusion criteria, focusing on factors such as relevance to blockchain adversarial strategies, publication type, clarity, and language. Each decision to include or exclude a study was meticulously documented, with reasons for exclusion noted where applicable. In cases where discrepancies arose between the two reviewers, these were resolved through discussion or by consulting a third reviewer. This transparent process ensured that only studies meeting the necessary criteria progressed to the next stage of screening. At the end of this process a sum of 461 publications were submitted to the next phase.

3.4.2. Full-Text Screening

During the full-text screening phase, the full texts of the studies that passed the initial screening were thoroughly reviewed to ensure they met all inclusion criteria. Each study was evaluated in detail, focusing on the methodology, relevance to blockchain adversarial strategies, and the quality of evidence provided. A standardized quality assessment tool was used to assess the methodological rigor of each study. Decisions to include or exclude studies were meticulously documented, with reasons for exclusion clearly noted. Any disagreements between reviewers were resolved through discussion or by consulting a third reviewer. This comprehensive review process ensured that only the most relevant and high-quality studies were included in the final analysis. At the end of this process, a total of 30 publications were selected for the SLR. A PRISMA flow diagram was used to document the study selection process, including the number of studies identified, screened, eligible, and included in the review.


Phase 1: Identification of studies via databases
Phase 2: Backward Snowballing
Identification
Screening
Included
Papers identified from Databases (n = 518)
Papers removed before screening: Duplicate records removed (n = 267)
Papers submitted for Initial Screening (n = 261)
Papers not meeting the inclusion criteria (n = 190)
Papers submitted for Full-Text Screening (n = 71)
Papers not meeting the inclusion criteria (n = 53)
Papers assessed for eligibility (n = 21)
Papers included in review (n = 30)
Papers passed Phase 1 (n = 21)
Papers identified through Backward Snowballing (n = 9)

4. Results

In this section, we outline the outcomes derived from the systematic review process described earlier.

Authors of [6] formulated a strategy called Selfish Mining that can be used by a minority pool to obtain more revenue than the pool's ratio of the total mining power. Under selfish mining the authors of [6] emphasized the strategy of a pool to keep deliberately forking the chain by keeping its discovered blocks private. Meanwhile, honest nodes persist in mining on the public chain, while the pool focuses on its separate, private branch. By consistently uncovering more blocks, the pool establishes a greater lead on the public chain, maintaining the secrecy of these new blocks. Once the public chain nears the length of the pool's private branch, the selfish miners disclose blocks from their private chain to the public. Consequently, this approach induces honest miners adhering to the Bitcoin protocol to expend resources on solving cryptographic puzzles that ultimately serve no purpose. Their analysis [6] indicates that although both honest and selfish participants waste resources to some extent, the honest miners incur a comparatively higher waste, and the rewards for the selfish pool surpass its share of the network's mining capacity. This situation provides the selfish pool with a competitive edge and motivates rational miners to join it. [6] showed that once a selfish mining pool reaches a certain threshold, rational miners will preferentially join selfish miners to earn higher revenues compared to other pools.

. Authors of [9] formalized a bitcoin mining model incorporating the relevant features of mining and system parameters. Suppose the system contains a set of miners $1, \dots, n$ where each m_i denotes the mining power of i^{th} miner such that,

$$\sum_{i=1}^{n} m_i = 1$$

Without loss of generality, let us assume that these miners are divided into two groups as selfish miners (a minority pool those who keep their mined blocks private) and honest miners (A majority those who adhere to the standard bitcoin protocol and mines on the long public branch). Suppose that the pool controls α fraction of the computing power of the whole network and γ be the communication capability of the pool: The fraction of the honest miners that will first receive a block published by the pool if the pool and one honest miner choose to release their blocks approximately at the same time.

The main idea of selfish mining [9] is described as follows.

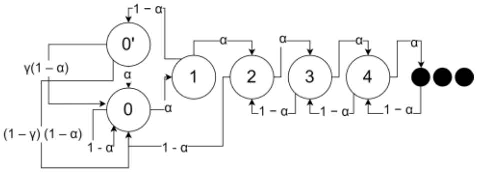
When the selfish mining group discovers a block, they gain an advantageous position by having a one-block lead over the public branch where honest miners are working. Instead of immediately sharing this private block and informing other miners about the discovery, selfish miners opt to keep it within their pool. At this juncture, two scenarios can unfold:

I. The honest miners find a new block on the public branch, thereby canceling out the pool's lead.
II. The pool manages to mine a second block, further widening its lead over the honest miners.

In scenario I, the pool chooses to publish its block to match the honest network. The selfish miners unanimously decide to adopt and extend the previously private branch, whereas honest miners make their choice based on the dissemination of notifications, opting to mine on either branch. If the selfish pool succeeds in mining a subsequent block before the honest miners who haven't adopted the pool's recently revealed block, it promptly publishes it to benefit from the earnings of both the first and second blocks of its branch. Conversely, if the honest miners mine a block following the pool's revealed block, the pool reaps the rewards of its block while the others gain revenue from their own block. Finally, if the honest miners mine a block subsequent to their own, they reap the benefits of their two blocks while the pool receives nothing.

In scenario II, should the selfish pool succeed in securing a second block, it establishes a two-block lead. Once reaching this stage, the pool persists in mining at the forefront of its private branch, releasing one block from its private branch for each block discovered by others. As the selfish pool represents a minority, its lead will likely diminish to just one block over time. At this juncture, the pool unveils its private branch. Given that the private branch surpasses the public branch by a block, all miners unanimously adopt it as the primary branch, allowing the pool to profit from all its blocks.

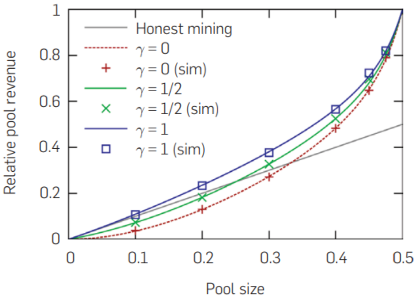
To investigate the above operation further, [9] presents a state machine capturing the transition frequencies involving α and γ.



Each state in the above state machine represents the lead: The difference between the length of the pool's private branch and the length of the public branch. Note the distinction between state 0 and 0'. State 0 is the state where there is only a single, global, public longest chain and the state 0' is the state where there are two public branches of length one: the main branch, and the branch that was private to the selfish miners and published to match the main branch. Transition between each state is triggered by a block mining operation either by the pool with a probability of α or honest network with a probability of $1-\alpha$.

Authors of [9] formulated the revenue rate of each agent: *revenue rate ratio* in the pool as given below.

$$R_{pool} = \frac{r_{pool}}{r_{pool} + r_{others}}$$
$$= \frac{\alpha(1-\alpha)^2(4\alpha + \gamma(1-2\alpha)) - \alpha^3}{1 - \alpha(1 + 2(2-\alpha)\alpha)}$$



The above figure illustrates the pool's revenue for different γ values with α varying from 0 to 0.5.

When $\gamma = 1$ selfish mining is much better than honest mining as the pool can quickly propagate its one-block to all the honest miners, so all honest miners will mine on the pool's block. However, when $\gamma = 0$ Honest mining looks more profitable for smaller pool sizes.

However as highlighted in [6], [11] the drawback of selfish mining is that it is not always more profitable than honest mining. It means there are certain scenarios where honest mining may shine. This limitation happens when nodes do not employ a large enough share of the computing resources and also due to propagation limitations in the network. In their study [6], they showed that an attacker can earn a higher profit if the pool has a mining power which exceeds 25% profit-threshold i.e. The minimal computational power an attacker needs in order to gain more than its fair share.

Selfish mining within the Bitcoin ecosystem has undergone thorough examination, with a range of mining strategies being put forth for consideration. Nevertheless, the prevailing selfish mining models cannot be readily transferred to Ethereum due to disparities in the mining framework compared to Bitcoin. This distinction arises from Ethereum's provision of two block rewards: uncle block reward, and nephew block reward in addition to the standard block reward in the Bitcoin system. Building upon the findings of [6], the authors of [10] have investigated the selfish mining strategy within the context of Ethereum. They introduced a two-dimensional Markov Decision Process (MDP) and tracked the block rewards in a probabilistic way to effectively model the behavior associated with selfish mining strategies capturing the impact of uncle and nephew rewards specific to Ethereum. Utilizing this mathematical framework, they have derived that the threshold of computational power required for an attacker to gain from selfish mining stands at 16.3%. This figure is notably lower than the profit threshold of 25% established in [6]. Indeed, the inclusion of uncle and nephew rewards in Ethereum's mining scheme amplifies the threat posed by selfish mining, making it a more significant concern for the Ethereum network compared to Bitcoin. This indicates that Ethereum exhibits a higher susceptibility to selfish mining compared to Bitcoin.

However, various authors have denied the profitability of selfish mining, based on inaccurate models of the Bitcoin protocol and pools work [37],[38],[39]. They argue that selfish mining is unprofitable because the time spent forking blocks ultimately reduces the growth rate of the main blockchain. For instance, during an attack, if the attacker generates *x* blocks and the rest of the network generates y blocks, all of the attacker's blocks will eventually be included in the blockchain, while only y−*x* of the other blocks will be included. As a result, the blockchain ultimately grows by only y blocks. Each block mined by the attacker is released strategically when another block is found by the network, thereby replacing the competitive block within the chain. This means that if the attacker mines *x* blocks, *x* blocks from the rest of the network will be discarded and replaced by the attacker's blocks. Consequently, the total blockchain growth rate is reduced to 1−*p* times the normal rate, where *p* is the proportion of the network's mining power controlled by the attacker. According to this view, any increase in relative revenue is offset by a decrease in profit per unit of time, rendering the strategy ineffective. Furthermore, critics argue that for selfish mining to be profitable, it would require long-duration attacks that extend beyond a single difficulty adjustment period. They contend that the necessity for prolonged attacks diminishes the feasibility and profitability of the selfish mining strategy. However, these claims proved to be false in later studies [29].

[8] presented a family of mining strategies called *stubborn mining* namely Lead Stubborn, Equal Stubborn and Trail Stubborn that go beyond and surpass the selfish mining approach allowing the miners to earn even higher rewards. The intuition behind stubborn mining strategies is the attacker can often gain more profit by mining on his private chain more frequently even if the attacker's private chain falls behind the honest chain. This stands in contrast to the selfish mining strategy, where the selfish miner withholds mined blocks and publishes them solely when they find themselves trailing the honest chain. Depending on the environmental parameters, their analysis [8] shows that stubborn mining strategies can surpass selfish mining by as much as 25%, even without employing any network-level attacks. Their analysis subsequently reveals that employing the trail-stubborn mining strategy can yield a 13% increase in gains compared to a non-trail-stubborn counterpart. Authors of [8] further show that an attacker can increase his reward by following non-trivial combinations of stubborn mining and network-level eclipse attacks [19, 62] by exploiting the network layer subsequently inspecting and controlling incoming and outgoing connections of the victim to further increase his/her revenue. Depending on the parameters, their work shows that these strategies can at times yield gains of up to 30% when compared with the naive utilization of eclipsed nodes.

As the selfish mining strategy proposed in [6] was not optimal, [9] extended the MDP mining models of [6], [8] to a more generalized form. In [9] authors showed that there are selfish mining strategies that allow miners to earn a higher reward and are also profitable for small miners compared to [6]. To do that, first, [9] formulated the mining problem as a single-player decision problem of the form $M \coloneqq \langle S, A, P, R \rangle$ where S is the state space, A is the action space, P is the stochastic transition matrix, and R is the reward matrix. However, M cannot be considered as an MDP as the objective function is nonlinear: The player aims to maximize its share of the accepted blocks, rather than the absolute number of its own accepted ones.

Let a be the number of blocks that have been built by the attacker after the latest fork (length of attacker chain) and h be the number of those built by honest nodes (length of the honest chain).

Action: Action space consists of four possible actions as given below.

- *Adopt*: The attacker accepts the honest chain and mines on the last block of the honest chain
- *Override:* The attacker publishes blocks to override the conflicting blocks of the honest chain, and is feasible whenever a > h.
- *Match*: The attacker publishes the same number of blocks as the honest chain to the whole network. This action creates a fork deliberately and initiates an open mining competition between the two branches of the adversary and the honest network.
- *Wait*: The attacker does not publish blocks and keeps mining on its own chain.

State: Each state in the state space is denoted by a tuple of size 3 of the form $(a,\ h,\ fork)$. The fork entry can take three values as given below.

- *Relevant*: $fork\ =\ relevant$, if the latest block is mined by the honest network. For instance, if the previous state was $(a,\ h-1,\ \bullet)$ and the honest network was able to mine a block then state changes to $(a,\ h,\ relevant)$.
- *Irrelevant*: $fork\ =\ irrelevant$, if the latest block is mined by the attacker. For instance, if the previous state was $(a-1,\ h,\ \bullet)$ and the pool was able to mine a block then state changes to $(a,\ h,\ irrelevant)$.

- Active: $fork\ =\ active$, if the pool has executed the action match from the previous state, and the blockchain is now split into two branches. It simply means a fork has been known to the public and there is active competition between the two branches.

Note that the initial state of the network is $(1,\ 0,\ irrelevant)$ with probability α or $(0, 1,\ relevant)$ with $1-\alpha$.

Transition and Reward Matrices: Execution of an action, every state transition corresponds to the creation of a new block either by the honest network or attacker. A description of the corresponding transition and reward matrices is shown in Table 1.

Objective Function: As derived in [9], we define the relative revenue of the attacker as given below.

$$REV\ :=E\left[\frac{\sum_{t=1}^{T} r_t^1(\pi)}{\sum_{t=1}^{T} r_t^1(\pi)+\sum_{t=1}^{T} r_t^2(\pi)}\right]$$

where $r_t^1(\pi)$, $r_t^2(\pi)$ is the immediate reward issued in the block interval t under the action defined by policy π, and T is the size of the observing window.

Based on the provided formula, the adversary's goal is to maximize the ratio of its total rewards to the total rewards achieved by the entire network but not to maximize its absolute total reward. In other words, the adversary aims to maximize its portion of accepted blocks rather than simply the absolute count of its own accepted blocks. This is because blockchain adjusts its mining difficulty regularly to maintain a consistent block production rate, for instance, approximately one block every 10 minutes in Bitcoin. This adjustment is crucial for the network's stability and security. As more miners join or leave the network, the total computing power dedicated to mining fluctuates. By adjusting the difficulty, Bitcoin ensures that blocks are neither generated too quickly nor too slowly, maintaining the intended block time and, consequently, the overall functionality and reliability of the network. To derive an optimal policy, since the objective function is nonlinear, [9] first transformed the model into a family of MDPs with linear objectives and then used a standard MDP solver combined with a numerical search over the family of MDPs to find the optimal mining strategy. Utilizing these MDP families [9] obtained upper and lower bounds on the attacker's profit. Based on their simulations [9], it was demonstrated that an attacker can achieve a greater portion of the rewards by employing a lower profit threshold of 23.21%, in contrast to the 25% profit threshold illustrated in the previous study [6].

However, a limitation of this approach is, this is a model-based approach. It means that this approach requires the knowledge of parameters of the network such as the attacker's computational power (α) and communication capability (γ). In real blockchain networks, these values are difficult to determine as they may change over time. Conversely, this model presumes that all miners, except for the adversarial pool, behave honestly, which is not a realistic assumption in actual blockchain environments. Other nodes may engage in malicious activities, such as eclipse attacks, deviating from honest mining behavior. Additionally, this model fails to account for various blockchain scenarios with differing stale block rates, confirmations, and real-world parameters like network delays. Consequently, there exists a substantial gap between this model and real-world blockchain networks.

Studies [6],[8],[9] have employed models to analyze the selfish mining strategy, operating under the premise of a singular selfish miner, while considering the potential presence of multiple colluding pools nearing the profitability threshold. Diverging from the aforementioned assumption, the authors of [56] proposed a novel model of a PoW blockchain that accommodates multiple independent attackers. In this context, "independent attackers" refers to attackers whose decision-making processes are independent of one another, although their state transitions are influenced by other miners. According to their formulated model, each attacker encounters a single-player decision problem. Given that each attacker must maintain their state, they defined the attacker's state as a 3-tuple $T\ =\ (lead, f_1, f_2)$ where $lead$ indicates the attacker's lead over the honest chain, f_1 denotes whether there is a fork in the main chain (indicating the existence of competition), and f_2 indicates if the attacker is involved in this competition. The action space for each attacker includes *Hold, Match, Override, Adopt*, and *Publish*. This is similar to the previously discussed MDP model, but with an additional action, *Publish*, which pertains to the event where $attacker_i$ publishes the head of his blocks. To evaluate the performance of their mining strategy for each attacker, they adopted the relative stale block rate associated with each attacker. Based on this model, they proposed a new strategy called publish-n (P_n). The intuition behind this strategy is to enable the attacker to shorten their private chain, which is advantageous in situations where they hold a long private chain but still lag behind the main chain. The value n acts as a trigger; when his state reaches n, when the attacker's state reaches n, they will either publish the first block of their private chain or execute the override action, depending on whether they have found the next block. In their demonstration, they evaluated original selfish mining, stubborn mining, and their new strategy, publish-n in a PoW blockchain environment with multiple attackers. Their results shows that publish-n can surpass selfish mining by an efficiency up to 26.3%.

Authors of [11] introduced a novel MDP to depict the state transitions between public and private chains, taking into account the presence of two selfish miners within the Bitcoin network and setting out a limit on the maximum length of the private chain. The selfish mining scenario was modeled with the inclusion of an honest pool representing all legitimate miners within the network, alongside two independent selfish miners unaware of each other's noncompliant behavior. In their paper, the authors addressed the question of profitability by considering both the hash rate of the attackers and the adjustments made to the mining difficulty. They demonstrated that for two selfish miners to achieve profitability, the minimum required threshold of profitability (hashrate) is symmetrically set at 21.48% and becomes more challenging when one of the selfish miners increases their hash rate. They also demonstrated that if the hash rates of selfish miners are both set at 22%, then the attackers can reap the benefits of selfish mining after 51 rounds of mining difficulty adjustments (equivalent to 714 days in Bitcoin), and this period reduces to 5 rounds (equivalent to 70 days) when the hash rates are increased to 33%.

[57]

Selfish mining strategy, however, confers no benefit to the adversary until a difficulty adjustment [37]. Selfish mining becomes particularly advantageous in scenarios where the difficulty adjustment does not immediately reflect the actual network conditions due to the withheld blocks. When a selfish miner withholds blocks and only releases them strategically, they create irregularities in block discovery times declining the block discovery rate. When the difficulty adjustment period arrives, the network adjusts the difficulty downward due to the perceived slower block discovery rate. As a consequence of this adjustment, the selfish miner now mines under a lower difficulty. They continue their strategy but now gain a higher proportion of the total block rewards with less computational effort. It's worth noting, however, that if the difficulty remains constant throughout this period, the immediate gains derived from the strategy might not sufficiently offset the associated costs. In the study [29], the authors introduced a profitable variant of the selfish mining strategy known as Intermittent Selfish Mining (ISM). In this variant, the attacker ceases to engage in selfish mining

State x Action	State	Probability	Reward
$(a, h, \bullet), adopt$	$(1,0, irrelevant)$	α	$(0, h)$
	$(0,1, irrelevant)$	$1-\alpha$	
$(a, h, \bullet), override$	$(a-h, 0, irrelevant)$	α	$(h+1,0)$
	$(a-h-1,1, relevant)$	$1-\alpha$	$(0,0)$
$(a, h, irrelevant), wait$ $(a, h, relevant), wait$	$(a+1,0, irrelevant)$	α	$(0,0)$
	$(a, h+1, relevant)$	$1-\alpha$	$(0,0)$
$(a, h, active), wait$ $(a, h, relevant), match$	$(a+1, h, active)$	α	$(0,0)$
	$(a-h, 1, relevant)$	$\gamma(1-\alpha)$	$(0,0)$
	$(a, h+1, relevant)$	$(1-\alpha).(1-\alpha)$	$(0,0)$

Table 1: State transition and reward matrices of the study [9] for optimal selfish mining

behavior immediately after a difficulty adjustment still managing to earn higher profits than honest miners meaning that the miner alternates between selfish and honest mining at every difficulty adjustment in Bitcoin. ISM consists of two distinct phases. In the first phase, the intermittent selfish miner (ISM) engages in selfish mining. During this phase, the ISM aims to undermine the honest miners by withholding blocks and strategically releasing them to maximize their own block rewards. The primary objective of this phase is to invalidate the blocks mined by honest participants, thereby positioning the attacker to gain a significant advantage in the subsequent epoch. Following the difficulty adjustment, the ISM transitions to honest mining in the second phase. With the difficulty level now reduced due to the slower block discovery rate induced by the earlier selfish mining, the mining process becomes more efficient. This lower difficulty results in a faster mining rate, benefiting all miners, but particularly allowing the ISM miner to gain a higher relative reward during two phases.

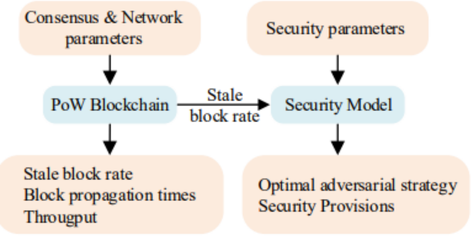
Mitigating issue of requiring the knowledge of various blockchain parameters such as adversary computational power and communication capability of [9], authors of [12] proposed a model-free, reinforcement learning (RL) based approach which allows a RL agent to dynamically learn a mining strategy with performance approaching that of the optimal mining strategy. For their work they adopted the MDP mining model proposed in [9] with Q-Learning algorithm to derive the optimal mining strategy. In their research, they employed a refined version of the Q-Learning algorithm, termed the Multidimensional Q-Learning algorithm, owing to the nonlinearity inherent in the objective function (Equation 1). Leveraging this multidimensional Q-Learning algorithm, they successfully optimized the nonlinear objective function to attain the optimal mining strategy. However, this approach faced limitations in its applicability to a real blockchain environment due to two primary reasons. Firstly, it relied on the model outlined in [9], which lacked consideration for real-world blockchain parameters like stale block rates, eclipsed attacks, propagation parameters, among others. Even if they were to adopt the mining model proposed in [12], it would present significant challenges for the RL agent to learn an optimal mining strategy, primarily due to the vast state-action space. In their study, they employed a tabular Q-Learning algorithm, which would prove highly inefficient for handling such large state-action spaces. Indeed, this issue directly impacts the convergence of the algorithm. If the algorithm requires a substantial amount of time to discover the optimal policy, it becomes economically unviable for miners. This is because prolonged computation time translates to increased expenses for hardware and computing power, diminishing the economic feasibility of mining operations.

As delineated in Section 1, within the Bitcoin ecosystem, every transaction is subject to validation by the nodes comprising the Bitcoin network before including in a publicly accessible distributed ledger, commonly referred to as the blockchain. Bitcoin blockchain is a decentralized, distributed ledger that records transactions across a network of computers in a secure and tamper-resistant manner. Each Bitcoin transaction is verified by a consensus mechanism among the network participants, and once validated, it is added to a chronological chain of blocks. As delineated in the study [18], malicious entities possess the capability to disrupt the synchronization of the ledger across multiple nodes, facilitating a form of attack known as the double-spending attack. This tactic involves redirecting previously validated transactions, thereby enabling the attackers to utilize the same coins twice. In this scheme, the attacker might initiate a transaction with a merchant and subsequently illicitly construct a longer chain of blocks excluding this transaction. By publishing their longer chain, they can prompt the ledger to undergo replacement, effectively nullifying the original transaction or redirecting the payment to another destination. In other words, in a double spending attack, a malicious actor spends the same cryptocurrency tokens twice by creating conflicting transactions in different parts of the network. This can be profitable if the attacker can control a significant portion of the network's hash rate. Furthermore, in the seminal paper by Satoshi Nakamoto [1], it is shown that if an attacker controls less than 50% of the computational power within the network, the likelihood of successful double-spend attacks diminishes exponentially over time. Typically, merchants and exchanges require multiple confirmations (blocks added after the transaction) to consider a transaction final and irreversible. The analysis presented in [18] indicates that the success of this attack is contingent not solely on the number of confirmations but also on the duration until the transaction is authorized. Given that blocks are typically generated approximately once every 10 minutes in Bitcoin, there exists a significant delay before a specific transaction is incorporated into the blockchain. While this form of attack is not inherently a mining attack, it can be enabled through manipulation of the mining process.

The 51% attack [17], [20], a form of mining attack wherein the attacker commands 51% of the computational power, or hashing power, of the entire network. This enables the attacker to mine blocks at a faster rate than other miners. In this scenario, the attacker initiates the creation of blocks privately, without disseminating them across the network, thereby maintaining its own version of the blockchain. Subsequently, the attacker reveals their private chain to the network at a later stage. Given that the attacker possesses more than half of the hashing power of the entire network, they can establish the longest chain by influencing network nodes to adopt their published chain [1]. This capability grants the attacker the opportunity to engage in double-spending. Specifically, if the attacker executes multiple transactions before revealing their private chain, and neglects to include those transactions in their own private chain, even though those transactions are confirmed and added to the public chain, the network will accept the attacker's private chain once it is published, superseding the public chain. Consequently, the attacker can exploit this situation to double-spend their coins. For instance, as outlined in [17], the GHash.IO pool temporarily commanded 54% of the total hashrate, surpassing the critical threshold of 51% widely recognized as the theoretical point for potential network vulnerability. Promptly, the community took remedial actions by redistributing mining resources to alternative pools. However, the underlying incentives driving the formation of sizable pools persist, thereby posing an ongoing risk of potential network disruption, should such concentrations of mining power arise again.

In the study [12], authors introduced an innovative quantitative framework designed to assess the security and performance associated with different consensus and network parameters within Proof of Work (PoW) blockchains. Their framework comprised two essential components, namely a blockchain instance and a blockchain security model. In their model, they instantiated the PoW blockchain with various consensus and network parameters, such as network delays, block propagation times, block sizes, and information propagation mechanisms. This approach aimed to more realistically capture real-world blockchain instances. The output of their blockchain instance was the stale block rate, which served as input to their security framework which was based on MDPs for selfish mining and double-spending building upon the work laid out by [9], but encompassing more real-world features such as stale block rates, mining power, mining costs, the number of block confirmations, propagation ability, eclipse attacks [19] to quantify the optimal adversarial strategies for and selfish mining and double-spending.

In their study [12], similar to the work presented in [9], the authors modeled the blockchain framework as a single-player decision problem, denoted as $M \coloneqq \langle S, A, P, R \rangle$. Here, S represents the state space, A denotes the action space, P is the stochastic transition matrix, and R is the reward matrix. The complete state transition and reward matrices for optimal selfish mining and double-spending is shown in Table 2. In addition to the *Adopt*, *Override*, *Match*, and *Wait* actions included in the model from [9], their action space also incorporated a new action called *Exit*. This action is specifically relevant for analyzing double-spending attacks, applicable when there is a successful double-spending with k confirmations and is feasible only if $l_a > l_h$, and $l_a > k$. Furthermore, their work distinguishes itself from that of [9] by defining the state space S as a four-tuple in the form $(l_a, l_h, b_e, fork)$ where l_a and l_h represent the lengths of the adversarial and honest chains, respectively; b_e represents the blocks mined by the eclipsed victim; and, similar to the MDP model by [9], $fork$label is categorized as either *relevant*, *irrelevant*, or *active*.



In their analysis they considered selfish mining and double-spending strategies separately because a selfish miner aims to maximize the ratio of his cumulative rewards over the cumulative rewards of the whole network while following the double-spending strategy adversary aims to maximize his absolute reward. To derive the optimal selfish mining policy, they employed the same approach as [9], transforming the single-player decision problem M into a family of MDPs with linear objectives. They restricted the state space of the MDP family to a cutoff length of 20 i.e. only allowing either chain to be length at most 30 resulting in a finite state MDP and then used a standard MDP solver combined with a numerical search over the family of MDPs to determine the optimal strategy for selfish mining.

In their study, they analyzed the impact of stale block rates, using values of 1% and 10%, which reflect different block sizes, block intervals, network delays, information propagation mechanisms, and network configurations. Their analysis indicates that as adversarial mining power increases, the relative revenue of a selfish miner grows, surpassing the upper bound formulated by [9]. They also studied the impact

State x Action	State	Probability	Reward
$(l_a, l_h, b_e, \cdot), adopt$	$(1{,}0{,}0, i)$	α	$(-c_m, l_h)$
	$(1{,}0{,}1, i)$	ω	
	$(0{,}1{,}0, i)$	$(1-\alpha-\omega)\cdot(1-r_s)$	
	$(0{,}0{,}0, i)$	$(1-\alpha-\omega)\cdot(r_s)$	
$(l_a, l_h, b_e,$ $\cdot), override$	$\left(l_a - l_h, 0, b_e - \left\lceil (l_h+1)\frac{b_e}{l_a} \right\rceil, i\right)$	α	$\left(\left\lfloor (l_h+1)\frac{l_a - b_e}{l_a} \right\rfloor - c_m, 0\right)$
	$\left(l_a - l_h, 0, b_e - \left\lceil (l_h+1)\frac{b_e}{l_a} \right\rceil, i\right)$	ω	$\left(\left\lfloor (l_h+1)\frac{l_a - b_e}{l_a} \right\rfloor - c_m, 0\right)$
	$\left(l_a - l_h - 1{,}1, b_e - \left\lceil (l_h + 1)\frac{b_e}{l_a} \right\rceil, r\right)$	$(1-\alpha-\omega)\cdot(1-r_s)$	$\left(\left\lfloor (l_h+1)\frac{l_a - b_e}{l_a} \right\rfloor - c_m, 0\right)$
	$\left(l_a - l_h - 1{,}1, b_e - \left\lceil (l_h + 1)\frac{b_e}{l_a} \right\rceil, i\right)$	$(1-\alpha-\omega)\cdot(r_s)$	$\left(\left\lfloor (l_h+1)\frac{l_a - b_e}{l_a} \right\rfloor - c_m, 0\right)$
$(l_a, l_h, b_e, i), wait$ $(l_a, l_h, b_e, r), wait$	$(l_a + 1, l_h, b_e, i)$	α	$(-c_m, 0)$
	$(l_a + 1, l_h, b_e + 1, i)$	ω	
	$(l_a, l_h + 1, b_e, r)$	$(1-\alpha-\omega)\cdot(1-r_s)$	
	(l_a, l_h, b_e, i)	$(1-\alpha-\omega)\cdot(r_s)$	
$(l_a, l_h, b_e, a), wait$ $(l_a, l_h, b_e, r), match$	$(l_a + 1, l_h, b_e, a)$	α	$(-c_m, 0)$
	$(l_a + 1, l_h, b_e + 1, a)$	ω	$(-c_m, 0)$
	$\left(l_a - l_h, 1, b_e - \left\lceil (l_h)\frac{b_e}{l_a} \right\rceil, r\right)$	$\gamma\cdot(1-\alpha-\omega)\cdot(1-r_s)$	$\left(\left\lfloor (l_h)\frac{l_a - b_e}{l_a} \right\rfloor - c_m, 0\right)$
	$(l_a, l_h + 1, b_e, r)$	$(1-\gamma)\cdot(1-\alpha-\omega)\cdot(1-r_s)$	$(-c_m, 0)$
	(l_a, l_h, b_e, a)	$(1-\alpha-\omega)\cdot(r_s)$	$(-c_m, 0)$
$(l_a, l_h, b_e, \cdot), exit$	$exit$	1	$(l_a - b_e + v_d, 0)$

Table 2: State transition and reward matrices of the study [12] for optimal selfish mining and double-spending

of eclipse attacks on selfish mining, considering scenarios where the adversary exploits the victim's mining power ω and uses all the victim's blocks to advance their private chain. The analysis indicates that higher the ω stronger the adversary's selfish mining capabilities become. Their analysis further indicates that an adversary possessing 30% of the mining power can acquire an average of 209 block rewards by following their MDP mining strategy when 1000 blocks are mined by the entire network, whereas according to [10], the average yield is 205.8 block rewards.

Another form of mining attack, known as Pool Hopping attack [23], involves miners strategically timing their participation in mining pools, alternately directing their computational resources towards the pool, and diverting them elsewhere. By doing so, these miners aim to gain rewards that exceed what would be considered equitable based on their proportional contribution to the pool's total computational power, making other miners earn a lesser reward. The most recognized manifestation of pool-hopping occurs within pools employing the proportional method (Section 1), renowned for its simplicity, widespread adoption, and susceptibility to hopping tactics. As delineated in Section 1, the reward allocated for each share is denoted as $\frac{B}{N}$ and the distribution of a block's reward among miners is proportional to the quantity of shares each miner has submitted since the preceding block ($\frac{nB}{N}$). Consequently, the value of a share submitted at any given moment is influenced by the total number of shares submitted since the last block was found.

In other words, as the duration of a mining round extends, the value of each individual share becomes less worth. Consequently, it becomes advantageous for miners to strategically submit shares to the pool during shorter rounds and redirect their efforts elsewhere during longer rounds. This strategy is driven by the understanding that participating in a pool with a substantial accumulation of shares and no corresponding discovery of a block diminishes the expected reward, as it will be distributed among all contributing miners. Thus, there arises a point where it becomes financially advantageous to cease mining within such a pool and allocate resources elsewhere.

In their study [24], the authors introduce two types of mining attacks: the Block Discarding attack and the Difficulty Raising attack. The concept behind the Block Discarding attack is akin to the Selfish Mining strategy proposed in [6], where an attacker withholds newly mined blocks until a competitor finds a block, at which point the attacker releases their withheld block. However, there are several key distinctions between these approaches. Firstly, the authors of [24] introduce a parameter called Network Superiority (ns), analogous to the communication capability of the adversary described in [6]. They analyze the Block Discarding attack through a hierarchical family of strategies denoted as st_k, where k represents different strategies ($k = 0, 1, 2, ...$) tailored to various combinations of attacker hashrates (p) and ns values. In this framework, the Selfish Mining strategy from [6] is considered the simplest case, corresponding to $k = 1$. Secondly, unlike the Selfish Mining strategy, which was initially designed for mining pools, the authors of [24] argue that maintaining block secrecy within pools is highly challenging. Therefore, they propose that the Block Discarding attack is more feasible for execution by solo miners rather than mining pools. This distinction emphasizes the tailored approach for solo miners, aiming to exploit the network's vulnerabilities more effectively without the complexities involved in coordinating a mining pool's efforts. Another notable difference is that the study in [6] describes a process that continues indefinitely until the attacker's pool completely ousts all other miners. In contrast, the study in [24] argues that a more likely outcome is the establishment of a new equilibrium where there are fewer honest miners, but those remaining continue to maintain the same profit levels due to the difficulty adjustment making mining easier by lowering the difficulty level to maintain the same block rate. Alternatively, the study suggests a scenario where all honest miners might eventually leave the system entirely.

The Difficulty Raising Attack which is the second kind of attack proposed in [24], involves adjusting the difficulty of their own chain. Here also the attacker operates a competitive blockchain characterized by blocks that lack correlation with the honest chain. Concurrently, the attacker manipulates the automatic difficulty adjustment mechanism within their clandestine chain to augment the likelihood of surpassing the public honest chain. Upon achieving this objective, the attacker discloses their secret chain.

Block withholding (BWH) attack is another adversarial mining strategy which was initially proposed in the study [23] where a malicious miner participates in a mining pool but withholds valid blocks, causing financial harm to the pool. The malicious attacker initiates his attack by joining a mining pool just like any regular miner and contributing his computational power to the mining pool. As discussed in the previous sections, the attacker starts to mine and submits partial PoW solutions (PPoW) to the pool, which are required to demonstrate their participation and effort. However, if the miner finds a Full Proof-of-Work (FPoW) solution, that meets the network's difficulty target and can be added to the blockchain, instead of submitting this valid block to the pool operator, the attacker withholds it. This means the block is not broadcast to the network or the pool. Since the valid block is withheld, the mining pool does not receive the block reward and transaction fees associated with that block. As a result, the total earnings of the pool decrease, financially harm the pool, making it less profitable and potentially driving miners to leave the pool. Numerous studies have examined various methods of block withholding attacks, including those targeting *dual mining pools* [30-35], *multiple mining pools* [40-43], and *hybrid block withholding attacks* [44-47]. These studies provide a comprehensive understanding of the mechanisms and impacts of such attacks on the blockchain ecosystem.

A BWH attack in dual mining pool is a sophisticated form of BWH attack that combines elements of block withholding and exploits the dynamics of mining pools by two mining pools to launch BWH on each other [30]. In this attack, the attacker opts to withhold blocks from one pool while submitting them to the other, reducing the profitability of competing pools that can drive miners to their own pool, increasing their relative hash power and influence. As discussed in [30], a mining pool can utilize this attack to gain competitive advantage by making miners to infiltrate or damage the other mining pools and conduct block withholding attacks on other mining pools to obtain revenue in order to increase the profitability of their own mining pools. For instance, if a pool P_1 wants to infiltrate a potential competitive pool P_2, P_1 sends a malicious miner M to P_2 making M to submit partial PoW solutions to P_2 and if M finds a valid block that meets the network's difficulty, instead of submitting the valid block to P_2, M withholds it and does not broadcast it to the network. Therefore, without effectively mining, yet still receiving rewards which are then redirected back to the original mining pool P_1, thereby increasing the income of mining pool P_1 at the expense of mining pool P_2. Sometimes, the miners who were originally supposed to infiltrate pool P_2 will betray pool P_1 by honestly mining on P_2 which in turn reducing the revenue of P_1.

The authors of [30] proposed a game theory [48] based multi-pool mining model for mining pools operating under the PoW consensus algorithm. This model incorporates a reward and punishment system designed to capture the attack behaviors of mining pools in a blockchain network. By analyzing both pure strategy (where a mining pool consistently selects a specific strategy) and mixed strategy (where a mining pool randomly selects a strategy with certain probabilities) Nash equilibriums, they provide a comprehensive framework for understanding these interactions. In their model, when a mining pool decides not to engage in attacks, it receives an additional reward, denoted as a (where $0 \leq a \leq 1$), provided by the system. Conversely, a mining pool that opts to attack is subjected to a penalty of ka (where $k \geq 1$ is the penalty-to-reward ratio). Furthermore, their study formulated a mining pool game model from the perspective of block withholding attacks between mining pools considering the infiltrate rate and betrayal rate of the mining pool to analyze the Nash equilibrium and the value of infiltrate rate under the Nash equilibrium. In the study [31], the authors proposed a more generalized model in which two participants can choose either to cooperate with each other or to employ a BWH within a mining pool. This model is designed to calculate the equilibrium analysis by considering two distinct costs: the cost of cooperation and the cost of generating partial proofs of work. To enhance the measurement of performance over a given period, they introduced a parameter called "payoff per time." Furthermore, the authors extended their model to account for scenarios where the payoff per time, the cost of cooperation, and the cost of generating partial proofs of work are directly related to computational power.

Previous studies have predominantly analyzed Block Withholding (BWH) attacks in the context of interactions between two mining pools. However, in the real world, multiple mining pools exist, and these pools can launch BWH attacks against one another. As the number of mining pools and miners increases, directly applying dual mining pool methods for optimizing block withholding attacks becomes impractical. Most researchers have focused on the simplistic scenario of a one-shot game between only two mining pools attacking each other. In reality, the competitive landscape involves multiple pools of varying sizes.

In the study [41], the authors addressed this complexity by formulating the BWH attack as a stochastic game with finitely many states and actions. They introduced the concept of dynamic migration of miners among mining pools. When miners decide to migrate from one pool to another, for instance because of their average revenue reduces, their destination is not predetermined, as they lack information about which pools are currently being attacked or are likely to be attacked in the future. Consequently, migrating miners make stochastic choices regarding their new destination, adding a layer of unpredictability to the game. This approach provides a more realistic framework for understanding and analyzing BWH attacks in a multi-pool environment. In their game model, the authors considered a total number of S_T miners, distributed across n mining pools of varying sizes as well as solo mining. Each pool has an associated parameter called attractiveness ($A \in \mathbb{R}$), which determines the willingness of miners to join or remain in the pool. The attractiveness parameter ranges from $A = 0$, indicating the minimum possible attractiveness, to $A = 1$, indicating the maximum possible attractiveness. When a pool is subject to a Block Withholding (BWH) attack, its revenue decreases, thereby reducing its attractiveness. Miners perceive that this pool is less fortunate or is being targeted by other pools, leading to a decrease in its attractiveness. The attractiveness of a pool directly influences its size, as miners decide at the end of each round whether to remain in their current pool or migrate to another based on the attractiveness levels. To analyze the dynamics of this game, the authors incorporated reinforcement learning, specifically Tile Coding as proposed by [55]. Since the ultimate goal of the pool is to increase its income by increasing its size i.e. the total number of miners in the pool, to evaluate the game and they modeled the utility of each pool in each round proportional to the change in its size.

$$u_i^t = \frac{S_i^t - S_i^{t-1}}{S_T}$$

Where S_i^t, S_i^{t-1} denotes the size of the pool i in the rounds t and $t-1$ respectively.

In the study [36], the authors introduce a novel sophisticated attack known as the Fork After Withholding (FAW) attack, which combines elements of the Block Withholding (BWH) attack with selfish mining strategies. This hybrid approach enables an adversary to achieve higher profits than a standard BWH attack, irrespective of their computational power or network capability. In a FAW attack, the adversary joins a target mining pool and strategically divides their computational power between legitimate mining and infiltration mining, similar to a BWH attack. In a traditional BWH attack, if the attacker discovers an FPoW solution, they withhold it without submitting it to the pool manager. In contrast, during a FAW attack, the attacker does not immediately propagate the FPoW solution to the pool manager. Instead, the attacker waits for an external honest miner to publish their FPoW solution first. Once this occurs, the attacker then propagates their withheld FPoW solution to the pool manager. If the pool manager accepts the submitted FPoW from the attacker, the manager propagates it to the network, resulting in the creation of a fork. All participants in the Bitcoin network must then choose between the competing branches. If the attacker's block is selected as the valid chain, the target pool receives the reward, and the attacker is also rewarded by the pool. By employing this method, an FAW attacker can secure additional rewards regardless of the specific outcome, as the target pool's successful inclusion of the attacker's block ensures that the attacker benefits. This makes the FAW attack particularly potent and profitable compared to a BWH attack, as it maximizes the attacker's gains through strategic block propagation. In their study, the authors demonstrate that an FAW attacker can earn significantly higher rewards—ranging from one to four times more—compared to a BWH attacker within a large pool that controls approximately 20% of the computational power of the entire Bitcoin network. Furthermore, the study extends the FAW attack to multiple pools, thereby enabling the attacker to accumulate even greater rewards. Their analysis reveals that if an attacker targets four popular mining pools with the FAW attack, their additional reward can be approximately 56% greater than that of a BWH attacker.

Bribery attacks (61)

5. Future Work

Blockchain technology, which underpins cryptocurrencies such as Bitcoin, relies on decentralized consensus mechanisms. Mining plays a crucial role in maintaining the integrity and security of these blockchains. However, selfish mining poses a significant threat to the stability and fairness of these networks. Despite extensive research on various selfish mining strategies, the application of advanced machine learning techniques, particularly reinforcement learning (RL), remains largely unexplored. Reinforcement learning, with its capability to optimize decision-making processes through iterative trial and error, presents a promising approach for developing more sophisticated and potentially more effective selfish mining strategies. As discussed in the previous section, there is potentially a significant number of RL applications modeling BWH attacks which has proven to be effective. However, currently, there is only one study that has directly applied reinforcement learning to achieve optimal selfish mining strategies indicating a notable gap in the literature and presenting substantial opportunities for further investigation [13]. However, the MDP model that the RL agent has been trained in has a significant gap modeling a real blockchain environment. Furthermore, the tabular Q-Learning model, the authors incorporated in their study is rather inefficient in terms of convergence when performing in a real blockchain environment. By focusing on achieving an optimal selfish mining strategy through reinforcement learning, our future work aims to address this critical research gap. By exploring the application of RL in selfish mining, our future research will contribute to a deeper understanding of how reinforcement learning can be utilized to achieve optimal selfish mining in blockchain networks while filling the existing research gaps.

6. Conclusion

References

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[2] A. M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Crypto-Currencies, ed., vol. , , : O'Reilly Media, , p.

[3] Hattab, siham, & Taha Alyaseen, I. F. . (2019). Consensus Algorithms Blockchain: A comparative study. *International Journal on Perceptive and Cognitive Computing*, *5*(2), 66–71. https://doi.org/10.31436/ijpcc.v5i2.103

[4] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, "A review on consensus algorithm of blockchain," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 2567-2572, doi: 10.1109/SMC.2017.8123011.

[5] N. Chaudhry and M. M. Yousaf, "Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities," 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan, 2018, pp. 54-63, doi: 10.1109/ICOSST.2018.8632190.

[6] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable,” Communications of the ACM, vol. 61, no. 7, pp. 95–102, 2018.

[7] D. Kraft, “Difficulty control for blockchain-based consensus systems,” Peer-to-Peer Networking and Applications, vol. 9, no. 2, pp. 397–413, 2016.

[8] K. Nayak, S. Kumar, A. Miller, and E. Shi, “Stubborn mining: Generalizing selfish mining and combining with an eclipse attack,” in 2016 IEEE European Symposium on Security and Privacy (EuroS P). Saarbrucken, Germany: IEEE, March 2016, pp. 305–320.

[9] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, “Optimal selfish mining strategies in bitcoin,” in International Conference on Financial Cryptography and Data Security. Springer, 2016, pp. 515–532

[10] Chen Feng, Jianyu Niu: Selfish Mining in Ethereum. ICDCS 2019: 1306-1316.

[11] Qianlan Bai, Xinyan Zhou, Xing Wang, Yuedong Xu, Xin Wang, Qingsheng Kong: A Deep Dive Into Blockchain Selfish Mining. ICC 2019: 1-6.

[12] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, ¨ and S. Capkun, “On the security and performance of proof of work blockchains,” in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, 2016, pp. 3–16.

[13] Taotao Wang, Soung Chang Liew, Shengli Zhang: When Blockchain Meets AI: Optimal Mining Strategy Achieved By Machine Learning. CoRR abs/1911.12942 (2019).

[14] Romiti, M., Judmayer, A., Zamyatin, A., & Haslhofer, B. (2019). A deep dive into bitcoin mining pools: An empirical analysis of mining shares. arXiv preprint arXiv:1905.05999.

[15] Yonatan Sompolinsky, Aviv Zohar: Secure High-Rate Transaction Processing in Bitcoin. Financial Cryptography 2015: 507-527.

[16] Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

[17] Martijn Bastiaan: Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin[C]//Availab le at http://referaat. cs. utwente. nl/conference/22/paper/7473/preventingthe-51-attack-astochasticanalysis-oftwo-phase-proof-of-work-in-bitcoin. pdf. 2015.

[18] Yonatan Sompolinsky, Aviv Zohar: Secure High-Rate Transaction Processing in Bitcoin. Financial Cryptography 2015: 507-527.

[19] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. 2015.

[20] Sayeed, Sarwar, and Hector Marco-Gisbert. "Assessing blockchain consensus and security mechanisms against the 51% attack." Applied sciences 9.9 (2019): 1788.

[21] Seb Neumayer, Mayank Varia, Ittay Eyal: An Analysis of Acceptance Policies For Blockchain Transactions. IACR Cryptology ePrint Archive 2018: 40 (2018).

[22] Marianna Belotti, Sofiane Kirati, Stefano Secci. Bitcoin Pool-Hopping Detection. 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), Sep 2018, Palerme, Italy. ff10.1109/RTSI.2018.8548376ff. ffhal-02481221f
[23] Meni Rosenfeld: Analysis of Bitcoin Pooled Mining Reward Systems, Preprit Dec 2011, http://arxiv.org/abs/1112.4980

[24] Lear Bahack: Theoretical Bitcoin Attacks with less than Half of the Computational Power (draft), http://eprint.iacr.org/2013/868.

[25] Wood, Daniel Davis. "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER." (2014).

[26] Cash, Bitcoin. "Bitcoin cash." Development 2 (2019).

[27] Jumaili, Mustafa Lateef Fadhil, and Sulaiman M. Karim. "Comparison of tow two cryptocurrencies: Bitcoin and Litecoin." Journal of Physics: Conference Series. Vol. 1963. No. 1. IOP Publishing, 2021.

[28]https://ocw.mit.edu/courses/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/47a24e96943c8ee02a774dc52f300e29_Lecture20FinalPart1.pdf

[29] Negy, Kevin Alarcón, Peter R. Rizun, and Emin Gün Sirer. "Selfish mining re-examined." International Conference on Financial Cryptography and Data Security. Cham: Springer International Publishing, 2020.

[30] W. Li, M. Cao, Y. Wang, C. Tang and F. Lin, "Mining Pool Game Model and Nash Equilibrium Analysis for PoW-Based Blockchain Networks," in IEEE Access, vol. 8, pp. 101049-101060,2020,doi: 10.1109/ACCESS.2020.2997996.

[31] Wu, D., Liu, X. D., Yan, X. B., Peng, R., & Li, G. (2019). Equilibrium analysis of bitcoin block withholding attack: A generalized model. Reliability Engineering & System Safety, 185, 318-328.

[32] R. Qin, Y. Yuan, F. Wang Optimal block withholding strategies for blockchain mining pools IEEE Trans. Comput. Social Syst., 7 (3) (2020), pp. 709-717

[33] T. Yang, Z. Xue Game theory among mining pools in blockchain system

[34] Hu, Q., Wang, S., & Cheng, X. (2019, June). A game theoretic analysis on block withholding attacks using the zero-determinant strategy. In Proceedings of the International Symposium on Quality of Service (pp. 1-10).

[35] Wang, Y., Tang, C., Lin, F., Zheng, Z., & Chen, Z. (2019). Pool strategies selection in pow-based blockchain networks: Game-theoretic analysis. *IEEE Access*, *7*, 8427-8436.

[36] Kwon, Y., Kim, D., Son, Y., Vasserman, E., & Kim, Y. (2017, October). Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 195-209).

[37] Grunspan, C., & Pérez-Marco, R. (2018). On profitability of selfish mining. arXiv preprint arXiv:1805.08281.

[38] Courtois, N.T., Bahack, L.: On subversive miner strategies and block withholding attack in bitcoin digital currency. arXiv preprint arXiv:1402.1718 (2014)

[39] Wright, C.S.: The Fallacy of the Selfish Miner (1): Economic Argument Critiqued (2017)

[40] Matthew, J., Page, J.E., McKenzie, P.M., Bossuyt, I., Boutron, T.C., Hoffmann, C.D., Mulrow, L., Shamseer, J.M., Tetzlaff, Elie, A., Akl, Sue, E., Brennan, et al.: The prisma 2020 statement: An updated guideline for reporting systematic reviews. Bmj, 372, (2021)

[41] Haghighat, A. T., & Shajari, M. (2019). Block withholding game among bitcoin mining pools. *Future Generation Computer Systems*, *97*, 482-491.

[42] Kim, S., & Hahn, S. G. (2019). Mining pool manipulation in blockchain network over evolutionary block withholding attack. *IEEE Access*, *7*, 144230-144244.

[43] WANG, T., YU, S., & XU, B. (2019). Research on proof of work mining dilemma based on policy gradient algorithm. *Journal of Computer Applications*, *39*(5), 1336.

[44] Dong, X., Wu, F., Faree, A., Guo, D., Shen, Y., & Ma, J. (2019). Selfholding: A combined attack model using selfish mining with block withholding attack. *Computers & Security*, *87*, 101584.

[45] Ke, J., Szalachowski, P., Zhou, J., Xu, Q., Yang, Z. (2019). IBWH: An Intermittent Block Withholding Attack with Optimal Mining Reward Rate. In: Lin, Z., Papamanthou, C., Polychronakis, M. (eds) Information Security. ISC 2019. Lecture Notes in Computer Science(), vol 11723. Springer, Cham. https://doi.org/10.1007/978-3-030-30215-3_1

[46] Chang, SY., Park, Y., Wuthier, S., Chen, CW. (2019). Uncle-Block Attack: Blockchain Mining Threat Beyond Block Withholding for Rational and Uncooperative Miners. In: Deng, R., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds) Applied Cryptography and Network Security. ACNS 2019. Lecture Notes in Computer Science(), vol 11464. Springer, Cham. https://doi.org/10.1007/978-3-030-21568-2_12

[47] Wang, Y., Yang, G., Li, T., Zhang, L., Wang, Y., Ke, L., ... & Yu, X. (2020). Optimal mixed block withholding attacks based on reinforcement learning. *International Journal of Intelligent Systems*, *35*(12), 2032-2048.

[48] Fudenberg, Drew, and Jean Tirole. *Game theory*. MIT press, 1991.

[49] Vidal-Tomás, D. (2022). Which cryptocurrency data sources should scholars use?. *International Review of Financial Analysis*, *81*, 102061.

[50] Glenski, M., Saldanha, E., & Volkova, S. (2019, May). Characterizing speed and scale of cryptocurrency discussion spread on reddit. In *The World Wide Web Conference* (pp. 560-570).

[51] Fang, L., Azmi, E., Hor, B., & Win, K. W. (2021). *How to DeFi: Advanced* (Vol. 1). CoinGecko.

[52] Bitcoin hashrate distribution, https://blockchain.info/pools, [Online; accessed 8-December-2017].

[53] Statistics - etherchain.org - the ethereum blockchain explorer, https:// etherchain.org/statistics/miners, [Online; accessed 8-December- 2017].

[54] Xmr mining network, http://minexmr.com/pools.html, [Online; ac- cessed 8-December-2017].

[55] Bowling, M., & Veloso, M. (2002, July). Scalable learning in stochastic games. In *AAAI Workshop on Game Theoretic and Decision Theoretic Agents* (pp. 11-18).

[56] Q.H. Liu, N. Ruan, et al. "On the Strategy and Behavior of Bitcoin Mining with N-attackers". Proc. of the Asia Conference on Computer and Communications Security, pp. 357-368, 2018.

[57] S. Zhang, et al., "Analysing the Benefit of Selfish Mining with Multiple Players," in 2020 IEEE International Conference on Blockchain (Blockchain), Rhodes Island, Greece, Nov. 2020, pp. 36–44.

[58] C. Hou et al., "SquirRL: Automating Attack Analysis on Blockchain Incentive Mechanisms with Deep Reinforcement Learning," presented at the Network and Distributed System Security Symposium, Virtual, 2021.

[59] Bai, Q., Xu, Y., Liu, N., & Wang, X. (2023). Blockchain mining with multiple selfish miners. *IEEE Transactions on Information Forensics and Security*.

[60] T. Li, et al., "Semi-selfish mining based on hidden Markov decision process," Int J Intell Syst, vol. 36, no. 7, pp. 3596–3612, Jul. 2021.

[61] S. Gao, et al., "Power Adjusting and Bribery Racing: Novel Mining Attacks in the Bitcoin System," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London United Kingdom, Nov. 2019, pp. 833–850.

[62] Arthur Gervais, Hubert Ritzdorf, Ghassan O Karame, and Srdjan Capkun. 2015. Tampering with the delivery of blocks and transactions in bitcoin. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 692–705