Q-Defense: When Q-Learning Comes to Help Proof-of-Work Against the Selfish Mining Attack

Ali Nikhalat-Jahromi[1][a], Ali Mohammad Saghiri[1,2][b] and Mohammad Reza Meybodi[1][c]

[1]*Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran*

[2]*Department of Computer Science, William Paterson University, New Jersey, U.S.A.*

Keywords: Blockchain, Proof-of-Work, Selfish Mining, Defense Mechanism, Reinforcement Learning, Q-Learning.

Abstract: The Proof-of-Work (PoW) consensus protocol is widely utilized in various blockchain implementations, including Bitcoin. The security of this protocol relies heavily on the incentive-compatibility of participating *miner*, who compete against each other to discover new blocks. However, the assumption that competition will naturally evolve into collaboration, ensuring blockchain security, is not always valid. Certain colluding miners, known as ”*selfish miners*,” attempt to unfairly obtain rewards by deviating from the prescribed protocol. In this paper, we propose a novel learning-based mechanism to address this challenge and enhance the PoW protocol. Specifically, we apply *Q-Learning*, a prominent technique in *reinforcement learning*, to each miner in order to mitigate the impact of selfish collaboration among colluding miners. To best of our knowledge, this is the first defense mechanism based on Q-Learning in the literature. Our comprehensive analysis demonstrates that the proposed modification to the PoW protocol can increase the threshold for successful selfish mining attacks from 25% to 40%. Furthermore, simulation results comparing our defense mechanism with tie-breaking, a well-known defense approach, validate the effectiveness of our proposed mechanism.

1 INTRODUCTION

In recent years, the emergence of blockchain (Nakamoto, 2008) technology has garnered considerable interest and has been recognized for its transformative potential across numerous industries. By offering a secure and decentralized platform (Narayanan et al., 2016), (Antonopoulos, 2014) for transactions and data storage, blockchain has paved the way for significant advancements. Initially introduced in 2009 by an enigmatic individual named Satoshi Nakamoto, blockchain found its first implementation through Bitcoin(Nakamoto, 2008). Since its inception, this technology has gained widespread popularity, with the introduction of various cryptocurrencies, most of which are based on the proof-of-work consensus mechanism (Bonneau et al., 2015), (Tschorsch and Scheuermann, 2016), (Wang et al., 2019), (Judmayer et al., 2022).

In the proof-of-work consensus mechanism (Mingxiao et al., 2017), (Chaudhry and Yousaf, 2018), (Panda et al., 2019), exemplified by Bitcoin's current operation, a group of nodes known as miners engage in a competitive effort to uncover new blocks. The process of discovering a new block serves two important purposes. Firstly, it involves recording transactions, ensuring their inclusion in the blockchain. Secondly, it extends the chain of blocks by introducing a new block, reinforcing the integrity and continuity of the blockchain (Narayanan et al., 2016), (Antonopoulos, 2014).

In the process of mining (Mingxiao et al., 2017), ensuring the principle of incentive-compatibility (Courtois and Bahack, 2014), (Eyal, 2015), (Babaioff et al., 2012) is crucial for maintaining network security. Miners actively compete against each other in the quest to discover blocks, driven by the desire to obtain two types of rewards. Firstly, they seek the reward associated with successfully mining a block. Secondly, they aim to collect transaction fees that are included in the blocks they discover (Antonopoulos, 2014), (Eyal and Sirer, 2018).

The assumption that all miners in the network consistently adhere to the rules is flawed (Eyal and Sirer, 2018), (Bai et al., 2019). In reality, there are instances where a group of colluding miners forms a

[a] https://orcid.org/0000-0001-8398-8058
[b] https://orcid.org/0000-0003-0797-314X
[c] https://orcid.org/0000-0003-3775-5565

Nikhalat-Jahromi, A., Saghiri, A. and Meybodi, M.
Q-Defense: When Q-Learning Comes to Help Proof-of-Work Against the Selfish Mining Attack.
DOI: 10.5220/0012378600003636
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 1*, pages 37-46
ISBN: 978-989-758-680-4; ISSN: 2184-433X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

pool with the intention of deviating from the rules to unfairly obtain a greater share of the rewards relative to their computational power. These miners exhibit selfish behavior, contrasting with the rational miners who strictly abide by the mining rules. This form of self-serving conduct by miners is commonly referred to as a selfish mining attack (Eyal, 2015), (Eyal and Sirer, 2018).

In addition to undermining incentive-compatibility, the selfish mining attack carries hidden consequences. One of these is the potential to entice rational miners to align with the selfish miners, amplifying their combined computational power (Eyal, 2015), (Eyal and Sirer, 2018). This incremental participation process further consolidates the dominance of the selfish miners, posing a significant threat to the decentralization of the targeted network. Moreover, the imposition of the selfish miners' new mining rule renders the victim network vulnerable to other attacks (Nayak et al., 2016), including the risk of double-spending (Eyal, 2015).

The covert nature of the selfish mining attack makes it challenging to detect for rational participants in the network. In such a dire situation, finding effective solutions becomes an arduous task. Previous attempts to address this type of attack have proven to offer minimal or no substantial relief. To best of our knowledge, this problem has been remained opened as long as the introduction of this kind of attack. Consequently, in our perspective, it is imperative to approach the problem in conjunction with the proof-of-work consensus mechanism, recognizing the need for a comprehensive solution.

Our research endeavors to devise an effective solution by going deeper into the intrinsic nature of the proof-of-work mechanism. To achieve this objective, we conducted a thorough examination of Nakamoto's proof-of-work-based blockchains, including Bitcoin, Litecoin (CLARKE et al., 2018), and Zcash (Hopwood et al., 2016). Our primary emphasis lies on Bitcoin, given its preeminence within this coin family. Through meticulous study and analysis, we aim to uncover insights that contribute to addressing the challenges posed by the proof-of-work consensus mechanism.

Our proposed solution involves equipping all miners in the network with Q-Learning (Sutton and Barto, 2018), (Watkins and Dayan, 1992), (Clifton and Laber, 2020) agents. Traditionally, miners adhere to the proof-of-work protocol by selecting the longest chain of blocks when faced with multiple competing chains. However, our novel solution, leveraging the embedded reinforcement learning agent, aims to intelligently select the optimal chain, deviating from the conventional policy. By incorporating Q-Learning, we enable miners to make more informed and strategic decisions in choosing the most advantageous chain. In order to assess the efficacy of the newly proposed defense method, a series of simulations have been performed. Specially, the distinct properties of the proposed defense mechanism are thoroughly examined. We should specify that the code is available on Github [1].

The contributions of this paper can be summarized as follows: we propose a novel defense algorithm against selfish mining that incorporates Q-Learning. The effectiveness of the proposed algorithm is evaluated through a series of experiments and compared to a well-known existing defense mechanism, tie-breaking.

The structure of the paper is organized as follows: in Section 2, we discuss related works. In Section 3, we provide the necessary preliminaries and background information. The details of the proposed algorithm are presented in Section 4. The performance of the defense algorithm is thoroughly evaluated in Section 5. Section 6 would discuss the paper's limitations and problems. Finally, we conclude the paper in Section 7, highlighting the key findings and contributions.

2 RELATED WORK

In this section, we provide a summary of existing defenses against selfish mining attacks. The defenses are categorized based on the similarity of the methods used. We focus on the most popular defenses that have been proposed.

First and foremost, it is important to highlight that our work draws significant inspiration from previous research. Specifically, we have proposed the Nik defense (Nikhalat-Jahromi et al., 2023a), which stands as the pioneering defense against selfish mining attacks employing learning automaton agents. Additionally, we have devised VDHLA (Variable Depth Hybrid Learning Automaton) (Nikhalat-Jahromi et al., 2023b), a novel family of learning automata that plays a central role in the implementation of the Nik defense. The subsequent paragraphs will categorize and present other defenses based on their similarities.

Certain defenses require fundamental changes to the blockchain structure, often necessitating significant updates to blockchain nodes that are incompatible with previous versions. One defense proposed by

[1] http://github.com/AliNikhalat/SelfishMining

Bahack (Bahack, 2013) involves imposing a punishment rule for all miners, including honest ones. In this defense, miners who fork the blockchain are subject to punishment. However, a drawback of this defense is that it also punishes honest miners. Solat et al. (Solat and Potop-Butucaru, 2016) introduced Zeroblock, which enforces the timely release of blocks by miners. If miners withhold their blocks for selfish mining and fail to broadcast them within the expected time, network peers generate their own dummy blocks. These defenses require changes to block validity and reward distribution, necessitating client updates by network nodes to comply with the new protocol. Another defense proposed by (Saad et al., 2019) suggests altering the structure of transactions. This defense introduces an additional parameter called ”Expected Confirmation Height” in transactions, which is compared with the expected value for the published block height to detect selfish mining attacks. The following paragraph discusses defenses that are effective when a new fork is observed in the network.

Defenses aimed at reducing the chances of selfish miners creating a fork have been developed. The most widely accepted solution is the tie-breaking defense, proposed by Eyal and Sirer (Eyal and Sirer, 2018). According to this defense, when a miner discovers competing branches of the same length, they should propagate all of them and randomly choose one to mine on. As demonstrated in their paper, selfish mining can only be initiated with a minimum mining power of approximately 25%. Heilman (Heilman, 2014) proposed another backward-compatible defense called Freshness Preferred (FP), which penalizes miners withholding blocks by utilizing an unforgeable timestamp parameter. The latest value of the unforgeable timestamp is compared to detect block withholding. Heilman claimed that the lower bound threshold for selfish mining would increase from 25% to 32%. However, a limitation of this solution is the reliance on a trusted party in the network, which contradicts the decentralized nature of Bitcoin. The following paragraph introduces defenses based on fork-resolving policies.

Defenses operating on fork-resolving policies aim to modify protocols so that the defense mechanisms are triggered when the selfish chain becomes longer than the public chain, in contrast to the tie-breaking defense. The first solution in this category is Publish or Perish proposed by Zhang and Preneel (Zhang and Preneel, 2017). In Publish or Perish, blocks not published in time are disregarded, while blocks that include links to competing blocks of their predecessors are given preference. Consequently, a block remains secure until a competing block is published, contributing to neither or both branches, thereby providing no advantage in winning the block race. The following paragraph introduces a machine learning-based algorithm for detecting selfish mining attacks.

Research has been conducted to identify factors that can detect selfish mining attacks (Chicarino et al., 2020),(Peterson et al., 2022). These studies utilize existing data on selfish mining attacks to create training and test datasets. The research investigates various factors and explores future research directions in this context.

3 PRELIMINARIES

In this section, we provide the necessary background information about the proposed algorithm, encompassing key aspects of blockchain, selfish mining strategies, and Q-Learning.

Bitcoin (Nakamoto, 2008) is a decentralized cryptocurrency that enables users to transfer digital currency by generating new transactions and appending them to the blockchain ledger. The blockchain serves as an immutable ledger maintained by a network of miners who secure it against data tampering. Miners are incentivized (Eyal and Sirer, 2018) for their contributions in safeguarding the blockchain by receiving rewards. Each transaction within the blockchain consists of one or more inputs and outputs, with the difference between the total inputs and outputs referred to as the transaction fee. This fee is directed to the miner responsible for incorporating the transaction into the blockchain (Nakamoto, 2008), (Narayanan et al., 2016), (Antonopoulos, 2014). Subsequently, we thoroughly explore a comprehensive explanation of the mining process, followed by an exploration of selfish mining attacks. Lastly, we discuss Q-Learning. Further details regarding the mining process will be provided in the subsequent subsection.

3.1 Mining Process

The state of the blockchain is altered through the execution of transactions, which are then grouped into blocks and appended to the blockchain. A typical block in the blockchain comprises two key components: the header and the body (Wang et al., 2021). The block's header contains vital information such as the hash of the previous block, the hash of the current block, the Merkle root representing all the transactions included in the block, and a nonce value. On the other hand, the block's body consists of the transactions that the miner has chosen to include in the block (Nakamoto, 2008), (Narayanan et al., 2016), (Antonopoulos, 2014).

To validate a block, a solution to a cryptographic puzzle must be found (Eyal and Sirer, 2018), (Sapirshtein et al., 2017). Miners strive to discover the correct nonce value to be placed in the block's header, resulting in a block hash that is smaller than the predefined block difficulty target. The block difficulty is dynamically adjusted to maintain an average block generation rate of approximately one block every ten minutes. Upon successfully solving the puzzle and adding the mined block to the longest chain, the miner is rewarded with newly created Bitcoins that did not exist previously, as well as the transaction fees associated with the newly included transactions.

The probability of mining a new block is directly proportional to the computational resources utilized in solving the puzzle (Eyal and Sirer, 2018), (Sapirshtein et al., 2017). However, due to the inherent nature of the mining process, the time interval between mining events exhibits significant variance from the perspective of an individual miner. As a result, miners often opt to join mining pools, where members collaborate to collectively mine each block and share the generated rewards whenever one of them successfully mines a block. While participating in a mining pool does not alter a miner's expected revenue, it helps reduce the revenue variance and provides a more predictable monthly income (Eyal and Sirer, 2018), (Sapirshtein et al., 2017).

3.2 Selfish Mining Attack

Bitcoin's documentation provides a comprehensive overview of the block release process after mining. When a miner successfully mines a new valid block, it is expected to promptly share it with the network. However, Eyal and Sirer (Eyal and Sirer, 2018) introduced the concept of "selfish mining," which involves certain miners deviating from Bitcoin's standard mining rules to unfairly increase their revenue. This particular strategy, known as "SM1," is the first and most widely recognized form of selfish mining.

In the SM1 selfish mining strategy (Eyal and Sirer, 2018), miners attempting to act selfishly conceal their newly discovered blocks by refraining from broadcasting them to the network. This deliberate concealment creates a fork in the blockchain. One fork is openly known to all network participants, while the other remains hidden and known only to the selfish miners. In this scenario, honest miners continue to work on the stale chain, unaware of the hidden chain pursued by the selfish miners. Consequently, honest chain will be discarded and the relative revenue of the selfish miners increases, which in turn motivates other miners to join their ranks.

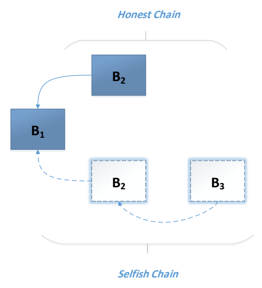


Figure 1: Development of private chain using selfish miners(white blocks).

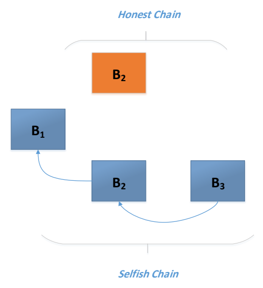


Figure 2: After publishing private chain and adoption of private chain.

To gain a better understanding of the SM1 strategy, let's consider a simplified scenario. In this scenario, selfish miners aim to carry out their mining activities covertly. When they are two blocks ahead of the honest miners (white blocks in Figure 1), upon discovering the first block mined by the honest miners (blue block in Figure 1), they reveal their private chain. As a result, the work done by the honest miners is discarded (orange block in Figure 2), and the network adopts the secret blocks mined by the selfish miners (blue blocks in Figure 2).

As well as SM1 strategy, new strategies for selfish mining come to play. Sapirshtein et al.(Sapirshtein et al., 2017) employed Markov Decision Process (MDP) to study the profit threshold, which represents the minimum fraction of resources necessary for a profitable attack. They determined a bound that ensures system security against such attacks and modified the protocol to evaluate its susceptibility to selfish mining by computing the optimal attack under different scenarios. They revealed situations in which selfish miners could retain control of a selfish chain, even if the public chain appears longer.

Recently, new research areas have emerged in the field of selfish mining, particularly leveraging machine learning techniques (Wang et al., 2021), (Hou et al., 2019). Many of these studies have utilized reinforcement learning methods to enhance the optimality of the attack (Sapirshtein et al., 2017). For instance, (Bar-Zur et al., 2022), (Bar-Zur et al., 2023) developed an improved MDP-based solution by applying reinforcement learning algorithms to maximize revenue. They introduced a novel deep reinforcement learning framework to analyze the incentives of rational miners under various conditions and established an upper bound for the security threshold of proof-of-work-based blockchains.

3.3 Q-Learning

Q-Learning (Sutton and Barto, 2018), (Watkins and Dayan, 1992), (Clifton and Laber, 2020) is a widely used reinforcement learning algorithm that has gained significant attention in the field of machine learning. It is a model-free approach that enables an agent to learn optimal actions in an environment through trial and error.

At its core, Q-Learning involves the exploration and exploitation of a state-action space. The agent interacts with the environment by taking actions and receiving rewards or penalties based on those actions. The goal is to maximize the cumulative reward over time.

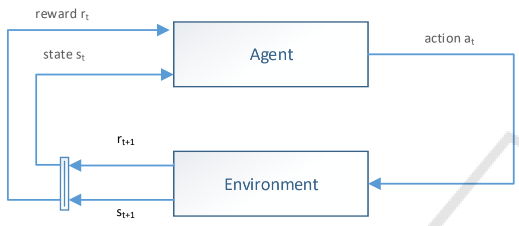


Figure 3: Interaction between Q-Learning agent and environment.

The algorithm maintains a Q-table, which is a matrix that stores the expected cumulative rewards for each state-action pair. Initially, the Q-values in the table are arbitrary or set to zero. As the agent explores the environment, it updates the Q-values based on the rewards received and the new information obtained. The following equation has shown the updating rule of the Q-learning:

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

Where:

- $Q(s,a)$ represents the Q-value of state s and action a
- α_q is the learning rate, controlling the impact of new information on the Q-values
- r is the immediate reward received after taking action a in state s
- γ_q is the discount factor, determining the importance of future rewards
- s' is the next state after taking action a in state s
- a' is the action with the highest Q-value in state s'

This updating rule calculates the new Q-value based on a weighted combination of the previous Q-value and the new information obtained from the reward and the maximum Q-value of the next state. The learning rate determines the balance between the new information and the existing knowledge.

By repeatedly applying this updating rule during the learning process, the Q-values converge to their optimal values, allowing the agent to make informed decisions and maximize its cumulative reward over time.

4 PROPOSED ALGORITHM: Q-DEFENSE

This section goes deeper into the details of the proposed method. We begin by providing the system model and necessary definitions to establish a clear understanding of our proposed algorithm. Subsequently, we present the proposed defense algorithm in a step-by-step manner, highlighting its key components and mechanisms.

4.1 System Model

Every algorithm proposition requires a thorough understanding of the environment in which it is intended to operate. Therefore, it is essential to establish a clear model of our target blockchain system. Specifically, we need to analyze the attack scenario from the perspective of honest miners.

Firstly, let us consider a network consisting of two groups of miners. The first group comprises selfish miners who deviate from the prescribed mining rules. We denote their mining power as α percent of the total mining power. Conversely, the second group consists of rational miners who diligently adhere to the mining rules and possess $1-\alpha$ percent of the total mining power.

The impact of network propagation delay is disregarded in our analysis. It is assumed that miners in the network strive to propagate blocks swiftly to minimize disruptions in the subsequent mining process.

Moreover, the blocks in the network are organized in a tree structure to construct a chain. If two blocks share the same previous block hash, it indicates the creation of a new fork in the chain. However, in our analysis, we only consider one type of fork, which arises from selfish mining. From the fork, two chains emerge: the selfish chain and the honest chain.

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

4.2 Required Definitions

This subsection is dedicated to presenting the prerequisite definitions that are essential for a better understanding of the proposed algorithm. It is important to note that these definitions are based on the characteristics of each branch of the fork resulting from selfish mining. The following definitions are provided:

- A fork branch refers to one of the competing chains in the network that is created as a result of selfish mining. It consists of a sequence of blocks starting from the common ancestor block up to the current block in that branch.
- Let L represent the length of a branch, indicating the number of blocks it contains. The length of a branch serves as a crucial parameter for assessing the size and extent of competing chains resulting from selfish mining.
- Let W denote the weight of a branch, which is determined based on a comparison of blocks within that branch and blocks of the same height in other branches. The branch with the most recent creation time will have its weight increased by one, encompassing all blocks from the first to the last in that particular branch.
- Let K denote the fail-safe parameter, which assists the miner in selecting a branch based on either its length (L) or its weight (W). If the length of a branch in the fork exceeds the others by at least K, that branch is chosen. Otherwise, the weight parameter W is used to determine the preferred branch.
- Let τ denote the decision-making time, which represents the time taken by a miner to check for the presence of forks. If a fork is detected, the miner must select one of the branches, taking into account the parameter K.
- The time parameter θ is defined to determine the next value for K using the automaton. It is composed of multiple τ values.

4.3 Algorithm

Having established the necessary definitions, we can now proceed to define our algorithm for defending against selfish mining attacks. The proposed algorithm is outlined in the following steps:

1. Calculate the length L of each branch.
2. Calculate the weight W of each branch.
3. If the difference between the length of the longest branch and the length of the second longest branch is greater than K, choose the longest branch. Otherwise, choose the heaviest branch based on the calculated W parameter.
4. When τ reaches the end, the Q-Learning agent determines the next value for K. Typically, K oscillates between $K_{\min}$ and $K_{\max}$. It is important to note that the Q-Learning agent has one state and three actions. The actions correspond to: 1-**Grow**, which increases K by one; 2-**Stop**, which keeps K unchanged; and 3-**Shrink**, which decreases K by one.
5. When θ reaches the end, the Q-Learning agent receives feedback from the environment. We have designed a virtual environment for the Q-Learning agent to provide information about its decisions. The reward is calculated by dividing the number of weight decisions (W) by the total number of decisions. The total number of decisions includes the number of height decisions (L) plus the number of weight decisions (W). The following equation illustrates the calculation of the r parameter (reward) of the Q-Leaning agent:

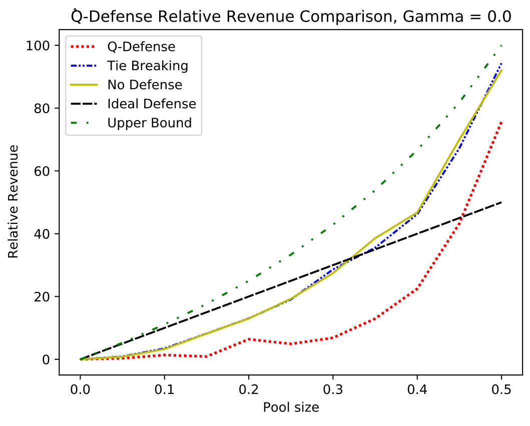
$$r = \frac{Number\ of\ Weight\ Decisions}{Total\ Number\ of\ Decisions} \tag{2}$$

5 EVALUATION

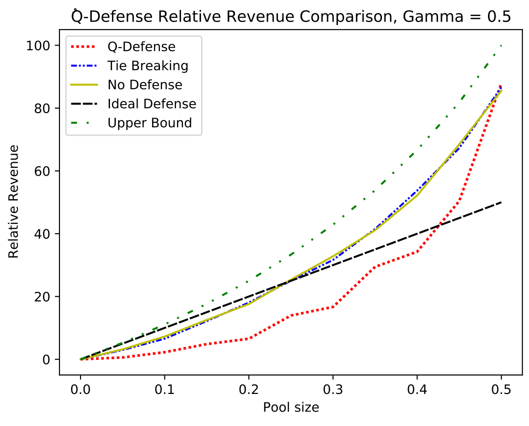
To evaluate the proposed algorithm, we conducted a series of experiments that explore its performance from various perspectives. In these experiments, we considered a network with two types of miners: selfish miner and rational miner. In our simulations, the mining process follows a Poisson process, where a new block is mined with a probability of α by the selfish miner and a probability of $1-\alpha$ by the rational miner.

Moreover, we modified the behavior of the selfish miner to operate within a learning-based mining environment. This allowed us to observe the dynamics of the proposed algorithm in a realistic setting.

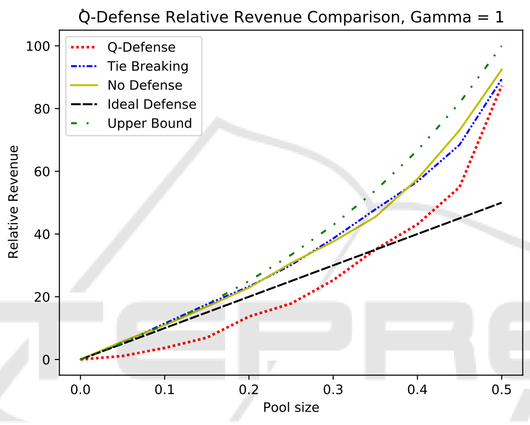
In order to fulfill our objectives, we conducted experiments involving a total of 10,000 blocks. The fail-safe parameter, denoted by K, was allowed to vary within the range $K_{\min} = 1$ and $K_{\max} = 3$. This range of K was chosen to allow Q-Learning agents to reach consensus in a short amount of time since choosing K is a time-consuming process. The Q-Learning agents



(a) $\gamma = 0$



(b) $\gamma = 0.5$



(c) $\gamma = 1$

Figure 4: Comparing Q-defense with tie-breaking using various values of α and γ.

were configured with specific values: a learning rate of 0.1 and a discount factor of 0.75. These settings were chosen to optimize the learning and decision-making process of the agents within the proposed algorithm.

To demonstrate the effectiveness of the proposed defense, we compared it with the tie-breaking defense mechanism. In these experiments, τ was set to the mining time of five blocks, and θ consisted of ten τ. These values are set for θ and τ to give Q-Learning agents enough time to explore and exploit accurately. The results are shown in Figure 4.

The results of the experiments highlight several interesting findings. Firstly, they demonstrate the superiority of the proposed defense over tie-breaking in terms of the relative revenue of selfish miners. This indicates that the proposed defense effectively reduces the revenue gained by selfish miners. Particularly, at $\gamma = 1$, where selfish miners have maximum power, the proposed defense performs exceptionally well in decreasing the relative revenue. These results provide strong evidence of the effectiveness of Q-Learning in complex environments like the blockchain.

In addition to relative revenue, another important metric, the lower bound threshold, was examined. The results reveal another significant observation. The proposed defense, leveraging the power of Q-Learning, significantly increases the lower bound threshold for initiating the attack. Specifically, the lower bound threshold is raised from 0.25 to approximately 0.4. This indicates that the proposed defense enhances the security and stability of the blockchain network by making it more difficult for selfish miners to launch successful attacks.

Furthermore, the detailed results of these experiments, including the relative revenue for various values of α and γ, are provided in Appendix 7.

6 DISCUSSION

Our work, while promising, acknowledges some inherent limitations that warrant discussion. One potential concern is the computational complexity introduced by machine learning methods, which can lead to overheads. To address this, we deliberately opted for Q-learning due to its favorable characteristics. Q-learning's off-policy nature allows it to efficiently explore and learn from suboptimal actions, preventing entrapment in local optima. Being model-free, it can handle diverse problems without requiring knowledge of the environment's dynamics. Moreover, its implementation simplicity and efficient memory usage make it well-suited for our specific application, effectively mitigating computational concerns and facilitating practical adoption.

Another aspect deserving attention is the potential non-deterministic nature of our approach. The value of parameter K heavily relies on the learning agent at each node, which could introduce variability. To address this, we limited the range of K, as demonstrated in the experiment section, opting for a controlled range, such as [1, 3]. This limitation helps to reduce inconsistency and ensures a more stable performance across different scenarios.

Considering scalability in large-scale networks is crucial. As all nodes must reach consensus on one chain in forks, the uniform convergence to the same K value becomes vital. With a larger number of nodes in the network, there is a higher likelihood of K reaching a consistent value across more nodes. This solution addresses both scalability issues and potential inconsistencies related to K, making our approach more robust and applicable in real-world, complex network environments.

Lastly, our system effectively prevents block withholding attempts through the τ and θ parameters. All miners actively participate in chain selection and evaluating their K value, making block withholding an ineffective strategy. This ensures the security and fairness of the consensus process and strengthens the overall integrity of our approach. By incorporating τ and θ, we have established a robust defense against block withholding and enhanced the reliability of the system.

Consequently, our work demonstrates significant promise despite the acknowledged limitations. By deliberately selecting Q-learning for its advantages, controlling parameter variability, addressing scalability concerns, and recognizing challenges like block withholding, our approach provides valuable insights and sets the foundation for future advancements in consensus algorithms for large-scale networks. As with any research, acknowledging and discussing limitations contribute to the scientific discourse and pave the way for even more robust solutions in the field.

7 CONCLUSION

One of the most significant challenges to the proof-of-work consensus mechanism is posed by colluding miners who aim to deviate from the prescribed mining rules. These colluding miners, commonly known as selfish miners, have remained a persistent threat since the early days of Bitcoin. In order to address this problem, we were inspired to leverage the power of Q-Learning, a well-established method in the field of reinforcement learning.

Our proposed model has undergone comprehensive evaluation from various perspectives. In terms of relative revenue, our method has successfully reduced the revenue obtained by selfish miners compared to conventional proof-of-work approaches that lack learning mechanisms. Furthermore, our evaluations reveal a significant increase in the initial threshold of the selfish attack, rising from 25% of the total mining power to 40% approximately. The results show the superiority of the proposed approach over tie-breaking.

The suggested mechanism opens up new avenues at the intersection of artificial intelligence and blockchain technology. As a future work, we will consider more implementation details and modelings to define comprehensive defense mechanism for proof-of-work consensus algorithms.

REFERENCES

Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.".

Babaioff, M., Dobzinski, S., Oren, S., and Zohar, A. (2012). On bitcoin and red balloons. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 56–73.

Bahack, L. (2013). Theoretical bitcoin attacks with less than half of the computational power (draft). *arXiv preprint arXiv:1312.7013*.

Bai, Q., Zhou, X., Wang, X., Xu, Y., Wang, X., and Kong, Q. (2019). A deep dive into blockchain selfish mining. In *Icc 2019-2019 Ieee International Conference On Communications (Icc)*, pages 1–6. IEEE.

Bar-Zur, R., Abu-Hanna, A., Eyal, I., and Tamar, A. (2022). Werlman: to tackle whale (transactions), go deep (rl). In *Proceedings of the 15th ACM International Conference on Systems and Storage*, pages 148–148.

Bar-Zur, R., Dori, D., Vardi, S., Eyal, I., and Tamar, A. (2023). Deep bribe: Predicting the rise of bribery in blockchain mining with deep rl. *Cryptology ePrint Archive*.

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE symposium on security and privacy*, pages 104–121. IEEE.

Chaudhry, N. and Yousaf, M. M. (2018). Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities. In *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, pages 54–63. IEEE.

Chicarino, V., Albuquerque, C., Jesus, E., and Rocha, A. (2020). On the detection of selfish mining and stalker attacks in blockchain networks. *Annals of Telecommunications*, 75:143–152.

CLARKE, S., CRAIG, I., and WYSZYNSKI, M. (2018). Litecoin cash: The best of all worlds sha256 cryptocurrency. *URL: https://litecoinca. sh/downloads/lcc whitepaper. pdf.[Last accessed on 2018 Sep 25].*

Clifton, J. and Laber, E. (2020). Q-learning: Theory and applications. *Annual Review of Statistics and Its Application*, 7:279–301.

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

Eyal, I. (2015). The miner's dilemma. In *2015 IEEE symposium on security and privacy*, pages 89–103. IEEE.

Eyal, I. and Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102.

Heilman, E. (2014). One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner. In *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18*, pages 161–162. Springer.

Hopwood, D., Bowe, S., Hornby, T., Wilcox, N., et al. (2016). Zcash protocol specification. *GitHub: San Francisco, CA, USA*, 4(220):32.

Hou, C., Zhou, M., Ji, Y., Daian, P., Tramer, F., Fanti, G., and Juels, A. (2019). Squirrl: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning. *arXiv preprint arXiv:1912.01798*.

Judmayer, A., Stifter, N., Krombholz, K., and Weippl, E. (2022). *Blocks and chains: introduction to bitcoin, cryptocurrencies, and their consensus mechanisms*. Springer Nature.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., and Qijun, C. (2017). A review on consensus algorithm of blockchain. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2567–2572. IEEE.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press.

Nayak, K., Kumar, S., Miller, A., and Shi, E. (2016). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 305–320. IEEE.

Nikhalat-Jahromi, A., Saghiri, A. M., and Meybodi, M. R. (2023a). Nik defense: An artificial intelligence based defense mechanism against selfish mining in bitcoin. *arXiv e-prints*, pages arXiv–2301.

Nikhalat-Jahromi, A., Saghiri, A. M., and Meybodi, M. R. (2023b). Vdhla: Variable depth hybrid learning automaton and its application to defense against the selfish mining attack in bitcoin. *arXiv preprint arXiv:2302.12096*.

Panda, S. S., Mohanta, B. K., Satapathy, U., Jena, D., Gountia, D., and Patra, T. K. (2019). Study of blockchain based decentralized consensus algorithms. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 908–913. IEEE.

Peterson, M., Andel, T., and Benton, R. (2022). Towards detection of selfish mining using machine learning. In *International Conference on Cyber Warfare and Security*, volume 17, pages 237–243.

Saad, M., Njilla, L., Kamhoua, C., and Mohaisen, A. (2019). Countering selfish mining in blockchains. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 360–364. IEEE.

Sapirshtein, A., Sompolinsky, Y., and Zohar, A. (2017). Optimal selfish mining strategies in bitcoin. In *Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016, Revised Selected Papers 20*, pages 515–532. Springer.

Solat, S. and Potop-Butucaru, M. (2016). Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin. *arXiv preprint arXiv:1605.02435*.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tschorsch, F. and Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123.

Wang, T., Liew, S. C., and Zhang, S. (2021). When blockchain meets ai: Optimal mining strategy achieved by machine learning. *International Journal of Intelligent Systems*, 36(5):2183–2207.

Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., and Kim, D. I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *Ieee Access*, 7:22328–22370.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8:279–292.

Zhang, R. and Preneel, B. (2017). Publish or perish: A backward-compatible defense against selfish mining in bitcoin. In *Topics in Cryptology–CT-RSA 2017: The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings*, pages 277–292. Springer.

APPENDIX

In this appendix, we present detailed results of the evaluation conducted in the Evaluation section (Section 5), focusing on the impact of varying the α and γ parameters. The analysis provides valuable insights into the performance and effectiveness of our proposed approach under different configurations of these parameters.

Table 1: Q-Defense results for different values of α and γ.

Selfish Miner Computational Power α	Rational Miner Revenue	Expected Rational Miner Revenue	Selfish Miner Revenue	Expected Selfish Miner Revenue
		$\gamma = 0.$		
0.0	100	100.0	0.0	0.0
0.05	99.67	95.00	0.33	5.00
0.10	98.62	90.00	1.38	10.00
0.15	99.10	85.00	0.90	15.00
0.20	93.60	80.00	6.4	20.00
0.25	95.08	75.00	4.92	25.00
0.30	93.14	70.00	6.85	30.00
0.35	87.04	65.00	12.96	35.00
0.40	77.51	60.00	22.49	40.00
0.45	56.73	55.00	43.26	45.00
0.50	24.33	50.00	75.66	50.00
		$\gamma = 0.5$		
0.0	100.0	100.0	0.0	0.0.00
0.05	99.41	95.00	0.59	5.00
0.10	97.73	90.00	2.27	10.00
0.15	95.15	85.00	4.85	15.00
0.20	93.44	80.00	6.56	20.00
0.25	86.06	75.00	13.94	25.00
0.30	83.31	70.00	16.69	30.00
0.35	70.55	65.00	29.45	35.00
0.40	65.73	60.00	34.27	40.00
0.45	49.58	55.00	50.49	45.00
0.50	12.07	50.00	87.93	50.00
		$\gamma = 1.0$		
0.0	100.0	100.0	0	0.00
0.05	98.89	95.00	1.11	5.00
0.10	96.30	90.00	3.70	10.00
0.15	93.04	85.00	6.96	15.00
0.20	86.31	80.00	13.69	20.00
0.25	82.17	75.00	17.83	25.00
0.30	74.72	70.00	25.28	30.00
0.35	64.95	65.00	35.05	35.00
0.40	56.91	60.00	43.09	40.00
0.45	45.05	55.00	54.95	45.00
0.50	12.68	50.00	87.32	50.00