The University of Hong Kong

Department of Computer Science, Faculty of Engineering

Blockchain Mining with Machine Learning

Interim Report

Supervisor: Dr. Liu Qi

Member: Cheung Yau Shing Jonathan (3035783560)

Group: FYP23033
Date of submission: 21/01/2024

Abstract

Blockchain miners today heavily rely on the brute force method for mining. However, this consumes immense computational resources and has led to various environmental problems. Inspired by the recent advancements in artificial intelligence, this project seeks to explore the application of machine learning in blockchain mining. More specifically, this work introduces the use of machine learning to 1) enhance nonce finding, 2) optimize transaction selection, and 3) discover chain-level mining strategy. Historical mining data were first collected from blockchain explorers and APIs. The pre-processed data was then used to train the respective models. After model fine-tuning and improvement, evaluation will be performed on the Bitcoin Testnet to obtain real-time results. A comparative study will finally be conducted to demonstrate the effectiveness and efficiency of the algorithm in comparison to existing mining techniques. It is hoped that insights gained from the project could encourage miners to adopt more environmentally conscious mining methods in the future.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Liu, Qi. for the guidance and support provided throughout the project. His expertise and mentorship have been instrumental in shaping the direction and success of this work. I would also like to extend my thanks to the University of Hong Kong for providing the necessary computational resources and budget crucial for my work.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

Abbreviation	Definition
PoW	Proof of work
DNN	Deep neural network
RL	Reinforcement learning
ML	Machine learning
PPO	Proximal Policy Optimization
DQL	Deep Q-Learning
Regtest	Regression test mode
TWh	Terawatt-hours
ASICs	Application-Specific Integrated Circuits
API	Application Programming Interface
GPU	Graphics processing unit
HKU	The University of Hong Kong

1. Introduction

In this section, we provide an introduction of the project, including the background, motivation, project objectives, proposed deliverables, and an outline of this progress report.

1.1 Background

Blockchain technology is becoming increasingly popular. As a decentralized system, it could increase trust, security, transparency and traceability of data across a business network [1]. It is therefore applied in various industries. In finance, it is used for cross-border payments and smart contracts. In Supply Chain Management, blockchain allows businesses and consumers to track the origin, movement, and authenticity of products. In healthcare, Blockchain helps securely store and share patient medical records.

Mining is crucial in blockchain for block creation and transaction validation [2]. The block creation process differs across various consensus mechanisms employed in the blockchain. Proof of Work is the one of the most prominent consensus mechanisms. To create a block under PoW, miners have to 1) collect pending transactions, 2) verify their validity, and 3) construct the block header by solving the hash problem [3]. The hash problem involves miners searching for a nonce (numerical value) that generates a hash value complying to predefined criteria [3]. This process ensures the security and integrity of the blockchain by discouraging malicious actors from altering the blockchain's history [4].

1.2 Motivation

There has been a long-held belief that trial and error is the only feasible and profitable block-mining strategy for PoW [5]. Therefore, miners with greater computational resources have a higher capacity to explore a larger number of solutions, thus increasing their chances of winning. This resource-based competition has led to excessive energy consumption. Figure 1 displays the annual energy consumption of Bitcoin in 2021, measured in terawatt-hours (TWh). It can be seen that Bitcoin consumed over 100 TWh of energy annually. This surpassed the total energy usage of prominent countries like Sweden, Ukraine, Norway, and Argentina. Remarkably, it was nearly half of the energy usage of the United Kingdom. The heavy reliance on fossil fuel-based electricity to power mining operations has led to the release of greenhouse gases, exacerbating climate change concerns. Moreover, miners often face the need to constantly upgrade their equipment. This causes the disposal of outdated mining rigs and worsens the growing issue of electronic waste. This thereby makes proof of work blockchain mining one of the most environmentally detrimental practices.

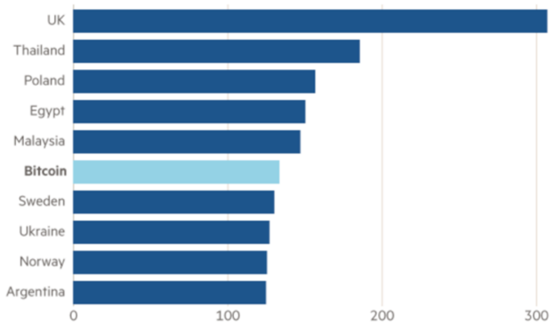


Figure 1. Energy consumption of bitcoin, compared with countries' reading (TWh) in 2021. Adopted from *[6]*

1.3 Project Objectives

This project aims to explore the application of machine learning in blockchain mining. A literature review was first conducted to uncover existing trends and techniques in blockchain mining. Then, historical blockchain data were collected from blockchain explorers and APIs. This data will then be used to train the proposed machine learning models to 1) enhance nonce finding, 2) optimize transaction selection, and 3) discover chain-level mining strategies. The algorithms will then be deployed on the Blockchain Testnet to obtain real-time mining results for evaluation, and a comparative study will finally be conducted to assess the proposed algorithms against existing methods.

1.4 Project Deliverables

By the end of the project, the following 3 deliverables will be presented:

1) Research Report: A comprehensive research report will be produced, providing an in-depth analysis of the background, objectives, methodology, and findings of the project. The report will examine and compare existing approaches with the developed algorithm to reflect its effectiveness in mining.

2) Code: The project will provide the entire codebase, including relevant scripts, modules, and libraries utilized in developing the machine learning model and conducting the experiments. This code will enable others to replicate the work, extend it further, and validate the results. Proper documentation and comments within the code will ensure its comprehensibility.

3) Presentation and Demonstration: A comprehensive presentation summarizing the project will be prepared, highlighting the key findings, methodology, and outcomes. Additionally, a demonstration on the machine learning model’s functionality and performance will be provided to showcase its practical application in blockchain mining. The presentation and demonstration will be completed in a video format and be uploaded to the project website.

1.5 Paper Outline

This report will discuss the current completion status of the project in detail. Section 1 will provide an introduction to the project. Section 2 will focus on literature review, highlighting the existing mining trends and techniques. Section 3 will outline the methodology employed in this project, providing insights into the research approach. Section 4 will cover experiments and results, while section 5 will focus difficulties encountered in the project and the corresponding solutions. Finally, section 6 will cover future plans and conclusion will be given in section 7.

2. Literature Review

In this literature review section, we first discuss artificial intelligence methods, then transition to the fundamental components of blockchain, including its framework and mining methods. Finally, we examine the popular blockchain system utilized for development and testing.

2.1 Deep Neural Network

Deep neural networks were first introduced by Hinton and Salakhutdinov in their paper "A fast learning algorithm for deep belief nets" [7]. They developed the network to capture complex patterns and relationships. The proposed DNN by Hinton and Salakhutdinov consisted of multiple layers between the input and output. Each layer had interconnected nodes called neurons. Each neuron processed input from the previous layer, and passed it to the next layer. Backpropagation was then used to adjust the network's parameters, optimizing the model's performance [7]. Deep neural networks achieve remarkable success in various domains, such as image recognition [8] and natural language processing [9] today. The ability of DNN in discovering relationship from multidimensional data allows them to adapt to different mining conditions [10]. This thereby enhances the performance of mining.

2.2 Reinforcement Learning

Reinforcement learning (RL) was first proposed by Sutton and Barto in their book 'Reinforcement Learning: An Introduction' [11]. This machine learning paradigm enabled an agent to make optimal decisions through interaction with an environment. More specifically, it involved an agent taking actions in a dynamic environment to maximize cumulative rewards. By leveraging trial and error, the agent received feedback in the form of rewards or penalties, allowing it to update its policy for future actions [10]. RL is well-suited for the project as it enables the agent to learn and adapt to the changing dynamics of the blockchain network [10]. It could optimize mining strategies, and makes informed decisions to maximize mining efficiency and rewards [10].

2.3 Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a reinforcement learning approach widely recognized for its stability and efficiency in policy training. Introduced by Schulman et al. in 2017, PPO aims to address the complexities and instabilities of policy optimization by utilizing a clipped surrogate objective function, which prevents large policy updates and promotes gradual learning [12]. This makes PPO a preferred choice in various applications, including its successful deployment in OpenAI's Dota 2-playing bots [13]. The algorithm's compatibility with both continuous and discrete action spaces further contributes to its broad applicability and popularity in the machine learning community [14].

2.4 Deep Q-Learning

Deep Q-Learning (DQL) is a pivotal algorithm in the field of deep reinforcement learning, merging Q-learning with deep neural networks to handle high-dimensional state spaces. The algorithm uses a technique known as experience replay and a separate target network to stabilize the learning process [15]. Subsequent enhancements to DQN have addressed issues such as overestimation bias with the introduction of Double DQN [16] and improved exploration through duelling network architectures [17].

2.5 Proof of Work Blockchain

The proof of work blockchain was proposed by Nakamoto in the paper 'Bitcoin: A Peer-to-Peer Electronic Cash System' [3]. Figure 2 depicts the structure of a PoW blockchain. From the figure, we can see that a PoW block consists of a header that contains the current and previous block hash. These hashes establish connections to the next and previous blocks. The figure also reflects that the Merkle Root links to the block body and includes the transactions and their respective hashes. To finalize the block, miners must discover the nonce [3]. The nonce is a random number that, when combined with other block data, produces a hash value that meets specific criteria [3]. PoW mining ensures the decentralization and security of the blockchain by making it challenging and time-consuming to tamper with existing blocks [18]. However, due to the significant energy consumption associated with PoW mining, it has faced criticism, and efforts are underway to explore more energy-efficient alternatives.

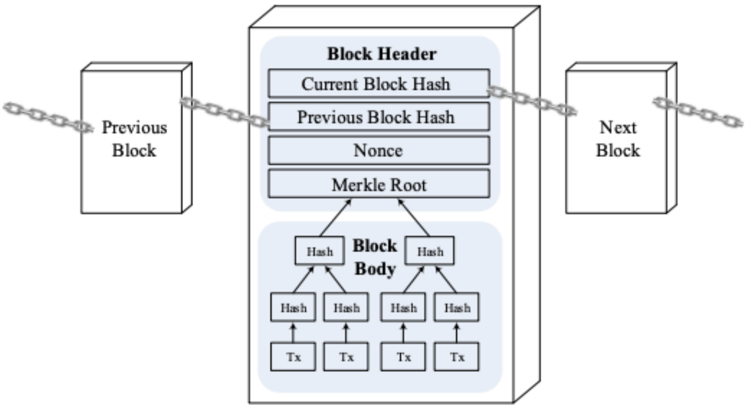


Figure 2. Data structure of a proof of work blockchain. Adopted from *[10]*

2.6 Brute-force Mining

According to the survey conducted by Wang et al. in 2019 [18], brute-force mining was recognized as one of the predominant mining approaches. Wang et al. described brute-force mining as miners iterating through an extensive range of nonce values in order to discover a valid solution to the hash puzzle. The study also revealed that brute-force mining demanded a significant amount of computational power. More-specifically, miners employed specialized hardware, such as Application-Specific Integrated Circuits (ASICs), to maximize their mining efficiency [18]. This therefore makes mining in Proof of Work blockchains one of the most environmentally deteriorating practices.

2.7 Honest Mining

In honest mining, miners act in good faith by confirming and recording transactions correctly [19]. They extend the blockchain by appending new blocks onto the longest recognized chain. This collective adherence is crucial for network stability and the prevention of double-spending, as it relies on a consensus mechanism where the majority of network participants are assumed to act honestly [19]. Enhancements to the protocol, such as the GHOST protocol, incentivize miners to maintain honesty by providing rewards for blocks that contribute to the network's security but do not become part of the main chain [20].

2.8 Selfish Mining

Selfish mining occurs when miners find a new block but intentionally withhold broadcasting it to the network [21]. By doing so, they aim to gain a head start on the next block while others continue mining on the old chain [21]. If successful, selfish miners can potentially claim more rewards as they secretly build a longer chain, which they reveal at a strategic time to override the network's current chain. This tactic can disrupt the fair distribution of rewards and threaten the security of the blockchain, prompting discussions around countermeasures and protocol revisions to deter such behavior [21].

2.9 Bitcoin Regression Test Mode

According to Bitcoin [22], the Bitcoin Regression Test Mode (Regtest) is a local testing mode that allows developers to create and control a private, isolated blockchain for testing purposes. The proposed platform operates locally on an individual's machine, making it faster and more customizable for testing scenarios. Regtest is best suited for program development. This is because it enables developers to quickly set up a local blockchain, generate blocks, and create custom testing scenarios [22].

2.10 Bitcoin Testnet

As per the Bitcoin Wiki [23], the Bitcoin Testnet is specifically created as an independent network for the purpose of testing and experimentation. It provides developers and users with a platform to evaluate their applications and protocols without the need for real bitcoins or any impact on the main Bitcoin network [23]. The Bitcoin Testnet is, therefore, ideal for evaluation in our project as it enables the model to operate based on real-time network conditions while mitigating the risk of affecting the actual Bitcoin network or using real funds [23].

3. Methodology

In the methodology section, we delve into the key steps undertaken in our project. This includes historical data collection, blockchain system setup, model development and training and model performance evaluation.

3.1 Historical Data Collection

Historical blockchain data was collected by accessing blockchain explorers and APIs. Two types of data were collected: Bitcoin Block and Bitcoin Transaction data. Bitcoin Block data includes the hash, version, previous block hash, Merkle Root, timestamp, and nonce of mined blocks, while Bitcoin Transaction data includes the transaction ID, fee, and weight of Bitcoin transactions. This information will be essential for the development and testing of the developed algorithm.

3.2 Blockchain System Setup

The project will be developed based on the Bitcoin blockchain system. Setting up the Bitcoin Regtest and Testnet environment will be essential for the development, experiment, and validation of the algorithm. To set up the Bitcoin Regtest network, the blockchain client software will be installed and configured to Regtest mode. Then, we set up the Bitcoin Testnet by configuring the connection parameters of the blockchain software.

3.3 Model Development and Training

We explore the use of machine learning in three areas of blockchain mining: to enhance nonce discovery, optimize transaction selection, and develop chain-wide mining strategy.

3.3.1 Enhance Nonce Finding

To successfully mine blocks, miners must discover a nonce that satisfies specific criteria. In this project, we explored the application of machine learning to identify the nonce value with the fewest attempts. First, we investigated different iteration methods, comparing the traditional sequential increment by one with an increment by two and the application of the Collatz conjecture for iterating nonce values nonlinearly. Second, we employed supervised learning to predict the starting seed for iteration. Our models were trained on historical data from previously mined Bitcoin blocks. The normalized inputs for the model include the Merkle root (converted to integer format), the timestamp, and the previous block hash, while the output is the nonce value. We applied non-linear regression models such as Polynomial Regression and Random Forest Regression for the task since they can capture complex patterns and relationships.

3.3.2 Transaction Selection Optimization with Reinforcement Learning

Miners must select transactions from the transaction pool, and their profit is contingent on the accumulated transaction fees. Therefore, we developed a reinforcement learning algorithm to optimize the selection of transactions, while adhering to the block's weight constraints. The action space is delineated by either incorporating a transaction into the block or omitting it. A positive reward is granted when the inclusion of a transaction leads to an increase in the

Running Average of Total Aggregated Fees. Conversely, a negative reward is assigned if the inclusion of a transaction leads to a decrease in this running average. The equation of Running Average of Total Aggregated Fees is given in Equation 1.

$$RATAF = \frac{Sum\ of\ Fees\ from\ Each\ Block}{Total\ Number\ of\ Blocks}$$

Equation 1. Formula for Running Average of Total Aggregated Fees (RATAF)

The termination condition for the algorithm is reached when the cumulative weight of the selected transactions meets or exceeds the block's weight limit. Proximal Policy Optimization (PPO) and Deep Q-Learning (DQN) are employed to optimize the selection policy.

We also introduced two naive methods for comparison. The first method ‘Random’, entails the random selection of transactions until the weight limit for the block is reached. This approach does not prioritize transactions by fee or weight, instead relying on chance to fill the block's capacity. The second method 'Sorted', involves sorting the transactions in descending order by their fees and in ascending order by weight. By doing so, we prioritize transactions that offer higher fees and occupy less weight, aiming to optimize the block's reward before reaching the weight limit.

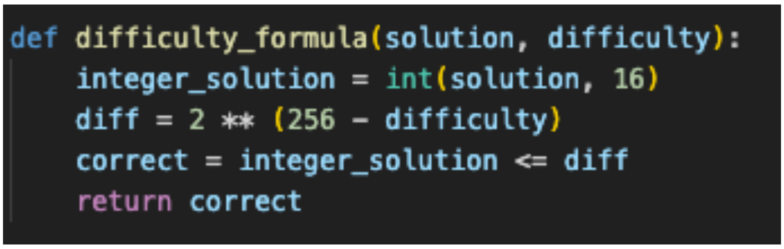
3.3.3 Chain-wide Mining Strategy

Recent research introduced new mining strategies [21] that achieved improved overall mining outcomes compared to honest mining [19]. Therefore, we will reimplement the reinforcement learning algorithm proposed in 'When Blockchain Meets AI: Optimal Mining Strategy Achieved By Machine Learning’ [10] to discover effective chain-wide mining strategy. The actions of the algorithm include:

- Adopt: The agent accepts the current longest chain in the network, referred to as the honest chain, and proceeds to mine on top of the latest block of this chain, effectively aligning with the network consensus.
- Override: The agent attempts to gain an advantage by publishing a chain that is one block longer than the honest chain, effectively making the adversary's chain the new longest chain and forcing the network to reorganize to this chain.
- Match: The agent deliberately creates a situation of uncertainty by publishing a number of blocks equal to the length of the honest chain. This action results in a fork, instigating an open mining competition between the two branches — one maintained by the honest network and the other by the adversary.
- Wait: The agent chooses a passive approach, holding off on publishing any blocks. It continues to mine in private, extending its secret chain in the hope of later executing an override or match action.

3.3.4 Custom Difficulty Formula

Since the official Bitcoin difficulty formula requires extensive computational resources for mining and is not feasible for our experiments, we adopted a custom difficulty formula that was more suitable for our experiments while ensuring the validity of our results. Figure 3 displays the code of the formula.



```python
def difficulty_formula(solution, difficulty):
    integer_solution = int(solution, 16)
    diff = 2 ** (256 - difficulty)
    correct = integer_solution <= diff
    return correct
```

Figure 3. Custom difficulty formula

3.4 Model Performance Evaluation

We will combine the three proposed methods into a final pipeline and perform detailed evaluation on the Bitcoin Testnet. Baseline results from brute-force mining will first be obtained, serving as reference points. We will then assess the performance our pipeline according to the defined metrics. A comprehensive comparison will finally be conducted to provide valuable insights into the effectiveness and efficiency of our work in enhancing mining outcomes. Graphs and tables will also be included to present the performance metrics, facilitating a clear comparison and analysis of different mining strategies.

4 Experiments and Results

We present the details and the results of our completed experiments in applying machine learning to enhance nonce finding and optimize transaction selection.

4.1 Enhance Nonce Finding

In this experiment, we compared the average number of attempts needed to find the correct nonce over 15 past Bitcoin blocks with the difficulty set to 20 in the custom difficulty formula as shown in Figure 3.

4.1.1 Iteration Methods

Table 1 displays the average number of nonces tested for different iteration methods. It is shown that the traditional method of sequential increment by 1 performs the best, requiring only 909,741 attempts to attain a correct nonce, followed by sequential increment by 2 with 1,150,721 attempts, and the Collatz conjecture with 1,607,630 attempts.

Method	**Average Number of Attempts**
Sequential (+1)	909,741
Sequential (+2)	1,150,721
Collatz conjecture	1,607,630

Table 1. Average number of nonces tested for different iteration methods

Figure 4 displays the number of nonces tested per block using different iteration methods. The results show that there are cases in which the Collatz conjecture performed significantly worse compared to linear methods. More specifically, for block 8, it required more than 10 times the number of attempts compared to sequential methods. This indicates that using a non-linear iteration method may not be ideal, as it could overlook many suitable nonce values.

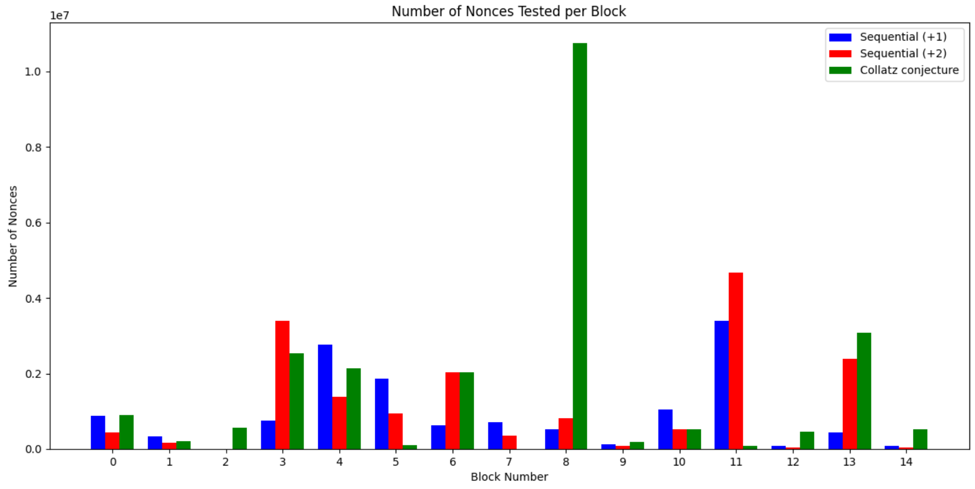


Figure 4. Number of nonces tested per block for different iteration methods

4.1.2 Starting Seed

We explore the use of machine learning to determine the starting seed for nonce searching. Section 4.1.1 demonstrates that iterating the nonce value by sequential increments of 1 is optimal; thus, upon establishing the starting seed, we continue with this sequential approach.

Table 2 displays the average number of nonces tested for different starting seeds. It is shown that Random Forest Regression performs the best, requiring only 710,550 attempts, followed by Sequential (+1) with a starting seed of 0, which needs 909,741 attempts, and finally Polynomial Regression with 1,040,948 attempts.

Method	Average Number of Attempts
Sequential (+1)	909,741
Polynomial Regression	1,040,948
Random Forest Regression	710,550

Table 2. Average number of nonces tested for different starting seed

Figure 5 displays the number of nonces tested per block. The results indicate that there are multiple instances where both regression methods performed better, such as in blocks 3, 4, and 10. Therefore, with more extensive training, we are confident that applying machine learning to find the starting seed can yield significant improvements in nonce discovery.

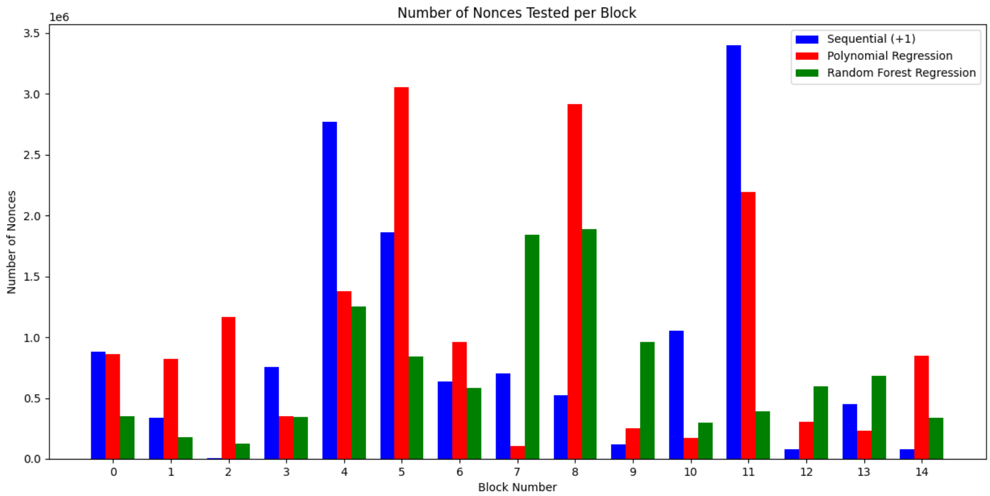


Figure 5. Number of nonces tested per block for different starting seed

4.2 Mining Fee Optimization

In this experiment, we compare our proposed reinforcement learning algorithm with the two naive methods 'Random' and 'Sorted' in maximizing the total fee accrued over the course of mining 15 blocks. We consider a dataset comprising 5,000 transactions, each characterized by its unique Transaction ID, associated fee, and weight. We perform our experiments in two scenarios: one with a weight limit of 250,000 and the other with 500,000.

First, we analyze our transaction dataset. From Figure 6, we can see that the transaction pool distribution is complex. There are 'good' transactions, where the weight is low and the fee is high, and 'bad' transactions, where the weight is high, and the fee is low. Therefore, these transactions should be carefully managed to maximize rewards.

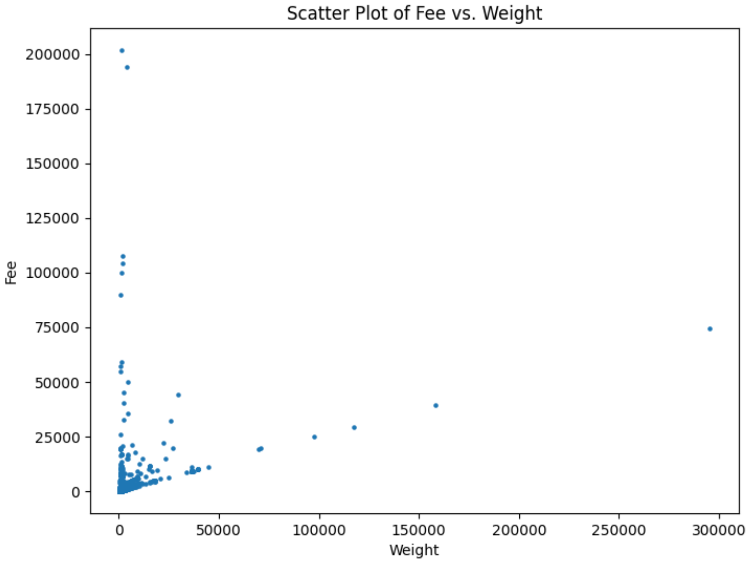


Figure 6. Relationship between fee and weight for the transaction dataset

Table 3 displays the total mining rewards for different transaction selection methods. We can see that the total mining rewards are lowest for ‘Random’, followed by ‘Sorted’, and are maximized when we apply our reinforcement learning algorithm to select transactions. More specifically, our reinforcement learning algorithm yields more than a 10% and nearly a 5% increase in rewards compared to 'Sorted' for weight limits of 250,000 and 500,000, respectively.

Random	Sorted	Reinforcement Learning
3,038,849	3,488,453	3,870,329

Weight Limit: 250,000

Random	Sorted	Reinforcement Learning
5,649,362	6,399,276	6,713,240

Weight Limit: 500,000

Table 3. Total mining reward for different transaction selection methods

Figure 7 displays the mining reward per block for various transaction selection methods. 'Sorted' selects the best transactions first, yet this leads to a significant degradation in performance over time. 'Random' selects transactions randomly and this results in suboptimal outcomes. Finally, our reinforcement learning algorithm is capable of balancing 'good' and 'bad' transactions and therefore could maximize rewards over time.

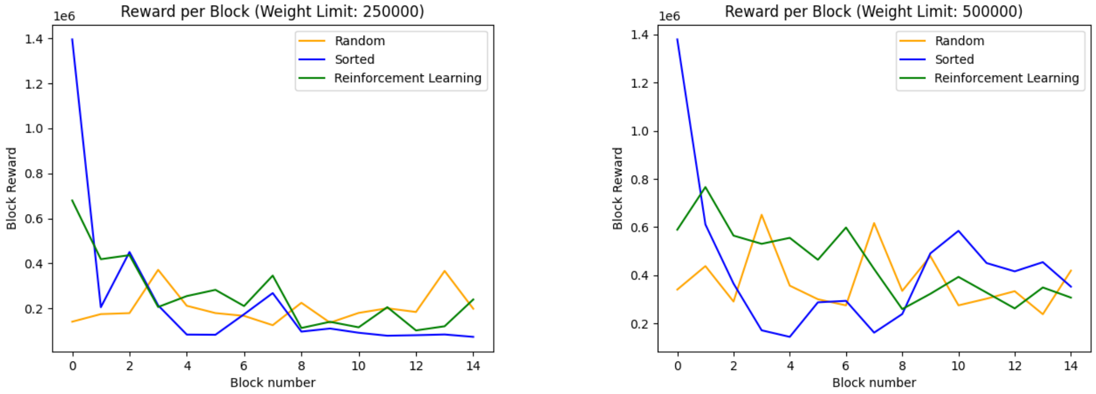


Figure 7. Mining reward per block for different transaction selection methods

5 Difficulties and Mitigations

The main challenge of our project lies in securing and managing the computational resources necessary for development.

Our project requires a significant amount of computational resources due to the complex and computationally intensive nature of training machine learning algorithms, as well as for blockchain mining. Our current strategy is to leverage the high-performance GPUs available at the HKU GPU farm, which will enable efficient computations and expedite the training process. However, recognizing the potential for high demand for these resources or the possibility that they may not be sufficient, we are prepared to seek additional support from other institutions, departments, or external partners. Furthermore, if the need arises, we are ready to allocate a portion of our project budget to secure additional GPUs from GPU cloud services. These platforms provide flexibility and scalability, allowing us to adjust resources according to the specific needs of our project at any given time. By ensuring adequate computational resources, we aim to adhere to our project timeline and guarantee the highest standard of model training.

6 Future Plans

Table 4 displays the project schedule. As of now, the project is on schedule, with the project setup, literature review, data collection and environment preparation completed. Current efforts are focused on model training and improvement.

The first task underway involves conducting experiments with a larger dataset and an enhanced model architecture. In the paper 'Machine Learning for Alternative Mining in PoW-based Blockchains: Theory, Implications and Applications,' [24], the authors trained a machine learning algorithm using 780k Bitcoin data blocks to predict the starting seed for nonce iteration, achieving significant performance gains. Building on this, we are collecting more data, testing various model architectures, and conducting more comprehensive experiments. We are optimistic that these efforts will yield promising improvements.

The second ongoing task is to reimplement the reinforcement learning algorithm for chain-wide learning. Efforts are now concentrated on reviewing foundational knowledge and establishing the necessary infrastructure.

Time	Objectives
Sep 2023 (60 learning hours)	Focus: Project Setup and Detailed Project Plan - Define project objectives, scope, and deliverable - Develop a detailed project plan, including timelines and resource allocation - Set up the WordPress website for progress updates
Oct 2023 (60 learning hours)	Focus: Literature Review and Data collection - Research on existing and related works on blockchain block mining with machine learning - Collect historical blockchain data from various sources
Nov 2023 – Jan 2024 (200 learning hours)	Focus: Environment Setup and Model Training - Set up the Bitcoin Regtest and Testnet - Develop and experiment with different machine learning algorithms - Train and fine-tune model parameters. - Define metrics for performance evaluation
Feb – Mar 2024 (150 learning hours)	Focus: Model Evaluation and Improvements - Deploy model on Bitcoin Testnet for evaluation - Perform detailed evaluation on models' performance on defined metrics - Compare with baseline results to derive insights - Explore improvements in the mining algorithm
Apr 2024 (80 learning hours)	Focus: Documentation and Reporting: - Document the project findings, methodologies, and outcomes. - Prepare the final project report and presentation summarizing the research, analysis, and results.

Table 4. Project schedule

7 Conclusion

This project aims to apply machine learning to enhance the sustainability and efficiency of blockchain mining. More specifically, we introduce the use of machine learning to 1) enhance nonce finding, 2) optimize transaction selection, and 3) discover chain-level mining strategy. Although the project is still underway, the initial results are promising. The completed literature review provided a comprehensive overview of blockchain mining, laying the groundwork for the subsequent steps. A substantial amount of high-quality data was also successfully gathered, and our preliminary experiments showed that machine learning could help speed up nonce discovery and optimize transaction selection. In the following months, the primary focus will be on model development and improvement. More extensive experiments will be carried out, and enhanced model architectures will be constructed. Following that, the combined pipeline will be deployed on the Bitcoin Testnet for detailed evaluation. Lastly, a comprehensive final report will be prepared, and a presentation will be delivered to summarize the project's outcomes. By developing a more efficient and sustainable block mining strategy, this project aims to encourage miners to adopt environmentally conscious mining methods, thereby contributing to a greener future in blockchain mining.

Reference List

[1] Joseph Bonneau; Andrew Miller; Jeremy Clark; Arvind Narayanan; Joshua A. Kroll; Edward W. Felten, "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," in *IEEE symposium on security and privacy*, 2015.

[2] Ittay Eyal, Emin Gun Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," *Communications of the ACM,* vol. 61, no. 7, pp. 95-102, 2018.

[3] S. Nakamoto, A Peer-to-Peer Electronic Cash System, Decentralized business review, 2008.

[4] Juan A. Garay, Aggelos Kiayias, Nikos Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2015.

[5] Raza, Ali, Kyunghyun Han, and Seong Oun Hwang, "A framework for privacy preserving, distributed search engine using topology of DLT and onion routing," *IEEE Access ,* vol. 8, pp. 43001-43012, 2020.

[6] Katie Martin, Billy Nauman, "Bitcoin's growing energy problem: 'It's a dirty currency'," Financial Times, 2021.

[7] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation,* vol. 18, no. 7, pp. 1527-1554, 2006.

[8] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Neural Information Processing Systems*, 2012.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,* arXiv preprint, 2018.

[10] Taotao Wang, Soung Chang Liew, and Shengli Zhang, "When Blockchain Meets AI: Optimal Mining Strategy Achieved By Machine Learning," in *International Journal of Intelligent Systems*, 2019.

[11] Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA: MIT Press, 2018.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford and Oleg Klimov, *Proximal Policy Optimization Algorithms,* arXiv preprint, 2017.

[13] OpenAI: Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P., *Dota 2 with Large Scale Deep Reinforcement Learning,* arXiv preprint, 2019.

[14] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, Nando de Freitas, *Sample efficient actor-critic with experience replay,* arXiv preprint, 2016.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, *Playing Atari with Deep Reinforcement Learning,* arXiv preprint, 2013.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King,

Dharshan Ku, “Human-level control through deep reinforcement learning,” *nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[17] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, Nando de Freitas, “Dueling network architectures for deep reinforcement learning,” in *International conference on machine learning*, 2016.

[18] Wenbo Wang; Dinh Thai Hoang; Peizhao Hu; Zehui Xiong; Dusit Niyato; Ping Wang; Yonggang Wen; Dong In Kim, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” *IEEE journal,* vol. 7, pp. 22328-22370, 2019.

[19] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, Srdjan Capkun, “On the Security and Performance of Proof of Work Blockchains,” *2016 ACM SIGSAC Conference on Computer and Communications Security,* pp. 3-16, 2016.

[20] Yonatan Sompolinsky and Aviv Zohar, “Secure high-rate transaction processing in Bitcoin”.

[21] Ittay Eyal, Emin Gun Sirer, “Majority is not enough: Bitcoin mining is vulnerable,” *Communications of the ACM,* vol. 61, no. 7, pp. 95-102, 2018.

[22] Bitcoin, “Bitcoin Regtest,” GitHub, [Online]. Available: https://github.com/bitcoin/bitcoin/tree/master/src/test. [Accessed 20 November 2023].

[23] “Testnet,” Bitcoin Wiki, [Online]. Available: https://en.bitcoin.it/wiki/Testnet. [Accessed 29 November 2023].

[24] Hamza Baniata, Radu Prodan, Attila Kertesz, *Machine Learning for Alternative Mining in PoW-based Blockchains: Theory, Implications and Applications,* Authorea Preprints, 2023.