Hindawi
Security and Communication Networks
Volume 2021, Article ID 5959698, 8 pages
https://doi.org/10.1155/2021/5959698

Research Article

ForkDec: Accurate Detection for Selfish Mining Attacks

Zhaojie Wang[1], **Qingzhe Lv**[1], **Zhaobo Lu**[1], **Yilei Wang**[1,2] and **Shengjie Yue**[3]

[1]*School of Computer Science, Qufu Normal University, Rizhao 276826, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, 541004, China*
[3]*School of Information Science and Engineering, University of Jinan, Jinan 250022, China*

Correspondence should be addressed to Yilei Wang; wang_yilei2019@qfnu.edu.cn

Received 25 October 2021; Accepted 19 November 2021; Published 30 November 2021

Academic Editor: Yuling Chen

Incentive mechanism is the key to the success of the Bitcoin system as a permissionless blockchain. It encourages participants to contribute their computing resources to ensure the correctness and consistency of user transaction records. Selfish mining attacks, however, prove that Bitcoin's incentive mechanism is not incentive-compatible, which is contrary to traditional views. Selfish mining attacks may cause the loss of mining power, especially those of honest participants, which brings great security challenges to the Bitcoin system. Although there are a series of studies against selfish mining behaviors, these works have certain limitations: either the existing protocol needs to be modified or the detection effect for attacks is not satisfactory. We propose the ForkDec, a high-accuracy system for selfish mining detection based on the fully connected neural network, for the purpose of effectively deterring selfish attackers. The neural network contains a total of 100 neurons (10 hidden layers and 10 neurons per layer), learned on a training set containing about 200,000 fork samples. The data set, used to train the model, is generated by a Bitcoin mining simulator that we preconstructed. We also applied ForkDec to the test set to evaluate the attack detection and achieved a detection accuracy of 99.03%. The evaluation experiment demonstrates that ForkDec has certain application value and excellent research prospects.

1. Introduction

Bitcoin is essentially a decentralized, distributed public ledger, which allows anyone or institution to participate in publishing transactions in a client-side manner [1]. The transaction will be collected by the participants (called miners) in the network and then added to the public ledger through a consensus protocol. The consensus protocol adopted by Bitcoin is called Proof-of-Work. All miners compete to solve a difficult-to-solve but easy-to-verify cryptographic puzzle. The miner who successfully solves the puzzle first is allowed to add transactions to the ledger and receive Bitcoin rewards [2]. Incentive mechanism is central to the functionality of Bitcoin, which ensures the security and liveness of Bitcoin by encouraging a large number of honest miners to participate in the consensus process [3]. Traditionally, it is believed that Bitcoin's incentive mechanism is incentive-compatible, but the emergence of selfish mining proves that this opinion is inaccurate [2]. By strategically publishing previously withholding blocks to invalidate blocks mined by honest miners, selfish attackers can collect additional reward shares that should belong to honest miners. The harm of selfish mining attacks is not limited to this. Unfair reward distribution will induce some rational participants to be selfish. A large number of selfish participants may also launch collusive attacks to infringe the revenue of other honest participants, which will seriously damage Bitcoin's reputation. Resulting in plenty of honest miners quitting will weaken the security significantly and give other attacks (e.g., double-spending attacks) an opportunity to take advantage of. Although selfish mining attacks have not been discovered in the real world, with the continuous improvement of potential attackers' computing power and the iterative upgrade of attack algorithms [4–10], the possibility of this attack is gradually increasing. We consequently must attach great importance to the detection

of this attack to ensure that it can be discovered and countermeasures are taken as soon as possible when an attack occurs.

1.1. Related Works. Ethan Heilman proposed a method based on unforgeable timestamps against selfish mining [11], called *Freshness Preferred*. It requires miners to add unforgeable timestamps to blocks, and it invalidates the blocks withheld by attackers by encouraging honest miners to choose blocks with the latest timestamp. The disadvantage of this method, however, is that it requires a credible timestamp agency to generate unforgeable timestamps and requires honest miners to record all recent timestamp release logs. Solat et al. [12] proposed a new solution that does not use unforgeable timestamps, called the *ZeroBlock*. The idea is that if selfish miners withhold blocks for more than a preset time interval, all honest miners will directly reject the block. The *ZeroBlock* scheme forces the selfish attacker to be unable to withhold blocks for a long time. Zhang et al. proposed the *Weighted Fork-Resolving Policy*. When a fork occurs, a weight is calculated for each branch. And, it is recommended that honest miners no longer simply rely on the length of the branch when determining the main chain but choose the branch with the largest weight [13]. Saad et al. [14] assigned an expected confirmation height (i.e., the expected height of the block containing the specified transaction) to each transaction by measuring the transaction size, transaction fee, and other factors. The smaller the gap between the actual confirmation height and the expected height, the lower the possibility of selfish mining behavior. Lee et al. increased the profit threshold of selfish mining from 25% to 33% by adding transaction creation time to the transaction data structure [15]. Chicarino et al. [16] analyzed the impact of selfish mining on Bitcoin's fork height and judged whether a selfish mining attack occurred by monitoring the abnormal changes in the fork height.

1.2. Motivation and Contribution. Since Eyal and Sirer proposed the concept of selfish mining and pointed out its harmfulness; a series of studies on this attack have appeared [4, 10, 17, 18, 19, 20]. The main focus of most research, however, is to increase the attacker's rewards or reduce the mining power threshold. By contrast, there are relatively few research studies on selfish mining defense measures [11–16], and many works require upgrading the existing protocol, which is costly to implement. The selfish mining detector [16] proposed by Chicarino et al. realized the detection of selfish mining without modifying the Bitcoin protocol. However, it only considers the factor of fork height and does not take other factors into consideration, which leads to a certain misjudgment rate. To improve the detection accuracy, in this work, we propose a selfish mining attack detection system based on a machine learning classification model, called ForkDec. The system can detect selfish mining attacks in the Bitcoin network with an accuracy rate of 99.03%. Our primary contributions are threefold as follows:

(1) We construct a data set containing approximately 200,000 fork samples. Considering that selfish mining has not been discovered in reality, we build a Bitcoin mining simulator to simulate the Bitcoin mining process in the presence of propagation delays and selfish miners. In the simulation process, the simulator records all the fork features, and then the feature extractor extracts feature vectors based on the fork features to construct fork samples.

(2) We present ForkDec as an accurate detection system for detecting selfish mining attacks in Bitcoin. To accurately detect selfish mining, we trained a classification model based on logistic regression and a fully connected neural network (with 10 hidden layers and 10 neurons per layer) on the training set, respectively, and then applied the learned model to ForkDec for attack detection.

(3) We applied ForkDec to the test set to evaluate its performance. The evaluation results show that ForkDec is better than the selfish mining detector [16] in accuracy. In addition, we also found that the classification model based on the fully connected neural network has a better overall performance.

1.3. Roadmap of This Paper. The rest of the paper is organized as follows. In Section 2, we introduce the details of the ForkDec system, including the construction of the data set and the selection of the classification model. In Section 3, the evaluation results and discussion of the proposed system are given. Finally, we conclude this work in Section 4.

2. ForkDec: System Description

Figure 1 presents the basic architecture of the ForkDec system. It mainly includes three modules: data set construction, model training, and attack detection. Firstly, we built a simulator to simulate the Bitcoin network with selfish attackers. The simulator will record the information of each block (block height, miner, and timestamp), especially fork features, and then each fork will be delivered to the feature extractor to extract the feature vector to construct the fork data set. We, subsequently, use the built training set to train the classification model and embed the learned model into ForkDec for attack detection.

2.1. Data Set Construction. The classification model relies on the training set to learn sample features and to identify unknown samples. To get an excellent attack detection model, we must have a training set with abundant selfish mining samples. Since machine learning has not been applied to selfish mining detection before, there is no existing data set available. To solve this issue, we constructed a data set containing 200,000 fork samples for model training, in which the ratio of natural fork samples to attacking fork samples is 3 : 7.

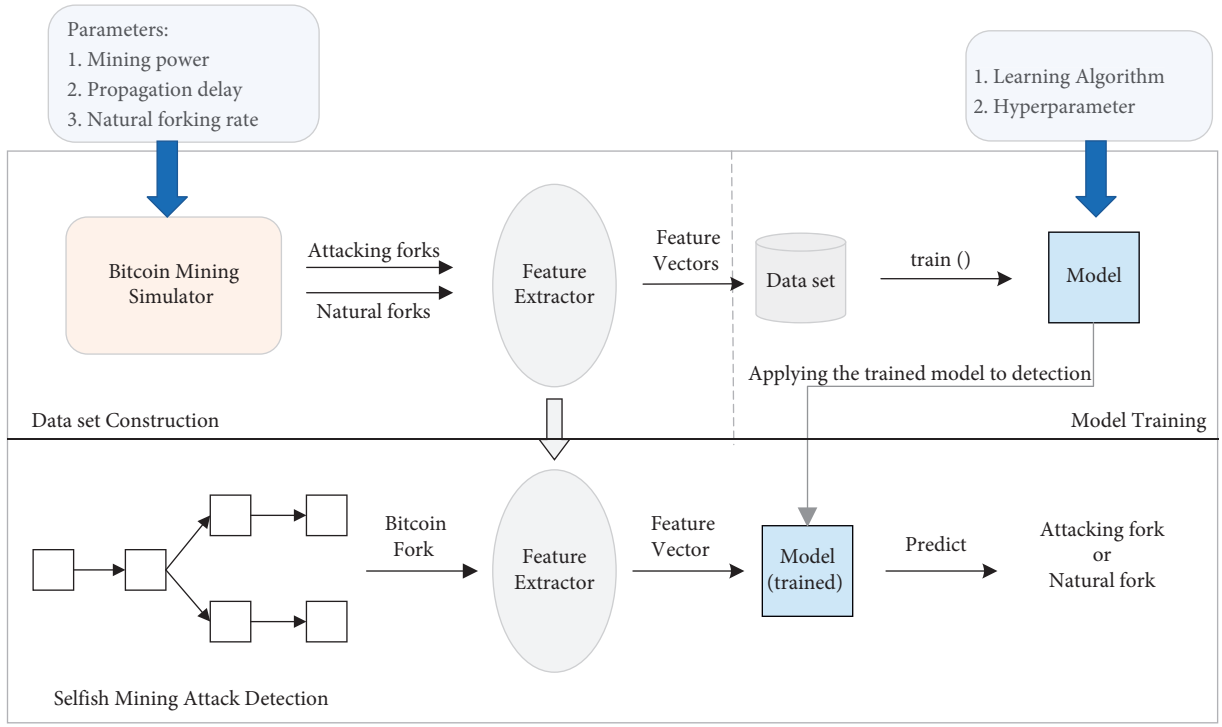

Figure 1: Schematic of the ForkDec detection system.

2.1.1. Feature Vector Extraction. All miners in the Bitcoin network utilize Proof-of-Work to compete for accounting rights to create new blocks at an average rate of 10 minutes. After being created, the new block will be broadcast immediately by all honest miners in the Peer-to-Peer network. Unlike honest miners, the selfish attacker will secretly withhold newly mined blocks to create conflicting branches. Then, the attacker invalidates the blocks mined by honest counterparts through strategically publishing the withheld blocks. In this way, the attacker could increase his proportion of rewards distribution. By studying the strategy of the selfish attacker, it can be known that the attacker carries out attacks by making forks. Therefore, the key to detecting this type of attack is to track the fork data in the blockchain. Based on this, we construct a feature extractor to represent the fork data as a feature vector. The classification model learns the characteristics of the selfish mining attack through the feature vector, thereby detecting the attacks. In the Bitcoin, we define the structure of the feature vector as follows: $\{h, l, i_b, i_t\}$. The meaning of each feature is as follows:

(i) h is the block height of the fork

(ii) l is the length of the fork, i.e., the number of blocks on the conflicting branch

(iii) i_b is the number of blocks between this fork and the previous fork

(iv) i_t is the absolute value of the difference between the timestamps of the first block of each branch

Subsequently, we use an example to present the general process of feature vector extraction, as illustrated in Figure 2. For simplicity and without loss of generality, we assume that b_0 is the first block after the previous fork is resolved, and its timestamp is t_0. After b_1 is accepted by all participants, two valid blocks b_2 (with timestamp t_2) and b_2' (with timestamp t_2') are propagated in the P2P network. Consequently, the blockchain makes a fork since b_2 and b_2' have the same block height, i.e., $h(b_2)$. Note that we utilize $h(x)$ to indicate the height of block x. The Bitcoin mining simulator will capture and record information about this fork. Then, the extractor converts this information into a 4-dimensional vector, which is the feature vector on the far-right side of Figure 2.

2.1.2. Fork Sample Generation. Under the setting that only considers selfish mining attacks, there are two types of forks in the Bitcoin network: natural forks and attacking forks. Natural fork means that when a block is propagated in the network, other miners create and broadcast a block with the same height, which leads to inconsistencies in the distributed ledger. This inconsistency is not caused by the attack but by network propagation delays [21]. Christian Decker and Roger Wattenhofer pointed out that the average delay of a block in Bitcoin is 12.6 seconds, and after the new block is broadcast for 40 seconds, 95% of the nodes have received the block [21]. In other words, the timestamp difference between most conflicting blocks in the Bitcoin network is close to the average propagation delay. Based on this, we adopt an exponential distribution with the expected value of 12.6 seconds to approximate the block propagation delay distribution, as shown in Figure 3. The simulator then randomly samples based on the distribution to simulate the timestamp interval of a natural fork.

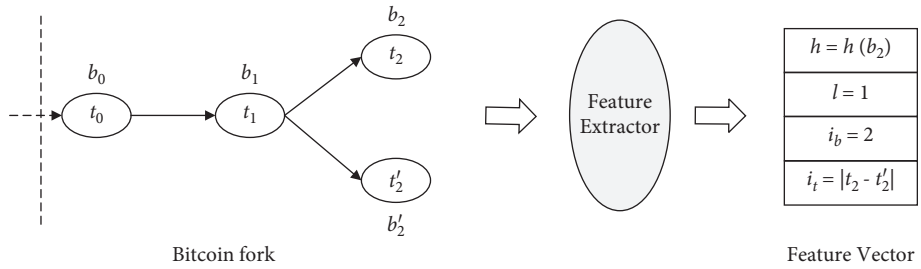

Figure 2: A clearly expressed example of feature vector extraction.

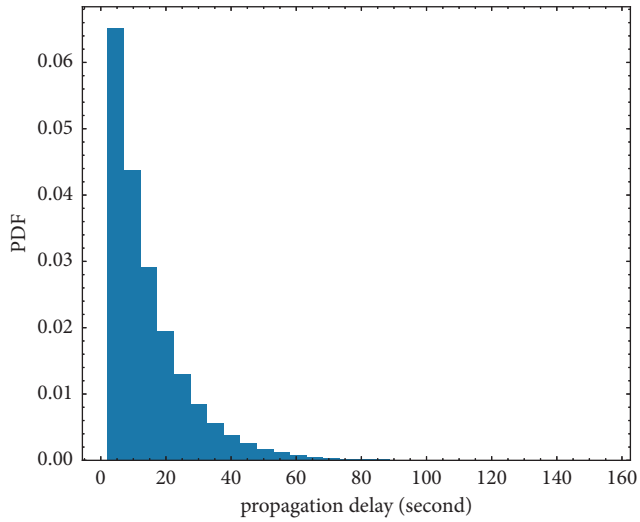

Figure 3: The sampling distribution of propagation delay.

The opposite is the attacking fork which is caused by a malicious attack. Figure 4 shows the formation of an attacking fork. Assume that the selfish attacker firstly mines block A_1 at time t. According to the SM1 strategy [2], the attacker will secretly withhold block A_1. Since honest miners will not perceive the existence of A_1 until it is published, the honest miners may mine a new block A_2 at any time $t\prime$ after time t. Then, there is $0 < t\prime - t$; considering the average block creation time is 600 seconds (10 minutes), we set $0 < t\prime - t \leq 600$. That is when the simulator is simulating an attacking fork, the timestamp interval of conflicting blocks is randomly sampled between 0 and 600 seconds.

2.2. Classification Model. The selection of the learning algorithm is another key point for ForkDec to realize high-accurate detection. It is impossible to get an efficient model if the learning algorithm is not well selected and even if there are rich sample data sets to utilize. We, respectively, test the detection effect of ForkDec when logistic regression and a fully connected neural network are used as the classification model. Among them, the logistic regression features a faster model convergence rate while the fully connected neural network performs better in accuracy rate.

2.2.1. Logistic Regression. Logistic regression is a classification model that utilizes a linear model to predict binary classification problems. The idea is to map the output of the linear model (any continuous value) to a value between 0 and 1 by adding the sigmoid function after the linear model. Equation (1) presents the mathematical expression of logistic regression, where x^T represents the sample to be classified, (w, b) represents the model parameter, and $\hat{y}$ represents the prediction results of the model (also called the confidence level):

$$\begin{aligned}\hat{y} &= \text{Sigmoid}\left(x^T w + b\right) \\ &= \frac{1}{1 + \exp\left(x^T w + b\right)}.\end{aligned} \tag{1}$$

By setting the threshold to 0.5, the ForkDec classifies fork samples with a confidence level of more than 0.5 as attacking forks, otherwise as natural forks. In addition, to prevent overfitting, we use minimizing the cost function (with the L2 penalty term) as the optimization problem during model training and then apply the L-BFGS algorithm, a kind of quasi-Newton method, to solve the optimization problem.

2.2.2. Fully Connected Neural Network. Logistic regression has the characteristics of clear structure and simplicity. However, on the other hand, a simple model may not be able to make full use of the rich training samples and cannot achieve top-notch detection results. To further improve the accuracy in attack detection, we additionally consider the use

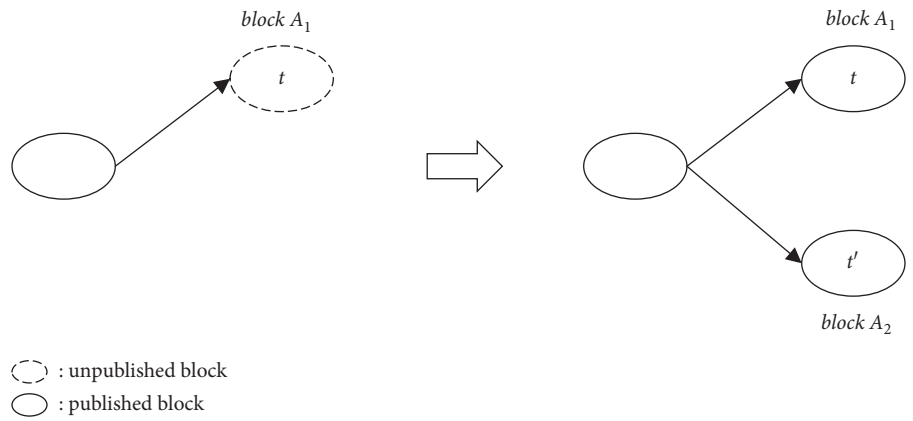

Figure 4: The example of an attacking fork.

of fully connected neural network, also known as multilayer perceptron, as the classification model. Figure 5 presents the structure of the fully connected neural network.

The input layer on the far left is composed of a group of neurons $\{x_i|x_1, x_2, \ldots, x_m\}$ representing the characteristics of the sample. Unlike logistic regression, there can be one or more nonlinear layers between the input layer and output layer of a neural network, called hidden layers. The neurons in each hidden layer perform a weighted linear summation conversion on the values of the previous layer. The converted value firstly passes through the activation function and then is delivered to the next layer until the final output layer. In the ForkDec system, we utilize backpropagation to train the neural network, and finally, we get a fully connected neural network with 10 hidden layers and 10 neurons in each layer.

3. Evaluation

In this section, we evaluate the performance of ForkDec in detecting selfish mining attacks. The ForkDec system utilizes Scikit-learn (version 1.0) to implement the model training. Scikit-learn, an open-source and efficient machine learning tool library, is implemented based on the Python program language. Subsequently, we embed the trained model into the ForkDec system and test it on a test set containing 76,686 samples. The test results show that the ForkDec system can achieve a detection accuracy of 99.03% when the fully connected neural network is used as the classification model and 98.76% when using logistic regression. We additionally compare the performance of the ForkDec detection system with the selfish mining detector (hereinafter referred to as SM detector) proposed in [16]. We also train the fully connected neural networks under different hyperparameters to find the optimal model and then detect selfish attackers with different abilities.

3.1. Comprehensive Performance. By applying ForkDec to a test set containing 76,686 samples, the confusion matrix of ForkDec in detecting selfish mining attacks can be obtained, which is presented in Table 1. In the confusion matrix, the classification results of ForkDec and the real distribution of the samples are shown, where positive represents the attacking fork category and negative represents the natural fork category. To facilitate the description, we name the ForkDec system, respectively, according to the different classification models:

(i) ForkDec-DNN is the ForkDec system with the fully connected neural network as the classification model

(ii) ForkDec-LR is the ForkDec system with logistic regression as the classification model

(iii) ForkDec is the collective name of ForkDec-DNN and ForkDec-LR

From Table 1, we can see that the advantage of ForkDec-DNN is that it does not misclassify natural forks as attacking forks, while ForkDec-LR misclassifies 542 natural forks as attacking forks. However, ForkDec-DNN also has its disadvantages; that is, 745 attacking forks are misidentified as natural forks by ForkDec-DNN, while this value is only 407 for ForkDec-LR.

To more intuitively evaluate the performance of ForkDec, we present the accuracy, precision, recall, and F_1 value of ForkDec on the test set in Figure 6. The meanings of these indicators are as follows:

(i) Accuracy: it is the proportion of correctly classified samples to the total sample.

(ii) Precision: among all attacking fork samples detected by the model, precision is the proportion of real attacking samples.

(iii) Recall: among all the attacking samples, recall is the proportion detected by the model.

(iv) F_1: the F_1 value, shown in (2), can be used to measure the comprehensive performance of the model in terms of precision and recall. The reason is that the F_1 value is only high when both precision and recall are high:

$$\frac{1}{\text{precision}} + \frac{1}{\text{recall}} = \frac{2}{F_1}. \tag{2}$$

It can be found in Figure 6 that ForkDec-DNN and SM Detector both have top scores in precision rate, which

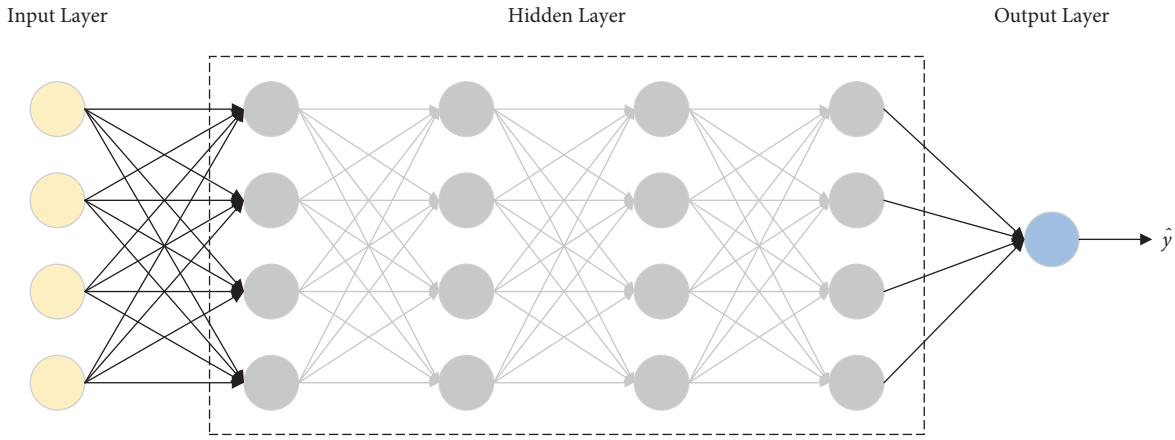

Figure 5: The example of a fully connected neural network with 4 hidden layers and 4 neurons per layer.

Table 1: The confusion matrix of ForkDec. In the values (x, y), x indicates the classification result of ForkDec-DNN and y represents the result of ForkDec-LR.

The real distribution of samples	The classification results of ForkDec	
	Positive	Negative
Positive	(57238, 57576)	(745, 407)
Negative	(0, 542)	(18703, 18161)

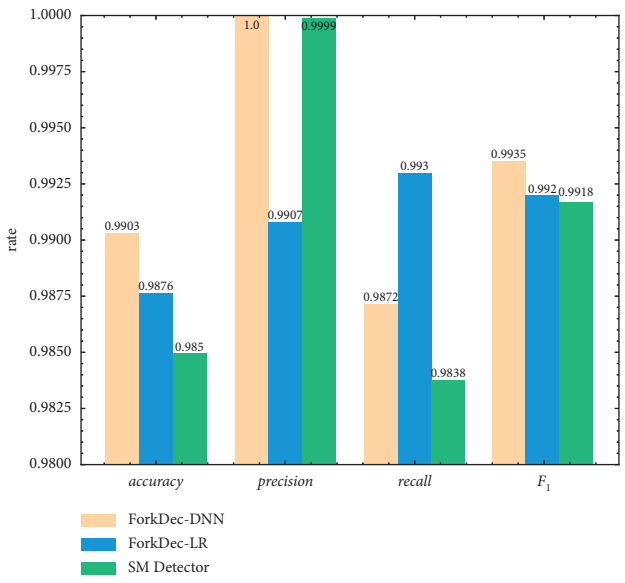

Figure 6: The performance of ForkDec-DNN, ForkDec-LR, and SM Detector.

indicates that both can ensure that there are almost no false positives in all detected attacking forks. Precision rate and recall rate are a pair of contradictory indicators. ForkDec-DNN and SM Detector pursue the ultimate precision rate, which also means that both will have a loss in the recall rate. However, the loss of SM Detector's recall rate is greater, so this leads to a lower F_1 value of SM Detector. Moreover, ForkDec-DNN also has the highest accuracy rate among the three, which cannot be achieved by SM Detector. Unlike ForkDec-DNN's extreme performance in precision rate, ForkDec-LR balances various indicators. In particular, ForkDec-LR has the highest recall rate among the three. In other words, ForkDec-LR can detect attacking forks as many as possible, with only a few false negatives.

3.2. The Optimal Model. To find the optimal model, we train the fully connected neural networks under different hyperparameters. Then, we apply these trained models to the test set. The performance of these models on the test set is shown in Table 2. It can be concluded from Table 2 that a neural network with 10 hidden layers and 10 neurons per layer has the best performance. And, more neurons do not mean better classification performance. It is worth mentioning that a neural network with 10 hidden layers and 10 neurons per layer may not be optimal, but its performance is close to the optimal model.

3.3. Detection for Attacker with Varying Power. In order to fully evaluate the detection effect of ForkDec, we additionally considered the detection of selfish attackers under specific mining power. We first utilize α to represent the fraction of attacker's mining power in the power of the entire Bitcoin network. Figure 7 presents the detection effect of ForkDec against different power attackers. We notice that the accuracy rate, recall rate, and F_1 value drop rapidly when $\alpha > 0.25$. This is because, as the attacker's mining power increases, the frequency of the selfish mining attack is getting higher and higher, resulting in a large number of forks with close timestamps in the blockchain. Many of these forks are not correctly detected by the model, leading to a drop in recall rate, as the characteristics of such attacking forks are very similar to natural forks. Then, the accuracy rate and F_1

Table 2: The performance of neural networks with different hyperparameters on the test set.

Model	Accuracy	Precision	Recall	F_1
8 layers × 10 neurons	0.9902329	1.0	0.9870824	0.9934992
9 layers × 10 neurons	0.9902459	1.0	0.9870997	0.9935080
10 layers × 10 neurons	0.9902851	1.0	0.9871514	0.9935342
11 layers × 10 neurons	0.9902590	1.0	0.9871169	0.9935167
12 layers × 10 neurons	0.9902459	1.0	0.9870997	0.9935080
12 layers × 12 neurons	0.9902199	1.0	0.9870652	0.9934905

m layers × n neurons indicates a neural network with m hidden layers and n neurons per layer.

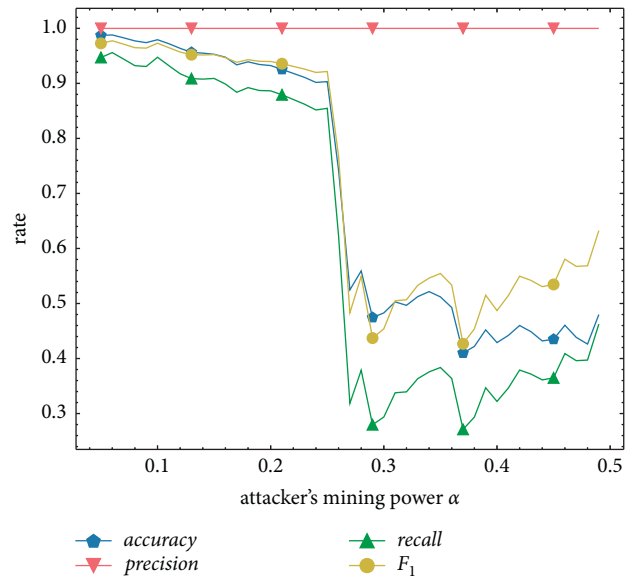

Figure 7: The detection of selfish attacker with varying mining power.

value also drop. However, even in the face of powerful attackers, ForkDec still maintains a very high-accuracy rate. It can still ensure that there are almost no false positives during the detection process.

4. Conclusion

In this work, we propose a detection system for selfish mining attacks in Bitcoin, called ForkDec. The system is based on the machine learning classification model to realize intelligent detection of attacks. To ensure that ForkDec has a high detection accuracy, we construct a data set containing about 200,000 Bitcoin fork samples for model training. We then apply ForkDec to the test set for evaluation. The evaluation results show that ForkDec can achieve an accuracy of 99.03% for the detection of selfish mining in Bitcoin. What needs to be clear is that ForkDec can only detect the presence of an attack but cannot identify the miner who launched the attack. In future work, we will further analyze the attacker's strategy and improve ForkDec to accurately locate the attacker. In addition, the blockchain also applies in the fields of privacy protection [22] and data traceability. Attackers may use other methods to attack the blockchain. Hence, we also have to study the application of ForkDec to the detection of other attacks, e.g., double-spending attacks [23], time-bandit attacks [24], and blockchain DoS attacks [25].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Foundation of National Natural Science Foundation of China (Grant nos. 62072273, 72111530206, 61962009, 61873117, 61832012, 61771231, and 61771289); Natural Science Foundation of Shandong Province (Grant no. ZR2019MF062); Shandong University Science and Technology Program Project (Grant no. J18A326); Guangxi Key Laboratory of Cryptography and Information Security (Grant no. GCIS202112); Major Basic Research Project of Natural Science Foundation of Shandong Province of China (Grant no. ZR2018ZC0438); Major Scientific and Technological Special Project of Guizhou Province (Grant no. 20183001); Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Grant no. 2019BD-KFJJ009); and Talent Project of Guizhou Big Data Academy, Guizhou Provincial Key Laboratory of Public Big Data (Grant no. [2018]01).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, Article ID 21260, 2008.

[2] I. Eyal and E. G. Sirer, "Majority is not enough: bitcoin mining is vulnerable," in *Proceedings of the International conference on Financial Cryptography and Data Security*, pp. 436–454, Kota Kinabalu, Malaysia, February 2014.

[3] C. Hou, M. Zhou, Y. Ji et al., "SquirRl: automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning," 2019, https://arxiv.org/abs/1912.01798.

[4] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceedings of the International conference on Financial Cryptography and Data Security*, pp. 515–532, Christ Church, Barbados, February 2016.

[5] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "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, Dallas TX USA, October 2017.

[6] S. Gao, Z. Li, Z. Peng, and B. Xiao, "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*, pp. 833–850, Auckland, New Zealand, July 2019.

[7] L. Tao, W. Zhaojie, Y. Guoyu, and C. Yang, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, pp. 3596–3612, 2021.

[8] L. Tao, C. Yuling, W. Yanli et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.

[9] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li, "IPBSM: an optimal bribery selfish mining in the presence of intelligent and pure attackers," *International Journal of Intelligent Systems*, vol. 35, no. 11, pp. 1735–1748, 2020.

[10] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: generalizing selfish mining and combining with an eclipse attack," in *Proceedings of 2016 IEEE European Symposium on Security and Privacy*, pp. 305–320, Saarbruecken, Germany, March 2016.

[11] E. Heilman, "One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract)," in *Proceedings of the International conference on Financial Cryptography and Data Security*, pp. 161-162, Okinawa, Japan, April 2014.

[12] S. Solat and M. Potop-Butucaru, "Zeroblock: timestamp-free prevention of block-withholding attack in bitcoin," 2016, https://arxiv.org/abs/1605.02435.

[13] R. Zhang and B. Preneel, "Publish or perish: a backward-compatible defense against selfish mining in bitcoin," in *Proceedings of Cryptographers'Track at the RSA Conference*, pp. 277–292, San Francisco, CA, USA, February 2017.

[14] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *Proceedings of 2019 International Conference on Computing, Networking and Communications*, pp. 360–364, Honolulu, HI, USA, February 2019.

[15] J. Lee and Y. Kim, "Preventing bitcoin selfish mining using transaction creation time," in *Proceedings of 2018 International Conference on Software Security and Assurance*, pp. 19–24, Seoul, Korea, July 2018.

[16] V. Chicarino, C. Albuquerque, E. Jesus, and A. Rocha, "On the detection of selfish mining and stalker attacks in blockchain networks," *Annals of Telecommunications*, vol. 75, no. 3-4, pp. 143–152, 2020.

[17] L. Tao, W. Zhaojie, C. Yuling, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, .

[18] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in *Proceedings of the International conference on financial cryptography and data security*, Kota Kinabalu, Malaysia, February 2020.

[19] G. Cyril and R. Pérez-Marco, "On profitability of selfish mining," 2018, https://arxiv.org/abs/1805.08281.

[20] R. B. Zur, I. Eyal, and A. Tamar, "Efficient MDP analysis for selfish-mining in blockchains," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pp. 113–131, New York, NY, USA, October 2020.

[21] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *Proceedings of IEEE P2P*, pp. 1–10, Trento, Italy, September 2013.

[22] C. Yuling, S. Jing, Y. Yixian, T. Li, X. Niu, and H. Zhou, "PRSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, .

[23] M. Rosenfeld, "Analysis of hashrate-based double spending," 2014, https://arxiv.org/abs/1402.2009.

[24] P. Daian, S. Goldfeder, T. Kell et al., "Flash boys 2.0: front-running, transaction reordering, and consensus instability in decentralized exchanges," 2019, https://arxiv.org/abs/1904.05234.

[25] M. Mirkin, Y. Ji, J. Pang, A. Mundt, I. Eyal, and A. Juels, "BDoS: blockchain denial-of-service," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 601–619, New York, NY, USA, November 2020.