Research Article

Nakamoto Consensus to Accelerate Supervised Classification Algorithms for Multiparty Computing

Zhen Zhang ¹, Bing Guo,¹ Yan Shen,² Chengjie Li ³, Xinhua Suo,¹ and Hong Su ¹

¹College of Computer Science, Sichuan University, Chengdu 610065, China

²School of Computer Science, Chengdu University of Information Technology, 610225, China

³School of Computer Science and Technology, Southwest Minzu University, Chengdu 610225, China

Correspondence should be addressed to Chengjie Li; chengjieabc@hotmail.com

Received 12 December 2020; Revised 12 February 2021; Accepted 28 February 2021; Published 11 March 2021

Academic Editor: Omar Cheikhrouhou

Copyright © 2021 Zhen Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bitcoin mining consumes tremendous amounts of electricity to solve the hash problem. At the same time, large-scale applications of artificial intelligence (AI) require efficient and secure computing. There are many computing devices in use, and the hardware resources are highly heterogeneous. This means a cooperation mechanism is needed to realize cooperation among computing devices, and a good calculation structure is required in the case of data dispersion. In this paper, we propose an architecture where devices (also called nodes) can reach a consensus on task results using off-chain smart contracts and private data. The proposed distributed computing architecture can accelerate computing-intensive and data-intensive supervised classification algorithms with limited resources. This architecture can significantly increase privacy protection and prevent leakage of distributed data. Our proposed architecture can support heterogeneous data, making computing on each device more efficient. We used mathematical formulas to prove the correctness and robustness of our system and deduced the condition to stop a given task. In the experiments, we transformed Bitcoin hash collision into distributed computing on several nodes and evaluated the training and prediction accuracy for handwritten digit images (MNIST). The experimental results demonstrate the effectiveness of the proposed method.

1. Introduction

Artificial intelligence (AI) has significantly affected human life in various aspects, solving various tasks, such as image classification and object detection based on supervised classification algorithms. Supervised classification algorithms use computational methods to learn information directly from data, where a positive and proportional relationship exists between the number of training samples and the accuracy of prediction results. The increase in the number of training samples would mean increasing the algorithm's training sample time. Thus, having an architecture where the nodes can quickly get results and reach a consensus through off-chain smart contracts and private data would be extremely useful.

At present, the research on computational power based on blockchains can be summarized as follows:

- (1) Some proposed works leverage auction mechanisms to off-load tasks [1–6]. In these works, an application is divided into multiple tasks, and the tasks are off-loaded to a cloud server or edge servers. Consuming time, energy, and edge servers' reputation are the indexes of auctions. To our knowledge, heterogeneous devices and privacy issues are not considered.
- (2) Several methodologies use deep learning to derive task off-loading for heterogeneous devices [7–9]. However, privacy issues have not been well solved, or computing devices must work in a permitted network.
- (3) Distributed computing based on Federated Learning (FL) has been proposed because it can protect privacy and reduce network burden [10–13]. FL can complete AI computation without disclosing data, but FL is not suitable for heterogeneous devices.

Smart contracts are naturally distributed computation technologies. As a mature technology, on-chain smart contracts have some shortcomings in running complex programs. For example, Bitcoin scripts are not Turing-complete [13], and Ethereum does not support the execution of complex computations. [14–16].

A consistent result must be obtained in calculating distributed and heterogeneous data while considering speed, energy consumption, and privacy protection. Reputation is an important index in evaluating nodes. Nodes with a higher reputation can process more tasks and get more rewards. Many blockchain-based computing models punish malicious nodes [17–19]. However, with different training samples and devices, even honest nodes can make mistakes in AI calculations (e.g., supervised classification algorithms). Such penalties will greatly fluctuate the reputation of nodes and affect the calculation results.

In our proposed model, off-chain smart contracts and private data (edge data center) are leveraged for multiparty computing. Our method can speed up training and improve prediction accuracy and privacy. Our experimental results on MNIST show that the cooperation of many low-power nodes is not weaker compared to centralized servers. The key points of this paper can be summarized as follows:

- (1) We propose strong privacy protection and compatibility-computing model for supervised classification algorithms to accelerate data training in supervised classification and improve the accuracy of prediction results.
- (2) In the proposed architecture, we proved that the prediction result with the most supporters is most likely to be the right one.
- (3) Using Nakamoto consensus, we calculate the impact of the number of nodes and the accuracy of single node prediction for the entire blockchain. We also obtained the condition for task termination through calculations.
- (4) The influences of malicious nodes and lazy nodes on the prediction results are discussed further, and we proved the robustness against malicious nodes and lazy nodes.

The remainder of this article is organized as follows. In Section 2, the background of the study is discussed, and the rationale behind the design is explained. In Section 3, the novel distributed computing architecture is demonstrated. Section 4 explains how the methodology improves prediction accuracy and the robustness of the supervised classification algorithm. Section 5 presents the experimental results and discusses the performance of the proposed architecture. Finally, Section 6 presents the conclusions of this study.

2. Background

Similar to traditional programs, smart contracts can be stored and executed. However, smart contracts are distributed programs residing in the blockchain. They are

automatically triggered according to instructions and do not require the participation of a third party. For a trigger event, the result of distributed execution needs to be unique, such that all nodes need to admit the solution known as the consensus mechanism. This section introduces the consensus mechanism in Bitcoin and presents relevant research on smart contracts.

2.1. Bitcoin and Nakamoto Consensus. At present, blockchain technology has become a research hot spot in finance, IoT, copyright protection, and information technology. It is a decentralized peer-to-peer (P2P) architecture, where the nodes consist as network participants. Blockchain establishes transparency and trust without third-party insurance.

As the first widely deployed and decentralized global currency, Bitcoin has attracted increasing attention. Nodes in Bitcoin compete to perform challenging Proof of Work (PoW) problems, in which solutions to the problems are worked out about every ten minutes. The winners who solve the problems get rewards through bonuses, which are stored in blocks. The blocks propagate among Bitcoin nodes through the network, so the bonuses are recorded by each node redundancy. When a block is accepted and added to the blockchain, the height of the blockchain is increased by one. In some cases, multiple nodes solve the problem before receiving solutions from other nodes, and so multiple blocks may be generated at a certain height.

A consensus mechanism is an algorithm in which a group of nodes can reach an agreement on events and sequences simultaneously. There are a lot of consensus mechanisms, such as PBFT [20], Paxos [21], Raft [22], Proof of Stake (PoS) [23], Tangle [24], and Shimmer [19]. The core technology to reach a consensus on Bitcoin is the Nakamoto consensus, as presented in Figure 1. In the figure, there are blocks chained in succession from $Block_0$ to $Block_{n-1}$, where $Block_0$ is at height 0 and $Block_{n-1}$ is at height $n-1$. At height n , node Alice and node Bob declare their blocks $Block_n$ and $Block_n^*$ concurrently. Node Carol receives $Block_n$ before $Block_n^*$, so $Block_n$ is added to Carol's fork as a tip, and $Block_n^*$ is stored as a backup. The same goes for Alice and Bob, wherein they build forks using their blocks, and subsequent blocks are stored in their memories as backup blocks. Alice and Carol keep the same tips in their memories, while Bob keeps them differently. At height $n+1$, Carol publishes its block $Block_{n+1}$ while $Block_n$'s hash is kept in $Block_{n+1}$'s header. Because the fork with $Block_{n+1}$ is longer than that with $Block_n^*$, when Bob receives $Block_{n+1}$, it keeps $Block_{n+1}$ in memory, activates $Block_{n+1}$'s previous blocks (i.e., $Block_n^*$), and then reserves $Block_n^*$ as a backup. At height $n+2$, Bob publishes its new block $Block_{n+2}$ with the header pointing to $Block_{n+1}$. As a result, Alice's bonus in $Block_n$ is accepted by every node, and Bob's bonus in $Block_n^*$ is ignored.

At height n , there are two blocks called forks. After several rounds of competition, the longest fork is considered to be the best chain in Bitcoin. The computing competition of the PoW problem for computing power incentives in Bitcoin is also called mining [25]. Bitcoin adjusts the

FIGURE 1: Nakamoto consensus mechanism.

difficulty of mining to ensure that a result is worked out every 10 minutes. In early October 2020, Bitcoin difficulty is 19.30 T, and the hash rate reaches 138.09 exahashes per second (EH/s) [26]. Such giant power makes Bitcoin the most energy-consuming application. According to Digiconomist [27], the estimated power used by miners to verify Bitcoin blockchain transactions is 70.89TWh a year, which is greater than the annual electric consumption of Colombia and 41 other countries. Therefore, the power waste in the hash collision has become an emerging concern.

2.2. Smart Contracts. The concept of smart contract was proposed by Nick Szabo in the 1990s [28]. He proposed embedding the concept of contracts into computer components. However, this concept was only theoretical because the technologies and protocols needed were not available at the time. Today, these requirements are available, allowing the implementation of smart contracts with blockchain technology.

There are two types of smart contracts: on-chain smart contracts and off-chain smart contracts. On-chain smart contracts are executed by all nodes in the network, such as Bitcoin scripts [29], Ethereum smart contracts [30], and Fabric chain codes [31]. On-chain smart contracts have three disadvantages. First, they must be run by all nodes, which means that they scale poorly. Second, the Turing downtime problem (also called the statement loop problem) directly causes the smart contract environment to execute script files in an infinite loop, resulting in increased running pressure until the system crashes. Third, although external data can be fed to smart contracts by oracles, data are visible to all nodes.

Off-chain smart contracts are executed outside of the core protocol. Only a subset of nodes need to execute them, such as the ongoing IOTA Smart Contracts [32], FastKitten [17], Ekiden [33], and ZoKrates [34]. In these systems, the

calculation of tasks is performed off-chain by using multi-party computation (MPC) [35, 36], and the consensus is reached on-chain. While off-chain smart contracts do not put burden on the network and can handle heterogeneous data, their overall security depends on the security of each device. The summary of details of these works is presented in Table 1. Though these works adopt off-chain smart contracts to enable efficient decentralized task execution at low cost, none consider the heterogeneous data and devices.

3. Novel Distributed Computing Architecture

Traditionally, computers spend much time training a large number of samples. There are many types of computing devices used ubiquitously around the world, such as smartphones, smart vehicles, and wearable devices. Therefore, we propose a blockchain-based architecture for supervised classification. The model can gather the computing power of scattered equipment and reduce calculation time while ensuring accuracy. Devices are heterogeneous; some are powerful with their hardware, some have strong operating systems, and some are efficient with their training samples. To retain each device's advantage, we propose a blockchain framework named RapidTrainChain with flexible off-chain smart contracts and compatible consensus named Proof of Prediction (PoP) for node cooperation. The longest chain is selected as the best chain in PoP. In each block of the chain, the transactions and task solutions are stored. The same solution is linked in the same fork, while different solutions are in different forks.

RapidTrainChain is designed as a distributed computing system to maximize overall performance and protect the data. The system architecture is shown in Figure 2. Off-chain algorithms and private data are managed by devices. Devices are also called nodes in the blockchain. When RapidTrainChain receives a task, the node determines whether to

TABLE 1: Works with off-chain smart contracts.

Works	Consensus mechanism	Platform	Environment	Privacy-preserving?	Computing-intensive tasks oriented?
[32]	Tangle	IOTA	Untrusted environment	Not mentioned	Not mentioned
[34]	Consensus-agnostic	Ethereum and other blockchains	Untrusted environment	High	No
[17]	Proof of correctness	Bitcoin	Trusted execution environment	High	No
[33]	Consensus-agnostic	Ekiden	Trusted execution environment	High	No
[37]	Consensus-agnostic	Arbitrum	Trusted execution environment	High	No
[38]	PoW	Lightning Network	Trusted execution environment	High	No
Ours	PoP	Bitcoin	Untrusted environment	High	Yes

FIGURE 2: System architecture for RapidTrainChain.

start a new task. If the task needs to be started, the node triggers the off-chain smart contract through an interface to start computing. The node can start working on a task if the node is free. After the nodes complete a task, the prediction results are stored in the blocks of RapidTrainChain. Same solutions are stored in the same fork, and the longest fork is considered the best chain. Solutions stored in blocks of the best chain are the final solution for the task. The node can decide when to stop working, as discussed in Section 4.5.

In contrast with the hash collision in Bitcoin or Ethereum, the accuracy of prediction results cannot be verified by nodes, so each legal prediction result from every node is stored in RapidTrainChain. Multiple nodes can also generate the same prediction results simultaneously, giving rise to blocks with the same prediction results stored on different forks. In this case, the prediction results with the most supporters cannot constitute the best chain. To ensure that the blocks supporting the same prediction are in the same fork, every node checks whether it is consistent with the predicted result in the latest received block. If the node is consistent and its block is not in the fork, it publishes a new

block following the received one. The workflow is shown in Figure 3.

All nodes begin to train at t_0 . Bob initially finishes training and publishes its prediction result in the yellow block at t_1 . Later, Alice and Carol publish their findings at t_2 . Alice's and Carol's prediction results are the same but they are different from Bob's. As Alice and Carol publish their blocks simultaneously, they cannot follow each other's block. Also, they are unable to follow Bob's block as their prediction results are different. At t_3 , Alice and Carol publish their blocks $Block_{n+1}$ and $Block_{n+1}^*$ separately again, so their blocks are both the longest. At t_4 , Dave publishes its prediction result the same as Alice's and Carol's. Dave receives $Block_{n+1}$ earlier than $Block_{n+1}^*$, so $Block_{n+2}$ follows $Block_{n+1}$. The fork with $Block_{n+2}$ is the best chain. Every node has its own special off-chain smart contracts and private training data. When a new task is started, nodes begin to train their private samples with their off-chain smart contracts and publish the prediction results into blocks. Blocks with the same prediction results are connected to the same fork, while different prediction results are stored in different forks. The

TABLE 2: Computation parameters.

Names of parameters	Description
$task_x$	Task x to be solved
$C_x = \{task = x c_0, c_1, \dots, c_{n-1}\}$	Set of possible classifications of $task_x$; $ C_x $ is the number of classifications
$Nodes = \{node_0, node_1, \dots, node_{m-1}\}$	Set of nodes; $ Nodes $ is the number of nodes
$Nodes_{type}$	Set of nodes of type
$target_x$	Accurate prediction result, $target_x \in C_x$
$Nodes_x^{c_i}$	Fork of nodes. The solution in all the blocks in the fork is c_i
$p_x^{node_i} target_x$	Accuracy of $node_i$ for $task_x$.
$P_x target_x$	Accuracy of RapidTrainChain for $task_x$.

curves in Figure 4 with $|C_x| = 4$, $p_x^{node_i} target_x$ from 0.004 to 1, and $|Nodes|$ from 1 to 30. When $p_x^{node_i} target_x = 0.25$, no matter how large $|Nodes|$ is, the curve does not go up or down. When $p_x^{node_i} target_x > 0.25$, with plenty of nodes working together, the curves ascend to 1. When $p_x^{node_i} target_x < 0.25$, with adequate nodes, the curves go down to 0. If enough nodes work together, the nodes are more likely to choose the correct classification after the samples are trained (i.e., $p_x^{node_i} target_x > 1/|C_x|$). Equation (1) shows the trend of $P_x target_x$.

Suppose that the accuracy of a node named $node_j$ is $p_x^{node_j} target_x$; the total accuracy of RapidTrainChain ($P_x target_x$) can be characterized as follows:

- (1) If $node_j$ properly trains samples and computes more accurately than $node_i$, that is, $p_x^{node_j} target_x > p_x^{node_i} target_x > 1/|C_x|$, while the number of nodes remains constant, the accuracy $P_x target_x$ is improved.
- (2) If $node_j$ properly trains samples but computes less accurately than $node_i$, that is, $p_x^{node_j} target_x > p_x^{node_i} target_x > 1/|C_x|$, given enough nodes computing at least as accurately as $node_j$, the accuracy $P_x target_x$ is also high.
- (3) If $node_j$ skips training and directly makes predictions, the classifications are randomly selected; that is, $p_x^{node_j} target_x = 1/|C_x|$. The situation is discussed in Section 4.2.
- (4) If $node_j$ improperly trains samples, it is more likely to pick up wrong solutions; that is, $p_x^{node_j} target_x < 1/|C_x|$. Such a situation is discussed in Section 4.3 and Section 4.4.

$$\begin{aligned}
 \lim_{|Nodes| \rightarrow \infty} P_x target_x &= 1, \text{ s.t. } p_x^{node_i} target_x > \frac{1}{|C_x|}, \\
 \lim_{|Nodes| \rightarrow \infty} P_x target_x &= \frac{1}{|C_x|}, \text{ s.t. } p_x^{node_i} target_x = \frac{1}{|C_x|}, \\
 \lim_{|Nodes| \rightarrow \infty} P_x target_x &= 0, \text{ s.t. } p_x^{node_i} target_x < \frac{1}{|C_x|}.
 \end{aligned} \tag{1}$$

4.2. *Robustness with Lazy Nodes.* Some nodes may skip training and make predictions directly. The prediction results are randomly picked from the set of all solutions (C_x), and these nodes are called lazy nodes. Nodes that properly train samples are called honest nodes. The accuracies for these types of nodes are as follows:

- (1) For lazy nodes, $p_x^{lazy\_node} target_x = 1/|C_x|$, and,
- (2) For honest nodes, $p_x^{honest\_node} target_x > 1/|C_x|$,

where $lazy\_node \in Nodes_{lazy}$ and $honest\_node \in Nodes_{honest}$. Therefore, the following equations can be used to calculate the expected fork lengths for the different solutions:

$$|Node s_{honest}| \times p_x^{honest\_node} target_x + |Node s_{lazy}| \times \frac{1}{|C_x|}, \tag{2}$$

$$\frac{|Node s_{honest}| \times (1 - p_x^{honest\_node} target_x)}{|C_x| - 1} + |Node s_{lazy}| \times \frac{1}{|C_x|}. \tag{3}$$

Equation (2) is the expected length of a fork with $target_x$, that is, $|Node s_x^{target_x}|$. Equation (3) is the expected length of other forks. The distance between the tips of forks is equal to equation (2) minus equation (3); that is, $|Node s_{honest}| \times (p_x^{honest\_node} target_x \times |C_x| - 1) / |C_x| - 1$. As long as $p_x^{honest\_node} target_x > 1/|C_x|$ and $|C_x| > 1$, the greater the number of honest nodes is, the more likely the fork with $target_x$ would be the longest. The number of lazy nodes does not affect accuracy.

4.3. *Solutions Competitions.* Suppose that there is one solution that is easier to calculate than all other solutions except $target_x$, that is, $\exists c_k \in C_x - \{target_x, c_k\}$, $p_x^{honest\_node} c_k \geq p_x^{honest\_node} c_l$. The race between the fork with c_k ($Node s_x^{c_k}$) and that with $target_x$ ($Node s_x^{target_x}$) can be characterized as a Binomial Random Walk. The probability (pro_z) indicating the length of forks with c_k catching up that of $target_x$ from z blocks behind pro_z is given in the following equation:

Result: Accuracy of RapidTrainChain, P

Input:

Number of nodes, $|Nodes|$;

Accuracy of a node, p ;

Number of classifications, $|C_x|$;

Function:

Decompose (nodes_number_range_of_a_classification, remaining_nodes, remaining_classifications)

if nodes_number_range_of_a_classification > 1 **then**

foreach $x \in$ nodes_number_range_of_a_classification **do**

      tmpcnt += $C_{remaining\_nodes}^x$

    Decompose ($[remaining\_nodes - x / remaining\_classifications - 1, x]$, remaining_nodes - x , remaining_classifications - 1);

      map< x > ++;

end

else

    1;

end;return

cnt = tmp = 1;

foreach iterator $\in$ map **do**

  tmp = tmp $\times C_{|C_x|-cnt}^{iterator- > second}$;

  cnt = cnt + iterator- > second;

end

return

Decompose ($[|Nodes|/C_x, |Nodes|]$, $|Nodes|$, $|C_x|$) $\times$ cnt;

ALGORITHM 1: RapidTrainChain's accuracy.

FIGURE 4: Accuracy of RapidTrainChain changes with $|Nodes|$ and $p_x^{node_i target_x}$.

$$pro_z = \left\{ \begin{array}{l} 1, p_x^{honest\_node target_x} \leq p_x^{honest\_node c_k} \\ \frac{p_x^{honest\_node c_k}}{p_x^{honest\_node target_x}}^z, p_x^{honest\_node target_x} > p_x^{honest\_node c_k} \end{array} \right\}. \quad (4)$$

If $p_x^{honest\_node target_x} > p_x^{honest\_node c_k}$, pro_z drops exponentially as z increases. As the number of accurate blocks

increases, the chances of $p_x^{honest\_node c_k}$ become extremely smaller.

FIGURE 5: Probability of catching up changes along with $p_x^{honest\_node} c_k / p_x^{honest\_node} target_x$ and z .

Consider when RapidTrainChain is sufficiently certain that c_k cannot win the task. $target_x$ is added to a block, and there are at least z more nodes supporting $target_x$ than those of c_k . Assuming that the honest nodes spend almost the same period in making predictions, c_k 's potential progress will be a Poisson distribution with expected value:

$$\lambda = z \times \frac{p_x^{honest\_node} c_k}{p_x^{honest\_node} target_x}. \quad (5)$$

Equation (5) is the same as "Calculations" in Bitcoin Whitepaper [39]. Using Nakamoto's methods, the probability can be expressed by the following formula:

$$1 - \sum_{k=0}^z \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - \frac{p_x^{honest\_node} c_k}{p_x^{honest\_node} target_x} \right)^{z-k}. \quad (6)$$

According to equation (6), probability changes with $p_x^{honest\_node} c_k / p_x^{honest\_node} target_x$ and z . The curves show that probability drops exponentially with z in Figure 5. The closer $p_x^{honest\_node} c_k$ and $p_x^{honest\_node} target_x$ are to each other, the more blocks are needed to ensure that the fork with c_k ($Node s_{c_k}^{c_k}$) does not catch up with the fork with $target_x$ ($Node s_{target_x}^{target_x}$). If the probability of getting solution c_k is half that of $target_x$, that is, $p_x^{honest\_node} c_k / p_x^{honest\_node} target_x = 0.5$, the probability of catching up is 0.010943 when $z = 21$. In this case, when one fork is 21 blocks longer than another, the solution in the longest fork is very likely to be $target_x$.

4.4. Competition with Sybil Attackers. In distributed systems, there are many kinds of attackers, such as Sybil attackers. Sybil attackers propose a certain wrong prediction (c_w) continuously; that is, $p_x^{Sybil\_node} c_w = 1$. Suppose that the number of Sybil attackers is $|Nodes_{Sybil}|$.

$$\begin{aligned} & |Node s_{honest}| \times p_x^{honest\_node} target_x + |Node s_{Sybil}| \times 0 \\ & + |Node s_{lazy}| \times \frac{1}{|C_x|}, \end{aligned} \quad (7)$$

$$\begin{aligned} & |Node s_{honest}| \times p_x^{honest\_node} c_w + |Node s_{Sybil}| \times 1 \\ & + |Node s_{lazy}| \times \frac{1}{|C_x|}. \end{aligned} \quad (8)$$

Equation (7) is the expected length of the fork with $target_x$, while equation (8) is the expected length of the fork with c_w . Equation (7) minus equation (8) is the distance of tips between the two forks, that is, $|Node s_{honest}| \times (p_x^{honest\_node} target_x - p_x^{honest\_node} c_w) - |Node s_{Sybil}|$.

There needs to be more than $|Node s_{Sybil}| / (p_x^{honest\_node} target_x - p_x^{honest\_node} c_w)$ honest nodes to ensure RapidTrainChain's accuracy. Suppose $p_x^{honest\_node} c_w / p_x^{honest\_node} target_x \leq 0.5$. After a certain period, when most honest nodes finish making predictions, if one fork is 21 blocks longer than the others (as described in Section 4.3), the solution in this fork is very likely to win.

TABLE 3: Hardware configuration.

Items	Memory (GB)	CPU (pcs)	CPU (MHz)
Nodes in RapidTrainChain	2	1	3000
The powerful computer	20	4	3000

FIGURE 6: Performances of nodes in RapidTrainChain powerful computer.

4.5. Task Duration. RapidTrainChain works on one task until one fork is unambiguously the longest, such that the length of the fork satisfies the condition:

$$\exists c_k \in C_x, \forall c_l \in C_x - \{c_k\} + \{null\}, |Node s_x^{c_k}| \geq |Node s_x^{c_l}|, \quad (9)$$

where c_k is a solution belonging to C_x . If a fork with c_k is the longest (i.e., no fork can possibly be longer), RapidTrainChain stops the task. However, there are infinite nodes in the public network. If forks are very unlikely to catch up with one fork, RapidTrainChain should stop working on this task. As discussed in Section 4.3 and Section 4.4, if there are more than $|Node s_{sybil}|/p_x^{honest\_node} target_x - p_x^{honest\_node} c_w$ honest nodes after a certain period and supposing $p_x^{honest\_node} c_w / p_x^{honest\_node} target_x \leq 0.5$, RapidTrainChain stops the working task when one fork is 21 blocks longer than the others.

5. Implementation and Evaluation

We enhanced the mining function in Bitcoin source codes (generatetoaddress in mining.cpp), validation functions (CheckProofOfWork in pow.cpp), and other Bitcoin-related functions to invoke off-chain smart contracts. Off-chain smart contracts can train samples and make predictions and store these prediction results into blocks. We set up a powerful computer and 10 less powerful nodes in RapidTrainChain, as shown in Table 3. Each node in RapidTrainChain possesses 5,000 training samples, while a powerful computer holds 50,000 training samples. We monitored the performance of nodes in RapidTrainChain and compared it with the computer's performance. To make the performance data comparable, nodes in RapidTrainChain and the powerful computer adopted the same algorithm in training their samples and making predictions repeatedly.

FIGURE 7: Accuracy of RapidTrainChain and average accuracy.

The summary of node performances is presented in Figure 6. The main highlights are as follows:

- (1) From Figure 6(a), the RapidTrainChain and powerful computer start simultaneously at time = 20 seconds. RapidTrainChain stops at time = 1240 seconds, while the powerful computer stops at time = 5420 seconds. The powerful computer takes more than *four* times as long as RapidTrainChain.
- (2) Figures 6(a) and 6(c) present the CPU and memory performance charts, showing that a single node in RapidTrainChain consumes much less CPU computing and memory resources. This is because the node trains fewer samples, and the calculation burden is shared.
- (3) Figure 6(d) shows the data storage performance chart. A single node in RapidTrainChain requires a little more storage than the powerful computer because it needs to store messages from other nodes.
- (4) As shown in the network performance chart in Figure 6(b), the powerful computer consumes little bandwidth, while RapidTrainChain uses some bandwidth for block transfer.

We used a convolution neural network (CNN) algorithm to predict on MNIST. Because the existing CNN algorithm performs very well in digital recognition, the accuracy of prediction would be high even if only 5,000 training samples were used. Thus, we divided the 50,000 samples into 20 parts (i.e., each containing 2,500 samples) and assigned the parts to 40 nodes. As shown in Figure 7, the accuracy of RapidTrainChain is much higher than the average accuracy of nodes and increases as the number of nodes increases.

6. Conclusion

In this paper, we presented a novel supervised learning approach based on Bitcoin (RapidTrainChain). In the proposed algorithm, we introduced a rapid, compatible consensus mechanism (PoP), which helps RapidTrainChain make an accurate prediction. We formalized the cooperation

mechanism to reduce the workload of a single node while maintaining overall accuracy, improving overall efficiency, and ensuring overall privacy. In the experiment, we showed the influence of honest nodes, lazy nodes, and Sybil attackers on the overall accuracy. We implemented our proposed algorithm and evaluated its efficiency. Our results suggest that RapidTrainChain does not depend heavily on the computational power of single nodes and is friendly to heterogeneous devices. We found that the more nodes are used in RapidTrainChain, the more secure the system becomes. Moreover, our results showed that the number of nodes does not affect the processing time for a given task and that RapidTrainChain can be applied in the public network.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

Acknowledgments

This research was funded in part by the National Natural Science Foundation of China (Grant no. 61772352), National Key Research and Development Project (Grant nos. 2020YFB1711800 and 2020YFB1707900), the Science and Technology Project of Sichuan Province (Grant nos. 2019YFG0400, 2021YFG0152, 2020YFG0479, 2020YFG0322, and 2020GFW035), the R&D Project of Chengdu City (Grant no. 2019-YF05-01790-GX), National Natural Science Foundation of China (Grant no. 61871422), and Science and Technology Program of Sichuan Province (Grant no. 2020YFH0071).

References

- [1] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 1975–1989, 2019.

- [2] S. Luo, "Blockchain based task offloading in drone-aided mobile edge computing," *IEEE Network*, vol. 14, no. 8, pp. 1–6, 2020.
- [3] G. Qiao, S. Leng, H. Chai, A. Asadi, and Y. Zhang, "Blockchain empowered resource trading in mobile edge computing and networks," in *Proceedings of the ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019.
- [4] Y. He, Y. Wang, C. Qiu, Q. Lin, J. Li, and Z. Ming, "Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2226–2237, 2021.
- [5] Y. Hao, Y. Miao, L. Hu, M. S. Hossain, G. Muhammad, and S. U. Amin, "Smart-edge-CoCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT," *IEEE Network*, vol. 33, no. 2, pp. 58–64, 2019.
- [6] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.
- [7] R. Zhao, X. Wang, J. Xia, and L. Fan, "Deep reinforcement learning based mobile edge computing for intelligent Internet of Things," *Physical Communication*, vol. 43, p. 101184, 2020.
- [8] S. Rathore, Y. Pan, and J. H. Park, "BlockDeepNet: a blockchain-based secure deep learning for IoT network," *Open Access Journal*, vol. 11, pp. 1–15, 2019.
- [9] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate edge and cloud computing with distributed deep learning for smart city internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8099–8110, 2020.
- [10] Z. Shae and J. Tsai, "Transform blockchain into distributed parallel computing architecture for precision medicine," in *Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1290–1299, Vienna, Austria, July 2018.
- [11] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: a blockchain for auditable federated learning with trust and incentive," in *Proceedings of the 2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 151–159, QingDao, China, August 2019.
- [12] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2020.
- [13] A. B. Kurtulmus and K. Daniel, "Trustless Machine Learning Contracts; Evaluating and Exchanging Machine Learning Models on the Ethereum Blockchain," 2018, <http://arxiv.org/abs/1802.10185>.
- [14] J. Kim and S.-M. Moon, "Blockchain-based edge computing for deep neural network applications," in *Proceedings of the Workshop on INTElligent Embedded Systems Architectures and Applications - INTESA '18*, pp. 53–55, Turin, Italy, October 2018.
- [15] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: a comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.
- [16] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, Harvard University, Cambridge, MA, USA, 2014.
- [17] P. Das, "Fastkitten: practical smart contracts on bitcoin," in *Proceedings of the 28th USENIX Security Symposium*, pp. 801–818, Santa Clara, CA, USA, August 2019.
- [18] C. Li, B. Palanisamy, and R. Xu, "Scalable and privacy-preserving design of on/off-chain smart contracts," in *Proceeding of the 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, pp. 7–12, Macao, April 2019.
- [19] S. Popov, *The Coorcidice*, IOTA Found, Berlin, Germany, 2020.
- [20] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.
- [21] L. Lamport, "Fast Paxos," *Distributed Computing*, vol. 19, no. 2, pp. 79–103, 2006.
- [22] D. Ongaro, *Consensus: Bridging Theory and Practice*, Stanford University, Stanford, CA, USA, 2014.
- [23] B. Cao, Y. Li, L. Zhang et al., "When internet of things meets blockchain: challenges in distributed consensus," *IEEE Network*, vol. 33, no. 6, pp. 133–139, 2019.
- [24] S. Popov, "IOTA whitepaper v1.4.3," *New Yorker*, vol. 81, no. 8, pp. 1–28, 2018.
- [25] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceeding of the International Conference on Financial Cryptography and Data Security*, pp. 515–532, Sliema, Malta, April 2017.
- [26] BTC_diff BTC.com. <https://btc.com/stats/diff>.
- [27] Digiconomist. Bitcoin energy consumption index. <https://digiconomist.net/bitcoin-energy-consumption>.
- [28] N. Szabo, *Forming and Securing Relationships on Public Networks*, vol. 2, 1997.
- [29] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: architecture, applications, and future trends," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium*, Suzhou, China, July 2018.
- [30] P. Wackerow. Introduction to smart contracts. <https://ethereum.org/en/developers/docs/smart-contracts/>.
- [31] Chaincode Tutorials. 2018. <https://hyperledger-fabric.readthedocs.io/en/release-1.3/chaincode.html>.
- [32] E. Drasutis, *An Introduction to Iota Smart Contracts*, <https://blog.iota.org/an-introduction-to-iota-smart-contracts-16ea6f247936>, 2020.
- [33] R. Cheng, F. Zhang, J. Kos et al., "Ekiden: a platform for confidentiality-preserving, trustworthy, and performant smart contracts," in *Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroSecP)*, pp. 185–200, Stockholm, June 2019.
- [34] J. Eberhardt and S. Tai, "ZoKrates - Scalable Privacy-Preserving Off-Chain Computations," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber*, Halifax, NS, Canada, August 2018.
- [35] R. Kumaresan and I. Bentov, "How to use bitcoin to incentivize correct computations," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale, AZ, USA, November 2014.
- [36] R. Kumaresan, V. Vaikuntanathan, and P. N. Vasudevan, "Improvements TO secure computation with penalties," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, vol. 24–28, pp. 406–417, 2016.
- [37] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: scalable, private smart contracts," in *Proceeding of the 27th USENIX Security Symposium*, pp. 1353–1370, Berkeley, CA, USA, May 2018.
- [38] M. Leinweber, M. Grundmann, L. Schönborn, and H. Hartenstein, "TEE-based distributed watchtowers for fraud protection in the lightning network," *Lecture Notes in Computer Science*, LNCS, vol. 11737, , pp. 177–194, 2019.
- [39] S. Nakamoto, *Bitcoin: A Peer-To-Peer Electronic Cash System*, Springer, Berlin, Germany, 2008, <https://nakamotoinstitute.org/bitcoin/>.