mathematics

MDPI

Article

A Secure and Fair Federated Learning Framework Based on Consensus Incentive Mechanism

Feng Zhu, Feng Hu *, Yanchao Zhao, Bing Chen and Xiaoyang Tan

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; fengzhu@nuaa.edu.cn (F.Z.); ychao@nuaa.edu.cn (Y.Z.); cb_china@nuaa.edu.cn (B.C.); x.tan@nuaa.edu.cn (X.T.)

* Correspondence: huf@nuaa.edu.cn

Abstract: Federated learning facilitates collaborative computation among multiple participants while safeguarding user privacy. However, current federated learning algorithms operate under the assumption that all participants are trustworthy and their systems are secure. Nonetheless, real-world scenarios present several challenges: (1) Malicious clients disrupt federated learning through model poisoning and data poisoning attacks. Although some research has proposed secure aggregation methods to address this issue, many methods have inherent limitations. (2) Clients may refuse or passively participate in the training process due to considerations of self-interest, and may even interfere with the training process due to competitive relationships. To overcome these obstacles, we have devised a reliable federated framework aimed at ensuring secure computing throughout the entirety of federated task processes. Initially, we propose a method for detecting malicious models to safeguard the integrity of model aggregation. Furthermore, we have proposed a fair contribution assessment method and awarded the right to write blocks to the creator of the optimal model, ensuring the active participation of participants in both local training and model aggregation. Finally, we establish a computational framework grounded in blockchain and smart contracts to uphold the integrity and fairness of federated tasks. To assess the efficacy of our framework, we conduct simulations involving various types of client attacks and contribution assessment scenarios using multiple open-source datasets. Results from these experiments demonstrate that our framework effectively ensures the credibility of federated tasks while achieving impartial evaluation of client contributions.

Keywords: federated learning; blockchain; malicious model detection; contribution evaluation

MSC: 68M25

Citation: Zhu, F.; Hu, F.; Zhao, Y.; Chen, B.; Tan, X. A Secure and Fair Federated Learning Framework Based on Consensus Incentive Mechanism. *Mathematics* **2024**, *12*, 3068. https://doi.org/10.3390/math12193068

Academic Editor: Radi Romansky

Received: 21 August 2024
Revised: 20 September 2024
Accepted: 25 September 2024
Published: 30 September 2024

1. Introduction

With the continuous advancement of technology, the issues of data security and privacy in machine learning are becoming increasingly prominent. As more and more data is collected and utilized for training models, concerns about the protection and misuse of personal information have come to the forefront. The growing reliance on machine learning algorithms to make critical decisions in various domains further underscores the need for robust safeguards to ensure the confidentiality and integrity of sensitive data. As such, addressing the challenges posed by data security and privacy has become a pressing priority in the field of machine learning. Scholars have proposed a series of privacy protection schemes to address data security risks in machine learning. These primarily include federated learning (FL), secure multiparty computation (SMPC), homomorphic encryption (HE), and differential privacy (DP). Among these techniques, federated learning stands out by utilizing distributed offline training methods, ensuring privacy while effectively saving communication and computational resources. It is particularly suitable for scenarios with large data volumes, widely distributed data sources, and high information sensitivity.

The concept of federated learning initially appeared in literature and has evolved into three fundamental frameworks: (1) vertical federated learning, suitable for scenarios where participating parties have significant data overlap; (2) horizontal federated learning, addresses situations where nodes share similar data features; and (3) federated transfer learning, ideal when participant sample spaces partially overlap but feature distributions differ. In environments with uneven computational power, these three approaches are often implemented using a client/server (C/S) model. Leveraging high-throughput, high-performance devices as central nodes, the C/S model offers advantages such as higher training efficiency, equitable benefit distribution, and enhanced local data security compared with distributed learning. However, in untrusted environments, the C/S-based federated learning method is vulnerable to threats like identity forgery, data tampering, and denial-of-service (DoS) attacks.

To address these trust issues, researchers have proposed a blockchain-based federated learning scheme, instantiating abstract trusted service nodes as distributed consensus incentives [1]. The nodes in the framework can play the roles of data provider and block miner at the same time or separately. All participants complete the training of the submodel on the local data set, and then upload it to the randomly selected or voted miners. The miner is responsible for verifying and integrating all local models, and then generating new blocks according to the PoW or PoS consensus mechanism. These blocks are responsible for recording the mining rewards of miners and the page rewards of data providers, and storing the updated parameters of the model. Then, the participants download the aggregated model again and repeat the above process until they get a satisfactory global machine learning model. It can be seen that the essence of this machine learning method lies in indirect data sharing and effective cooperation incentives, so the reliability of the consensus algorithm and the fairness of the reward mechanism will directly affect the performance of the whole system.

However, there are still challenges in the existing blockchain-based FL frameworks:

- Limitations of blockchain-based federated learning solutions focused on a single aspect: Current blockchain-based federated learning methods primarily concentrate on addressing specific aspects of trustworthy federated learning. Some approaches focus on mitigating the impact of malicious attacks through consensus mechanisms, while others emphasize selecting high-quality clients based on reputation scores. However, these approaches fail to provide a comprehensive consideration of potential threats throughout the entire federated learning process.
- Limitations in identifying malicious clients: Although blockchain technology can record and verify the behavior of nodes, relying solely on its transparency and immutability for identifying malicious clients has limitations. For instance, malicious clients may evade detection by submitting fabricated model updates, and blockchain alone is insufficient to fully prevent such stealthy attacks.
- Lack of fair consensus and incentive mechanisms: The reward distribution process is vulnerable to free-rider behavior from inactive or low-quality participants, and existing methods are unable to fully eliminate this unfairness. Furthermore, malicious clients may intentionally exaggerate or falsely report their contributions, leading to inaccurate assessments of their performance.

Addressing these challenges will be crucial to the continued development and adoption of blockchain-enhanced federated learning systems. Based on the above challenges, this paper proposes a trusted federated learning framework based on blockchain, which supports malicious client detection and has a fair blockchain consensus and incentive mechanism.

The contributions of this paper include:

- **Blockchain-based framework for trustworthy federated learning**: We propose a federated learning framework that ensures both trustworthiness and fairness. This framework leverages blockchain technology and smart contract mechanisms to secure the entire collaborative process, offering a comprehensive federated learning solution

that addresses client management, malicious client detection, data and model security, and fairness in incentive mechanisms.

- **Generalized secure global aggregation**: To ensure secure global model aggregation in federated learning, we introduce a feature fusion-based approach for detecting malicious clients. This method enhances the reliability of global model aggregation while mitigating the impact of potential malicious participants.
- **Efficient and fair blockchain consensus incentive mechanism**: We propose a fair blockchain consensus incentive mechanism. By employing robust model watermarking techniques and parameter distance algorithms, the mechanism ensures equitable distribution of rewards based on actual contributions, addressing the limitations of traditional incentive schemes that rely solely on reputation or self-reporting. During the model aggregation phase, the creator of the best model is granted the right to write to the blockchain, promoting active participation in both local training and model aggregation.

2. Related Work

2.1. Trustworthy Federated Learning

There are two main types of attacks in federated learning: model poisoning attacks, also known as backdoor attacks, and data poisoning attacks. Several studies have explored secure aggregation methods for client-uploaded models. For example, Li et al. introduced a spectral anomaly detection technique [2], training a server-side detection model to identify malicious client models. Chen et al. proposed a reinforcement learning-based method for selecting trustworthy clients (FedDRL) [3]. Zhang et al. developed FLDetector [4], a model comparison-based approach that flags a client as malicious if its uploaded model consistently differs from the predicted model. Another approach is BaFFLe [5], introduced by Andreina, which validates the newly aggregated model on the client's dataset to identify potential poisoning.

Additionally, there have been efforts to achieve secure model aggregation. Blanchard et al. proposed the MutilKrum method [6], which calculates the Euclidean distances between models to identify malicious ones and achieve secure aggregation. Yin et al. presented the TrimmedMean algorithm [7], which trims the parameters of models to eliminate abnormal values during model fusion.

While these methods address the detection of malicious clients to some extent, they still face certain challenges. Firstly, some algorithms have strict usage conditions, requiring prior knowledge of the number of malicious clients. Secondly, certain methods only function reliably when the number of malicious clients is small. Our proposed blockchain framework utilizes model feature fusion to detect malicious clients. Compared with traditional methods, it demonstrates higher reliability and adaptability in identifying malicious clients, particularly when confronted with various types of malicious behaviors.

2.2. Trustworthy Federated Learning Base on Blockchain

Blockchain has increasingly found utility in addressing malicious client attacks within federated learning, owing to its advantageous characteristics such as decentralization, high fault tolerance, and resistance to tampering. Trustworthy research in federated learning leveraging blockchain technology encompasses the detection of malicious clients and the management of client reputation values. Some studies focus on enhancing the trustworthiness of federated learning through blockchain integration. For instance, Mohamed et al. [8] proposed the Fed-Trust framework, which employs semi-supervised techniques to detect attacks in Internet of Things (IoT) environments. Additionally, Li et al. presented the BFLC decentralized framework [9], which incorporates a committee consensus mechanism aimed at mitigating malicious attacks. Furthermore, Muhammad et al. developed the Biscotti peer-to-peer information interaction framework [10], designed to remain dependable in scenarios involving poisoning attacks. Ma et al. introduced a decentralized Federated Learning (FL) framework with blockchain support [11], which addresses malicious client

attacks and includes a detection algorithm to prevent fraudulent behavior by lazy clients seeking to manipulate rewards.

Certain scholars leverage blockchain technology to record clients' reputation values. For example, Kang et al. introduced a composite calculation approach for reputation valuation [12], which utilizes blockchain for the management of individual client reputation scores. Zhang et al. proposed a Proof-of-Reputation (PoR) consensus algorithm [13] aimed at assessing the quality of client models and determining associated reputation values. Additionally, Qi et al. [14] devised a methodology for client reputation assessment, facilitating the selection of high-quality clients for participation in model aggregation processes.

Although existing algorithms tackle client selection and detection of malicious clients, many of these approaches focus solely on one aspect of trustworthy federated learning. Hence, there's a pressing need to develop a comprehensive federated computing framework that offers a broader spectrum of reliable computing solutions. Our proposed blockchain framework not only covers malicious client detection but also integrates client management and data and model security. By combining blockchain and smart contracts, we offer a comprehensive solution that strengthens defense against various attacks. Unlike Fed-Trust and BFLC, our framework ensures trustworthy client behavior while introducing model watermarking techniques and parameter distance algorithms, providing a more efficient and fair consensus incentive mechanism. These improvements give our framework a distinct advantage in handling malicious clients and ensuring fairness in reward distribution.

2.3. Contribution Evaluation and Incentive Mechanism

Current methods for assessing the contribution of federated learning include the self-reporting method, the Shapley value method, and the client reputation method.In the self-reporting method, participating clients provide accurate reports on their data and computational resources, which are then analyzed by the task initiator to assess client contribution [15]. The Shapley value method computes client contribution through three steps: firstly, it evaluates the value of a client participating in the global model fusion; secondly, it calculates the value of the client not participating; and finally, it determines the marginal contribution of the client by comparing these two scenarios. Moreover, there is ongoing research aimed at enhancing the computational efficiency of the Shapley value method [16–18]. Regarding the reputation method, some studies utilize client reputation to gauge their contribution [12,13,19]. This method primarily involves assessing each client's behavior in federated tasks to derive their reputation value, ultimately leading to a fair allocation of rewards based on each client's reputation.

Once the contribution value of each client has been evaluated, the next critical task is to distribute rewards to each client. Gao et al. [20] devised a system for client reward distribution utilizing both client reputation and a smart contract mechanism. Yu et al. [21] introduced a method to assess the client's revenue, addressing the budget constraints related to total rewards. Meanwhile, Chen et al. [22] developed a game-theory-based approach to facilitate fair and reasonable reward distribution among clients.

Regrettably, the presence of lazy or low-quality data clients in model fusion can lead to instances of free-riding behavior, posing challenges for some Shapley Value methods to offer a fair assessment of client contribution under such circumstances. It's worth noting that blockchain technology, renowned for its tamper-proof and transparent nature, holds the potential to bolster the security and trustworthiness of federated learning. By integrating federated learning with blockchain, critical information such as models, reputation values, and rewards involved in federated tasks can be made auditable, with any unauthorized modifications detected and halted during the consensus process. Moreover, the utilization of smart contracts can effectively govern the execution of federation tasks. We propose a novel blockchain consensus incentive mechanism that achieves fair distribution of node contributions through model watermarking techniques and parameter distance algorithms. Compared with traditional methods, our mechanism not only overcomes the limitations of

relying solely on reputation or self-reporting but also incentivizes active participation in local training and model aggregation by linking the creation of the best model to the right to write to the blockchain. This innovative incentive mechanism improves the fairness and accuracy of reward distribution, thereby enhancing the overall efficiency and effectiveness of the federated learning system.

3. Problem Formulation

3.1. Architectural Design

In this section, we first describe the issues associated with conventional federated learning algorithms. We assume that n clients are participating in the federated task, and we define the ith client as C_i and the set of n clients as $\{C_1, C_2, ..., C_n\}$. The parameters of the models trained by client C_i are defined as θ_i, and the set of parameters for the models across all n clients is represented as $\{\theta_1, \theta_2, ..., \theta_n\}$.

In each communication round, the server selects k clients from the pool of n clients, the selected clients upload the local model to the central server, and then the server aggregates the models of each client to aggregate a global model, which is defined as θ_{global}. The primary goal of the server is to iteratively refine and obtain the best global model. The process is shown as (1).

$$\min_{\theta} G\left(\theta_{\text{global}}\right) = \sum_{i=1}^{k} w_i f(\theta_i), \sum_{i=1}^{k} w_i = 1, w_i \geq 0 \tag{1}$$

The w_i is the weight of client i. traditional federated learning algorithm assumes that the models uploaded by the selected k clients are trustworthy. However, these algorithms fail to consider the involvement of malicious clients in the model aggregation process. Figure 1 illustrates the process of malicious clients launching attacks on the server through data poisoning and model poisoning. If the server-side employs the models of malicious clients for global model fusion, those malicious models' parameters could significantly impact the aggregated model's parameters, which finally impacts the accuracy and convergence of the global model—ultimately causing the entire federated learning task to fail. Consequently, our foremost concern should be identifying and excluding malicious models from participating in the global model fusion process.

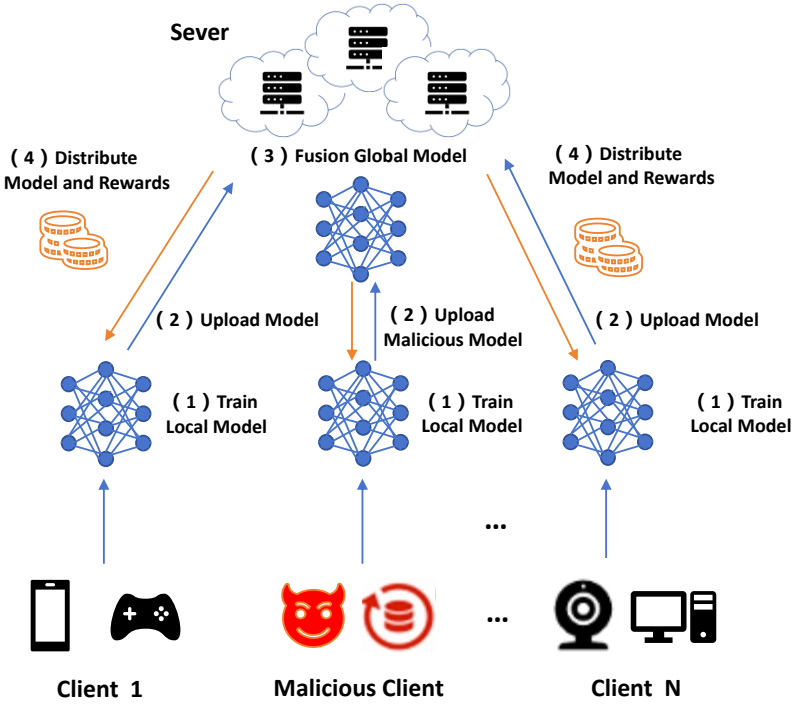


Figure 1. Attackflow of malicious clients in federated learning.

After identifying potentially malicious clients, we must objectively assess the contribution of each client. The resulting evaluation will determine the distribution of rewards. We assumed the federated task conducts m communication rounds, and we will evaluate the clients' contribution value in each round. We define this process as (2). Where *FairReward* denotes a fair reward distribution algorithm, t denotes the tth communication round, and $t \in [1, m]$.

$$\left\{C_1^t, C_2^t, \ldots, C_k^t\right\} \leftarrow FairReward\left(\left\{\theta_1^t, \theta_2^t, \ldots, \theta_k^t\right\}\right) \tag{2}$$

Upon addressing these two concerns, it becomes imperative to integrate blockchain technology for comprehensive authentication of federated learning tasks, tracking client reputation, and recording reward-related information. Consequently, relevant smart contract mechanisms need to be devised to guarantee the security and credibility of federated learning tasks. In summary, the following three challenges must be addressed:

(1) How to identify malicious clients and realize the aggregation of a secure and trustworthy global model.

(2) How to evaluate clients' contributions and realize fair reward distribution.

(3) How to ensure the security and trustworthiness of the whole process of federated task.

3.2. Threat Model

Blockchain can address the centralization issue in federated learning, while federated learning achieves data privacy protection on the blockchain. In order to ensure the reliable operation of the proposed solution in this paper, formal definitions of its performance and security are established first. Subsequent sections will elaborate and provide arguments around these definitions.

Definition 1. *Byzantineattack environment.*

The Byzantine environment refers to a scenario in distributed systems like federated learning, where some participants (nodes, agents, or devices) may behave maliciously or unpredictably.These participants may send incorrect or misleading data with the intent to disrupt the system's functionality, yet the system must remain resilient and achieve its objectives despite these adversarial agents. The term "Byzantine" comes from the Byzantine Generals' Problem, which describes a situation where communication failures and deceit among participants can prevent the system from functioning correctly. In the context of machine learning, particularly in federated learning or distributed training, a Byzantine assault refers to scenarios where some of the nodes participating in training intentionally provide corrupted or misleading updates (such as gradients). The challenge is for the system to learn despite these adversarial participants.

In a federated learning process, local models trained on different nodes are aggregated to form a global model. Gradient models help evaluate the contribution of each node's local data to the global model, especially in a decentralized environment with non-independent and identically distributed (non-IID) data. The contribution of a node's gradient model to the global model is calculated using the following formula:

$$F(N_k, \alpha) = \alpha \cdot f_l(x) + (1 - \alpha) \cdot f_k(x). \tag{3}$$

where N_k is the gradient model generated at node k, $f_l(x)$ and $f_k(x)$ represent the contribution of the local and global models, respectively. α is a weighting factor dependent on the similarity between datasets. This function ensures that the contribution from each node is weighted based on its performance, allowing the global model to be aggregated fairly from various local models. This framework is designed to handle the challenges of non-IID data distribution and poisoning attacks, ensuring robustness in federated learning.

Assume that the set of nodes participating in the j th round of consensus is G_j. For $\forall k \in G_j$, k has the computing power of probabilistic polynomial-time (PPT), and has a digital signature verification key set $PK = \{pk_i \in i | i \in G_j\}$ for verifying the authenticity of data transmitted by other nodes. Node k uploaded gradient model W_k possesses the following characteristics:

(1) The gradient model W_k itself needs to satisfy the watermark verification $V_w(W_k, (B_k, \theta_k)) = True$;

(2) The aggregation model N_d obtained by using W_k in the fusion process of aggregation node d can pass the watermark verification, that is $V_w(N_d, (B_k, \theta_k)) = True$;

(3) If the aggregation node d does not utilize W_k during the fusion process, the resulting global model N_d may fail to satisfy the watermark verification, that is $V_w(N_d, (B_k, \theta_k)) = False$.

In the presence of a Byzantine environment, there exists a subset of nodes denoted as A. For any $a \in A$, it possesses the capability to forge digital signatures of other nodes in this set and manipulate model watermarks. It may initiate attacks such as selective communication, delayed communication, and communication noise. However, for honest nodes $c \in G_j - A$, they will operate normally according to the PFconsensus protocol, remain online throughout the joint training, and do not experience delayed communication or communication noise.

Definition 2. *The gradient model to be fused.*

Let $\overline{t^w}$ represent the time required for the normal transmission of a gradient model, the time consuming for training the gradient model on node k is t_k^train, and t_k^avg represents the time spent on model aggregation. Select m gradient models from n participating nodes for fusion, and their execution process is required to meet the following conditions:

$$\frac{\sum_{k=0}^{n} t_k^{train}}{n} = \overline{t^{train}}; \tag{4}$$

$$\max(t_k^{train}) + \overline{t^w} = m\overline{t^{train}}; \tag{5}$$

$$\overline{t^{train}} \geq \overline{t^w} >> \overline{t^{avg}}; \tag{6}$$

$$\sqrt{\frac{\sum_{k=0}^{n}(t_k^{avg} - \overline{t^{avg}})^2}{n}} \cong 0. \tag{7}$$

These four conditions ensure that in a Byzantine environment, at least one honest node correctly executes the consensus, thus preventing the exclusion of all honest nodes from the consensus process due to performance differences or collusion. Specifically, Equation (2) ensures that honest nodes receive at least m correct gradient models during the commitment scoring stage, while Equation (4) guarantees that the aggregated model set inevitably contains a correctly aggregated model.

Definition 3. *Poisoning attack node.*

For Byzantine node $k \in A$ in a decentralized environment, it can publish malicious gradients $W_k{}'$, causing the performance of the aggregation model N' that aggregates $W_k{}'$ to degrade.

4. Proposed Framework

4.1. Architectural Design

In order to solve the three problems in the Section 3 and ensure the credibility and fairness of the entire federated task, we proposed the FedCFB framework by combining

federated learning with blockchain. Different from the traditional centralized federated learning architecture, we designed a rotating center architecture in which the task issuer plays the role of the central server. Figure 2 shows the overall architecture of the FedCFB, consisting of client, blockchain, and federation layers. Among them, the federation layer is responsible for coordinating and scheduling all federation tasks, the blockchain layer is responsible for storing federation task information, client models, rewards, etc, and the client layer mainly performs federation tasks.

To address the challenge of secure collaborative training in a non-trust environment for federated learning and the issue of significant resource wastage and declining node participation arises due to the consensus process, this paper proposes the federated blockchain structure as illustrated in Figure 2. In this structure, the data recorded on the chain primarily includes: (1) Hash of the previous block; (2) Model parameters after aggregation; (3) Local gradient set used in constructing the aggregated model; (4) Property rewards based on evaluation criteria; (5) Optimization objectives for the next training round.

At the beginning of the protocol, participating nodes will obtain the publicly released initial model and training target from the blockchain, and locally train a gradient model containing watermarks. Subsequently, the node will broadcast the gradient model through the gossip protocol, and after receiving enough gradient information, try to obtain the aggregate model through the aggregation algorithm. Finally, the aggregated model will be sent to each node for evaluation, and the optimal model generated by voting and the optimization goals of the next round of protocols will be written into the new block at the same time. Considering the anti-tampering problem of data, in addition to distributed storage, the Hash chain structure and the longest chain principle will be used to ensure the durability of the blockchain. It is worth mentioning that when designing the block data structure, this solution records the gradient model of each node on the block, which can ensure the credibility of the aggregate model.

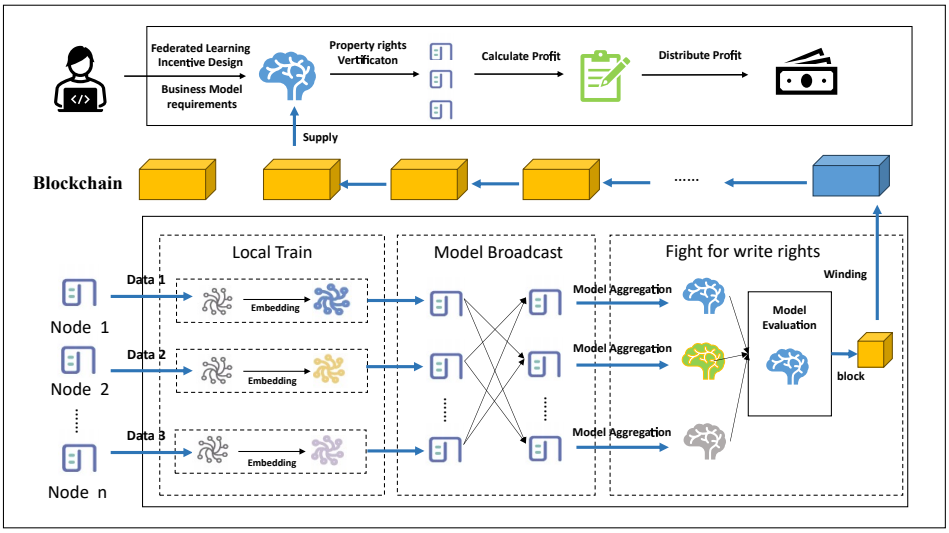


Figure 2. The structure of the blockchain system in this paper.

4.2. Secure Identification Framework

In the FedCFB framework, task publishers generate federated tasks and extend invitations to other clients for participation. Upon receiving the task message, each client assesses its computing and data resources to determine whether it will engage in the task. The task publisher assumes responsibility for the entirety of the federated task and allocates rewards according to each client's contribution. The overall process is as follows. The architecture is shown in Figure 3.

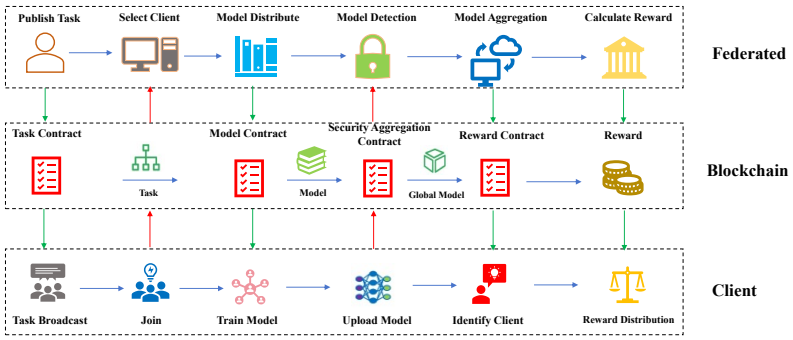


Figure 3. The architecture of FedCFB.

(1) **Task creation stage**: The task publisher crafts the task description, encompassing task details such as information, model specifications, data type, and reward particulars. Subsequently, the task publisher disseminates the task and constructs a federated task block, recording the task information on the blockchain.

(2) **Task initialization stage**: other participants receive the broadcast message and decide whether to participate in the task based on their computing and data resources.

(3) **Client selection stage**: The task publisher reviews the reputation records of each participant on the blockchain and chooses clients based on these records. Subsequently, the task publisher dispatches initialization models to the selected clients.

(4) **Model training stage**: The task publisher will publish the basic structure of the model and initialize the parameters. Subsequently, each client will use local data to train the task publisher's initial model and broadcast it after implanting the model watermark.When the model training is completed, each client calculates the MD5 of the model and stores its signature information in the blockchain.

(5) **Model detection stage**: The task publisher initially acquires the models submitted by each client and retrieves the signature and MD5 information of each model from the blockchain to verify completeness and trustworthiness. Then, it invokes the secure identification algorithm, which is to identify potential issues with each model, subsequently categorizing clients into groups of trusted and malicious clients based on the detection results.

4.3. Fair Federated Consensus Algorithm

4.3.1. Consensus

After the preliminary screening in the secure identification algorithm, we start to implement the PFconsensus federated consensus algorithm [23] (see Algorithm 1). This article uses the federated learning algorithm itself as the consensus, and combines blockchain and watermarking technology to solve the non-IID and poisoning problems in federated learning to a certain extent. It can be summarized into two parts: model performance screening and consensus writing.

(1) **Model screening stage**: Following the identification of models, the task publisher evaluates clients' reputation by analyzing the results of the models and computing new reputation scores for each client. Ultimately, the task publisher stores the updated reputation values in the blockchain.

At this time, members in the clients set will verify the accuracy of the resulting aggregate model, marking the first received model as N_k, and subsequent models as N_a, where a represents the publisher of the model. Compare the performance of N_k and N_a in the order received, If N_a performs better, calculate the commitment $(N_a, vote_k)$. At the same time, it will no longer accept models with performance less than N_a, and let $N_k = N_a$, Otherwise return message (N_k, k).

(2) **Reputation evaluation stage**: After the task publisher identifies the models, it evaluates the clients' reputation by assessing the models' results and calculating the new

reputation for each client. Finally, the task publisher stores the reputation value in the blockchain.

(3) **Model aggregation stage**: Following the identification of models, the task publisher evaluates clients' reputation by analyzing the results of the models and computing new reputation scores for each client. Ultimately, the task publisher stores the updated reputation values in the blockchain.

Algorithm 1 Algorithm of federated consensus mechanism

Input: The jth Block $Block_j$; The set of Node G_{j+1}; Then initial weight W_{init}^{j};
Output: *Average Model N*;

for $k \in \mathbb{G}_{j+1}$ **do**
  $Train\ W_k \leftarrow Train_k\left(W_{init}^{j},\ ID_k\right)$
  $\sigma_k \leftarrow Sign(sk_k, W_k)$
  $Gossip\ \left((W_k, \sigma_k)\right)$
  $num = 0$
  while $k \leftarrow (W_i, \sigma_i)$ **do**
    if then $VS(W_i, \sigma_i) = 1$
      $Append_k(W_i)$
      if then $FC_k(W_i) = 0$
        $num = num + 1$
      end if
    end if
    if $num \geqslant m$ **then**
      $N_k \leftarrow FedAvg(\mathrm{M}_k, ID_k)$
      $\sigma_k \leftarrow Sign(sk_k, N_k)$
      $Gossip\ \left((N_k, \sigma_k)\right)$
    end if
  end while
end for

4.3.2. Incentive Mechanism

The improvement process of the federated learning model aligns with the principle of diminishing marginal utility in economics. At the same time, the contribution of each client in the first stage of federated learning is much more significant than in the second stage. Therefore, We draw on the principle of diminishing marginal utility in economics and the Pareto principle (80/20 rule) to allocate the rewards for each round. When the boundary point is determined, we divide the communication rounds of federated learning into two subgroups according to the boundary point: G_1 and G_2. Then, we divide the total rewards among the two groups in the ratio of 8:2. In the subgroup G_1, accounting for 80%, we design a method that uses the distance between the gradient model and the final model parameters to convert the contribution. This method can obtain a convincing contribution index while protecting the data privacy of each node.Note that the aggregation model is N_{end} and the aggregation method is the FedAvg algorithm:

$$N_{end} \leftarrow \sum_{K=1}^{K} \frac{n_k}{n} W_k \tag{8}$$

We define $W_k \leftarrow ClientUpdate(n, W)$. Then to evaluate the contribution degree C_k of a gradient model W_k, the following two steps need to be taken:

$$\theta_k = \frac{\langle N_{end}, W_k \rangle}{|N_{end}| \cdot |W_k|}, \tag{9}$$

$$C_k = \frac{\theta_k}{\sum_{i=0}^{n} \theta_i}. \tag{10}$$

By calculating the angle between the aggregate model and different gradient models, you can measure its overall contribution. And In the subgroup G_2, accounting for 20%, we adopt the method of evenly distributing rewards.

Finally, the task publisher utilizes the contribution evaluation algorithm to calculate each client's contribution, determining individual rewards accordingly. Subsequently, the task publisher records the calculated reward values in the blockchain.

5. Experimental Evaluation

5.1. Experimental Setup

(1) **Dataset description**: Our experimental datasets encompass three distinct categories, namely MNIST, FASION-MNSIT, and CIFAR10. MNIST constitutes a 10-class image dataset featuring single-channel grayscale images, comprising 60,000 training samples and 10,000 test samples. FASION-MNSIT, also a 10-class grayscale image dataset, comprises 60,000 training and 10,000 test samples. Meanwhile, the CIFAR-10 dataset, a 10-class color image dataset, includes 50,000 training and 10,000 test samples.

(2) **Data partitioning**: To emulate the non-IID (non-independently and identically distributed) nature of data in the federated environment, we employed the Dirichlet function to partition the dataset based on the number of clients.

(3) **Network configuration**: Unique network models were tailored for each dataset. Specifically, a 4-layer MLP network was devised for MNIST and FASION-MNSIT, while a 6-layer CNN network was designed to handle the CIFAR10 dataset. The Stochastic Gradient Descent (SGD) method was applied with a learning rate set to 0.001 for local model training.

(4) **Experimental infrastructure**: Our experiments were conducted on a high-performance workstation equipped with an Intel i9-12900K CPU (Intel, Santa Clara, CA, USA), 64 GB RAM (Overlockers UK, Newcastle-under-Lyme, UK), and an NVIDIA RTX 3090 GPU (Nvidia, Toronto, ON, Canada).

To facilitate our research, we developed a federated learning simulation framework capable of generating diverse client types, each exhibiting distinct behaviors.

(5) **Practical cases**: In IoT applications, devices are often deployed in different environments, generating non-independent and identically distributed (non-IID) data. This is similar to the characteristics of standard datasets, which can be used to simulate unevenly distributed image data collected by IoT devices, allowing for the evaluation of federated learning models in such environments. In the healthcare sector, one of the main advantages of federated learning is its ability to train models across institutions without sharing sensitive patient data. Datasets can simulate tasks such as handwritten character recognition in electronic health records, highlighting the importance of federated learning in preserving data privacy.

5.2. Secure Identification Algorithm Evaluation

5.2.1. Baseline

Experimental comparison: Our methodology undergoes a comprehensive evaluation against six established approaches.

(1) Baseline-FedAvg-Normal: we assume the absence of any malicious clients participating in the global model fusion, employing the FedAvg algorithm.

(2) Baseline-FedAvg-Attack: introducing a varying number of malicious clients into the FedAvg algorithm during global model fusion characterizes this scenario.

(3) Krum: this method employs the Euclidean distance approach, selecting the model with the smallest cumulative distance value as the global model.

(4) MultiKrum: an enhanced algorithm derived from Krum, MultiKrum selects the m models with the smallest cumulative distance values at each communication round, subsequently utilizing them to fuse the global models.

(5) Median: This technique involves sorting the jth parameter in all client models, choosing the median value as the jth parameter value for the global model. All global model parameter values follow this calculation process.

(6) TrimmedMean [7]: The algorithm begins by sorting each parameter of the client model and then selects the middle K values to compute the mean value of the parameter. This mean value becomes the parameter of the global model. The process iterates until all parameter values of the global model are determined.

5.2.2. Attack Types

Attack scenarios: Our experimentation involved the categorization of clients into normal and malicious groups, with the latter engaging in various attacks, including model poisoning and data poisoning in each communication round. The simulated attacks encompass three distinctive types:

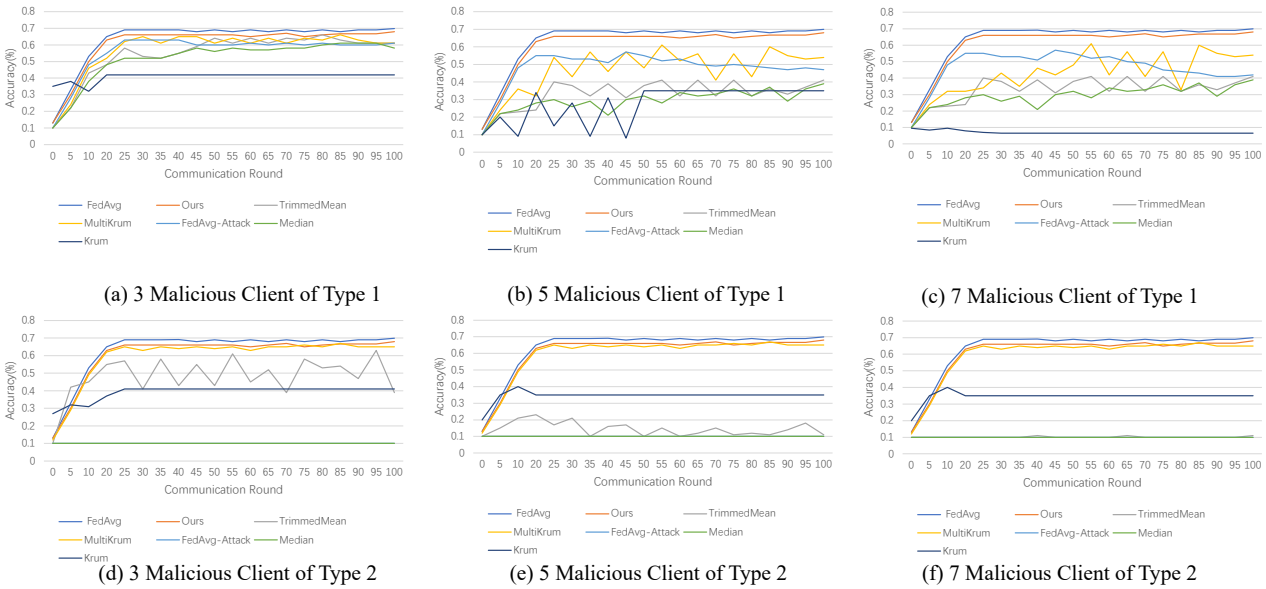
Label manipulation attack (Type 1): Malicious clients in this attack type manipulate the authentic labels of samples within their local datasets, substituting them with labels from different classes. Local datasets are tampered with through data poisoning techniques.

Parameter tampering attack (Type 2): Upon completion of client model training, malicious clients engage in model tampering by negatively modifying the model parameters.

Combined hybrid attack (Type 3): A synthesis of both Type 1 and Type 2 attacks, the hybrid attack involves orchestrating different malicious clients to execute each attack type independently. The quantity of malicious clients, set at 3, 5, and 7, corresponds to the number of clients for each attack type as (2,1), (3,2), and (4,3) respectively. This strategic approach introduces a nuanced combination of attacks, showcasing the adaptability of our framework under diverse adversarial scenarios.

5.2.3. Accuracy

Demonstrating the Efficacy of Each Algorithm in Federated Tasks: In order to provide a clearer insight into the performance of each algorithm in federated learning scenarios, we conducted a set of comparative experiments utilizing the Cifar-10 dataset. The results of these experiments are visually represented in Figure 4.



(a) 3 Malicious Client of Type 1 (b) 5 Malicious Client of Type 1 (c) 7 Malicious Client of Type 1

(d) 3 Malicious Client of Type 2 (e) 5 Malicious Client of Type 2 (f) 7 Malicious Client of Type 2

Figure 4. Algorithm accuracy under different attack types (To compare the accuracy of each algorithm in each communication round, we compare the three best-performance algorithms. In attack type 1, the accuracy of TrimmedMean and MutilKrum algorithms is severely degraded when the number of malicious clients is 5 and 7, respectively. In attack type 2, the TrimmedMean algorithm no longer works.

In the context of attack type 1, as the number of malicious clients increases, the accuracy of all algorithms exhibits fluctuations. Notably, when seven malicious nodes participate in model fusion, the global model convergence proves elusive for these algorithms.

For attack type 2, it is observed that the Krum, Median, and Trimmed-Mean algorithms encounter challenges in functioning effectively. Conversely, the Multi-Krum and our proposed algorithms demonstrate stability and achieve superior accuracy even under the influence of malicious activities.

This comparative analysis on the Cifar-10 dataset serves to underscore the algorithmic robustness and resilience, particularly in the face of adversarial scenarios, with our approach outperforming certain counterparts in maintaining stability and accuracy.

For malicious data, it often has fatal flaws such as large deviations, errors, and noise. There may even be a situation where the overall data distribution conforms to the sample, but there is a significant deviation in one or several segments of data, which can interfere and mislead the data trainer. As is shown in Figure 5.

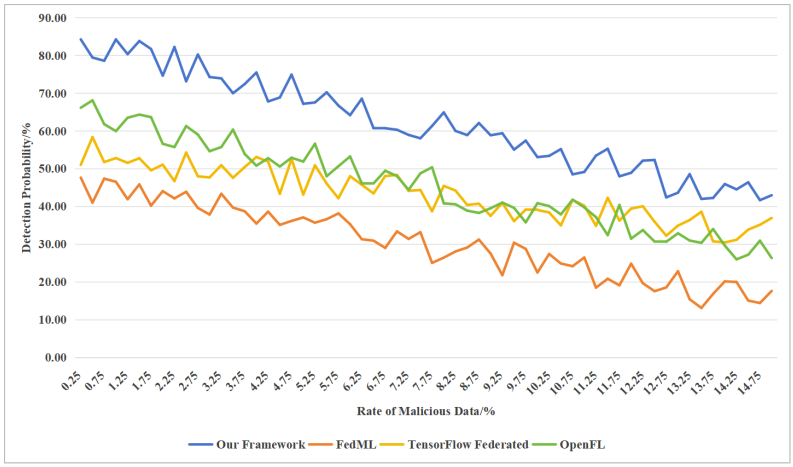


Figure 5. Regular and harmful cluster in sample dataset.

5.2.4. Clustering Results

Given that various attack behaviors are concealed within the parameters of client models, our proposed algorithm excels in identifying malicious clients by extracting distinctive features from each model, facilitating the clustering of diverse client models. To underscore the effectiveness of our algorithm, we conduct a detailed analysis of the clustering results in two distinct attack types.

In the context of attack type 1, illustrated using the MNIST dataset, the outcomes are presented in Figure 6. Across communication rounds, the features of normal clients converge, resulting in their proximity, while malicious clients resist clustering. This discrepancy arises from the convergence of normal client model features toward the characteristics of the global model over communication rounds. Conversely, the erratic tampering of data labels leads to inconsistent features among malicious clients.

For attack type 2, exemplified by the Cifar-10 dataset showcasing sign-flipping attacks, the malicious client alters the gradient direction, causing normal and malicious clients to form opposing groups. As communication rounds progress, both normal and malicious clients exhibit similar behavior, aggregating into two distinct groups. In this scenario, distinguishing the normal group becomes challenging. Hence, validation of aggregated models on a validation dataset or comparison of similarity with previous round models becomes crucial for resolving malicious model identification challenges. The detailed results are depicted in Figure 6.

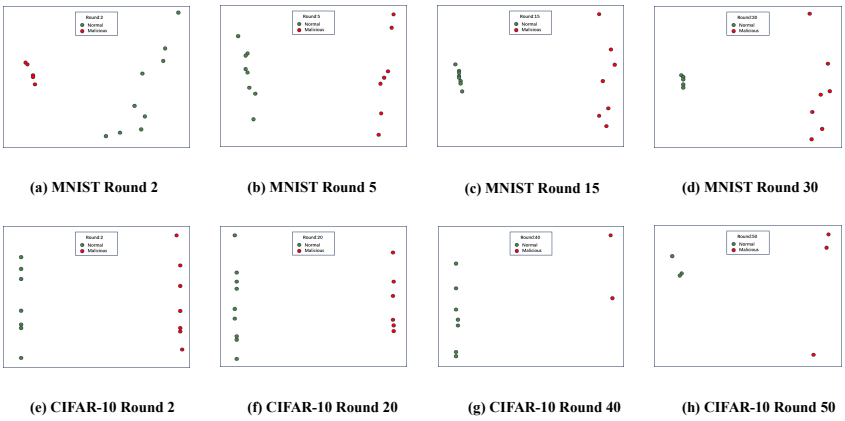


Figure 6. Clustering results for different attack types (In attack type 1, we take the MNIST dataset as an example and select the clustering results for a few critical communication rounds. In attack type 2, we take the CIFAR-10 dataset as an example and show the clustering results for each round).

5.2.5. Result Analysis

Traditional algorithms like Trimmed Mean operate by trimming extreme values from the updates received from participants, discarding the highest and lowest values, and calculating the mean of the remaining updates. This method is based on two assumptions: (1) the majority of participants are honest, and (2) the updates from malicious participants are significantly different from those of honest ones, making them easier to detect and discard. However, in specific attack scenarios, Trimmed Mean may fail. One of the main vulnerabilities is colluding attacks, where multiple malicious participants coordinate to design their updates to resemble those of honest participants, allowing them to bypass the trimming process. Over multiple rounds, attackers can gradually adjust their updates to approach the Trimmed Mean threshold.

Another issue arises in environments with low data homogeneity, where the data distribution across participants is significantly different. In such cases, the updates from honest participants may naturally vary widely, causing the Trimmed Mean algorithm to inadvertently discard useful updates from honest participants while retaining some from malicious ones. Additionally, in Byzantine attacks, attackers can inject subtly biased updates that mimic normal participant behavior, poisoning the global model over time. These limitations suggest that Trimmed Mean is less effective in defending against sophisticated and well-coordinated attacks, especially in scenarios with diverse data distributions.

The experimental data shown in Figure 7 confirm this result.

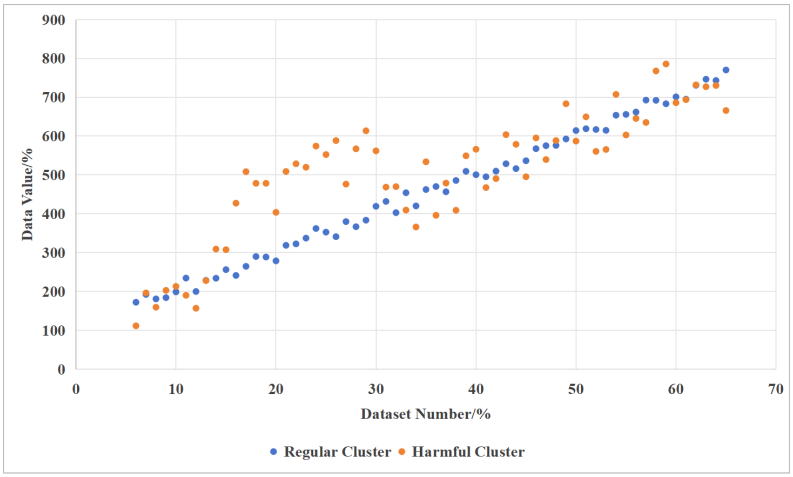


Figure 7. Detection probability of malicious data by FL under different frameworks.

5.3. Fair Federated Consensus Algorithm Evaluation

5.3.1. Election Probability

In this paper, we derived the probability, denoted as P, that a model generated by more than half of the nodes is considered N_{best}. Subsequently, based on the performance evaluation function $F(N_k, i)$, model selection was conducted by executing 100,000 rounds of the PF consensus protocol to compete for write permissions. The process recorded the instances where a random node produced NB, and these occurrences were then converted into p values. Figure 8 illustrates that when the data distribution among nodes remains IID, P continues to decrease as G increases, given that z is greater than 0.5. Simultaneously, due to the fact that the number of nodes and votes can only be integers in practice, this leads to periodic variations in P as G increases. These results collectively indicate the consistency between the theoretical analysis presented in this paper and real-world scenarios. By employing the same evaluation function to calculate accuracy metrics (weighted sum of ACC and F1) for machine learning models, further analysis can be conducted on the node packaging behavior of the system in practical applications.

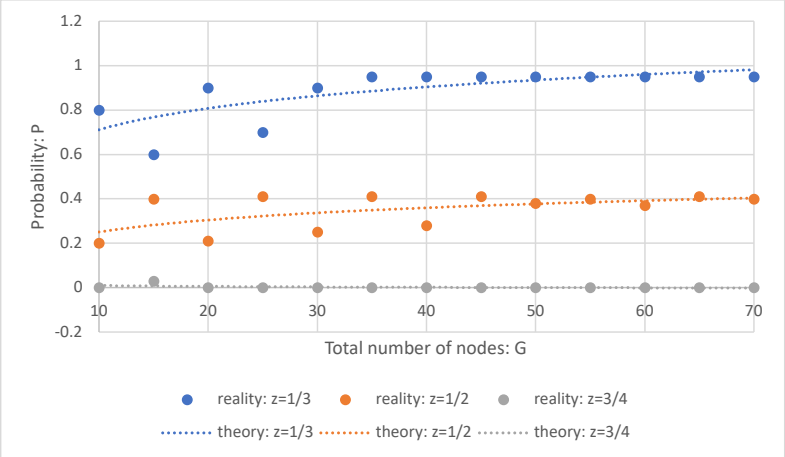


Figure 8. The influence of z and G on P when data distribution is IID.

5.3.2. Communication Overhead

By observing Figure 9, it can be noted that the communication frequency required for blockchain block generation conforms to the pattern. This conclusion affirms the accuracy of the theoretical analysis presented in Section 3. However, it is not sufficient to demonstrate the feasibility of the federated learning blockchain proposed in this paper.

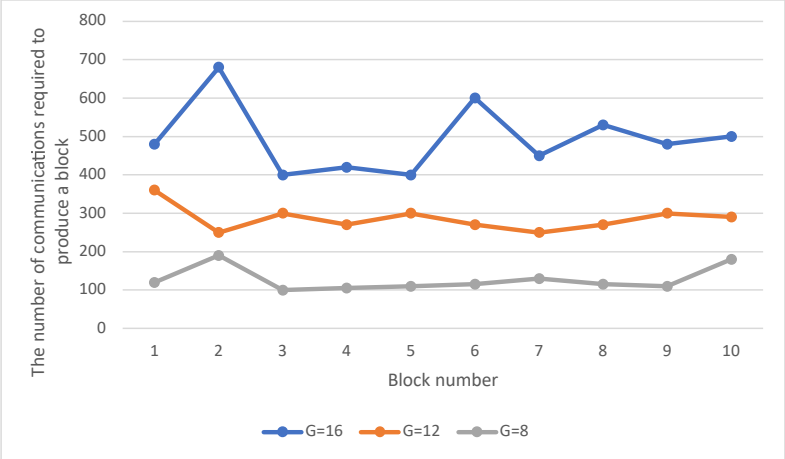
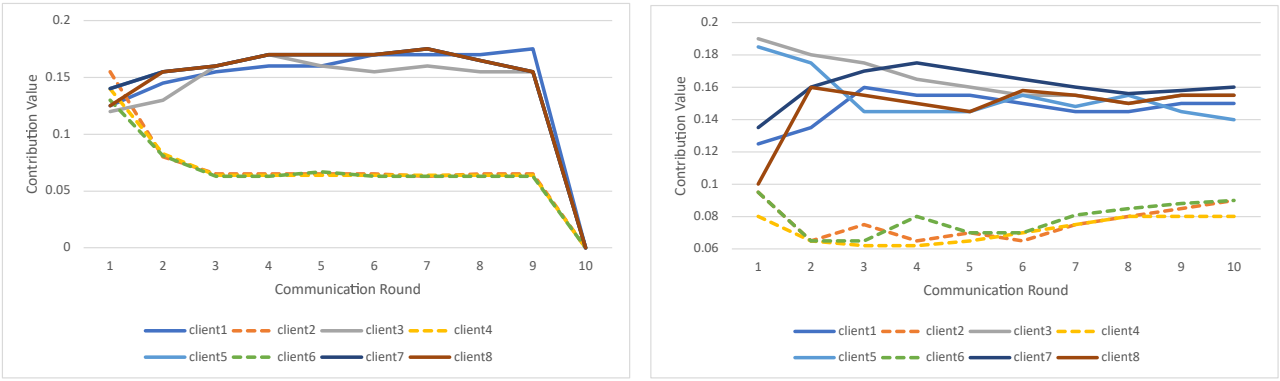


Figure 9. Changes in the number of communications that generate new blocks.

In Figure 10, the segmented calculation method for determining sample contribution values is presented to demonstrate how fairness is ensured within the proposed federated learning framework. The figure outlines two distinct scenarios: Scenario 1 employs a segmented method to reduce the contribution values of clients with smaller sample sizes. This ensures that clients contributing smaller amounts of data are not unfairly rewarded, thus maintaining fairness in the system. By lowering the impact of smaller contributions, the model avoids disproportionately rewarding these clients, aligning with the principle of contribution-based fairness.

Scenario 2 introduces penalty parameters for low-quality models, further emphasizing fairness. In this case, clients contributing poor-quality data receive reduced contribution values, ensuring that only high-quality contributions are adequately rewarded. This discourages free-riding behavior, where participants might contribute substandard models to gain rewards without meaningful participation.



(**a**) Scenario-1 Ours (**b**) Scenario-2 Ours

Figure 10. A Segmented calculation method for determining sample contribution values.

These two strategies, as illustrated in Figure 10, show that the framework ensures fairness by carefully evaluating both the quantity and quality of contributions, adjusting rewards accordingly. This dynamic adjustment mechanism creates an incentive for all participants to actively contribute valuable data and models, thus promoting fairness across all clients in the federated learning process.

5.3.3. Reward Allocation

According to the experimental configuration, we conducted two scenarios on the two datasets, our approach exhibits commendable allocation performance in diverse scenarios. In the first scenario, we introduce a segmented calculation method for determining sample contribution values, leading to a notable reduction in the contribution values of clients with small sample sizes. Simultaneously, in scenarios characterized by low-quality models (Scenario 2), we incorporate penalty parameters to diminish the contribution values associated with such models. This strategic inclusion aims to amplify the disparity in contribution values among distinct client types. Consequently, our method adeptly ensures a fair and equitable assessment of contribution values across the federated learning framework.

5.4. The Difference between Blockchain Based Trusted Federated Learning Framework and Non Blockchain Based Framework

Blockchain-based Trustworthy Federated Learning Framework vs. Non-Blockchain Framework The integration of blockchain technology in federated learning frameworks introduces distinct characteristics that set it apart from traditional, non-blockchain-based systems. Below are the key distinctions:

(1) **Data Security and Privacy Preservation**:

Blockchain-based: Blockchain's encrypted and distributed ledger provides enhanced security and immutability of data, mitigating the risks of data breaches and tampering in federated learning.

Non-blockchain: Centralized storage and processing in conventional federated learning may increase vulnerabilities to data theft and unauthorized alterations.

(2) **Decentralization**:

Blockchain-based: The decentralized nature of blockchain eliminates single points of control or failure, bolstering the robustness and resilience against attacks in federated learning ecosystems.

Non-blockchain: Centralized federated learning systems may rely on a few central nodes, potentially creating systemic risks and bottlenecks.

(3) **Transparency and Traceability**:

Blockchain-based: All transactions and model updates are recorded on the blockchain, ensuring high transparency and traceability, allowing participants to verify and audit the learning process.

Non-blockchain: The record-keeping of data and model updates may lack transparency, making auditing and verification more challenging.

(4) **Incentive Mechanisms**:

Blockchain-based: Smart contracts within blockchain automate the execution of incentive mechanisms, rewarding participants based on their contributions, thus encouraging active engagement.

Non-blockchain: Incentive schemes may require additional management and may not be as automated or equitable compared with blockchain-based systems.

(5) **Consensus Mechanisms**:

Blockchain-based: Blockchain systems employ consensus algorithms (e.g., Proof of Work, Proof of Stake) to achieve agreement within the network, which is crucial for model updates and validations in federated learning.

Non-blockchain: Consensus may be reached through centralized means or specific algorithms, lacking the decentralization and fairness inherent in blockchain consensus mechanisms.

(6) **Scalability**:

Blockchain-based: The distributed nature of blockchain facilitates system scalability, with the ability to enhance processing power and reliability as the number of participants grows.

Non-blockchain: Centralized systems may encounter performance limitations when scaling, as all computations and data flows are channeled through central nodes.

(7) **Cost**:

Blockchain-based: The deployment and maintenance of blockchain can be costly, particularly in networks requiring significant computational and storage resources.

Non-blockchain: Traditional federated learning systems may have lower initial deployment costs, but these can escalate with increasing data volumes and heightened security demands.

In summary, blockchain-based federated learning frameworks offer a more secure, transparent, and decentralized solution, albeit with potentially higher costs and complexity. Non-blockchain frameworks may be simpler and less costly to implement but may not match the security and transparency of blockchain-based systems.

Figure 11 shows the comparison of computation time between a blockchain based trusted federated learning framework and a non blockchain based framework at different training epochs. The blue solid line in the figure represents a blockchain based framework, while the green dashed line represents a non blockchain based framework. Each marker displays the computation time for different epochs, and as the training epochs increase, the computation time for blockchain frameworks increases faster than for non blockchain frameworks.

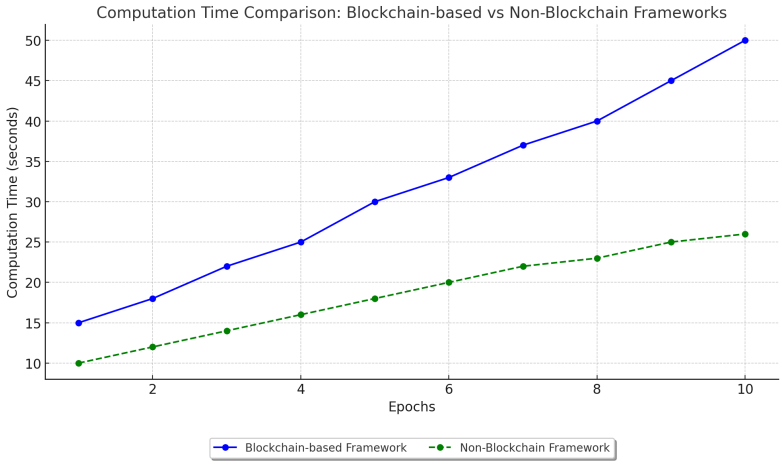


Figure 11. Computation time comparison:blockchain-based vs non-blockchain frameworks.

6. Conclusions

This paper presents a federated learning framework with a focus on trustworthiness and fairness. Initially, we establish a federated computing framework built upon principles of trustworthiness, ensuring computational security throughout the federated learning process through the integration of a meticulously designed smart contract mechanism and information storage structure. We introduce a method for detecting malicious models to guarantee secure model aggregation. Additionally, we devise a mechanism for evaluating contributions, with the objective of achieving equitable reward distribution. Experimental results demonstrate that our algorithm outperforms Euclidean distance-based and parameter-pruning-based methods in detecting malicious clients. Furthermore, the proposed contribution evaluation method offers a more equitable reward distribution system.

In real-world environments, as federated learning becomes increasingly widespread, our framework demonstrates significant potential for broad application. For instance, in banking systems where data security and privacy protection are paramount, our framework can ensure the secure sharing of sensitive data among multiple financial institutions while preventing malicious attacks, thereby enhancing the credibility of financial services and fostering user trust. In the healthcare sector, sharing medical data and training models are crucial for disease prediction and personalized treatment. Our framework can assist hospitals and research institutions in securely sharing data and aggregating models while protecting patient privacy, thereby advancing medical research and improving patient care. In future research, we aim to develop dynamically adaptive mechanisms that adjust security strategies and model evaluation methods in response to real-time threats. Additionally, we plan to explore the performance of the framework under large-scale datasets and high-load conditions, optimizing its scalability to support larger federated learning networks and more participating nodes.

Author Contributions: Conceptualization, F.Z. and F.H.; methodology, F.H.; software, Y.Z.; validation, F.Z. and F.H.; formal analysis, F.Z.; writing—original draft preparation, F.Z.; writing—review and editing, F.H. and X.T.; project administration, B.C.; funding acquisition, B.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China, under Grant 62203219 and 62176122.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qu, Y.; Uddin, M.P.; Gan, C.; Xiang, Y.; Gao, L.; Yearwood, J. Blockchain-enabled federated learning: A survey. *Acm Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]
2. Li, S.; Cheng, Y.; Wang, W.; Liu, Y.; Chen, T. Learning to detect malicious clients for robust federated learning. *arXiv* **2020**, arXiv:2002.00211.
3. Chen, L.; Dong, C.; Qiao, S.; Huang, Z.; Wang, K.; Nie, Y.; Hou, Z.; Tan, C. FedDRL: A Trustworthy Federated Learning Model Fusion Method Based on Staged Reinforcement Learning. *arXiv* **2023**, arXiv:2307.13716. [CrossRef]
4. Zhang, Z.; Cao, X.; Jia, J.; Gong, N.Z. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2545–2555.
5. Andreina, S.; Marson, G.A.; Möllering, H.; Karame, G. Baffle: Backdoor detection via feedback-based federated learning. In Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), Washington, DC, USA, 7–10 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 852–863.
6. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 119–129.
7. Yin, D.; Chen, Y.; Kannan, R.; Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5650–5659.
8. Abdel-Basset, M.; Moustafa, N.; Hawash, H. Privacy-preserved cyberattack detection in Industrial Edge of Things (IEoT): A blockchain-orchestrated federated learning approach. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7920–7934. [CrossRef]
9. Li, Y.; Chen, C.; Liu, N.; Huang, H.; Zheng, Z.; Yan, Q. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Netw.* **2020**, *35*, 234–241. [CrossRef]
10. Shayan, M.; Fung, C.; Yoon, C.J.M.; Beschastnikh, I. Biscotti: A Blockchain System for Private and Secure Federated Learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1513–1525. [CrossRef]
11. Ma, C.; Li, J.; Shi, L.; Ding, M.; Wang, T.; Han, Z.; Poor, H.V. When Federated Learning Meets Blockchain: A New Distributed Learning Paradigm. *IEEE Comput. Intell. Mag.* **2022**, *17*, 26–33. [CrossRef]
12. Kang, J.; Xiong, Z.; Niyato, D.; Zou, Y.; Zhang, Y.; Guizani, M. Reliable federated learning for mobile networks. *IEEE Wirel. Commun.* **2020**, *27*, 72–80. [CrossRef]
13. Zhang, Q.; Ding, Q.; Zhu, J.; Li, D. Blockchain empowered reliable federated learning by worker selection: A trustworthy reputation evaluation method. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Nanjing, China, 29 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
14. Qi, J.; Lin, F.; Chen, Z.; Tang, C.; Jia, R.; Li, M. High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation. *IEEE Internet Things J.* **2022**, *9*, 18378–18391. [CrossRef]
15. Martinez, I.; Francis, S.; Hafid, A.S. Record and reward federated learning contributions with blockchain. In Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Guilin, China, 17–19 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 50–57.
16. Wei, S.; Tong, Y.; Zhou, Z.; Song, T. Efficient and fair data valuation for horizontal federated learning. In *Federated Learning: Privacy and Incentive*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 139–152.
17. Yan, B.; Liu, B.; Wang, L.; Zhou, Y.; Liang, Z.; Liu, M.; Xu, C.Z. Fedcm: A real-time contribution measurement method for participants in federated learning. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; IEEE: Piscataway, NJ, USA; 2021; pp. 1–8.
18. Liu, Z.; Chen, Y.; Yu, H.; Liu, Y.; Cui, L. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *Acm Trans. Intell. Syst. Technol.* **2022**, *13*, 1–21. [CrossRef]
19. Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; Zhang, J. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet Things J.* **2019**, *6*, 10700–10714. [CrossRef]
20. Gao, L.; Li, L.; Chen, Y.; Xu, C.; Xu, M. FGFL: A blockchain-based fair incentive governor for Federated Learning. *J. Parallel Distrib. Comput.* **2022**, *163*, 283–299. [CrossRef]
21. Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; Yang, Q. A fairness-aware incentive scheme for federated learning. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, 7–8 February 2020; pp. 393–399.
22. Chen, Z.; Liu, Z.; Ng, K.L.; Yu, H.; Liu, Y.; Yang, Q. A gamified research tool for incentive mechanism design in federated learning. In *Federated Learning: Privacy and Incentive*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 168–175.
23. García-Pérez, Á.; Gotsman, A.; Meshman, Y.; Sergey, I. Paxos consensus, deconstructed and abstracted. In *Programming Languages and Systems: 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018*; Springer International Publishing: Thessaloniki, Greece, 2018; pp. 912–939.