electronics

MDPI

Article

Research on Privacy Protection in Federated Learning Combining Distillation Defense and Blockchain

Changxu Wan [1], Ying Wang [1,*], Jianbo Xu [1], Junjie Wu [2], Tiantian Zhang [1] and Yulong Wang [1]

[1] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China; wancx@mail.hnust.edu.cn (C.W.); jbxu@hnust.edu.cn (J.X.); Zhangtt@mail.hnust.edu.cn (T.Z.); wyl@mail.hnust.edu.cn (Y.W.)

[2] Quanzhou Institute of Information Engineering, Quanzhou 362000, China; wujj@qzuie.edu.cn

* Correspondence: 1050060@hnust.edu.cn

Abstract: Traditional federated learning addresses the data security issues arising from the need to centralize client datasets on a central server for model training. However, this approach still poses privacy protection risks. For instance, central servers cannot verify privacy leaks resulting from poisoning attacks by malicious clients. Additionally, adversarial sample attacks can infer specific samples from the original data by testing the local models on client devices. This paper proposes a federated learning privacy protection method combining distillation defense technology with blockchain architecture. The method utilizes distillation defense technology to reduce the sensitivity of client devices participating in federated learning to perturbations and enhance their ability to resist adversarial sample attacks locally. This not only reduces communication overhead and improves learning efficiency but also enhances the model's generalization ability. Furthermore, the method leverages the "decentralized" nature of blockchain architecture as a trusted record-keeping mechanism to audit information interactions among clients and shared model parameters. This addresses privacy leakage issues resulting from poisoning attacks by some clients during the model construction process. Simulation experiment results demonstrate that the proposed method, compared with traditional federated learning, ensures model convergence, detects malicious clients, and improves the participation level of highly reputable clients. Moreover, by reducing the sensitivity of local clients to perturbations, it enhances their ability to effectively resist adversarial sample attacks.

Keywords: decentralization; distillation defense; blockchain; privacy protection; federated learning

check for updates

Citation: Wan, C.; Wang, Y.; Xu, J.; Wu, J.; Zhang, T.; Wang, Y. Research on Privacy Protection in Federated Learning Combining Distillation Defense and Blockchain. *Electronics* **2024**, *13*, 679. https://doi.org/10.3390/electronics13040679

Academic Editor: Andrei Kelarev

Received: 3 January 2024
Revised: 1 February 2024
Accepted: 2 February 2024
Published: 6 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, the rapid advancement of artificial intelligence in various domains such as finance, healthcare, IoT, urban planning, and autonomous driving has been propelled by the surge in big data. Within these realms, vast amounts of sensitive data, accumulated by individuals or organizations, hold immense commercial value and encompass users' personal privacy information. This has led to a delicate balance between safeguarding data privacy and leveraging it for commercial purposes.

To address this challenge, distributed learning methods, especially federated learning (FL) technology [1], have emerged. FL enables collaborative data efforts across organizations while mitigating the risk of privacy breaches. It involves multiple data holders—such as mobile phones, IoT devices, financial, and healthcare institutions—cooperatively training models without directly sharing data. Instead, they exchange training parameters solely at intermediate stages, thereby safeguarding data privacy and security. While, ideally, federated learning models that prioritize data privacy yield shared data models comparable to those trained on centrally stored datasets, traditional FL models still rely on a central server for aggregating training parameters. This dependency increases system vulnerability to single-point failures and distributed denial-of-service (DDoS) attacks. To

overcome the centralization weaknesses inherent in traditional federated learning, this study adopts a decentralized FL approach. This method eradicates dependence on a central server by enabling direct parameter exchange among individual nodes. Additionally, the decentralized approach's improvements in data ownership and privacy protection further enhance the potential applications of federated learning.

With the introduction of decentralized federated learning, blockchain technology has become a pivotal tool in ensuring system integrity and reliability [2]. This study leverages blockchain to establish a transparent and tamper-proof mechanism for updating local model parameters, thereby enhancing the security and transparency of the entire system. However, with the introduction of blockchain, incentive mechanisms and voting systems have become particularly crucial in ensuring active participation from highly reputable clients [3]. This study integrates blockchain-based incentive mechanisms and voting systems to guarantee the effective detection of malicious client activities, even in the presence of attacks from malicious nodes. This approach not only detects malicious clients effectively but also maintains model accuracy.

Finally, addressing client-side security concerns, this study introduces knowledge distillation techniques to defend against adversarial sample attacks. This distillation defense not only enhances the resistance of client-side models against attacks but also strengthens the models' capability to handle heterogeneous data [4]. This enhancement allows each client to train personalized models based on their unique characteristics. This paper combines distillation defense and decentralized federated learning achieved through blockchain technology to accomplish the following research objectives:

1. Utilizing distillation defense techniques to reduce the sensitivity of participating client devices in federated learning to perturbations. This enhancement increases the local client's effectiveness in countering adversarial sample attacks while reducing communication overhead. Consequently, it improves learning efficiency and enhances the model's generalization capabilities.
2. Leveraging decentralized blockchain technology as a trusted record-keeping mechanism to audit information exchange among clients and the shared model parameters. This facilitates the detection of privacy breaches resulting from poisoning attacks by certain clients during the model construction process.

This study delves more deeply into the framework of decentralized federated learning, integrating distillation defense and blockchain. Through experiments, it aims to validate the performance of this integration. The study seeks to advance the exploration of federated learning, seeking a new balance between privacy protection and model performance. It aims to pave the way for novel research directions in privacy preservation within distributed learning methodologies.

2. Related Work

2.1. Distillation Defense and Federated Learning

Deep learning has achieved remarkable success in fields like computer vision. However, scholars like Szegedy first observed a security issue. They found that despite the exceptional accuracy of deep neural networks in image classification tasks [5], they are exceptionally fragile and susceptible to minute adversarial perturbations. These disturbances are nearly imperceptible to the human visual system, yet they pose a significant threat [6]. These adversarial attacks can completely alter the classification predictions of deep neural network classifiers for images. What is more concerning is that the attacked models often report high confidence in these misclassified predictions. Additionally, the same image perturbation can deceive multiple different deep neural network classifiers [7].

Knowledge distillation is commonly employed in federated learning to defend against attacks targeting communication bottlenecks, thereby reducing communication overhead, saving storage space, and minimizing redundant parameters. In the context of privacy preservation, adversarial sample attacks are often used to test the model's response, aiming to identify specific samples within the training data, thereby leading to privacy data

breaches. In defending against adversarial attacks, knowledge distillation techniques have proven to enhance the model's generalization and robustness. While previous studies like that of Papernot et al. have made progress in countering adversarial sample attacks [8], the effective integration of distillation defense methods in the federated learning environment, especially within decentralized architectures, remains relatively unexplored. Furthermore, existing solutions in federated learning often overlook issues related to data heterogeneity, as well as the dynamics of model updates and the distillation process.

McMahan et al. proposed the Federated Averaging algorithm (FedAvg), where the central server aggregates and averages user-updated model parameters for the subsequent global iteration [1]. However, they did not account for nonindependent and identically distributed (Non-IID) client data or address security issues in centralized federated learning. Lin et al. combined knowledge distillation (KD) methods with federated learning, transferring knowledge from heterogeneous client models to a global model [9]. Yet, their focus did not particularly address the issue of data heterogeneity in federated learning. Divi et al. employed knowledge distillation (KD) to address data heterogeneity in federated learning and introduced the Personalized Federated Learning algorithm (PERSFL). This algorithm comprises two stages: the initial phase focuses solely on federated learning, while the second phase involves users selecting the best teacher model from each iteration's global model and conducting distillation for personalization [10]. However, these two stages operate independently, and the teacher model remains unchanged throughout the distillation process, lacking adjustment based on different student models. Gong et al. introduced a distillation-based federated learning framework, FedKD, resolving issues regarding protecting local data privacy and addressing data distribution imbalances by leveraging unlabeled and cross-domain public data for learning [11]. To mitigate communication bottlenecks, they employed a one-time offline knowledge distillation process. However, in practical applications, this approach might be constrained by limited public data, leading to suboptimal model training outcomes. Additionally, the method proposed by Hong and Tao et al. combines weight update aggregation with model distillation [12]. Typically conducted centrally, it involves client-side distillation and may potentially restrict the scale of the aggregated model.

The core concept behind distillation defense involves extracting model knowledge and transferring it into smaller networks to maintain accuracy similar to larger networks while enhancing the model's generalization capability on samples beyond the training dataset. This enhancement fortifies the model's resilience to perturbations. Distillation is a technique that augments the robustness of training networks through soft target training, aiming to reduce the impact of adversarial attacks on deep learning models and increase their credibility. This method presents an effective protective mechanism for classification tasks in adversarial environments. The emergence of federated distillation offers a new approach, enabling integrated learning across heterogeneous models, reducing communication costs, and improving performance on non-IID data.

2.2. Blockchain and Federated Learning

In the field of federated learning, traditional centralized architecture poses challenges regarding data privacy and security. Despite federated learning's intrinsic aim to safeguard privacy, vulnerabilities in the model parameter update process can lead to privacy breaches. To address this, FLChain introduces an immutable local model parameter update scheme using blockchain. It devises a federated learning architecture based on a blockchain peer-to-peer network, eliminating reliance on a central server and enhancing system integrity and robustness [13]. Awan et al. propose a blockchain-based privacy protection framework combined with encryption techniques to safeguard privacy data, eliminating the semihonest assumption of participants. However, this method still carries the risk of single-point failure in parameter servers and lacks optimization to address this issue [14]. Jin et al. introduced the Blockchain FL architecture to achieve decentralization and secure model storage [15]. FL-Block represents a blockchain-based federated learning approach

ensuring block generation efficiency by employing a distributed hash table and utilizing a proof-of-work mechanism to maintain global model consistency [16]. Kim et al. integrated blockchain into federated learning to address single-point failure but did not consider privacy protection for intermediate parameters [17]. Zhao et al. proposed a reputation-based federated learning system applied in mobile edge computing, incorporating differential privacy to safeguard customers' sensitive information [18].

Martinez et al. utilized REST-API (Representational State Transfer Application Programming Interface) to interact with Hyperledger Fabric blockchain for recording and incentivizing gradient uploads [19]. However, the study primarily focused on enhancing and optimizing the incentive mechanisms in federated learning through blockchain, lacking an in-depth exploration of data security and privacy protection issues. Among these studies, some overlooked the privacy protection of intermediate parameters, while others did not delve deeply into effective incentive mechanisms to encourage device participation. Additionally, certain research did not sufficiently address potential security issues in terminal devices [20]. In the computational environment of federated learning, constructing a global model and safeguarding the data privacy of honest participants becomes increasingly challenging when there are multiple untrusted participants. These malicious entities can collude and use the shared global model to perform inference attacks, deducing the private information of other participants. Consequently, they can exploit contributions from honest participants within the global model to extract sensitive data [21]. Hence, assessing collusion attacks in federated learning is crucial.

Furthermore, blockchain-based federated learning ensures system integrity, reliability, and robustness [22]. These methodologies within the research utilize blockchain to specifically address the over-reliance on central servers in FL. However, incentive mechanisms and voting systems are crucial components for ensuring the active participation of each device in blockchain-based federated learning. Designing reasonable incentive and voting mechanisms becomes essential to reward honest and contributing clients while penalizing malicious or inefficient client behavior.

Summarizing the above research, this study proposes a novel decentralized federated learning method. The method utilizes blockchain technology collectively maintained by all participating clients to achieve decentralized federated learning. It ensures the active participation of highly reputable clients through incentive and voting mechanisms. Simultaneously, it employs knowledge distillation technology to reduce the sensitivity of client models to perturbations, defending against privacy leaks caused by adversarial sample testing attacks. Experimental results indicate that the proposed method ensures model convergence and accuracy. Compared with traditional federated learning training models, this study effectively detects malicious clients while increasing the participation level of highly reputable clients. By reducing the sensitivity of local clients to perturbations, it enhances their ability to resist adversarial sample attacks. This research aims to provide a new solution for data privacy protection and performance balance in federated learning.

3. Methods

3.1. System Model

This research's system architecture, as shown in Figure 1, includes a blockchain maintained collectively by all clients. During a training round, all clients are randomly assigned to three roles: miners, working clients, and validating clients. The working clients engage in federated learning for data training, incorporating distillation defense mechanisms during the process to safeguard local data. To ensure model security and quality, the system employs validating clients to assess the performance of the models uploaded by working clients. It is important to note that since validators do not directly access local data from other clients, this study uses proxy comparison methods for data verification. Finally, the system records the process of each training round and its ultimate outcome through selected miner roles onto the collectively maintained blockchain.

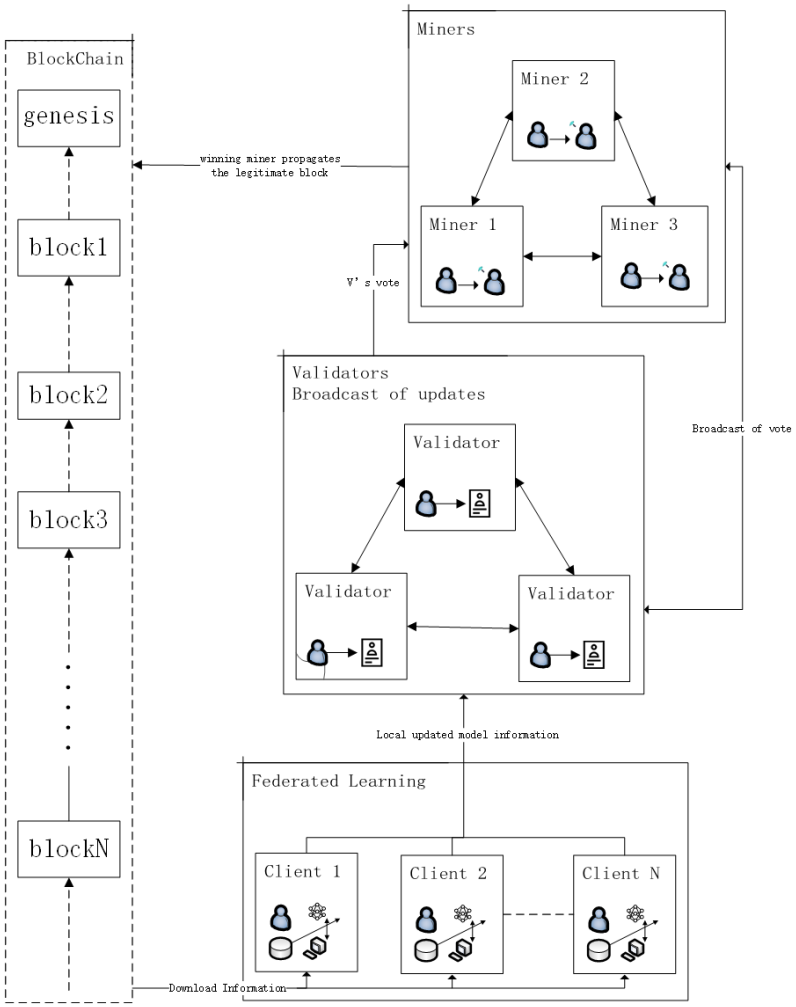


Figure 1. Distillation defense and decentralized federated learning system architecture diagram.

To ensure fairness and effectiveness, this study employs role rotation and role-based incentive mechanisms. These mechanisms are employed by the system to ensure each participant contributes their data to the federated learning training and receives corresponding incentives to sustain system operations. Moreover, the role-based incentive mechanism serves as the criterion for selecting the crucial role of miners within the system.

In traditional federated learning, the global model G is the weighted average of all local models M_i. The aggregation process can be represented as follows:

$$G = \sum_{i=1}^{n} w_i \cdot M_i \tag{1}$$

In this equation, n represents the number of clients participating in federated learning, and w_i signifies the weight of the local model M_i used for weighting the local model. In this study, during distillation, the prediction outcomes of each local model and the global model are fused for training the local model. The loss function for each local model M_i is represented as Equation (2):

$$L_i = w_i \cdot \left(L_{\mathbf{CE}}(M_i, D_i) + \frac{\alpha}{T_i} \cdot L_{\mathbf{KD}}(M_i, G) \right) \tag{2}$$

where L_i represents the loss function on the i-th device, $L_{\mathbf{CE}}$ is the cross-entropy loss function used to measure the performance of the local model M_i on local data D_i, $L_{\mathbf{KD}}$ is the knowledge distillation loss function employed to assess the disparity between the

local model M_i and the global model G, and $\alpha \in \{0,1\}$ is a hyperparameter used to weigh the local model's prediction against the global model's prediction, with a larger α emphasizing the influence of the global model more strongly. This study considers the local weight w_i during the aggregation process and introduces the temperature parameter T_i from knowledge distillation to control the degree of softening of the parameters. A larger T_i results in smoother model predictions, while a smaller one emphasizes the model's own predictions more. In the experimentation phase, this temperature parameter is adjusted by the local clients to control the strength of knowledge distillation.

In the process of distillation defense, each local model M_i retains distilled knowledge. This study incorporates this knowledge into the global model G using the following approach, as shown in Equation (3):

$$G = \sum_{i=1}^{n} w_i \cdot (\beta \cdot M_i + (1 - \beta) \cdot G) \tag{3}$$

In this context, $\beta \in \{0,1\}$ serves as a hyperparameter governing the influence of distilled knowledge on the weights of the global model. A higher β value tends to make the global model more affected by distilled knowledge, while a lower β value tends to preserve the model's own weights to a greater extent.

3.2. Attack Model

This study makes the following assumptions about adversaries: (1) attackers possess certain knowledge of adversarial samples, allowing them to choose and implement known adversarial sample generation methods, such as FGSM; (2) attackers have limited model information, namely model type, partial parameters, or general model architecture information, which can be public or obtained during the federated learning process; (3) attackers have a certain sensitivity to private information or personal data and aim to use attacks to assess the target model's understanding of private information; (4) attackers wish to maintain a certain level of stealth in their actions to avoid triggering the model's defense mechanisms or being detected; and (5) attackers have computational resources and time but may not necessarily have extensive computational capabilities or datasets. Our goal is to explore the potential of detecting model privacy through adversarial sample attacks, even against a highly restricted adversary.

3.3. Workflow

Assuming a set of clients $C = \{c_1, c_2, \ldots, c_n\}$ participating in federated learning, experiencing $R = \{r_1, r_2, \ldots, r_i, \ldots\}$ rounds of learning, each client is randomly designated as a miner m ($m \in M$), a validator v ($v \in V$), or a working client d ($d \in U$), where $M + V + U = C$. Clients perform their respective functions based on their assigned roles. Each client possesses a unique ID for identification, which is also the client's public key used for verifying generated transactions or transaction blocks. In each round r_i, the global model G_{i-1} is constructed. The primary steps and Algorithm 1 are outlined as follows:

Step 1: Each selected working client d for training utilizes the global model G_{i-1} to generate a local model L_{wi}. In traditional federated learning, L_{wi} is directly used to compute the global model G_i; in the proposed model of this study, L_{wi} must undergo validation by the validator.

Step 2: Based on the learning outcomes, the working clients calculate the expected reward r_{wi} and encapsulate the expected reward r_{wi} along with the local model L_{wi} into transactions tx_{wi} signed with the private key of each working client, which are then sent to the validator v.

Step 3: The validator v verifies the transactions tx_{wi} from the working clients and obtains the validation reward $r_{v-\text{verify}}$.

Step 4: Based on local data, the validator v votes $v_{tv}(L_{wi})$, indicating affirmation or negation towards the model.

$$v_{tv}(L_{wi}) = \begin{cases} 1, & \text{if the validator chooses positive} \\ 0, & \text{if the validator chooses negative} \end{cases} \tag{4}$$

Step 5: The validator updates the model locally for voting purposes.

Step 6: The validator's transaction $tx_{vi}(L_{wi})$ is broadcasted to the miners, who obtain all validator voting results, as shown in Equation (5).

$$r_{wj} = \{ \begin{matrix} \text{Reward}, & \text{if Rj} > 0 \\ 0, & \text{otherwise} \end{matrix} \tag{5}$$

Algorithm 1: Distillation Defense Combined With Blockchain In Decentralized Federated Learning

Input: Global model G, Rounds R, Client set C
Output: Optimized global model G'
for *round* $\leftarrow 1$ **to** R **do**
  Construct G according to federated learning rules: $G_i = f(G_{i-1}, M_i)$;
  for *each client* $c \in C$ **do**
    Assign a role to client;
    if *Role == 'Worker'* **then**
      Train a local model using global model: $M_i = \text{Train}_{\text{local}}(G_{i-1})$;
      Apply knowledge distillation defense on local model M_i;
      Calculate the local loss:
       local loss $= w_i * \left(L_{CE}(M_i, D_i) + \left(\frac{\alpha}{T_i} \right) * L_{KD}(M_i, G) \right)$;
      Sign transaction with client's private key, local model M_i, and calculated reward;
      Send transaction to validators;
    end
    else if *Role == 'Validator'* **then**
      Receive transaction;
      Verify signature of transaction;
      Cast vote based on transaction;
      Update local model with received vote;
      Broadcast vote to miners;
    end
    else if *Role == 'Miner'* **then**
      Collect votes from validators;
      Validate collected votes;
      Update blockchain with validated votes;
      Reward miner client;
    end
    Reward client c based on role;
  end
end
return Optimized global model G';

If L_{wi} is voted as positive within a legitimate block r_i, the working client involved in the training receives the reward r_{wi}. This process ensures model quality and legitimacy for federated learning tasks through reward and validation mechanisms.

In this validation mechanism, it is essential to note that the validator can only access L_{wi} but cannot directly access the C_w model used for training, the training data $train_w$, or the testing data $test_w$. Consequently, the validator cannot obtain $A_v(L_{wi})$ or $A_v(G_{i-1})$. Here, $A_v(L_{wi})$ denotes the validator's assessment of the accuracy and performance of the local model L_{wi}, while $A_v(G_{i-1})$ represents the validator's evaluation of the accuracy

and performance of the previous round's global model. To address this issue, this study introduces the validator v employing a proxy comparison method. Specifically, the validator compares the local model L_{wi} provided by the working client C_w on the validator client's testing dataset accuracy $A_{self}(L_{wi})$ with the previous round's global model G_{i-1} accuracy $A_{self}(G_{i-1})$ on the validator client's dataset. This comparison serves as a proxy, representing the precision comparison between the working client's model $A_w(L_{wi})$ and the previous round's global model $A_w(G_{i-1})$.

$$\begin{aligned} A_{self}(L_{wi}) - A_{self}(G_{i-1}) &> \text{Threshold} \Rightarrow \text{Vote} = \text{Negative} \\ A_{self}(L_{wi}) - A_{self}(G_{i-1}) &\leq \text{Threshold} \Rightarrow \text{Vote} = \text{Positive} \end{aligned} \tag{6}$$

In Equation (6), the Threshold represents a threshold used to determine the tolerance for testing accuracy decline, while Vote denotes the validator's vote, signifying their stance on the local model. If the decline in testing accuracy exceeds the threshold, the vote is against (Negative); otherwise, it is in favor (Positive). This proxy comparison method aids the validator in assessing the quality of the working client's model without directly accessing the training data on the working client's end, thus enhancing privacy and security assurance concerning data in federated learning.

3.4. Rotation of Roles Mechanism

To prevent malicious clients from continuously acquiring the same role, this study implements a rotation mechanism for roles. The clients engage in different role-playing through a rotation mechanism, participating in the learning training of each round. Through this role rotation approach, this study aims to achieve the following: (1) Ensure a reselection of miners in each training round, reducing the opportunity for malicious devices to repeatedly win and acquire the role of miner. (2) Enable each device to participate in training as a working device in multiple rounds, facilitating better model aggregation in federated learning. This enhances the accuracy of the global model and thereby improves the overall security and effectiveness of federated learning, ultimately serving the purpose of safeguarding data privacy.

3.5. Incentive Mechanism Based on Role Rewards

In order to encourage the active participation of nonmalicious clients in the training process of federated learning, this study introduces an incentive mechanism based on role rewards. The design of this incentive mechanism aims to promote full cooperation among participants in the decentralized federated learning on the blockchain. By rewarding the working clients, the system stimulates their desire to actively engage in model training, particularly through distillation defense rewards, ensuring the security and resilience of the model. Validators receive rewards by verifying the votes and transaction signatures of working clients, emphasizing the importance of validation and aiding in resisting manipulation. For miner clients, rewards are related to the number of valid transactions they validate, effectively reducing the threat of witch attacks. Through unit rewards and differentiated incentives, the system ensures fairness and balance, allowing each participant to receive appropriate rewards based on their contributions and tasks. This establishes an attractive, secure, and fair decentralized federated learning ecosystem based on blockchain.

3.5.1. Rewards for Working Clients

The rewards for the working clients engaged in model training consist of two parts: the reward obtained by submitting the local model L_{wi} and the distillation reward acquired by training with their own local model. Firstly, during the voting process, if the positive votes received by L_{wi} (denoted as $P(L_{wi})$) are greater than or equal to the negative votes (denoted as $N(L_{wi})$), then the participating working client C_w qualifies for a reward, as indicated by Equation (7):

$$C_{wi} = \begin{cases} e_{wi} \cdot \mid train_w \mid \cdot r & P(L_{wi}) \geq N(L_{wi}) \\ 0 & P(L_{wi}) < N(L_{wi}) \end{cases} \tag{7}$$

The reward is proportional to the number of data samples $|e_{wi}|$ in the local training process. Secondly, to incentivize the working client C_w engaged in model training to use distillation defense to enhance their local security and simultaneously improve training accuracy while reducing sensitivity to perturbations through distillation, this research introduces a distillation reward term Crs_i. This reward is associated with the effectiveness of C_w after distillation, ensuring that the working client selects an appropriate temperature during the distillation process that does not compromise training accuracy while increasing sensitivity to perturbations. The effectiveness of this distillation defense is measured by the difference between the testing accuracy $A_{self}(L_{wi})$ of the local model L_{wi} on the validation dataset and the testing accuracy $A_{self}(G_{i-1})$ of the previous round's global model G_{i-1} on the same validation dataset. The better the distillation effect, the more distillation reward the working client C_w receives, as depicted by Equation (8):

$$Crs_i = (A_{self}(L_{wi}) - A_{self}(G_{i-1})) \cdot \mid test_w \mid \cdot r_{self} \tag{8}$$

where $A_{self}(L_{wi})$ is the testing accuracy of the working client C_w using the local model L_{wi} after employing distillation defense on the validation dataset, $A_{self}(G_{i-1})$ is the testing accuracy of the previous round's global model G_{i-1} on the same validation dataset, $|test_w|$ represents the number of samples in the validation dataset of the working client C_w, and r_{self} denotes the unit reward for distillation defense. Each participating training working client C_w conducts distillation locally. Every working client involved in training uses its local model to generate soft targets. The specific process is as follows: The participating training working client uses its local data to train the original model θ_w, as depicted in Equation (9):

$$\theta_w = argminL(\theta_w, D_w) \tag{9}$$

Here, θ_w represents the original model parameters on the client C_w, and L denotes the loss function. The original model generates a probability vector as soft targets P used for training a smaller model M_w on the corresponding client. This smaller model shares the same network structure as the original model but has fewer parameters, allowing it to retain the knowledge of the original model while reducing computational resource requirements and sensitivity to disturbances. The process of generating soft targets P is shown in Equation (10):

$$p_w = Softmax(\theta_w / T) \tag{10}$$

Here, the Softmax function is used for normalization, providing the original probability distribution, and T is the temperature parameter utilized to regulate the smoothness of the soft target distribution. This study utilizes the output of the original model as soft targets, enabling the model to intricately learn relationships between different categories to defend against adversarial attacks and consequently enhance model performance. The process of training the smaller model φ_w is illustrated in Equation (11):

$$M_w = \text{argmin}\, L(\varphi_w, D_w, p_w) \tag{11}$$

Here, L represents the loss function, and p_w denotes the soft targets derived from the original model. In essence, the training of φ_w involves optimizing the loss function L to determine the parameters that result in the minimum loss on the client C_w's local dataset D_w under the softening target p_w.

The specifics of the temperature parameter adjustment process are exhibited in the experiment section. After the completion of training for each participating working client, they continue to engage in the decentralized federated learning iteration process, aiming to

enhance the model's robustness and reduce the risk of adversarial sample attacks through multiple iterations.

3.5.2. Rewards for Validators

The validator's reward consists of two parts: the validation reward $rv_{veri}i$ and the validation voting reward $rv_{vali}i$, as shown in Equation (12):

$$rv_i = \mid rv_{veri}i \mid \cdot re + \mid rv_{vali}i \mid \cdot re \tag{12}$$

For the validation reward $rv_{veri}i$, it is calculated based on the validator C_v's successful verification of the signed transactions tx_{wi}. If the validator successfully verifies tx_{wi}, they receive the validation reward. $\mid rv_{veri}i \mid$ represents the number of work transactions tx_{wi} successfully verified by the validator C_v, and re denotes the unit reward.

Regarding the validation voting reward $rv_{vali}i$, this reward is calculated based on the validator C_v's participation in assessing the votes generated by the working client C_w, denoted by $vtv(L_{wi})$. If the validator participates in the voting process based on the proxy comparison logic, they receive the validation voting reward. $\mid rv_{vali}i \mid$ represents the number of times the validator C_v has engaged in voting, specifically in assessing the votes generated by the working client $vtv(L_{wi})$.

The validator's reward mechanism primarily focuses on the validator client's duties in voting and verifying to uphold the security and legitimacy of the federated learning system. The rewards for validator clients are contingent on their level of activity and participation, ensuring their effective execution of validation tasks.

3.5.3. Rewards for Miners

The reward for the miner client is represented in Equation (13):

$$rCm_i = \mid tx_vj(L_{wi}) \mid \cdot re \tag{13}$$

Here, rCm_i denotes the reward for the miner client Cm_i in round r_i, and re represents the unit reward. The miner client's reward depends on the number of transactions $\mid tx_vj(L_{wi}) \mid$ it validates. This incentivizes miner clients to actively validate transactions related to the model updates L_{wi}.

In this research system, miner clients primarily serve the role of blockchain validation and propagation. They ensure the legitimacy of transactions by validating transaction signatures and packaging legitimate transactions into blocks. Their role is an integral part of maintaining the security and legitimacy of the system implemented in this research.

3.6. Distillation Defense Principle

This study employs distillation defense as a method to counter adversarial sample attacks. This section elaborates on the algorithm and theoretical justification of distillation defense. The algorithm is presented as shown in Algorithm 2.

The distillation defense method in this research is achieved by adjusting the temperature values of neural networks during training to enhance the model's defensibility. During the training phase, we set the model's temperature value relatively high, which smoothens the model's outputs, thereby reducing sensitivity to minor input variations and subsequently mitigating the impact of adversarial sample attacks. In the subsequent aggregation phase, we adjust the parameters appropriately. For the specific parameter adjustment process, please refer to the experimental section. Because the information from the smoothed outputs learned during training is embedded in the model parameters, even with temperature adjustments afterward, the model will maintain a certain level of robustness.

Next, this research theoretically demonstrates this process: Firstly, defining a Jacobian matrix considering the partial derivatives of the output vector $y = f(x)$ of the model f with respect to the input vector x, as shown in Equation (14):

$$J(x) = \frac{\partial y}{\partial x} \tag{14}$$

Algorithm 2: Temperature-Based Distillation Defense

Input: Training dataset (X, Y), Distillation temperature T_{train}
Output: Trained model parameters W
Initialization: Initialize model parameters: W;
Training Phase;
for *each epoch* **do**
  for *each batch* (X, Y) **do**
    Logits $Z = f_W(X_{\text{batch}})$;
    Softmax probabilities: $P = \text{softmax}\left(\frac{Z}{T_{\text{train}}}\right)$;
    Loss function $L = \text{CE}(P, Y_{\text{batch}})$;
    Update model parameters using backpropagation: $W \leftarrow W - \alpha \nabla_W L$, where α is the learning rate;
  end
end
Testing Phase;
$T_{\text{test}} = 1$;
Evaluate model performance using the test dataset;
  Compute logits Z using the model: $Z = f_W(X_{\text{test}})$;
  Compute softmax probabilities: $P = \text{softmax}\left(\frac{Z}{T_{\text{test}}}\right)$;
  Make predictions based on P;

Under high-temperature training, the model's output distribution becomes smoother, reflected by the model's probability distribution P. Under the influence of T_{train}, the output probabilities of the model become more uniform, reducing differences between output categories. Simultaneously, this smoothness affects the model's sensitivity to small changes in input, i.e., the change in output concerning the input. During high-temperature training, due to the smoother output distribution, the absolute values of elements in the Jacobian matrix decrease. As the elements of the Jacobian matrix describe the sensitivity of the model output concerning the input, the smoothening of the output distribution results in a weakened response to small changes in the input.

During adversarial sample attacks, the slight perturbations in input data are suppressed by the weakened sensitivity of the model. This is because the model has learned a smooth response to small input variations during training. The reduced sensitivity to minor input changes due to the stability of the output makes the model more robust against adversarial sample attacks that exhibit considerable jitter.

In the framework of differential privacy, this study introduces the definition of differential privacy for the model output $f(x)$ with respect to the input. For two adjacent inputs x and x', $\| x - x' \|_1 \leq \Delta$, we expect the difference between the outputs $f(x)$ and $f(x')$ to be bounded.

The formal definition of differential privacy can be expressed as

$$\Delta f(x) = \sup_{\|x - x'\|_1 \leq \Delta} \| f(x) - f(x') \| \leq \epsilon \tag{15}$$

In Equation (15), $\Delta f(x)$ represents the sensitivity of the model output to small changes in the input, Δ is the perturbation range of the input, and ϵ is the privacy parameter of differential privacy. A larger value of ϵ corresponds to a reduced sensitivity of the model to small changes in the input, resulting in more bounded differences in the output, thereby enhancing privacy protection.

Through the above formalization, this study theoretically proves that under high-temperature training, the sensitivity of the model's output to small changes in the input is protected by differential privacy. This protection makes the model more robust against adversarial sample attacks, as the sensitivity reduction of the model to small perturbations in input data suppresses the risk of privacy leakage.

4. Experiment

4.1. Experiment Setting

In order to ensure experiment reproducibility and consistency, this study utilized experimental code written in Python 3.6 and conducted simulated experiments in the Google Colab virtual environment. For the simulation experiments, this study selected the NVIDIA Tesla T4 GPU as the hardware device to ensure efficiency and precision in the experiments. The federated learning training environment of this study includes 20 simulated clients participating in the established federated learning framework. To ensure the even distribution of training data, the entire MNIST training dataset was randomly divided into parts equal to the number of clients, ensuring no overlap between these parts. This partitioning method helps simulate the heterogeneous data distribution in real federated learning scenarios.

During the experimental process, this study set a threshold value of 6. Specifically, if a client fails to reach this threshold within 6 communication rounds, this study considers it a potentially adversarial or low-quality client and excludes it in the next communication round. This mechanism aids in maintaining the stability and reliability of the federated learning system.

To validate the proposed method's resistance against targeted attacks, this study introduced targeted attacks on local clients in the experimental environment. This ensures that the proposed method not only performs well in normal training but also effectively defends against adversarial sample attacks. Through these experimental settings, this study aims to guarantee the authenticity and credibility of the experiments, simulating various scenarios in the experimental environment to comprehensively evaluate the performance of the proposed algorithm.

4.2. Results

4.2.1. Accuracy

Firstly, the experiment tested the method based on the distillation defense used in this study against the traditional FedAvg algorithm without malicious device intervention. This comparison was conducted in a scenario where no device introduced Gaussian noise during the aggregation process, and no external adversary influenced the federated learning process through adversarial sample attacks. In this study, accuracy is defined as the performance metric of the model on a specific task, typically represented by the proportion of correctly classified samples in a classification task. The experimental results in Figure 2 indicate that, before 30 rounds, the convergence rate of the model proposed in this study did not reach the same level as the FedAvg algorithm. This discrepancy can be attributed to the periodic distillation and knowledge transmission required in the model aggregation process employed in this study. Before 30 rounds, the knowledge obtained from these training sessions is not sufficiently diverse, and the coverage of clients is not comprehensive, leading to relatively poorer performance of the model in the initial stages. However, this value does not exceed 0.05. With the increase in training rounds, it can be observed that the model proposed in this study stabilizes in the later stages, and its accuracy surpasses that of the traditional FedAvg algorithm. This validates the superiority of our algorithm in long-term training and the effectiveness of the distillation defense introduced during the model aggregation process.

Next, within the same experimental setup, the experiment evaluated whether the distillation defense method used in this study could effectively defend against adversarial sample attacks targeting the client. The results from Figure 3 demonstrate that, with an

increase in client training rounds, the inclusion of the distillation defense process in this study did not impact the stability and convergence of the local client training model process; it converged as the training rounds progressed.

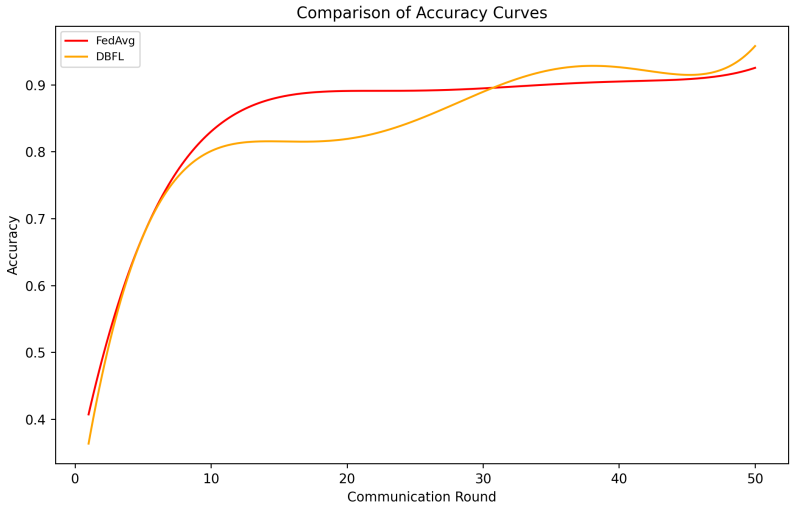


Figure 2. Comparison of accuracy between the system proposed in this study and traditional FedAvg algorithm in the absence of malicious device intervention.

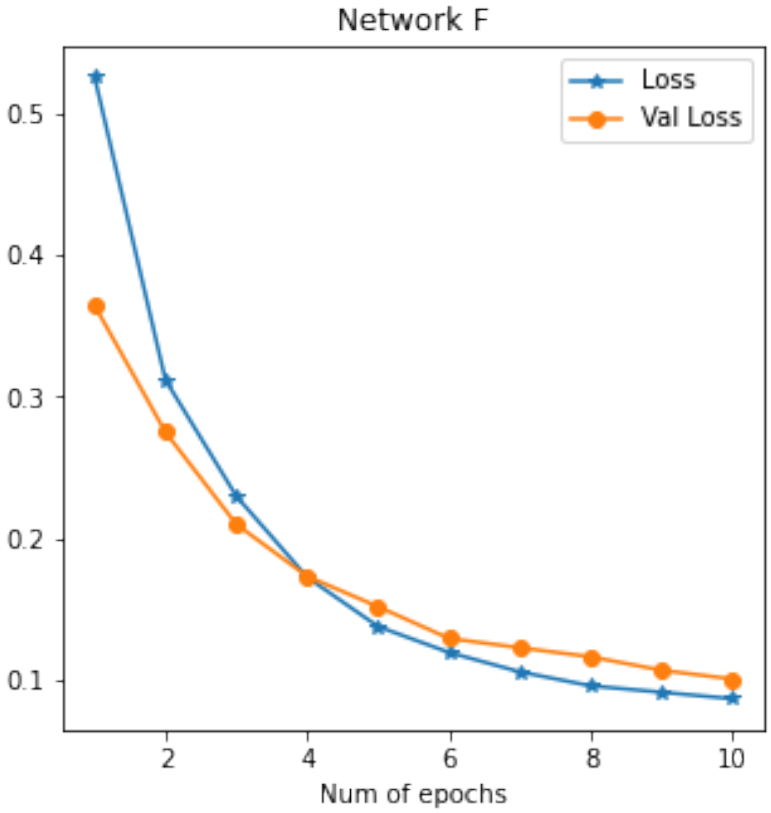


Figure 3. With the increase in rounds, there is a change in the training and validation losses of the working client models.

4.2.2. Adversarial Sample Attack Evaluation

Next, this paper assesses the robustness of the defense models proposed in this study through three types of attack experiments. The experiments employ the FGSM, IFGSM, and

MI-FGSM attack methods. The FGSM (Fast Gradient Sign Method) introduces perturbations to input data based on the gradient information of the model, maximizing the loss function to cause erroneous outputs from the model. The IFGSM (Iterative Fast Gradient Sign Method) generates adversarial samples through iterative adjustments applied to input data using the model's gradient information across multiple iterations, thereby increasing the model's error rate. The MI-FGSM (Momentum Iterative Fast Gradient Sign Method) is an advancement of the IFGSM that introduces a momentum term, enhancing the exploration of the adversarial space during iterations and making the attack more potent.

Figure 4 presents three sets of models subjected to three different attack methods from left to right. The top three graphs depict the results after employing the defense method, while the lower three graphs display the outcomes without utilizing the distillation defense. For clear trend visualization, the ordinate of the top three graphs is set above 0.880. From Figure 4, it is evident that employing the distillation defense method proposed in this study effectively counters adversarial sample attacks. In comparison with traditional federated learning without distillation defense, our proposed approach maintains the training accuracy consistently above 0.880 as the strength of adversarial samples intensifies. If the client side encounters stronger adversarial attacks, reaching a discernible level even to the naked eye where the perturbation is sufficiently large and significantly impacts the accuracy and security of the entire federated learning architecture and potentially leading to situations indicative of collusion with attackers, such occurrences will be identified during the model's secure aggregation process.

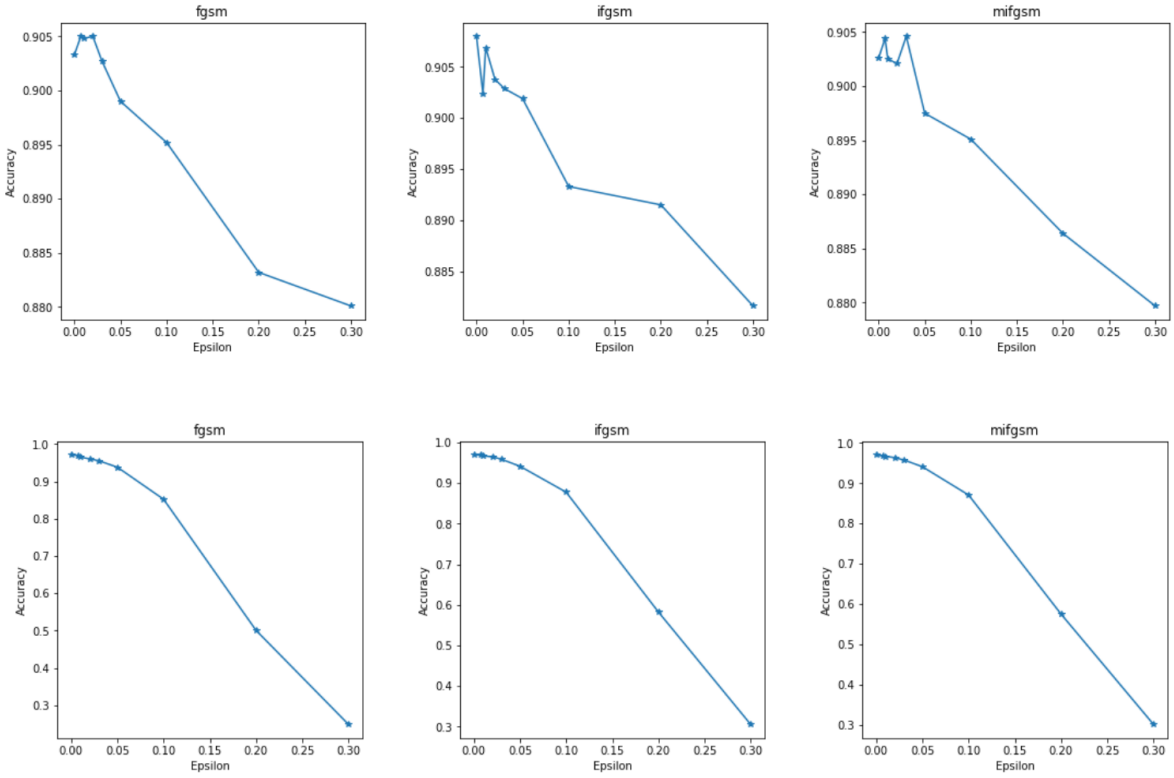


Figure 4. Assessment of client-side defense against adversarial sample attacks.

4.2.3. Black-Box Testing

The attacker possesses their dataset V and aims to detect whether the target model M leaks similar privacy information as in V. The attacker tests the learning capability of the target model M regarding privacy information in V by generating adversarial samples x_{adv}. The perturbation ϵ used in generating adversarial samples is kept as small as possible to avoid triggering the model's defense mechanisms.

$$x_{adv} = x + \epsilon \cdot \nabla_x J(\theta, x, y_{true}) \tag{16}$$

In Equation (16), the attacker utilizes auxiliary data from V to observe the performance of model M on V after the attack. The attacker uses Acc_{adv} to represent the postattack model's classification accuracy on V. If Acc_{adv} is significantly lower than expected, or the model produces incorrect classification results on V, the attacker can infer that the model has learned privacy features or attributes from V.

As shown in Figure 5, the model without distillation defense has a relatively low average accuracy before testing, ranging from 73% to 85%. This indicates that the model's performance is average when handling external data, and its recognition performance on this dataset is not stable. However, after testing, the model's average accuracy on specific data increases significantly to over 95%. This significant improvement indicates that the model has learned more information from the testing data, enabling it to detect privacy information similar to the data features, leading to privacy leakage.

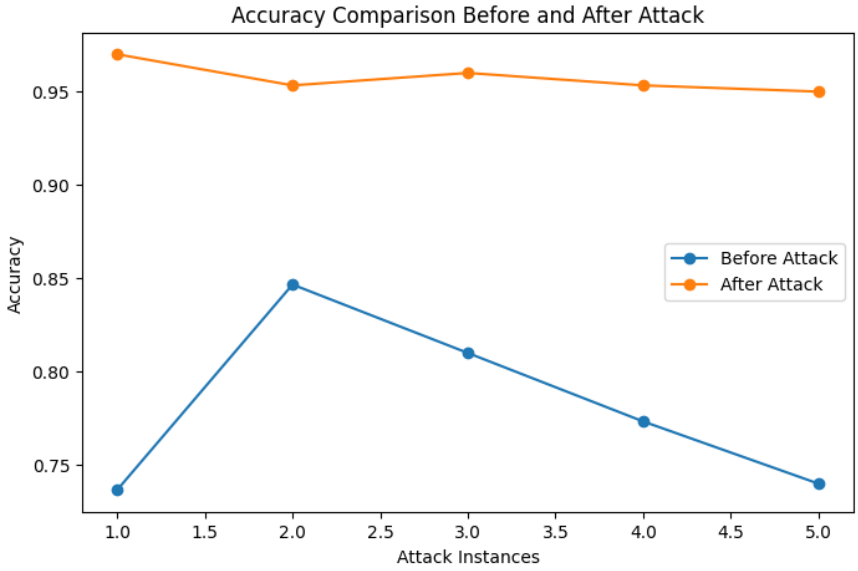


Figure 5. Accuracy comparison before and after model fusion with distillation defense.

This study integrated distillation defense locally on the client and conducted another round of experimental data testing. As shown in Table 1, after distillation, our data show no change in average accuracy. This indicates that our model can effectively handle interference from external data and does not expose privacy data similar to external data features during probing with similar external data.

Table 1. Privacy Leakage Detection Experiment Results.

Before Attack	After Attack
0.9599999785423279	0.9599999785423279
0.9800000190734863	0.9800000190734863
0.9750000385103486	0.9750000385103486
0.9700000286102295	0.9700000286102295
0.9800000190734863	0.9800000190734863

4.2.4. Security Evaluation of Models

Finally, to approximate realistic training effects, the experiment adjusted parameters throughout the entire model training process. The temperature parameter is employed to control the smoothness of soft labels. A higher temperature in the experiment makes the probability distribution of soft labels smoother, enabling the model to more easily capture

features of adversarial sample attacks. The weight parameter is adjusted to balance the loss from hard labels and soft labels, maintaining the robustness of knowledge distillation against adversarial samples. Parameter tuning ensures that the model can learn correct classification information from hard labels and acquire knowledge from the soft labels of the teacher model. Throughout this process, the experiment continued to use the aforementioned experimental environment, assuming that 15% of the clients had either been compromised or were colluding.

The experimental results in Figure 6 indicate that with a temperature of 0.1 and a weight of 0.3, the proposed algorithm in this paper outperforms the Federated Average (FedAvg) algorithm. This is mainly because the experiment used a lower temperature to prevent overfitting the model to the training data. At this temperature, the model focuses more on details and enhances its generalization ability, which is then aggregated into the global model. This federated learning approach based on distillation defense boosts the model's confidence, favoring high-confidence predictions. These highly confident predictions, when merged, ensure the final predictions of the model are more reliable and accurate. Hence, under these temperature and weight conditions, the experiment yields better results. However, it is important to note that the chosen parameters in this study are a balance between convergence speed and process. As seen in the graph, due to considerations of compromised clients during convergence, the proposed model as a whole requires multiple rounds of training to reach convergence. Therefore, fluctuations occur during convergence due to the randomness in role assignments and uncertainty regarding the allocation of roles to malicious clients. Nonetheless, these fluctuations tend to diminish as the training progresses steadily because the system identifies and screens out malicious clients involved in the training process, as proposed in this study.

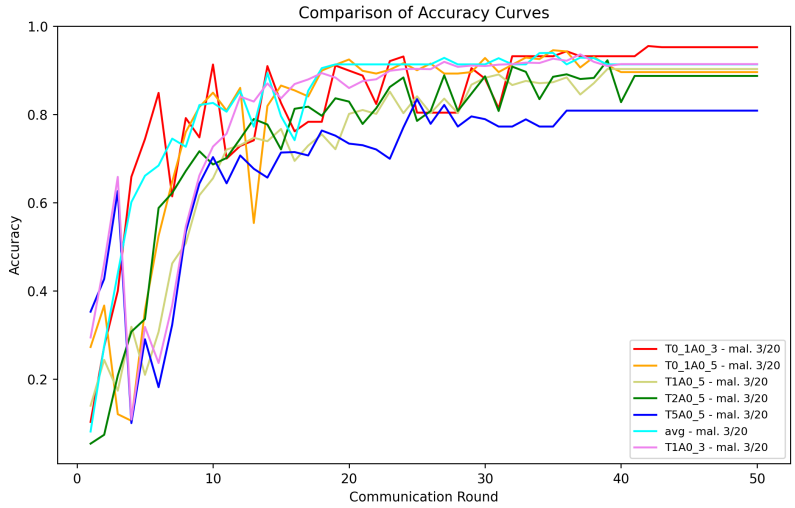


Figure 6. The tuning process of distillation defense and decentralized federated learning systems.

5. Conclusions

This study proposes a research approach that combines distillation defense with blockchain to enhance privacy in federated learning. Considering the susceptibility of traditional federated learning to poisoning attacks from malicious clients, this research integrates blockchain technology into federated learning. It establishes a decentralized blockchain maintained collectively by all clients to decentralize federated learning. Employing a distributed rotating validation mechanism, this study verifies the credibility of local model updates. This method helps prevent compromised or malicious clients from influencing the final global model aggregation due to repeated role-playing, ensuring optimal outcomes. Given the possibility of adversarial testing attacks on local clients during federated learning, resulting in privacy breaches of local client models, this study employs

the distillation defense method at the local client level. It addresses potential issues with malicious distortion by clients during local model updates or the introduction of noise due to damaged local clients during data training. Additionally, it mitigates the impact of adversarial sample attacks on local clients. Such attacks are attempts by adversaries to deceive the federated learning system proposed in this study through minute attacks, potentially leading to privacy breaches.

Finally, it should be noted that this study's model operates in a simulated environment. To comprehensively enhance the performance of the model proposed in this study, future research efforts should focus on developing more efficient blockchain data structures, chain synchronization algorithms, and fault-tolerant mechanisms. This will facilitate the testing of the feasibility, security, and reliability of the model methods proposed in this study in practical distributed system tests.

Author Contributions: Conceptualization, Y.W. (Ying Wang) and J.X.; Investigation, T.Z. and Y.W. (Yulong Wang); Data curation, J.W.; Writing—original draft, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Grant No. 61872138) and the Fujian Provincial Natural Science Foundation Project (Grant No. 2023J011800).

Data Availability Statement: The data for this study can be obtained through the corresponding author.

Acknowledgments: Special gratitude is extended to all authors for their invaluable insights contributed to this manuscript. We would also like to express our appreciation to the institution and laboratory members for their assistance with this article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
2. Cai, J.; Liang, W.; Li, X.; Li, K.; Gui, Z.; Khan, M.K. Gtxchain: A secure iot smart blockchain architecture based on graph neural network. *IEEE Internet Things J.* **2023**, *10*, 21502–21514. [CrossRef]
3. Qammar, A.; Karim, A.; Ning, H.; Ding, J. Securing federated learning with blockchain: A systematic literature review. *Artif. Intell. Rev.* **2023**, *56*, 3951–3985. [CrossRef] [PubMed]
4. Shen, T.; Zhang, J.; Jia, X.; Zhang, F.; Huang, G.; Zhou, P.; Kuang, K.; Wu, F.; Wu, C. Federated mutual learning. *arXiv* **2020**, arXiv:2006.16765.
5. Liang, W.; Xie, S.; Cai, J.; Xu, J.; Hu, Y.; Xu, Y.; Qiu, M. Deep Neural Network Security Collaborative Filtering Scheme for Service Recommendation in Intelligent Cyber-Physical Systems. *IEEE Internet Things J.* **2022**, *9*, 22123–22132. [CrossRef]
6. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
7. Liang, W.; Li, Y.; Xie, K.; Zhang, D.; Li, Ku.; Souri, A.; Li, K. Spatial-Temporal Aware Inductive Graph Neural Network for C-ITS Data Recovery. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 8431–8442. . [CrossRef]
8. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 23–25 May 2016; pp. 582–597.
9. Lin, T.; Kong, L.J.; Stich, S.U.; Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Curran Associates Incorporated: New York, NY, USA, 2020; pp. 2351–2363.
10. Divi, S.; Farrukh, H.; Celik, B. Unifying distillation with personalization in federated learning. *arXiv* **2021**, arXiv:2105.15191.
11. Gong, X.; Sharma, A.; Karanam, S.; Wu, Z.; Chen, T.; Doermann, D.; Innanje, A. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 11891–11899. [CrossRef]
12. Chen, H.Y.; Chao, W.L. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv* **2020**, arXiv:2009.01974.
13. Majeed, U.; Hong, C.S. FLchain: Federated learning via MEC-enabled blockchain network. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–4.

14. Awan, S.; Li, F.; Luo, B.; Liu, M. Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2561–2563.
15. Jin, H.; Dai, X.; Xiao, J.; Li, B.; Li, H.; Zhang, Y. Cross-cluster federated learning and blockchain for internet of medical things. *IEEE Internet Things J.* **2021**, *8*, 15776–15784. [CrossRef]
16. Qu, Y.; Gao, L.; Luan, T.H.; Xiang, Y.; Yu, S.; Li, B.; Zheng, G. Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet Things J.* **2020**, *7*, 5171–5183. [CrossRef]
17. Kim, H.; Park, J.; Bennis, M.; Kim, S.-L. Blockchained on-device federated learning. *IEEE Commun. Lett.* **2019**, *24*, 1279–1283. [CrossRef]
18. Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D.; Li, Z.; Lyu, L.; Liu, Y. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet Things J.* **2020**, *8*, 1817–1829. [CrossRef]
19. 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; pp. 50–57.
20. Liang, W.; Yang, Y.; Yang, C.; Hu, Y.; Xie, S.; Li, Ku.; Cao, J. PDPChain: A Consortium Blockchain-based Privacy Protection Scheme for Personal Data. *IEEE Trans. Reliab.* **2023**, *72*, 586–598. [CrossRef]
21. Xiong, X.; Zhuo, T.; Bin, X.; Li, K. A Survey on Privacy and Security Issue in Federated Learning. *J. Comput. Sci. Technol.* **2023**, *46*, 1019–1044. [CrossRef]
22. Wu, X.; Wang, Z.; Zhao, J.; Zhang, Y.; Wu, Y. FedBC: Blockchain-based decentralized federated learning. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 27–29 June 2020; pp. 217–221.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.