

Article

QuoTa: An Online Quality-Aware Incentive Mechanism for Fast Federated Learning

Hui Cai ¹, Chao Bian ¹, Biyun Sheng ¹, Jian Zhou ^{1,*}, Juan Li ² and Xin He ¹

¹ College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; carolinecai@njupt.edu.cn (H.C.); 1222046134@njupt.edu.cn (C.B.); biyunsheng@njupt.edu.cn (B.S.); hexin@njupt.edu.cn (X.H.)

² College of Computer, Nanjing University of Aeronautics and Astronautics, Nanjing 210003, China; juanli@nuaa.edu.cn

* Correspondence: zhoujian@njupt.edu.cn

Abstract: In addition to federated optimization, more current studies focus on incentive mechanism design problems for federated learning (FL), stimulating data owners to share their resources securely. Most existing works only considered data quantity but neglected other key factors like data quality and training time prediction. In combination with all the above factors, we proposed an online quality-aware incentive mechanism based on multi-dimensional reverse auction, QuoTa, for achieving fast FL. In particular, it first designs model quality detection to eliminate some malicious or dispensable devices based on their historical behaviors and marginal contributions. Due to the possible fluctuations of CPU frequency in realistic model training, it next predicts model training time based on the upper confidence bound algorithm. By combining the two modules, QuoTa incentivizes data owners with high data quality, high computing capability, and low cost to participate in the FL process. By rigorous theoretical proof and extensive experiments, we prove that QuoTa satisfies all desired economic properties and achieves higher model accuracy and less convergence time than the state-of-the-art work.

Keywords: incentive mechanism; federated learning; multi-dimensional reverse auction



Citation: Cai, H.; Bian, C.; Sheng, B.; Zhou, J.; Li, J.; He, X. QuoTa: An Online Quality-Aware Incentive Mechanism for Fast Federated Learning. *Appl. Sci.* **2024**, *14*, 833. <https://doi.org/10.3390/app14020833>

Academic Editors: Pan Zhou and Xiaofeng Ding

Received: 12 December 2023

Revised: 10 January 2024

Accepted: 15 January 2024

Published: 18 January 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

Federated learning (FL) has become a promising data circulation technique to break data isolation and further promote secure data sharing, in consideration of raw data stored in each data owner's local device rather than a centralized data server [1–4]. However, data owners' computing and communication resources have to be compromised during the FL training process, and thus they usually have no incentive to participate in the training process unless sufficient monetary compensation in return. Insufficient data owners' training data hinders the aggregated model accuracy, thus failing the whole FL training. Thus, how to incentivize abundant data owners' participation is a key problem in FL.

Most prior works [1,3] adopted a multi-dimensional reverse auction to incentivize participants and facilitate FL, where the central server is the buyer, and data owners are reversely considered as the sellers to trade the use of multiple resources by a bid to reflect the minimum acceptable payment once chosen as a winner. However, a recent work [5] selected data owners only by their claimed data quantity or privacy budget, but neglected some key factors like data quality and device training time. Although other few works [1,2,6] also measured data quality, they designed quality detection according to data distribution similarity or aggregated reputations and thus cannot be applied to our scenario directly. More importantly, they failed to choose data owners to achieve fast federated learning. To facilitate model training, Peng et al. [7] also considered model convergence speed, and they failed to achieve a satisfying training time guarantee.

To address the above limitations simultaneously, it is nontrivial to design an efficient incentive mechanism to select cost-effective data owners with high computing capability under a limited budget. However, three key challenges have to be addressed. The primary challenge is to quantify each data owner's contribution before the current FL training round. Wang et al. [1] measured data owners' quality according to their claimed data amount, privacy budget, and data distribution. Zhang et al. [2] aggregated weighted reputation scores from multiple data requesters who ever selected them as a winner as these data owner's quality indicator. When the expected data distribution is unknown, or there are still no other data requesters who ever recruited him, their solution cannot work in our case. The next challenge is how to predict model training time. The CPU frequency cannot usually reach the data owner's claimed value due to the fluctuation in an actual working environment. The actual low CPU frequency degrades the convergence time of the global model. Thus, it is essential to predict each data owner's actual CPU frequency used for FL training. The last but not least challenge is how to prevent data owners from misreporting their multiple-dimensional bids for higher profit. A strategic data owner probably submits a false bid to manipulate the market, thus leading to the failure of the FL process.

In response to the above challenges, we propose a quality-aware incentive mechanism, named QuoTa, for fast federated learning. It motivates data owners with high data quality, high computing capability, and low cost to participate in FL training. In particular, we first design a quality detection module to choose data owners with both large contribution and marginal contribution for the global model. Then, QuoTa exploited a training time prediction module to estimate each data owner's computing capability at the current round, in order to achieve faster convergence. Combined with these two modules, QuoTa propose an efficient data owner selection and payment strategy to achieve all desired economic properties.

We highlight main contributions as follows:

- We design a quality-aware incentive mechanism to achieve fast-federated learning. To the best of our knowledge, QuoTa is the first work to take data quality and model training time prediction into consideration.
- To mitigate negative impacts on global model accuracy by low-quality data owners, we propose a model quality detection module to update their quality indicator and eliminate malicious data owners. Moreover, to predict each data owner's actual training time, we propose reinforcement learning-based methods to estimate their actual CPU frequency at the current round, in order to achieve faster convergence.
- Through extensive evaluations and rigorous theoretical analysis, QuoTa improves model accuracy over the state-of-the-art and achieves all desired economic properties like individual rationality, truthfulness, and budget balance.

2. Related Work

Federated learning [8] was first proposed by Google in 2016, based on the new paradigm of data immobilization but model moving, which effectively guarantees data privacy of edge data owners. For distributed computing, different data distributions will greatly affect model aggregation. McMahan et al. [9] subsequently proposed a FedAvg algorithm with better robustness to improve the model aggregation effect by randomly selecting and averaging the aggregation method for non-Independent Identically Distribution (non-IID) data in federated learning. In addition, Zhao et al. [10] proposed a method to share a small portion of training data through global sharing and the definition of weight scatter, which refers to the distribution deviation of local model weights towards the local data due to the differences in the distribution of training data. To counteract this weight deviation, Wang et al. [11] proposed a reinforcement learning-based control framework FAVOR, which is realized by intelligently selecting data owner sequences in each communication round, with the benefit of accelerating model convergence.

Similar to the barrel effect, the convergence speed of global model in federated learning is greatly determined by the data owner with the slowest training time. Improper data

owner selection will greatly increase the global aggregation time. Chai et al. [12] proposed a data owner selection strategy based on dividing the training time hierarchy to cope with the 'slow' participants. To maximize the convergence speed of global model, Peng et al. [7] ensured holding a saturated amount of training data while minimizing the total energy consumption of the participant. Similarly, Lee et al. [13] proposed a Data Distribution Aware Online data owner Selection (DOCS) algorithm to minimize the convergence time while ensuring high learning accuracy. Different from the above work, considering the increasingly powerful mobile processors, the main factor influencing the data owner's training speed is the CPU frequency. In practice, the frequency of model training fluctuates so that it cannot reach the rated value. Our work proposes a training time prediction module to solve the above problem.

To cope with possible backdoor attacks, gradient attacks, and model poisoning attacks in federated learning, Fang et al. [14] first systematically investigated the impact of localized model poisoning attacks on federated learning. Then, Zhang et al. [15] proposed the FLDetector method to detect malicious data owners by comparing the consistency of data owners' model updates over multiple iterations through the observed differences in model updates of the attacked model. To prevent attacks, Gupta et al. [16] proposed MUD-HoG, a malicious data owner attack defense algorithm based on the long-short history gradient, which can differentially treat detected malicious and low-reputation data owners and identify whether there was a targeted attack from malicious data owners. Jebreel et al. [17] proposed a lightweight protocol that allows participants to privately exchange and mix their updated encrypted fragments, achieving client-side privacy protection, and designed a reputation-based defense measure to achieve security.

Proper incentives can motivate data owners to actively participate in model training in a federated system. Kang et al. [4] improved the incentive effect for highly reputable mobile devices with high-quality data by effectively combining reputation and contract theory, whereas Khan et al. [18] introduced the Stackelberg game to participate in the incentives. Liu et al. [19] measured the marginal contribution based on the data owner's local model. However, the existing scenarios cannot satisfy the case where the data owner's cost is unknown, and, once the payment is lower than the expected, the data owner's motivation will be greatly undermined. Based on this, Zeng et al. [3] proposed an incentive mechanism FMore with multi-dimensional procurement auctions with K winners for federated learning. They exploited Nash equilibrium strategies for edge nodes and achieved data owner payment guarantee and maximization of aggregation utility. Ying et al. [20] combined differential privacy algorithms with reverse auctions to ensure the authenticity of the bids with a security guarantee. Unfortunately, it is possible that malicious data owners participated in the model training while their resources did not match the claimed quality. Zhang et al. [2] then added model quality detection to the reverse auction and a reputation management mechanism that combined the blockchain to store data owners' reputations. Domingo et al. [21] proposed a decentralized computing protocol based on common utility, incentivizing nodes to correctly perform computing tasks through a reputation system. However, most of the existing federated learning and auction mechanisms do not take into account the scenario where data owners lie about resource quality, use unclaimed resources for model training, and even malicious data owners may destroy the aggregation of the global model. To address this problem, we will propose a model quality detection module and update the reputation value of the data owner in real-time based on the detection results in the previous communication round.

Existing works have provided certain solutions to the problems of aggregation impact due to non-independent and homogeneous distribution of data, defense against malicious attacks, and federated optimization. In this paper, we will combine the above problems and propose a comprehensive data owner selection strategy that guarantees high accuracy and fast convergence time, motivates data owner participation based on the multi-dimensional reverse auction mechanism, and ensures high efficiency for the FL system.

3. System Model and Problem Formulation

In this paper, we consider QuoTa consists of $\mathcal{N} = \{1, 2, \dots, N\}$ data owners and a server platform (due to privacy concerns, only data parameters will be sent to the platform rather than raw data from data owners. Moreover, the attack for data parameters is not considered in this paper, and belongs to our future work). The platform selects data owners to participate in FL training based on their quality indicators and estimated training time. As data owners' computing or communication resources have to be compromised once they are chosen to participate in FL training, they have to be compensated reasonably. Without loss of generality, we adopt a multi-dimensional reverse auction to incentivize data owners' participation.

Once a new data owner arrives, he must first register their personal information (the information includes their personal account and hardware information of devices (such as rated CPU frequency, etc.), and it will not change until the whole FL training ends.) on the platform. Before each FL training round begins, the server will first publish task specifications to each data owner, which mainly contain related data specifications, data amounts, required CPU frequencies, etc. Then, the data owner makes a bid according to their current spare resources. In addition, we consider there are probably malicious data owners (without loss of generality, we only consider the multi-label flipping attack in this paper, where all their data labels in the whole datasets will be flipped uniformly from one class to another for each malicious data owner. The single label-flipping attack is beyond the scope of this paper, and belongs to our future work.) who deliberately hamper global model accuracy, and thus they have to be eliminated. Another concern is that we consider that data owners with lower computing capabilities will affect the convergence speed of the model, and the CPU frequency usually fluctuates, making it impossible to achieve the claimed rated value. Thus, some data owners with lower actual computing capability should also be removed.

3.1. QuoTa Framework

Different from existing work, QuoTa is equipped with both a quality detection module and a CPU frequency prediction module to mitigate the potential impact of malicious data owners and CPU frequency fluctuations on model convergence, respectively. The framework of QuoTa is given in Figure 1, as follows.

- Step 1: The platform server broadcasts task specifications to each data owner, including (1) Data specifications: required for model training, such as pictures of cars, dogs, cats, birds, etc. Although the data provided by data owners are usually non-IID in a real scenario, the platform wants to find data owners whose data are equally distributed as far as possible for higher model accuracy (In general, the higher the degree of non-IID, the worse the quality of local model. Therefore, in model quality detection, for data owners with a high degree of non-IID, we will punish them and reduce their weight in model aggregation. More details will be shown in Section 4.1). (2) The bidding format that the data owner should submit to the server: the data owner's quotation, the data amount, and the rated CPU frequency. Note that the rated CPU frequency is known at the time of registration.
- Step 2: Each data owner submits a bid based on the bidding format and their spare resources. The bid submitted by a data owner i is: $\mathcal{B}_i = \{b_i, d_i, fre_i\}$, where b_i is their claimed cost including computing and communication cost, d_i is their own data amount, and fre_i is their claimed rated CPU frequency.
- Step 3: Based on the received bidding information, the server selects winning a data owner's set within a limited budget, B^t , based on their current reputation value and estimated CPU frequency f_i and then delivers the global model to selected data owners.
- Step 4: Each selected data owner uses its local resources to train the local model.
- Step 5: Each selected data owner uploads its local model parameter to the server.

- Step 6: The server performs model quality detection on the uploaded local model, updates each data owner’s reputation value, and uses reinforcement learning methods to estimate the CPU frequency of each data owner.
- Step 7: The server assigns different weights to different local models based on each data owner’s quality indicator and then aggregates them into a global model.
- Step 8: The server pays reasonable monetary payment $p_i(\mathcal{B})$ to each selected data owner i .

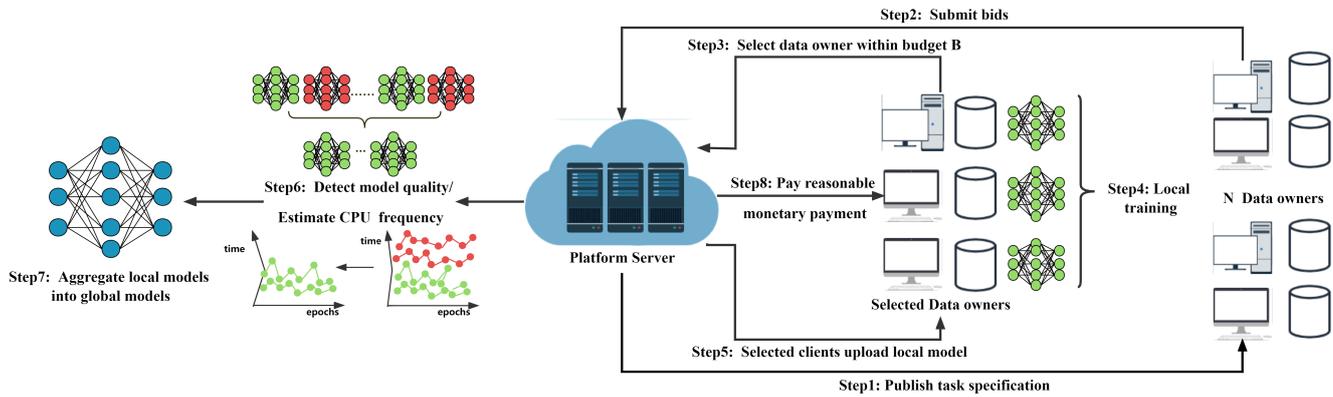


Figure 1. Framework of the proposed incentive mechanism QuoTa.

Let c_i denote each data owner’s real cost. Thus, their utility is defined as $u_i = p_i(\mathcal{B}) - c_i$. The main notations are shown in Table 1. Next, we introduce data amount model and computing capability model, which affect the global model accuracy and model convergence time, respectively.

Table 1. Key Notations.

Notation	Description
d_i^t	The amount of data bid by data owner i at round t
x_i^t	Whether data owner i is selected at round t
b_i^t	Data owner i ’s bid price at round t
B^t	Budget at round t
D^t	The maximum data amount that the server can select within the budget at round t
f_i^t	CPU frequency of data owner i at round t
C_i^t	The computing capability of data owner i at round t
GA^t	Global model accuracy at round t
LA_i^t	Local model accuracy for data owner i at round t
GA_{-i}^t	Global model accuracy after removing data owner i
S^t	The set of selected data owners at round t
Q^t	The maximum resource quality that the server can select within the budget at round t
rep_i^t	The direct reputation of data owner i at round t
Rep_i^t	The overall reputation of data owner i in previous rounds t

3.2. Data Amount Model

Inspired by previous work [22], we find the data amount in bidding information is one significant factor for model accuracy, assuming that each data owner reports their data amount truthfully. In particular, when the total amount D of training data in the federated system is larger, the global model accuracy GA is higher. In addition, more training data d_i provided by each individual data owner i would produce a higher aggregated global model accuracy. Thus, to obtain a higher global model accuracy, the server tries to select the data owner set with both larger local data amount of data owners and a larger total data amount with a limited budget, B . However, these two goals may not be achieved

simultaneously. The main idea in this paper is to choose data owners with as a large local data amount as possible while the total amount of training data reaches a given threshold. More details will be shown in Section 3.4.

3.3. Computing Capability Model

In consideration of high computing capability of the server, the model convergence time usually depends on the data owner with the lowest computing capability in the FL system. In order to quantify each data owner's training speed, computing capability C_i is defined as the estimated CPU frequency f_i divided by the data amount d_i in Formula (1):

$$C_i = \frac{f_i}{d_i}. \quad (1)$$

In order to ensure acceptable convergence time of the FL system, we restrict data owners with slower training speeds from participating in model training. Specifically, we set a computing capability threshold below which data owners will not be selected, in order to prevent them from slowing down model convergence time. Assuming that most data owners in the FL system have fair computing capability, the threshold can be set by a heuristic approach. For example, we sort all data owners' computing capability values in the decreasing order then take the k -th value in the sequence as the threshold: C_{low} .

The real CPU frequency for model training may fluctuate in a long period due to several factors, which cannot usually reach the claimed rated value. The possible reason is a too high CPU temperature or other concurrent program process. Thus, how to predict the real CPU frequency is one important problem. In consideration of low privacy sensitivity of CPU frequency, in this paper, a CPU cycle meter is implanted on each data owner's device, in order to detect their real required number of CPU cycle for training one local data sample [7], which is denoted as $cycle_i$. Thus, the required training time for all local training data are represented as $T_{loc_i} = (cycle_i * d_i) / \hat{f}_i$, where d_i is local data volume, and \hat{f}_i is their actual CPU frequency. Thus, once the data owner i completes model training at round t , the least number of CPU cycles $cycle_i^t$ and local training time $T_{loc_i}^t$ at round t can be measured. Then, their real CPU frequency at this round is computed as:

$$\hat{f}_i^t = \frac{cycle_i^t * d_i}{T_{loc_i}^t}. \quad (2)$$

However, according to the above Equation (2), we can only compute the value \hat{f}_i^t until the round t ends, which cannot thus be predicted in advance. Consequently, we first use the average CPU frequency \bar{f}_i^t in the previous t rounds to reflect the expected CPU frequency after the round t ends. Thus, the estimated CPU frequency f_i^t at round t is computed as:

$$f_i^t = \bar{f}_i^t = \frac{\sum_{\tau=0}^t \hat{f}_i^\tau * x_i^\tau}{n_i^t} \quad (3)$$

$$n_i^t = \sum_{\tau=0}^t x_i^\tau, \quad (4)$$

where n_i^t represents the number of times the data owner i was selected in the previous t rounds. By exploiting the above expected CPU frequency, we measure each data owner's computing capability as an important indicator of data owner selection in order to achieve faster model convergence. Although the claimed CPU frequency in bidding information is no longer used for a selection indicator once the real CPU frequency is computed after several rounds, it plays a part in winner selection at first. More details will be shown in Section 4.2.

3.4. Problem Formulation

As mentioned above, on the one hand, to guarantee a high global model accuracy, we select winning data owners with large claimed data volume in bidding information. On the other hand, to obtain faster model convergence time, we choose them according to data owners' estimated computing capability.

In this paper, we aim at designing a quality-aware incentive mechanism for achieving fast federated learning. Our objective is to maximize total data volume of selected data owners for any round t with limited budget and computing capability constraint, as shown in Equation (5):

$$\max \sum_{i=1}^n d_i^t * x_i^t \tag{5}$$

$$\text{s.t. } x_i^t \in \{0, 1\}, \forall i \in \mathcal{N} \tag{5a}$$

$$\sum_{i=1}^n b_i^t * x_i^t \leq B^t \tag{5b}$$

$$C_i^t * x_i^t \geq C_{low}, \forall i \in \mathcal{N}, \tag{5c}$$

where the constraint (5a) represents whether the data owner i is selected at round t , (5b) is budget constraint, and (5c) means that any winning data owner should have the least computing capability C_{low} .

In addition to the above objective function, we also need to select data owners with as large local data volume as possible. Thus, the objective can be rewritten as follows:

$$\min \sum_{i=1}^n x_i^t \tag{6}$$

$$\text{s.t. } (5a)(5b) \tag{6a}$$

$$\sum_{i=1}^n d_i^t * x_i^t \geq \alpha * D_t \tag{6b}$$

$$C_i^t * x_i^t \geq C_{low}, \forall i \in \mathcal{N}, \tag{6c}$$

where the total amount of selected data owners' training data at round t should exceed a given threshold, $\alpha * D_t$ (the threshold value is set as a fraction of the objective function value in Formula (5). D_t is the maximum total amount of selected data owners with a given budget and computing capability constraint before round t starts. $\alpha \in (0, 1)$ is a given parameter.) in (6b). The new objective function is to minimize the number of winning data owners with the maximum total data amount constraint in (6b). The transformation of the objective function is reasonable because we can guarantee each selected data owner with a large local data volume by minimizing the number of winning data owners under a given total amount of training data.

By solving the above optimization problem, we achieve a good balance between model accuracy and convergence time. In addition to that, the proposed incentive mechanism should also achieve the following properties:

- **Budget Balance:** The winning data owners' total payment $p_i(\mathcal{B})$ should not exceed the monetary budget B^t , i.e., $\sum_i p_i(\mathcal{B}) \leq B^t$.
- **Truthfulness:** Each data owner cannot obtain a higher utility by misreporting a false bid $\tilde{\mathcal{B}}_i$, i.e., $u(\mathcal{B}_i, \mathbf{B}_{-i}) \geq u(\tilde{\mathcal{B}}_i, \mathbf{B}_{-i})$, where $\mathcal{B}_i = \{b_i, d_i, fre_i\}$ is their true user bid, $\tilde{\mathcal{B}}_i$ is the false one, and \mathbf{B}_{-i} is the set of bids except \mathcal{B}_i .
- **Individual Rationality:** Any data owner i has non-negative utility for the truthful bid, i.e., $u_i(\mathcal{B}) \geq 0$.

4. Design of QuoTa

The proposed mechanism QuoTa consists of four key components, i.e., model quality detection, model training time prediction, data owner selection, and payment decision. In particular, model quality detection and training time prediction help select the winning data owner set.

4.1. Model Quality Detection

Data owners may use unclaimed data resources for model training and even make a malicious attack, thus affecting global model aggregation. However, due to privacy concerns, the server cannot access each data owner’s original data and thus fails to verify whether he uses the claimed training data. Instead, only the local model will be sent to the server after each FL round ends. Thus, we designed a model quality detection method for quantifying the quality of local models uploaded by data owners.

After the t -th round of communication ends, the server will use the local test set to perform accuracy detection for the local model submitted by data owner i and use Formula (7) to update the quality detection value for each data owner.

$$v_i^t = [(GA^t - LA_i^t) - \epsilon_1^t] - [(GA^t - GA_{-i}^t) - \epsilon_2^t], \tag{7}$$

where $GA^t - LA_i^t$ measures the difference between the data owner i ’s local model accuracy LA_i^t and the global model accuracy GA^t . The smaller above difference reflects their larger contribution for FL aggregation. In addition to that, $GA^t - GA_{-i}^t$ means the difference between two global model accuracy values with or without these data owner i , which quantifies their marginal distribution. Conversely, the smaller difference means their smaller marginal distribution. Thus, the former difference minus the latter one is used to quantify the whole difference in terms of both direct and marginal distribution.

Moreover, we compare the above two differences with two well-designed threshold values, i.e., ϵ_1^t and ϵ_2^t , respectively. Specifically, the thresholds ϵ_1^t and ϵ_2^t are determined by Formulas (8) and (9), which take the average of all the differences.

$$\epsilon_1^t = \frac{\sum_{i \in S^t} (GA^t - LA_i^t)}{|S^t|} \tag{8}$$

$$\epsilon_2^t = \frac{\sum_{i \in S^t} (GA^t - GA_{-i}^t)}{|S^t|} \tag{9}$$

Consequently, the above quality detection value can be used for quantifying each data owner’s direct quality at current round t . When v_i^t is positive, it means this data owner performs poorly at round t , and we subtract a penalty term from the previous comprehensive reputation value Rep_i^{t-1} in Formula (10). On the contrary, when v_i^t is negative, it means they perform better, and thus we exploit a reward item to be added to Rep_i^{t-1} in Formula (11).

$$rep_i^t = Rep_i^{t-1} - v_i^t \tag{10}$$

$$rep_i^t = Rep_i^{t-1} + \frac{(-v_i^t)}{2}, \tag{11}$$

where Rep_i^{t-1} represents their comprehensive reputation value of the data owner i in the past $t - 1$ rounds of communication, and rep_i^t means the direct reputation value at current t -th round. Specifically, Rep_i^t is updated according to Formula (12) (Here, we exploit an exponential moving average of data owner i ’s reputation value to update their

comprehensive reputation value considering that the reputation value closer to current round reflects their real model quality better).

$$Rep_i^t = \begin{cases} Rep_i^{t-1}, & \text{if } x_i^t = 0 \\ \lambda * Rep_i^{t-1} + (1 - \lambda) * rep_i^t, & \text{if } x_i^t = 1 \end{cases} \tag{12}$$

where Rep_i^t is the exponential moving average about rep_i^t within $[1, t]$, and λ is a decay factor. When the data owner is selected at this round, their comprehensive reputation will be updated. Otherwise, it remains unchanged.

When aggregating local models into global models, we compute the aggregated weight a_i^t of each data owner by Formula (13):

$$a_i^t = \frac{LA_i^t}{\sum_{i \in S^t} (LA_i^t)}, \tag{13}$$

where LA_i^t is their local model accuracy at round t . Compared with FedAvg, our approach is more effective at down-weighting malicious data owners and giving higher weight to benign data owners. In this way, the global model obtained by our aggregation strategy performs better than FedAvg.

4.2. Model Training Time Prediction

In real scenarios, the CPU frequency may fluctuate around the rated value due to multiple factors, such as device CPU scheduling and multiprocess running. According to the computing capability model in Section 3.3, the average CPU frequency in previous rounds can be considered as an indicator of its computing capability at the next round. The heuristic approach based on the above estimate would select data owner with the highest estimated value, but the above intuitive method possibly falls into a local optimum. When some data owners are barely selected as winners, the average estimate would fluctuate within a large confidence interval, thus failing to reflect their real CPU frequency. Thus, we design a model training time prediction algorithm to estimate each data owner’s actual CPU frequency more accurately.

To obtain a more accurate CPU frequency prediction for model training, we adopt the Upper Confidence Bound (UCB) algorithm [23] to update \tilde{f}_i^t :

$$\tilde{f}_i^t = \bar{f}_i^t + \sqrt{\frac{2 \ln t}{n_i^t}}, \tag{14}$$

where \bar{f}_i^t is the average CPU frequency in previous t rounds, and n_i^t is the total amount of being selected in these t rounds, as the exploitation part, indicating that data owners with larger average CPU frequencies have higher probability of being selected. However, to avoid choosing the same winning data owners repeatedly who have higher initial frequency values, we also adopt the exploration part $\sqrt{\frac{2 \ln t}{n_i^t}}$, i.e., we give more chances to select other data owners who are hardly selected in the past. For this reason, their estimated CPU frequency can be obtained by a more reliable confidence interval.

Before new round begins, the platform will use our prediction approach to calculate each data owner’s estimated CPU frequency and then update their computing capability \tilde{C}_i^t by Formula (15):

$$\tilde{C}_i^t = \frac{\tilde{f}_i^t}{d_i^t} \tag{15}$$

4.3. Data Owner Selection and Payment

As mentioned above, when making data owner selection, we should fully consider the possible existence of malicious data owners in the system, as well as the problem of CPU

frequency fluctuations in reality. In each communication round, we will first conduct model quality detection on the uploaded local model, update each data owner’s reputation value based on quality detection value, and further punish data owners with poor performance. In addition, we then update each data owner’s estimated CPU frequency in the next round according to the UCB algorithm. Combined with the above two components, we design an efficient data owner selection algorithm to achieve a better global model accuracy with less model training time.

First, we reformulate our optimization problem as follows:

$$\max \sum_{i=1}^n (d_i^t * Rep_i^{t-1}) * x_i^t \tag{16}$$

$$\text{s.t. } (5a)(5b) \tag{16a}$$

$$Rep_i^{t-1} * x_i^t \geq Rep_{low}, \forall i \in \mathcal{N} \tag{16b}$$

$$\tilde{C}_i^t * x_i^t \geq \tilde{C}_{low}, \forall i \in \mathcal{N} \tag{16c}$$

where we use $d_i^t * Rep_i^{t-1}$ to replace the previous objective function d_i^t in formula (5), and Rep_i^{t-1} is a weighted factor. In (16b), for data owners with too low reputation values, we will restrict their participation in model training by setting a threshold Rep_{low} (we adopt a similar heuristic approach to determine Rep_{low} with C_{low} in Section 3.3). By (16c), we use the computing capability value estimated by the UCB algorithm to limit the data owner with low computing capability.

As is similar to the problem reformulation (6) in Section 3.4, the server selects winning data owners that guarantee both global model accuracy and convergence time within the limited budget as follows:

$$\min \sum_{i=1}^n x_i^t \tag{17}$$

$$\text{s.t. } (5a)(5b)(16b)(16c) \tag{17a}$$

$$\sum_{i=1}^n (d_i^t * Rep_i^{t-1}) * x_i^t \geq Q_i^t \tag{17b}$$

where the constraint (17b) sets the threshold value Q_i^t as the lower upper of the objective function (16). We get the optimal data owner selection scheme by solving the above optimization problem according to the OPT tool.

Algorithm 1 describes the running process of the QuoTa mechanism. In each round of communication, the platform server first computes the optimal data owner set to participate in model training by the objective functions (16) and (17). After receiving the local model parameter uploaded by each winner $i \in \mathcal{S}^t$, the server first performs model quality detection to compute their quality detection value v_i^t and then calculates their direct reputation value rep_i^t based on v_i^t . After that, their comprehensive reputation value Rep_i^t is updated by their direct reputation and selection according to line 9–11 before round t ends. In addition, the server predicts each data owner’s computing capability \tilde{C}_i^t based on their average CPU frequency \tilde{f}_i^t by online exploration-exploitation algorithm in lines 13–15. After that, the global model parameter is aggregated by line 16. Moreover, each data owner is paid a critical payment, p_i [24]. For each winning data owner $i \in \mathcal{S}^t$, the server first excludes them from the whole candidate data owners \mathcal{N}^t , then tries to find an alternative one i^* based on \mathcal{N}^t , and finally uses the claimed cost of i^* to make a critical payment to them.

Algorithm 1: QuoTa Mechanism

Input: Budget B^t at each round t , set of candidate data owners \mathcal{N} , each data owner's bid profile $\mathcal{B}_i = \{b_i, d_i, fre_i\}$, initially estimated CPU frequency \tilde{f}_i^0 , initial comprehensive reputation value Rep_i^0 , initial global model w^0 .

Output: Set $\mathbf{S} = \{\mathcal{S}^1, \mathcal{S}^2, \dots\}$ of selected data owners at each round, their payment vector \mathbf{p} .

```

1  $\mathbf{p} \leftarrow \mathbf{0}, \mathbf{S} \leftarrow \vec{\emptyset}$ ;
2 for communication round  $t = 1, 2, \dots$  do
3   // Data Owner Selection ;
4   compute  $\mathcal{S}^t$  by Formulas (16) and (17);
5   deliver global model  $w^{t-1}$  to  $i \in \mathcal{S}^t$ ;
6   for  $i \in \mathcal{S}^t$  in parallel do
7      $w_k^t \leftarrow$  local model training by  $w^{t-1}$ ;
8     // Model Quality Detection;
9     calculate the detection value
10     $v_i^t = [(GA^t - LA_i^t) - \epsilon_1^t] - [(GA^t - GA_{-i}^t) - \epsilon_2^t]$ ;
11    calculated direct reputation  $rep_i^t$  by (10), (11) ;
12    update comprehensive reputation  $Rep_i^t$  by Formula (12);
13    calculate the aggregated weight  $a_i^t = \frac{LA_i^t}{\sum_{i \in \mathcal{S}^t} (LA_i^t)}$ ;
14    // Model Training Time Prediction;
15    calculate average CPU frequency  $\tilde{f}_i^t = \frac{\sum_{\tau=0}^t \hat{f}_i^\tau * x_i^\tau}{n_i^t}$ ;
16    update estimated computing capability  $\tilde{C}_i^t$  by Formulas (14) and (15);
17  aggregate global model  $w^t = a_i^t \sum_{k=1}^{|\mathcal{S}^t|} w_k^t$ ;
18  // Payment Decision;
19  for  $i \in \mathcal{S}^t, \mathcal{N}' \leftarrow \mathcal{N}$  do
20     $\mathcal{N}' \leftarrow \mathcal{N}' / i$ ;
21    Find an alternative data owner  $i^*$  with the smallest claimed cost  $b_{i^*}$ ;
22     $p_i = \min(\frac{B^t}{|\mathcal{S}^t|}, b_{i^*})$ ;
23 return  $(\mathbf{S}, \mathbf{p})$ ;

```

Theorem 1. The proposed QuoTa mechanism satisfies truthfulness, individual rationality, and budget balance.

Proof. We first prove that QuoTa is truthful. A winner with a smaller claimed bid is still selected because their bid is yet smallest among all candidate data owners. Hence, we show the monotonicity. In addition, according to line 21, when they claim a smaller bid \hat{b}_i , they still obtain the unchanged payment because the received payment is only related to the alternative data owner's bid, and thus their utility is never improved. Conversely, when he reports a higher bid, $\hat{b}_i \geq b_{i^*}$, another data owner will replace them as the new winner. Hence, their utility becomes zero. To this end, he has no incentive to misreport their cost.

Similarly, when they report a larger local data amount $\hat{d}_i \geq d_i$, they would still be selected and obtain the same payment. But, a smaller amount makes them lose. For fre_i , any misreporting about it would never improve their utility. Thus, QuoTa achieves truthfulness on multiple-dimensional claimed resources.

For any data owner who loses, their utility is zero. Additionally, for any winning data owner $i \in \mathcal{S}^t$, their payment is $p_i = \min(\frac{B^t}{|\mathcal{S}^t|}, b_{i^*}) \geq b_i = c_i$ in terms of truthfulness. Therefore, each data owner is individual rational.

Finally, since the total payment is $\sum_i p_i = |\mathcal{S}^t| \min(\frac{B^t}{|\mathcal{S}^t|}, b_{i^*}) \leq B^t$, QuoTa also achieves budget balance. \square

5. Evaluation

In this section, the performance of QuoTa will be demonstrated through simulation experiments to verify that the algorithm performs well in both model accuracy and model convergence time.

5.1. Experimental Settings

To effectively evaluate the QuoTa mechanism, we conducted simulation experiments on a laptop with an Intel® Core™ i7-12700H CPU, NVIDIA GeForce RTX 3060 Laptop GPU, and 16G RAM with WIN11 operating system and built a federated learning framework based on PyTorch deep learning framework.

We use the CIFAR10 and FMNIST datasets and the ResNet18 and LeNet5 models to simulate the federated learning task. We cut the CIFAR10 and FMNIST training sets into 20 sub-datasets of 5*1000, 5*2000, 5*3000, and 5*4000 and 5*500, 5*1000, 5*1500, and 5*2000 to simulate 20 data owners in the federated system. Subsequently, label mislabeling is used to simulate malicious data owners, i.e., the data output labels in the dataset are maliciously labeled as other values. For the simulation of CPU frequency, we use the numerical values to represent the rated CPU frequency and the real CPU frequency of the data owner's device. For example, the rated CPU frequency and the real CPU frequency of a data owner can be represented as 5 GHz and 4.2 GHz, respectively. We use the mathematical expectation of the rated CPU frequency based a normal distribution of variance of 0.2 to simulate the CPU frequency fluctuation.

For the time consumption calculation of the federated system, as the server's arithmetic power is stronger, the time consumption is negligible. Then, the time consumption of the federated system mainly depends on the slowest participating data owner, where the stronger the data owner's computing capability per unit (CPU frequency/data amount), the shorter the time consumption is. We use a linear function to map the relationship between the computing capability and the time consumption and then use the longest data owner's time consumption as the time consumption of training round. Additionally, we take 80 percent of a data owner's computing capability as the computing capability threshold.

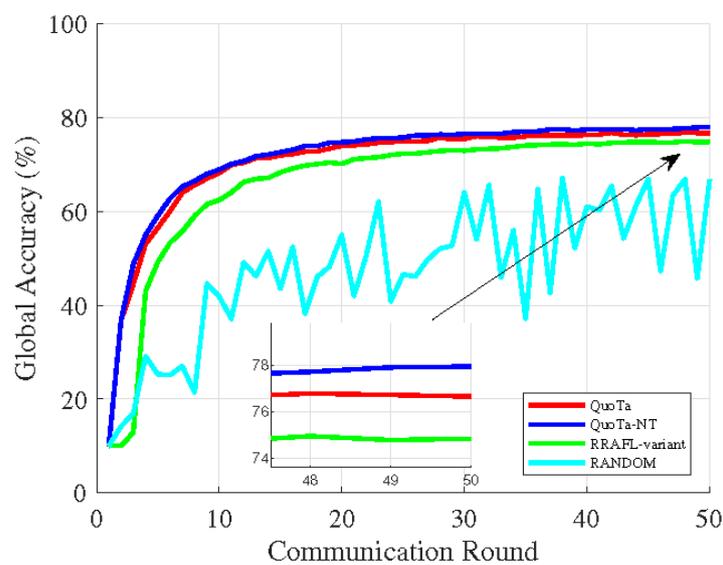
5.2. Comparison Algorithm

1. QuoTa-NT: Remove the computing capability constraints of this algorithm, i.e., the time constraint is not considered when selecting data owners.
2. RRAFL-Variant: As the RRAFL algorithm [2] cannot be directly applied to our scenario, we use RRAFL-Variant as a variant version, which uses the same selection method in terms of reputation/quotation, the model detection method in [2] to detect the local model, and the detected values to update the reputation value of each worker. But, it never considers the training time constraint.
3. RANDOM: The platform randomly selects data owners to participate in training within the budget.
4. SCORE: The bid of each data owner is scored as: $score_i = 0.7 \times d_i + 0.3 \times C_i - b_i$, where each term like d_i is normalized, and then the set of data owners with the highest scores are selected within the budget. We add the similar model quality detection method (below a certain reputation threshold can no longer be selected) and the calculation of computing capability using estimated CPU frequency.
5. GREEDY: The server selects the data owner with the lowest bid within the budget and never adopts our model quality detection method.

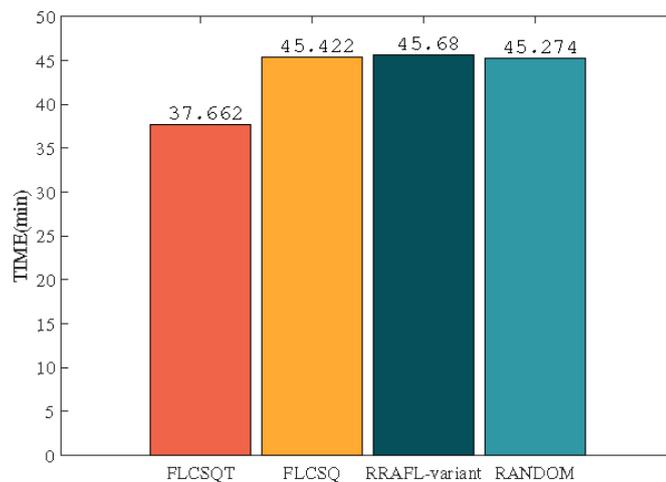
5.3. Model Accuracy and Model Convergence Time

Figure 2a,b depicts the scenario of training the ResNet18 model using the CIFAR10 dataset, where the system has 3*1000, 3*2000, 3*3000, and 3*4000 samples for a total of 12 normal data owners and 2*1000, 2*2000, 2*3000, and 2*4000 for a total of 8 malicious data owners, who are mislabeled with data labels.

Figure 2a represents the convergence accuracy achieved by the global model after 50 rounds of communication, and Figure 2b represents the total time taken to complete the model training after 50 rounds of communication. Similarly, as can be seen from Figure 2a, the QuoTa and QuoTa-NT algorithms both achieve better performance in terms of model accuracy. Due to the limitation of computing capability, some workers with large amounts of training data but poor computing capability may not be selected, so the final convergence accuracy of QuoTa may be slightly lower than that of QuoTa-NT. But, it can be seen from Figure 2b that QuoTa is significantly lower than QuoTa-NT in terms of final time consumption. The RRAFL-variant algorithm ends up with a significantly lower model accuracy than QuoTa and is a bit slower to converge. The main reason is that the data owner selection considers high reputation value per bid price as an indicator. Hence, if a low-reputation data owner makes a relatively low bid, it still has a high probability of being selected, thus affecting the model accuracy.



(a) Global accuracy with 50 rounds of communication

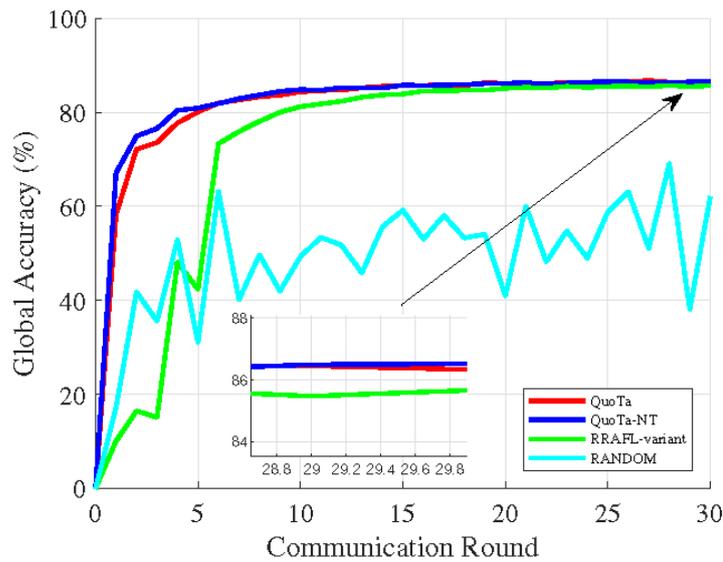


(b) Training time with 50 rounds of communication

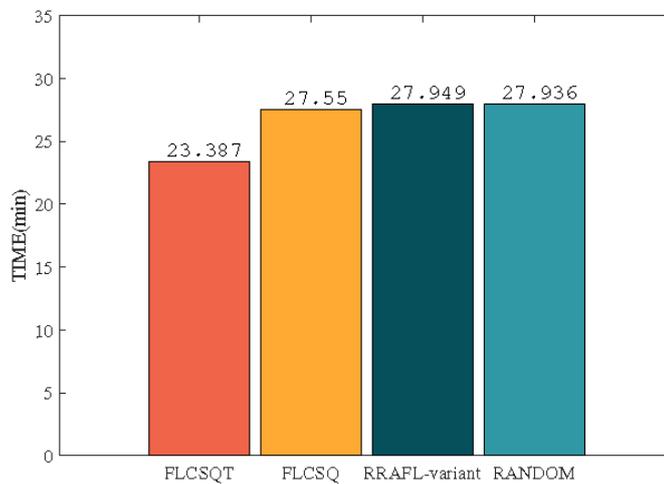
Figure 2. The global accuracy and time for ResNet18 with CIFAR-10.

As the RRAFL-variant algorithm does not take into account model training time by device heterogeneity, it takes longer time to achieve model convergence. The RANDOM algorithm possibly selects malicious data owners in each round of communication due to random selection, so the global model cannot converge.

Figure 3a,b depict the scenario of training the LeNet5 model using the FMNIST dataset, where there are 3*500, 3*1000, 3*1500, and 3*2000 samples for in total 12 normal worker data owners in the system, as well as 2*500, 2*1000, 2*1500, and 2*2000 samples for 8 malicious worker data owners, who are mislabeled with all data labels.



(a) Global accuracy with 30 rounds of communication



(b) Training time with 30 rounds of communication

Figure 3. The global accuracy and time for LeNet5 with FMNIST.

Figure 3a represents the convergence accuracy achieved by the global model after 30 rounds of communication, and Figure 3b represents the total time taken to complete the model training after 30 rounds of communication. As is similar to Figure 2a, QuoTa, QuoTa-NT algorithms have achieved better performance in terms of model accuracy. Due to the relatively simple structure of the LeNet5 model, the amount of data has less impact on the accuracy, so the two algorithms converge to the same final convergence accuracy,

and QuoTa is still dominant in convergence time. The RRAFL-variant algorithm is still lower than QuoTa in terms of final model accuracy and has longer convergence time. The global model of the RANDOM algorithm has also failed to converge. And, it takes near time when performing the same rounds of communication with RRAFL-variant.

5.4. Impact of Malicious Data Owners on Accuracy

In general, the number of malicious data owners in the system has a great impact on the global model accuracy. Figure 4 depicts the accuracy that the QuoTa algorithm achieves in 50 rounds of communication for the global model when the percentage of malicious data owners in the system is 20%, 40%, and 60%. It can be seen that the QuoTa algorithm has good robustness to different percentages of malicious data owners. The convergence speed and convergence accuracy of the global model decrease when the malicious data owners account for a high percentage. Fortunately, the decrease is within an acceptable range for the proposed QuoTa. Overall, our algorithm can effectively resist the influence of malicious data owners.

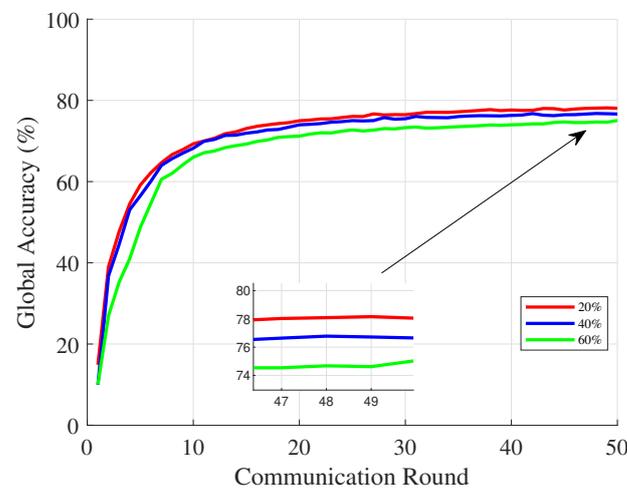


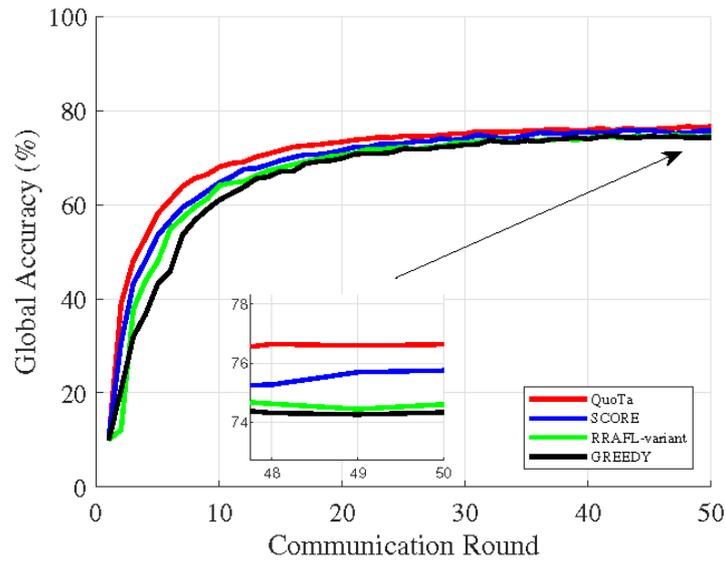
Figure 4. The impact of different numbers of malicious data owners on global accuracy.

5.5. Comparison of Data Owner Selection Schemes

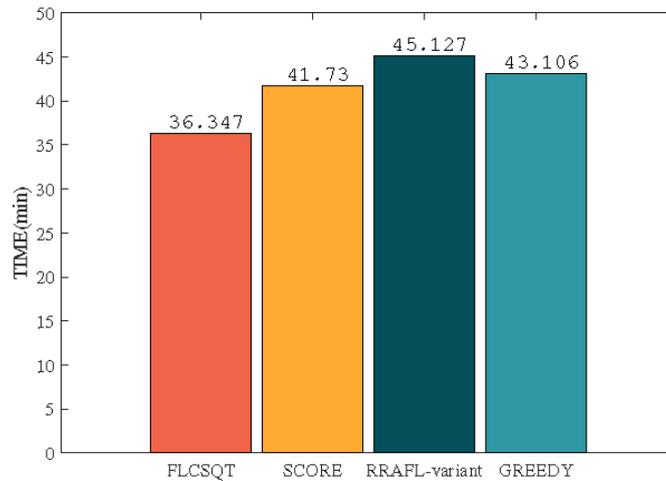
The QuoTa framework can prioritize a winning data owner set that has a large local data volume for a single worker and a large total amount of data in the federated system and directly set a computing capability threshold for the selection of data owners. We experimentally demonstrate that this selection scheme performs well in terms of model accuracy and convergence time. We mainly compare the three algorithms, SCORE, GREEDY, and RRAFL-Variant. Except for different data owner selection indicators, SCORE and GREEDY adopt the same model quality detection, reputation value calculation, and computing capability estimation.

From Figure 5a,b, we can see that QuoTa can converge to a higher accuracy faster with the same number of communication rounds and has a significant advantage in convergence time. The SCORE algorithm calculates the amount of data, computing capability, and the bid price into a single score, which allows for a more effective balance between model accuracy and convergence time. But, if the bid price of a data owner with the average model quality is low enough, it still has a higher probability of being selected, and thus it is inferior to QuoTa algorithm in terms of final global model performance. Similarly, if a data owner's bid price is low enough, the RRAFL-variant algorithm may still select them to participate in the model training even if its reputation value is low. Additionally, the RRAFL-Variant does not take training time into account, so it performs poorly in terms of model convergence. The GREEDY algorithm prioritizes data owners with the lower bid price. As data owners with average model quality may have a low bid price, the GREEDY algorithm performs moderately in terms of the final model accuracy. In consideration of

shorter convergence time than the RRAFL-variant algorithm, the reason lies in the fact that many data owners with less data volume have lower bid prices, thus producing shorter training time. Consequently, QuoTa performs better than other baseline algorithms in terms of global model accuracy and convergence time.



(a) Global accuracy with 50 rounds of communication



(b) Training time with 50 rounds of communication

Figure 5. The impact of different selection schemes on global accuracy and time.

6. Conclusions

In this paper, we proposed a quality-aware incentive mechanism, named QuoTa, for fast federated learning. Specifically, we effectively quantified the data owner’s contribution to the FL training and proposed a model quality detection module to solve the problem of data owners using unclaimed resources to train models or even maliciously destroy global model aggregation. To solve the problem that CPU frequency fluctuation could not reach the standard value during model training in practice, we introduced the model training time prediction module. We conducted extensive evaluations using the ResNet18 and LeNet5 models as well as the CIFAR10 and FMNIST data sets. The results showed that

the QuoTa mechanism performed well in terms of model accuracy and convergence time, effectively ensuring the effectiveness of the FL system.

7. Future Work

In our current work, we have not considered volatile data owners whose federated learning environments are unstable. For example, some data owners cannot upload local model parameters in time before a given training time deadline because their CPU frequency drops dramatically. To adaptively adjust to varying conditions, we can use a probability allocation indicator to record their historical success allocation. By combining it with the training time prediction module, we probably obtain a more accurate training time estimation. In addition, for model quality detection, we can further take the degree of heterogeneity of the data distribution into consideration, which is also regarded as a selection indicator for winning data owners. Thus, the quality detection module is further enhanced. It is worth noting that, in this paper, we do not consider the dynamic arrival of data owners and the fairness of data owner selection, which will be further studied in subsequent work.

Author Contributions: Conceptualization, H.C. and C.B.; methodology, H.C. and C.B.; investigation, H.C. and C.B.; software, C.B. and B.S.; supervision, H.C., J.Z. and J.L.; writing—original draft preparation, H.C. and C.B.; writing—review and editing, H.C., B.S., J.Z., J.L. and X.H.; project administration, H.C., C.B. and B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of China under Grants 62102195, 62102196, 61972210, and 62202224, in part by China Postdoctoral Science Foundation Funded Project under Grant 2022M711688, in part by Jiangsu Provincial Double-Innovation Doctor Program under Grant JSSCBS20210501, and in part by Natural Science Foundation of Jiangsu Province under Grant BK20220882.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: We want to thank Daixin Song for her valuable contributions to our paper, including but not limited to the interpretation of data for the work and partial evaluations.

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

References

1. Wang, D.; Ren, J.; Wang, Z.; Wang, Y.; Zhang, Y. Privaim: A dual-privacy preserving and quality-aware incentive mechanism for federated learning. *IEEE Trans. Comput.* **2022**, *72*, 1913–1927. [[CrossRef](#)]
2. Zhang, J.; Wu, Y.; Pan, R. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 947–956.
3. Zeng, R.; Zhang, S.; Wang, J.; Chu, X. Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 278–288.
4. 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](#)]
5. Liu, T.; Di, B.; An, P.; Song, L. Privacy-preserving incentive mechanism design for federated cloud-edge learning. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2588–2600. [[CrossRef](#)]
6. Tang, J.; Han, F.; Fan, K.; Xie, W.; Yin, P.; Qu, Z.; Liu, A.; Xiong, N.N.; Zhang, S.; Wang, T. Credit and quality intelligent learning based multi-armed bandit scheme for unknown worker selection in multimedia MCS. *Inf. Sci.* **2023**, *647*, 119444. [[CrossRef](#)]
7. Peng, C.; Hu, Q.; Wang, Z.; Liu, R.W.; Xiong, Z. Online-learning-based fast-convergent and energy-efficient device selection in federated edge learning. *IEEE Internet Things J.* **2022**, *10*, 5571–5582. [[CrossRef](#)]
8. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv* **2016**, arXiv:1610.02527.

9. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the PMLR 2017, Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
10. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582.
11. Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing federated learning on non-iid data with reinforcement learning. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1698–1707.
12. Chai, Z.; Ali, A.; Zawad, S.; Truex, S.; Anwar, A.; Baracaldo, N.; Zhou, Y.; Ludwig, H.; Yan, F.; Cheng, Y. Tifl: A tier-based federated learning system. In Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, Stockholm, Sweden, 23–26 June 2020; pp. 125–136.
13. Lee, J.; Ko, H.; Seo, S.; Pack, S. Data distribution-aware online client selection algorithm for federated learning in heterogeneous networks. *IEEE Trans. Veh. Technol.* **2022**, *72*, 1127–1136. [[CrossRef](#)]
14. Fang, M.; Cao, X.; Jia, J.; Gong, N. Local model poisoning attacks to {Byzantine-Robust} federated learning. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Boston, MA, USA, 12–14 August 2020; pp. 1605–1622.
15. 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.
16. Gupta, A.; Luo, T.; Ngo, M.V.; Das, S.K. Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning. In *European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 445–465.
17. Jebreel, N.M.; Domingo-Ferrer, J.; Blanco-Justicia, A.; Sánchez, D. Enhanced security and privacy via fragmented federated learning. *arXiv* **2022**, arXiv:2207.05978.
18. Khan, L.U.; Pandey, S.R.; Tran, N.H.; Saad, W.; Han, Z.; Nguyen, M.N.H.; Hong, C.S. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Commun. Mag.* **2020**, *58*, 88–93. [[CrossRef](#)]
19. 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. (TIST)* **2022**, *13*, 1–21. [[CrossRef](#)]
20. Ying, C.; Jin, H.; Wang, X.; Luo, Y. Double insurance: Incentivized federated learning with differential privacy in mobile crowdsensing. In Proceedings of the 2020 International Symposium on Reliable Distributed Systems (SRDS), IEEE, Shanghai, China, 21–24 September 2020; pp. 81–90.
21. Domingo-Ferrer, J.; Blanco-Justicia, A.; Sánchez, D.; Jebreel, N. Co-utile peer-to-peer decentralized computing. In Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGRID 2020, Melbourne, Australia, 11–14 May 2020; pp. 31–40.
22. Deng, Y.; Lyu, F.; Ren, J.; Wu, H.; Zhou, Y.; Zhang, Y.; Shen, X. Auction: Automated and quality-aware client selection framework for efficient federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 1996–2009. [[CrossRef](#)]
23. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* **2002**, *47*, 235–256. [[CrossRef](#)]
24. Noam, N.; Tim R.; Eva, T.; Vijay, V. *Algorithmic Game Theory*; Cambridge University Press: Cambridge, UK, 2007; Volume 1.

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.