


Article

A Truthful and Reliable Incentive Mechanism for Federated Learning Based on Reputation Mechanism and Reverse Auction

Ao Xiong¹, Yu Chen^{1,*} , Hao Chen², Jiewei Chen¹, Shaojie Yang¹, Jianping Huang², Zhongxu Li² and Shaoyong Guo¹

¹ State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

² State Grid Zhejiang Electric Power Co., Ltd., Hangzhou 310007, China

* Correspondence: cywhatme@bupt.edu.cn

Abstract: As a distributed machine learning paradigm, federated learning (FL) enables participating clients to share only model gradients instead of local data and achieves the secure sharing of private data. However, the lack of clients' willingness to participate in FL and the malicious influence of unreliable clients both seriously degrade the performance of FL. The current research on the incentive mechanism of FL lacks the accurate assessment of clients' truthfulness and reliability, and the incentive mechanism based on untruthful and unreliable clients is unreliable and inefficient. To solve this problem, we propose an incentive mechanism based on the reputation mechanism and reverse auction to achieve a more truthful, more reliable, and more efficient FL. First, we introduce the reputation mechanism to measure clients' truthfulness and reliability through multiple reputation evaluations and design a reliable client selection scheme. Then the reverse auction is introduced to select the optimal clients that maximize the social surplus while satisfying individual rationality, incentive compatibility, and weak budget balance. Extensive experimental results demonstrate that this incentive mechanism can motivate more clients with high-quality data and high reputations to participate in FL with less cost, which increases the FL tasks' economic benefit by 31% and improves the accuracy from 0.9356 to 0.9813, and then promote the efficient and stable development of the FL service trading market.

Keywords: federated learning; incentive mechanism; reputation mechanism; reverse auction



Citation: Xiong, A.; Chen, Y.; Chen, H.; Chen, J.; Yang, S.; Huang, J.; Li, Z.; Guo, S. A Truthful and Reliable Incentive Mechanism for Federated Learning Based on Reputation Mechanism and Reverse Auction. *Electronics* **2023**, *12*, 517. <https://doi.org/10.3390/electronics12030517>

Academic Editor: Alberto Fernandez Hilario

Received: 24 November 2022

Revised: 16 January 2023

Accepted: 17 January 2023

Published: 19 January 2023



Copyright: © 2023 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

With the rapid development of edge computing and distributed machine learning, and the emphasis on data privacy and security has become a worldwide trend [1], federated learning (FL) was first proposed by Google as a new distributed machine learning paradigm in privacy computing [2] and applied to Gboard, an artificial intelligence application for mobile users [3]. A typical FL system consists of the FL server and clients providing data. The FL server publishes the FL tasks and coordinates clients to participate in model training. The client uses local data to train locally and upload updated model parameters. McMahan et al. [4] proposed a practical model based on model averaging for FL, and the FedAvg algorithm is widely used for global model aggregation in FL. FL can perform large-scale machine learning without exposing the raw data of the clients, and the privacy-preserving and high learning performance characteristics of FL have attracted a lot of attention and research from academia and industry [5]. A typical application scenario of FL is to break the data barriers between upstream and downstream enterprises in the financial industry chain, and realize the secure sharing of private data between enterprises, thus promoting the integration and development of financial business.

Although FL is a very promising distributed learning technique, it still faces serious challenges. On the one hand, the clients' participation in FL tasks increases their system

costs, such as consuming power, computational resources, and communication resources of local devices. At the same time, clients also face the threat and risk of privacy leakage during the training process. Therefore, rational clients are reluctant to voluntarily participate in FL without any incentives and financial compensation [6]. On the other hand, some unreliable or malicious clients may unintentionally or intentionally perform undesired behaviors, such as deceptively training the model with low-quality data to reduce resource consumption or malicious behaviors, such as deliberately sending malicious updates to mislead the global model parameters during the training process. Malicious behaviors are harmful behaviors for the FL server, and negative behaviors can be thought of as “useless” behaviors, because negative behaviors do not interfere as much as malicious behaviors or contribute beneficially to the FL global model updates; furthermore, negative behaviors slow down the global model convergence. Then the clients can obtain payments that do not match the contribution through negative behaviors of falsifying data quality. It is the lack of strong aggressiveness like malicious behavior that makes negative behaviors difficult to be detected by FL servers. These unexpected behaviors can severely impact the performance of FL and even lead to the failure of the FL task.

However, most of the current work and research on FL mainly focus on algorithm optimization, efficiency improvement, and privacy protection. There are relatively few studies on the incentive mechanism of client participation and reliable client selection in FL [7]. Moreover, current research on the FL incentive mechanism lacks an accurate evaluation of the truthfulness and reliability of participating clients, and the incentive mechanism built on untruthful and unreliable clients is unreliable and inefficient. The performance of FL depends on the quality of clients’ local model updates. The clients have no motivation to participate in FL, and the malicious influence of unreliable clients will greatly reduce the accuracy of the global model, which seriously hinders the wider application of FL. In order to motivate more reliable clients with high-quality data to participate in FL, we commercialize the FL service between FL servers and clients and consider the FL network as an FL service trading market. The FL server needs to hire a group of clients with data and computational resources for model training to achieve a certain task. After receiving the task, the clients provide private data and computing resources for local training and submit the model updates to the FL server. As the buyer of FL services, the FL server requires the clients to provide high-quality, truthful, and reliable FL services; as the provider of the FL service, the clients require different rewards according to different qualities of the FL services. In order to promote the efficient and stable development of the FL service trading market, the main contributions of this paper are summarized as follows.

1. We construct a reputation mechanism to detect malicious behaviors and negative behaviors of participating clients in the FL training process, and measure the truthfulness and reliability of clients through multiple reputation evaluations. Based on the reputation mechanism, we design a truthful and reliable client-selection scheme.
2. We design a reverse auction that satisfies individual rationality, incentive compatibility, and weak budget balance, and use the deep reinforcement learning (DRL) algorithm D3QN to select the best set of clients from truthful and reliable clients to maximize the social surplus in the FL service trading market.
3. Experimental simulations show that the FL incentive mechanism based on the reputation mechanism and reverse auction can motivate more clients with high-quality data and high reputation to participate in FL with fewer rewards, and can significantly improve the accuracy of FL tasks.

The rest of this article is organized as follows. We present the related work in Section 2 and the preliminary knowledge and system model in Section 3. Section 4 describes the reliable client selection scheme based on reputation mechanism. Section 5 illustrates the optimal client selection scheme based on reverse auction, followed by the performance evaluation in Section 6. Finally, we summarize this article in Section 7.

2. Related Work

The ultimate goal of the incentive mechanism in FL is to motivate more high-quality clients to participate in FL tasks, thus improving the performance of FL. The incentive mechanism in FL consists of three components: contribution evaluation, node selection, and payment allocation [8]. In the incentive mechanism, the optimal strategies for clients and the rewards of FL servers are determined by solving optimization problems (such as the social welfare maximization problem). Moreover, the rewards paid to each participant should be distributed according to their contribution level. In addition, the incentive mechanism and client-selection problems are coupled and should be solved jointly to achieve FL, and the performance of FL can be maximized only if high-quality and reliable clients are selected and motivated to participate in FL. The methods used in the existing research on the FL incentive mechanism mainly include game theory, auction theory, and contract theory. Table 1 provides a summary of the main technique, features, and limitations of each proposed model in the existing incentive mechanisms for FL.

Table 1. Summary of the main technique, features, and limitations in the existing incentive mechanisms for FL.

Ref.	Main Technique	Features	Limitations
[9]	Stackelberg game	<ul style="list-style-type: none"> The mobile devices determine the price per unit of data to maximize their profit. The server chooses the size of the training data to optimize its profit. 	<ul style="list-style-type: none"> Do not consider the impact of non-IID data among different mobile devices. Do not consider mobile devices' unreliable behaviors.
[10]	Reverse auction	<ul style="list-style-type: none"> Design two auction mechanisms to maximize the social welfare of the FL service market. Regard clients' data volume and data distribution as the criteria for contribution evaluation. 	<ul style="list-style-type: none"> Do not consider that the clients falsify the local data volume and data distribution. Do not consider poisoning attacks during training as same as [9].
[11]	Contract theory	<ul style="list-style-type: none"> The mobile users can maximize their utility only when they choose contracts that match their type. Use reputation to measure the reliability and trustworthiness of mobile devices. 	<ul style="list-style-type: none"> Do not consider mobile devices' negative behaviors. Do not consider mobile devices' data size when evaluating contribution.

The authors in [9] proposed a Stackelberg game model for studying the interactions between the server and mobile devices in a cooperative relay communication network. The mobile devices determine the price per unit of data to maximize their profit, whereas the server chooses the size of the training data to optimize its profit. Finally, the interactions between the server and the mobile devices can reach a Nash equilibrium. However, this scheme in [9] only evaluates the contribution of mobile devices in terms of data size, and does not consider the impact of nonindependent and identical distribution (Non-IID) data among different mobile devices on the accuracy of the global model. At the same time, this scheme does not consider those mobile devices may have unreliable behaviors maliciously interfering with FL training.

The authors in [10] propose an auction-based FL service market to encourage clients to participate in FL. They design two auction mechanisms of approximate policy proof and automatic policy proof based on deep reinforcement learning and graph neural network for the FL platform to maximize the social welfare of the FL service market. This scheme regards clients' data volume and data distribution as the criteria for contribution evaluation, and derives the relationship between the accuracy rate of the FL global model and clients' data volume and data distribution through experiments. However, this scheme also does not consider that the clients falsify the local data volume and data distribution during auction bidding and launch poisoning attacks during training as same as [9].

The authors in [11] propose an approach based on contract theory that incentivizes mobile devices to participate in FL and contribute high-quality data. The mobile users can maximize their utility only when they choose contracts that match their type. Meanwhile, reputation is introduced in [11] as an indicator to measure the reliability and trustworthiness of mobile devices, and a reputation-based FL worker selection scheme is designed by using a subjective logic model. However, this scheme only considers the malicious behaviors of poisoning attacks when evaluating the reputation of mobile devices, and does not consider the negative behaviors that mobile devices may deliberately use low-quality data in local training. When evaluating the contribution of mobile devices, the scheme only considers the data distribution, not the data size.

Among the three approaches commonly used in the design of the FL incentive mechanism, unlike game theory and contract theory, auction theory allows clients to actively report their local data types, which helps eliminate information asymmetry between FL servers and clients and is more conducive to evaluating clients' contributions. However, it must be ensured that the clients cannot lie or falsify the bidding resource information during the auction; that is, ensuring that the clients have no negative behaviors. Similar to the situation in the Internet of things (IoT), different devices often need to share resources, but resource sharing faces huge threats and challenges due to the existence of malicious and untrusted devices. A fog-based trust evaluation method for IoT devices proposed in [12] is designed to identify malicious devices and help protect the system from several attacks. IoT devices are identified as malicious devices based on direct and indirect trust values obtained through collaborative IoT devices. The fog node then aggregates trust values from multiple IoT devices to calculate the final trust value that can be shared across all entities, which will help maintain the security of the IoT system. The ShareTrust mechanism proposed in [13] provides a central interface integrated with trust-management mechanisms to evaluate trust. The evaluation computes the trust degree to identify malicious and compromised nodes. ShareTrust calculates trust at the time of the event and allows the seeker and provider to calculate trust.

Therefore, a reputation mechanism can be introduced to detect malicious and negative behaviors of clients participating in FL training, and direct and indirect reputation evaluation can be used to measure the truthfulness and reliability of clients. In order to design a more truthful and reliable FL incentive mechanism based on the auction theory, we introduce a reputation mechanism to ensure that the clients participating in the auction have no malicious or negative behavior, thus realizing the truthfulness and reliability of the FL incentive mechanism by ensuring the truthfulness and reliability of the clients. At the same time, it can use auction theory to maximize the social surplus of the entire FL service trading market.

3. Preliminary Knowledge and System Model

In this section, we briefly introduce the basics of federated learning followed by the system model.

3.1. Preliminary Knowledge of Federated Learning

The typical FL scheme usually adopts the parameter server architecture, where a group of edge-side terminal devices called clients participate in distributed machine learning

training under the guidance of the FL server. FL aims to optimize the global loss function by minimizing the weighted average of the local loss functions on each client's local dataset. Each client i maintains a private local training dataset D_i and has a local FL runtime to train the local model w_i [9]. The client uses the local dataset to perform training tasks locally, and then uploads the trained local model parameters to the FL server for model aggregation. Different from the traditional collection of all data for centralized training, the FL server only collects and aggregates the model parameters after client training, which can effectively reduce data transmission costs and protect privacy at the same time [6]. The FL training process usually includes the following three steps, where step 2 and step 3 form an iterative loop between the FL server and the clients until the global loss function converges.

Step 1. Task initialization and distribution: The FL server determines the training task and the corresponding data requirements, while specifying the machine learning model and the hyperparameters of the training process. Then, the FL server transmits the task information and the initial global model w_g^0 to all clients in the FL network.

Step 2. Local model training and update: Based on the global model w_g^k , where k denotes the current global epoch index, each client uses local data to update the local model parameters w_i^k , respectively. The goal of the client i in epoch k is to derive parameters that minimize the predefined loss function $L(w_i^k)$, i.e.,

$$w_i^{k*} = \operatorname{argmin}_{w_i^k} L(w_i^k). \quad (1)$$

Step 3. Global model aggregation and update: The FL server receives and aggregates the local model parameters from clients, and then sends the updated global model parameters back to clients. The FL server aims to minimize the global loss function, i.e.,

$$L(w_g^k) = \frac{1}{N} \sum_{i \in \mathcal{N}} L(w_i^k). \quad (2)$$

3.2. Truthful and Reliable Incentive Mechanism for FL

In the FL task, the FL server most values the accuracy of the global model. Because the global model parameters are obtained by weighted averaging the parameters uploaded by each client [4], the performance of FL is affected by the training effect of each client. However, thousands of clients with heterogeneous data are involved in the FL service trading market. The datasets of different clients are usually unbalanced and non-IID, and the data heterogeneity will greatly affect the model accuracy and training speed of FL. The FL server cares most about the client's local data quality, which includes two aspects: data volume and data distribution. In machine learning, more data volume is usually better for the model training effect. Data distribution will affect the degree of divergence of model weights and lead to a decrease in model accuracy [14]. In order to characterize the local data distribution of client i , we choose the EMD distance ε_i to calculate the probability distance between the local data distribution and the global data distribution, i.e.,

$$\varepsilon_i = \sum_{j \in Y} \|P_i(y = j) - P_g(y = j)\|, \quad (3)$$

where Y is the label space of training data required for FL, and the global data distribution P_g is derived by the FL server based on existing datasets or historical experience. Because the computational and communication resources of clients are not the focus of this paper, it is assumed that these two types of resources of clients can meet the minimum requirements for the FL server. At the same time, edge-side clients are not always truthful and reliable. There exist malicious clients that launch poisoning attacks and low-quality clients that falsely claim to have high-quality data and execute malicious or unreliable model updates locally, thereby reducing the performance of FL. In order to filter and motivate truthful

and reliable high-quality clients, this paper proposes a truthful and reliable FL incentive mechanism, and its workflow is shown in Figure 1.

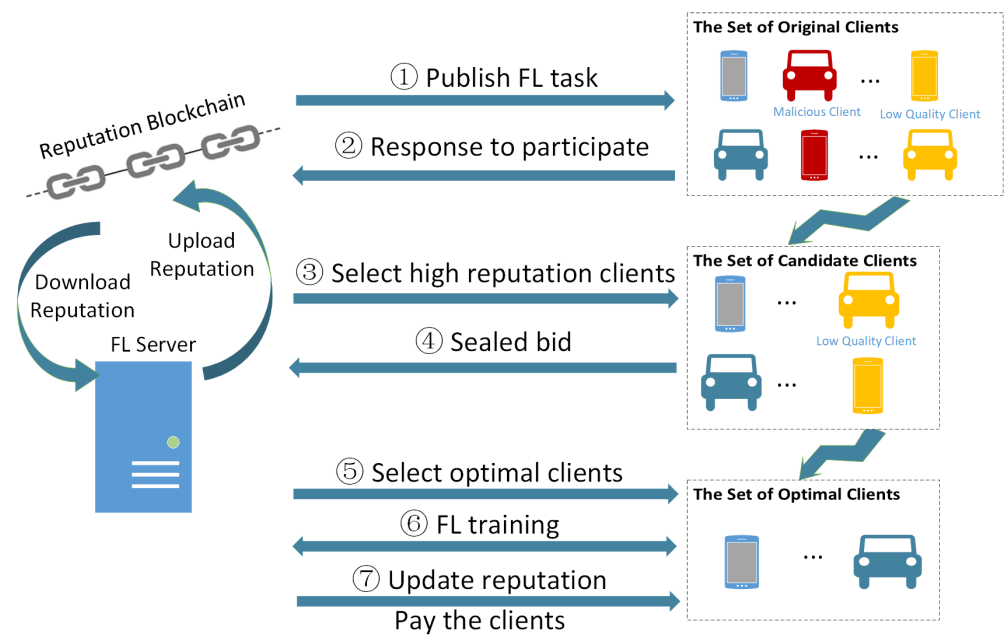


Figure 1. The workflow of the incentive mechanism.

1. Publish FL task. FL server broadcasts FL task information in the FL service trading market, including required data types, minimum requirements for computing resources, bidding rules, etc.
2. Response to participate. After receiving the task information, the interested clients express their willingness to participate in FL by combining their own data and computing resources.
3. Select high reputation clients. After receiving the participation response from clients, the FL server first downloads the clients' reputation value from the reputation blockchain. Then FL server calculates the reputation of each client according to the reputation mechanism, so as to select the clients with high reputations to form a set of reliable clients.
4. Sealed bid. The reliable clients make sealed bids to the FL server and report their own data quality and cost information.
5. Select optimal clients. The FL server solves the reverse auction by using the DRL algorithm D3QN to select the set of optimal clients to participate in FL.
6. FL training. The FL server aggregates and updates the global model, and the clients train and update the local models, repeating until the global loss function converges.
7. Update reputation and pay the clients. The FL server evaluates the reputation of each client based on the information interacted with the clients in this task and uploads it to the reputation blockchain. If the client's reputation declines, the server will not pay the client; otherwise, the client will be paid.

4. Reliable Client Selection Scheme Based on Reputation Mechanism

4.1. The Design of Reputation Mechanism

In the FL service trading market, high-reputation clients with high-quality data play an important role in the model training process, and an effective and accurate reputation calculation method will motivate more high-quality clients to provide high-quality model parameters. In this paper, the reputation assessment of clients is divided into direct reputation assessment, indirect reputation assessment, and comprehensive reputation

assessment, and the corresponding reputation assessment values are direct reputation, indirect reputation, and comprehensive reputation. The detailed definitions are as follows.

- **Direct Reputation:** The reputation value is obtained by the FL server in the current FL task through direct reputation assessment based on the interaction history with the clients.
- **Indirect Reputation:** The reputation value is derived from the reputation assessment of clients by other FL servers not in the current FL task based on their interaction history, which is an indirect reputation assessment compared to the FL server in the current task. Indirect reputation is also called recommended reputation, and the FL server that gives indirect reputation is also called recommended server.
- **Comprehensive Reputation:** When the direct reputation cannot evaluate the clients, the comprehensive reputation will be calculated by considering the direct reputation and the indirect reputation.

The interaction refers to the process that the clients download the global model parameters from the FL server and train and upload the local model parameters once. Interactions are divided into malicious interaction, negative interaction, and positive interaction depending on the clients' behaviors in the task. The interaction history is the collection of server–client interaction records within a time window. Next, the reputation mechanism will be described from four aspects: client behavior and detection, direct reputation assessment, indirect reputation assessment, and comprehensive reputation assessment.

4.1.1. Client Behavior and Detection

Due to the openness and complexity of the FL service trading market, some clients may perform malicious and unreliable model updates unintentionally or intentionally; for example, malicious clients may launch poisoning attacks during the training process, intentionally inject toxic data points into the training dataset, or modify the training datasets to decline the accuracy of the training data, thus increasing the probability of misclassification and possibly manipulating the training process; unreliable clients deceptively train locally with low-quality data to reduce resource consumption and obtain maximum benefits. These malicious and unreliable behaviors can affect the accuracy of the global model and even lead to the failure of the FL task. According to the impact brought by the clients during the FL training process, the client behaviors are classified as malicious behavior, negative behavior, and positive behavior.

1. **Malicious Behavior:** The client has poisoning attacks, including data poisoning and model poisoning. Data poisoning refers to the attacker contaminating the samples in the training set by adding wrong labels or flipping sample labels to reduce the accuracy of the data and increase the probability of misclassification, thus affecting the performance of the global model. Model poisoning differs from data poisoning in that the attacker does not directly manipulate the training data, but sends wrong parameters or corrupted models to corrupt the global model, thus affecting the change direction of the global model parameters, slowing down the convergence of the model, and even destroying the correctness of the global model, which seriously affects the performance of the model [15].
2. **Negative Behavior:** The client has no malicious behavior, but has deceptive behavior. In order to reduce resource consumption and obtain maximum individual benefits, the data size and EMD distance provided by the client during the actual training process don't match the bidding time. Negative behavior refers to falsifying the data quality, i.e., the data quantity d and EMD distance ϵ are falsified, making the submitted local model accuracy rate much lower than expected.
3. **Positive Behavior:** The client bids truthful data size and EMD distance, and is single-minded in completing the training task, with no malicious behaviors or negative behaviors during the training process.

In order to accurately assess the client's reputation and behavior during training, it is necessary to design a client behavior detection scheme based on the above behavior definitions. Due to the FL architecture and privacy protection requirements, the FL server cannot directly detect the clients' training dataset, so the FL server can only discriminate and detect the behaviors by performing quality assessments on the local model updates uploaded by the clients.

1. **Malicious Behavior Detection:** The FL server evaluates the quality of clients' local model updates through the poisoning attack-detection scheme to determine whether the clients have malicious behaviors. RONI [4] and FoolsGold [16] are attack-detection schemes for IID and non-IID data scenarios respectively. The RONI scheme verifies local model updates by comparing the impact of local model updates on the FL server's predefined database. If the performance degradation of local model updates on the database exceeds a specified threshold given by the system, the local model updates will be rejected during aggregated model updates. The FoolsGold scheme identifies malicious clients based on the diversity of local model gradients in the non-IID scenario, because malicious clients always repeatedly upload similar gradients as local model updates in each iteration. Through these two schemes, the FL server can identify the malicious behaviors and reject the local model parameters submitted by the malicious clients.
2. **Negative Behavior Detection:** The negative behaviors of clients are deceptive behaviors of false reporting in the bidding stage, leading to much lower than the expected behavioral performance of clients in the subsequent training stage, such as significantly lower model accuracy. Therefore, detection for negative behaviors can be relatively detected among clients by using differences in the clients' local model accuracy rates. Here, it is assumed that there is no client collusion and therefore no group cheating occurs. In order to obtain maximum benefits while reducing resource consumption, the clients have three specific behaviors for data quality fraud, which are as follows.
 - (a) Randomly generate model parameters and upload them to the FL server. In this case, the clients don't train locally, but directly generate model parameters through a random algorithm for upload. Due to the randomness of the parameters generated by the algorithm, the accuracy of the uploaded model is extremely low, much lower than that of other normal client uploaded models. Moreover, the parameters submitted by the client in each round are also random, which cannot reflect the direction of the local model updates at all.
 - (b) Directly upload the global model parameters in the next round. In this case, the clients choose to directly upload the global model parameters downloaded from the FL server in the next round. In this way, the client can not only avoid resource-consuming local training, but also submit a local model with a higher accuracy rate and the same update direction as the global model, which is not considered in most incentive designs.
 - (c) Use local low-quality data for training. Knowing that the local data quality is very poor (the amount of data d is small, and the EMD distance ϵ is large), the client submits false data-quality information when bidding in order to successfully pass the screening of the incentive mechanism, declaring that the amount of local data is large and the EMD distance is small. This type of client does not submit false model parameters during training, but trains with local low-quality data. Through the experiment in Section 6.2, it is found that the accuracy of the model submitted by this type of clients is very low in the first three rounds, which is much lower than the model submitted by the other normal clients, but after three rounds of global aggregation, the accuracy of the models submitted by this type of client does not differ much from the global model. Because the small amount of local data no longer affects the update of the global model.

Combined with the above description, negative behavior (b) can be detected directly based on the local parameters uploaded by the clients and the global parameters are the same; for negative behavior (a) and (c), the accuracy of the models submitted by the negative clients is much lower than that of the positive client due to the use of randomly generated model updates or low-quality data for training. In addition, the K-means clustering in the unsupervised clustering algorithm upholds the principle of “the closest distance within a group and the farthest distance between groups”, so that similar samples can be automatically classified into one category. Therefore, the K-means clustering algorithm can be used to detect negative behaviors (a) and (c) by classifying clients into positive and negative categories according to data size, EMD distance, and model accuracy rate. Clients classified into negative categories have negative behaviors and refuse to accept the model submitted by negative clients to participate in the aggregation. See Section 6.2 for specific experiments.

4.1.2. Direct Reputation Assessment

According to the client-behavior detection scheme given above, the interaction between the FL server and clients is defined as malicious interaction if the client is detected to have malicious behavior, negative interaction if the client is detected to have negative behavior, and positive interaction if there is no malicious behavior or negative behavior. In a time window T containing N time slots $\{t_1, t_2, \dots, t_N\}$, the direct reputation evaluation is determined by the proportion of positive interactions, negative interactions, and malicious interactions. The direct reputation calculation in the time slots t_n is shown in (4),

$$R_{t_n} = \frac{\alpha I_P^{t_n}}{\alpha I_P^{t_n} + \beta I_N^{t_n} + \gamma I_M^{t_n}}, \quad (4)$$

where $I_P^{t_n}$ denotes the number of positive interactions in t_n , $I_N^{t_n}$ denotes the number of negative interactions in and the number of malicious interactions in t_n , and α , β , and γ denote the weights of positive interactions, negative interactions, and malicious interactions in the reputation calculation, respectively. Because positive interactions can increase the clients' reputation value, negative interactions and malicious interactions both decrease the reputation value, and malicious interactions are more serious than negative interactions, so the weights of each type of interaction should be different, satisfying $\alpha < \beta < \gamma$ and $\alpha + \beta + \gamma = 1$. It should also be noted that the reputation value is affected by several factors, and we have considered two factors' activity and freshness in the reputation calculation.

1. Activeness: The more times client i interacts with FL server P , the more truthful and reliable the server's reputation assessment of the client is. The activeness is the ratio of the number of interactions between server P and client i in a time window and the average number of interactions between server P and all clients, as shown in (5),

$$A_{P \rightarrow i} = \frac{I_{P \rightarrow i}}{\frac{1}{|S|} \sum_{s \in S} I_{P \rightarrow s}}, \quad (5)$$

where $I_{P \rightarrow i} = I_P + I_N + I_M$ denotes the total number of interactions between server P and client i in a time window, and S denotes the set of clients that interacted with the FL server in the same time window.

2. Freshness: During interactions between FL servers and clients, clients are not always truthful and reliable. The trustworthiness of a client varies over time, and more recent interactions with greater freshness are given more weight than past interactions. To accurately reflect the timeliness of reputation evaluation, a freshness decay function is introduced to account for the freshness of interactions, as shown in (6),

$$v_{t_n} = z^{N-n}, \quad (6)$$

where $z \in (0, 1)$ is a given decay parameter about the freshness of the interaction and $n \in (0, 1)$ is the time slot that determines the decay degree of the interaction freshness. Therefore, the direct reputation of FL server P to client i within a time window is expressed as (7),

$$R_{P \rightarrow i} = A_{P \rightarrow i} \frac{\sum_n^N v_{t_n} R_{t_n}}{\sum_n^N v_{t_n}}. \quad (7)$$

4.1.3. Indirect Reputation Assessment

In the FL service trading market, each client has participated in tasks issued by different FL servers and has had many different types of interactions with them. The reputation assessment of clients should consider the indirect reputation assessment from other FL servers. Due to the complexity of the FL service trading market, the reliability of indirect reputations from different FL servers should be different. It needs to be considered that the similarity of business between the request server and recommended servers and the dispersion of the reputation given by the recommended servers from the overall recommended reputations.

1. Similarity: The more similar the business between FL servers, the higher the overlap in the set of recruited clients, and the higher reliability of the indirect reputation of clients that interact more frequently between two FL servers. We represent each FL server's reputation assessment of the interacted clients as a separate vector, and measure the similarity of the business between FL servers by solving the similarity of the vectors with Pearson's correlation coefficient. The greater the similarity, the higher the reliability of the reputation given by the recommended servers. According to the definition of Pearson's correlation coefficient [17], the similarity between server i and server j is shown in (8),

$$\text{sim}(i, j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i) (R_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i)^2} \sqrt{\sum_{c \in I_{ij}} (R_{j,c} - \bar{R}_j)^2}}, \quad (8)$$

where I_{ij} denotes the set of clients that have interacted with both FL server i and FL server j , that is, the set of clients for which both have reputation assessment; $R_{i,c}$ denotes the reputation assessment value of FL server i for client c ; $R_{j,c}$ denotes the reputation assessment value of FL server j for client c ; \bar{R}_i denotes the mean value of FL server i 's reputation assessment for clients in I_{ij} ; \bar{R}_j denotes the mean value of FL server j 's reputation assessment for clients in I_{ij} .

2. Dispersion: Considering that the reputation recommendation server may collude with some clients to cheat and maliciously increase the reputation value of some low-reputation clients, not all indirect reputation values are reliable. In this paper, we introduce information entropy to reflect the dispersion degree among indirect reputation values, that is, the degree of deviation of each indirect reputation value from the overall set of indirect reputation values. Information entropy can identify the excessively high and low reputation values in the overall indirect reputation, thus making the indirect reputation assessment more objective and accurate [18]. The dispersion of indirect reputation values $R_{j \rightarrow c}^{rec}$ of client c by recommended server j is calculated by using information entropy as shown in (9),

$$\omega_j = \frac{1 - \frac{H(R_{j \rightarrow c}^{rec})}{\lg R_{j \rightarrow c}^{rec}}}{\sum_{j=1}^n \left[1 - \frac{H(R_{j \rightarrow c}^{rec})}{\lg R_{j \rightarrow c}^{rec}} \right]}, \quad (9)$$

where $H(R_{j \rightarrow c}^{rec}) = -R_{j \rightarrow c}^{rec} \lg R_{j \rightarrow c}^{rec} - (1 - R_{j \rightarrow c}^{rec}) \lg (1 - R_{j \rightarrow c}^{rec})$ denotes the entropy of the indirect reputation value $R_{j \rightarrow c}^{rec}$ and n denotes the total number of recommended servers. The weight of the indirect reputation value $R_{j \rightarrow c}^{rec}$ from each recommended server j is set $\frac{\text{sim}(i,j) + \omega_j}{2}$ by considering the similarity between request server i and recommended server j and the dispersion between recommended servers. Finally, the global indirect reputation evaluation of client c is calculated as shown in (10),

$$R_c^{rec} = \sum_{j=1}^n \frac{\text{sim}(i,j) + \omega_j}{2} R_{j \rightarrow c}^{rec}. \quad (10)$$

4.1.4. Comprehensive Reputation Assessment

Comprehensive reputation evaluation is to use information entropy to determine the adaptive weights of the two evaluation methods of direct reputation and indirect reputation, that is, the weights of reputation values are corrected according to the degree of difference between the two assessment methods. Therefore, the calculation of the comprehensive reputation value of client i is as in (11),

$$R_{P \rightarrow i}^{final} = \theta_d R_{P \rightarrow i} + \theta_r R_i^{rec}, \quad (11)$$

where θ_d, θ_r are the adaptive weights of direct reputation $R_{P \rightarrow i}$ and indirect reputation R_i^{rec} respectively, and θ_d, θ_r are calculated as in (12) and (13).

$$\theta_d = \frac{1 - \frac{H(R_{P \rightarrow i})}{\lg R_{P \rightarrow i}}}{\left[1 - \frac{H(R_{P \rightarrow i})}{\lg R_{P \rightarrow i}}\right] + \left[1 - \frac{H(R_{j \rightarrow c}^{rec})}{\lg R_{j \rightarrow c}^{rec}}\right]} \quad (12)$$

$$\theta_r = \frac{1 - \frac{H(R_{j \rightarrow c}^{rec})}{\lg R_{j \rightarrow c}^{rec}}}{\left[1 - \frac{H(R_{P \rightarrow i})}{\lg R_{P \rightarrow i}}\right] + \left[1 - \frac{H(R_{j \rightarrow c}^{rec})}{\lg R_{j \rightarrow c}^{rec}}\right]}. \quad (13)$$

4.2. The Reliable Client Selection Scheme

As shown in Algorithm 1, we design a reliable client-selection scheme to filter out high-reputation clients from the original client set, which corresponds to the third step of the workflow of the incentive mechanism. The FL server sets three thresholds: the upper limit of direct reputation R_{direct}^{upper} , the lower limit of direct reputation R_{direct}^{down} and the threshold of reputation R^{TH} . The FL server first calculates the direct reputation of the original clients based on historical interactions, and if the direct reputation is lower than R_{direct}^{down} , the client is eliminated; if the direct reputation is higher than R_{direct}^{upper} , the client is directly selected; if the client's direct reputation lies between the upper and lower limits, the FL server continues to calculate the client's indirect reputation, and then generates a comprehensive reputation by considering the direct reputation and indirect reputation. Finally, if the comprehensive reputation is greater than the reputation threshold R^{TH} , the client is selected; otherwise, the client is eliminated.

Algorithm 1 Reliable client-selection scheme.

Input: The set of original clients \mathcal{N} , upper limit of direct reputation R_{direct}^{upper} , lower limit of direct reputation R_{direct}^{down} , threshold of reputation R^{TH} .

Output: The set of reliable clients \mathcal{R}

```

 $\mathcal{R} \leftarrow \emptyset$ 
1: for each client  $i$  in  $\mathcal{N}$  do
2:    $R_i^{dir} \leftarrow \text{DirectReputationCalculation}(i)$ 
3:   if  $R_i^{dir} > R_{direct}^{upper}$  then
4:      $\mathcal{R} \leftarrow \{i\} \cup \mathcal{R}$ 
5:     continue
6:   else if  $R_i^{dir} < R_{direct}^{down}$  then
7:     Delete( $i$ )
8:     continue
9:   else
10:     $R_i^{rec} \leftarrow \text{IndirectReputationCalculation}(i)$ 
11:     $R_i^{final} \leftarrow \text{ComprehensiveReputationCalculation}(i)$ 
12:   end if
13:   if  $R_i^{final} \geq R^{TH}$  then
14:      $\mathcal{R} \leftarrow \{i\} \cup \mathcal{R}$ 
15:     continue
16:   else
17:     Delete( $i$ )
18:     continue
19:   end if
20: end for

```

5. Optimal Client-Selection Scheme Based on Reverse Auction

5.1. The Design of Reverse Auction

To select the best set of clients with high-quality data from the set of reliable clients, the FL server organizes a reverse auction wherein clients make sealed bids reporting their data quality and cost information. The client bid $Bid_i = (d_i, \varepsilon_i, b_i)$, where d_i is the local data size of the client i , ε_i is the EMD distance of the client i , and b_i is the bid price of client i and represents the cost quotation of the client i to participate in FL. After receiving the bids from the clients, the rational FL server needs to balance the FL benefits brought by the clients and the payments that need to be paid to the clients, and select the clients to form the best set of clients under the satisfaction of nonnegative benefits, and decide their corresponding payments based on their contributions to FL.

In the FL service trading market, the transaction objects are an FL server and n clients $\mathcal{N} = \{1, 2, \dots, n\}$. Each client $i \in \mathcal{N}$ has a local dataset containing d_i data samples to participate in FL. The FL server sets the number of local epochs δ_l and the number of global epochs δ_g in the FL task. The number of CPU cycles to execute one data sample in local training is the same for all clients and is denoted as c . The CPU operating frequency of all clients is denoted as f . According to [19], the computation time required by the client i in one local epoch is $d_i c / f$. Therefore, the calculation time of the client i in one local epoch in the FL task is calculated as (14),

$$T_i^{cmp} = \delta_g \delta_l \frac{d_i c}{f}. \quad (14)$$

The energy consumption of client i in one local epoch in the FL task is shown in (15), where ζ is the effective capacitance parameter of the client's computational chipset [20],

$$E_i^{cmp} = \delta_g \delta_l \zeta c f^2 d_i. \quad (15)$$

During the global iterations, the client sends the local model updates to the FL server via wireless communication. In this paper, we consider the time division multiple access protocol (TDMA) for the clients to communicate with the FL server, while assuming that the client's position is constant during the transmission of the local model updates. The transmission rate of client i is $r_i = B \ln[1 + (\rho_i h_i / N_0)]$, where B is the transmission bandwidth, ρ_i is the transmission power of client i , h_i is the channel gain of the peer-to-peer link between client i and the FL server, and N_0 is the background noise [10]. Therefore, the communication time of client i in the FL task is calculated as in (16),

$$T_i^{com} = \frac{\sigma}{r_i} = \frac{\sigma}{B \ln[1 + (\rho_i h_i / N_0)]}, \quad (16)$$

where σ is the size of the local model updates. Because we do not consider the differences in communication resources of clients in this paper, and assume that the data size of the local model update is the same for all clients, the communication time in the FL task is the same for all clients. The communication energy consumption of client i is $E_i^{com} = \rho_i T_i^{com}$. Therefore, the cost of the client i to participate in the FL task is calculated as in (17), where μ is the energy consumption weight parameter,

$$C_i = \mu (E_i^{cmp} + E_i^{com}). \quad (17)$$

The benefit B of the FL server is related to the accuracy of the global model and is calculated as in (18),

$$B = B(Q(G)) = \lambda Q(G), \quad (18)$$

where $Q(G)$ is the satisfaction function of the FL server on the global model accuracy [9], and λ is the satisfaction weight parameter of the FL server on $Q(G)$, which is calculated as shown in (19),

$$Q(G) = Q(D, \Delta) = \alpha(\Delta) - k_1 e^{-k_2 (k_3 D)^{\alpha(\Delta)}}, \quad (19)$$

where $D = \sum_{i \in \mathcal{N}} d_i$ is the sum of the data size of all participating clients and $\Delta = \frac{1}{n} \sum_{i \in \mathcal{N}} \varepsilon_i$ is the average EMD distance of all clients, $\alpha(\Delta) = k_4 \exp\left(-\left(\frac{\Delta + k_5}{k_6}\right)^2\right) < 1$. The calculation of $Q(G)$ adopts the curve-fitting method for determining the machine learning quality function. The positive curve-fitting parameters $k_1 = 0.361$, $k_2 = 4.348$, $k_3 = 10^{-3}$, $k_4 = 0.993$, $k_5 = 0.31$, and $k_6 = 1.743$ can be obtained from [10].

In this paper, we do not consider the energy consumption and cost loss of the FL server during the aggregation global model; that is, the only cost of the server is the payments to the client. Thus, the utility function of the FL server is (20)

$$U = B \odot \eta - \sum_{i=1}^n P_i \eta_i, \quad (20)$$

where $\eta \in (0, 1)^n$ denotes the client-selection vector and $\eta_i = 1$ represents that the client is selected in the reverse auction; otherwise, it is not selected. The social surplus obtained by the FL task is defined as the revenue obtained by the FL server minus the cost of client training, which is calculated as in (21)

$$S = B \odot \eta - \sum_{i=1}^n b_i \eta_i. \quad (21)$$

Payment rules are formulated based on the client's marginal contribution to the social surplus in FL, and the payment of client i is as follows (22),

$$P_i = S^*(d, \varepsilon, b) - S^{-i*}(d^{-i}, \varepsilon^{-i}, b^{-i}) + b_i, \quad (22)$$

where S^* is the social surplus under the determination of the best set of clients, and S^{-i*} is the social surplus under determining the best set of clients without client i . d^{-i} , ε^{-i} , and b^{-i} refer to the data size, EMD distance, and bidding cost without client i , respectively. Thus, the utility function of the client is as in (23),

$$U_i = P_i - b_i = S^*(d, \varepsilon, b) - S^{-i*}(d^{-i}, \varepsilon^{-i}, b^{-i}). \quad (23)$$

The utility of the client is the social surplus difference between the client participating in the FL task and not participating in the FL task, that is, the marginal contribution of the client to the social surplus in the FL task. Payments based on the client's marginal contribution to the social surplus can achieve fairness in the distribution of payments, because the more contributions, the higher the payment. Because both the FL server and clients are rational, the design of the reverse auction needs to satisfy the following properties:

- Individual Rationality (IR): Only when all participating clients have nonnegative utility, the overall mechanism is IR. IR shows that clients are hesitant to join FL when the payment is lower than the cost.
- Incentive Compatibility (IC): All participating clients can get the best compensation only when they honestly report the type of resources and costs, and the overall mechanism is IC. In other words, each participating client cannot increase its revenue by submitting false information.
- Weak Budget Balance (WBB): The utility of the FL server is nonnegative, when the FL services transaction is completed.

The proofs of the above properties will be given in Section 5.2.

5.2. The Optimal Client-Selection Scheme

The goal of selecting the best clients is to maximize the social surplus defined by Equation (21) and then solve for the best client selection vector as in (24),

$$\eta^* = \operatorname{argmax}_{\eta \in (0,1)^n} \left[B \odot \eta - \sum_{i=1}^n b_i \eta_i \right]. \quad (24)$$

Because solving the social surplus maximization is an NP-hard problem, we use the DRL algorithm dueling double deep Q network (D3QN) to solve Equation (24) to obtain the best client-selection vector. The D3QN algorithm is a double Q network DRL algorithm with dueling architecture, which combines the double DQN [21] and dueling DQN [22] features to solve the Q overestimation problem while fitting to obtain more accurate Q values. The best client-selection scheme based on D3QN proposed in this paper is shown in Figure 2, and the relevant settings are as follows.

- States: The state of the execution to step m is $s^m = (s_1^m, \dots, s_i^m, \dots, s_n^m)$, $s_i^m = 1$ indicating that client i is selected into the candidate set V ; otherwise, it is not selected.
- Actions: The action a^m in step m is the client number i , which is not in the candidate set V , indicating that the client i is selected by the FL server in the current step.
- State transition: The state transition from s^m to s^{m+1} is determined by the action a^m at step m , i.e., $s_{a^m} = 1$, and a^m will be put into the candidate set V .
- Reward: The reward at step m is the social surplus increased by the action a^m , i.e., $r^m = S(V \cup a^m) - S(V)$. The cumulative reward $R = \sum_{m=1}^M r^m$ is the goal of optimization, which ultimately maximizes the social surplus of the FL task.
- Policy: D3QN uses the evaluation network $Q'(s^m, a^m; w_e)$ to obtain the action corresponding to the optimal action value in the state s^{m+1} , and then uses the target network $Q(s^m, a^m; w_t)$ to calculate the action value of the action to obtain the target value. The interactions between the two networks effectively avoid the algorithm

overestimation problem. The FL server interacts with the environment by using the ϵ -greedy policy, which can be expressed by Equation (25),

$$a^m = \begin{cases} \underset{a^m}{\operatorname{argmax}} Q(s^m, a^m; w_t), & \text{probability } \epsilon \\ \text{randomly select a client in } \mathcal{N}/V, & \text{probability } 1 - \epsilon \end{cases}. \quad (25)$$

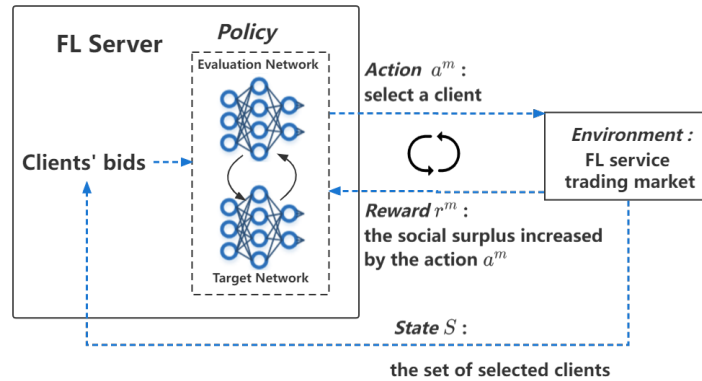


Figure 2. The optimal client selection scheme based on D3QN.

After solving for the optimal client selection vector by training D3QN, it is next proved that the reverse auction proposed in this paper satisfies IC, IR, and WBB.

Proposition 1. *The reverse auction is truthful (incentive compatible).*

Proof. There are two types of misrepresentation by clients in reverse auction.

(1) Client misrepresentation of bid costs.

When the bid cost b_i of the client i is true, its utility is as follows:

$$U_i = S^*(b_i, b^{-i}) - S^{-i*}(b^{-i}). \quad (26)$$

When the client i bid cost \hat{b}_i is not true, the utility is as follows,

$$\bar{U}_i = \bar{S}^*(\hat{b}_i, b^{-i}) - S^{-i*}(b^{-i}) + \hat{b}_i - b_i, \quad (27)$$

where

$$S^*(b_i, b^{-i}) = B \odot \eta^* - \sum b_k \eta_k^* \quad (28)$$

$$\bar{S}^*(\hat{b}_i, b^{-i}) = B \odot \bar{\eta}^* - \sum_{k \neq i} b_k \bar{\eta}_k^* - \hat{b}_i. \quad (29)$$

By substituting (26) and (27), we have

$$U_i - \bar{U}_i = (B \odot \eta^* - \sum b_k \eta_k^*) - (B \odot \bar{\eta}^* - \sum b_k \bar{\eta}_k^*). \quad (30)$$

Because η^* is the optimal solution of Equation (24), the social surplus corresponding to the best client selection vector η^* must be greater than or equal to the other solutions, e.g., $\bar{\eta}^*$. Therefore, $U_i - \bar{U}_i \geq 0$ shows that each client has no motivation to misrepresent the bid cost.

(2) Client misrepresentation of data quality.

Because the payment to the client is set as postpayment, a client that misrepresents the data size d and the EMD distance ϵ will be detected by the negative behavior detection in the reputation mechanism, and the FL server will not pay the client, and will reduce

the reputation of the client at the same time, which will affect the client's subsequent participation in other FL tasks. Therefore, a rational client will not exchange a small amount of interest in a particular FL task at a high cost of reducing reputation, indicating that each client has no motivation to misrepresent data quality.

Combining (1) and (2), it can be proven that the client cannot improve utility in this reverse auction by submitting false data quality and cost information. Therefore, the reverse auction is truthful and satisfies incentive compatibility. \square

Proposition 2. *Reverse auction satisfies individual rationality.*

Proof. The utility of client i is

$$U_i = P_i - b_i = S^*(d, \varepsilon, b) - S^{-i*}(d^{-i}, \varepsilon^{-i}, b^{-i}), \quad (31)$$

where $S^*(d, \varepsilon, b)$ is the maximum social surplus when client i is selected as the best client, $S^{-i*}(d^{-i}, \varepsilon^{-i}, b^{-i})$ is the maximum social surplus when client i is not selected, and the selection of client i means the client can increase the social surplus. Therefore,

$$S^*(d, \varepsilon, b) > S^{-i*}(d^{-i}, \varepsilon^{-i}, b^{-i}) \quad (32)$$

$U_i > 0$ shows that the utility of the client is nonnegative, and the reverse auction satisfies personal rationality. \square

Proposition 3. *Reverse auction satisfies weak budget balance.*

Proof. After selecting the best set of clients, the utility of the FL server is

$$U = B \odot \eta^* - \sum_{i=1}^n P_i \eta_i^* \quad (33)$$

where

$$P_i = S^*(d, \varepsilon, b) - S^{-i*}(d^{-i}, \varepsilon^{-i}, b^{-i}) + b_i \quad (34)$$

$$S^* = B \odot \eta^* - \sum_{i=1}^n b_i \eta_i^*. \quad (35)$$

It can be derived that

$$\sum_{i=1}^n P_i \eta_i^* = S^* + \sum_{i=1}^n b_i \eta_i^* = B \odot \eta^*. \quad (36)$$

Therefore, the utility of the FL server is nonnegative and the reverse auction satisfies the weak budget balance. \square

6. Performance Evaluation

6.1. The Settings of Simulation

To evaluate the performance and effectiveness of the FL incentive mechanism based on the reputation mechanism and reverse auction proposed in this paper, we choose to use the convolutional neural network (CNN) model on the classic MNIST dataset [23] to perform the FL task of handwritten digit recognition. The MNIST dataset is a sample set of handwritten digits containing 0~9, with 60,000 training samples and 10,000 test samples. In this paper, these 10,000 test samples are placed on the FL server side as the test dataset

for detecting the client model, and the distribution of this test set is used as the global data distribution P_g for calculating the EMD distance,

$$P_g = \{p_0 = 0.098, p_1 = 0.1135, p_2 = 0.1032, p_3 = 0.101, p_4 = 0.0982, p_5 = 0.0892, p_6 = 0.0958, p_7 = 0.1028, p_8 = 0.0974, p_9 = 0.1009\}. \quad (37)$$

It can be seen that the occurrence probability of each type of label is approximately equal, which is in line with the real situation of the image-recognition task. The used convolutional neural network is a simple CNN containing two convolutional layers and two fully connected layers.

In order to realistically simulate the heterogeneity of clients in the FL network, a total of 50 clients are set up in the experiment, including 10 clients with reputations below the reputation threshold and 40 clients with reputations above the reputation threshold, where the 40 clients with high reputation consisted of 20 positive clients, 15 negative clients and five malicious clients. The positive clients are clients without negative and malicious behaviors, negative clients are those with deceptive negative behaviors during training, and malicious clients are those that would initiate poisoning attacks. In order to better match the real situation of client non-IID settings, as in [24] we choose the Dirichlet distribution with distribution parameter $\alpha = 1.0$ to randomly assign datasets to each positive client. The smaller the parameter α is, the degree of non-IID is higher. The 15 negative clients that falsify data quality may have no local dataset or be randomly assigned with only a small amount of data containing a certain type of label. The five malicious clients have the same local dataset assignment as that of the positive clients, but the labels in them are replaced and the labels are added 1 and modulo 10 to the real labels, for example, label 1 is replaced with label 2 and label 9 is replaced with 0. Figure 3 shows the data distribution of some clients, and it can be seen that the positive clients cover more sample labels and have higher data quality, whereas the negative clients contain only one type of label and have poor data quality. The data distribution of malicious clients is the same as the positive client, but the labels have all been replaced, which will interfere with the global model update.

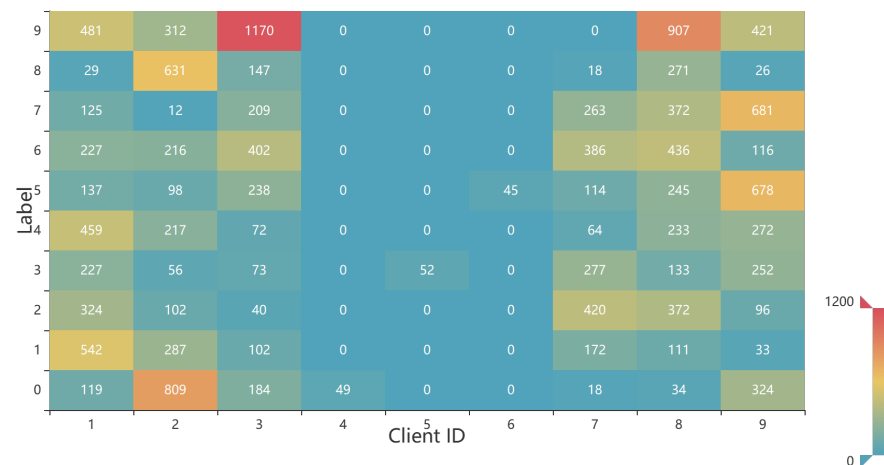


Figure 3. Data distribution of some clients. ID 1~3 are positive clients, ID 4~6 are negative clients, and ID 7~9 are malicious clients.

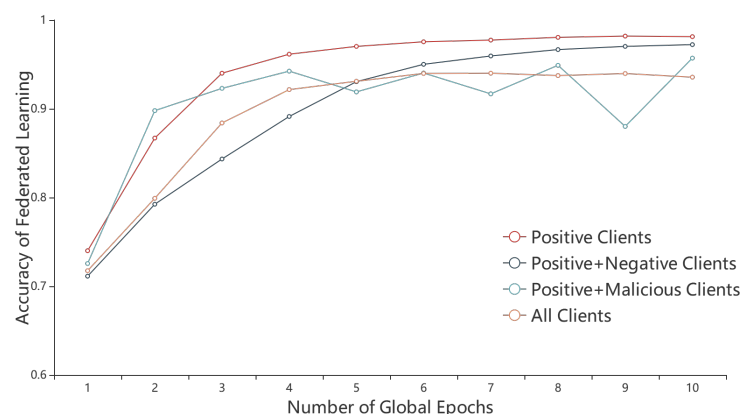
In the simulation, the aggregation algorithm selected in FL is FedAvg, and the optimization algorithm is SGD. More parameters set in the simulation are shown in Table 2.

Table 2. Parameter settings in the experimental simulation.

Parameter	Setting
Learning rate	$lr = 0.01$
Training batch size	batch_size=64
The number of local iteration epochs	$\delta_l = 5$
The number of global iteration epochs	$\delta_g = 5$
The number of CPU cycles to execute one data sample in local training	$c = 5$
The CPU operating frequency of client	$f = 2$
The effective capacitance parameter of the client's computational chipset	$\zeta = 2$
The energy consumption of uploading local model updates for all clients	$E_i^{com} = 20$
The energy consumption weight parameter for client	$\mu = 1$
The satisfaction weight parameter on the accuracy of the global model	$\lambda = 2000$

6.2. Performances of the Reputation Mechanism

We first evaluate the impact of malicious clients and negative clients on the accuracy of the FL task. It can be seen from Figure 4 that the addition of malicious clients and negative clients seriously reduces the accuracy of the FL task. The accuracy of FL is the lowest when all types of clients exist, which is 0.9356, and the highest accuracy rate is 0.9813 in the presence of only positive clients. The global model accuracy rate of the combination of positive clients and negative clients is higher than that of the combination of positive clients and negative clients, and the global model with the combination of positive clients and malicious clients still does not converge after 10 global epochs. Obviously, the influence of malicious clients on FL is much greater than that of negative clients. Meanwhile, it can be seen from Figure 3 that the global model accuracy rate of the combination of positive clients and negative clients reaches 0.97 at the ninth global epoch, and the accuracy rate of the positive clients reaches 0.97 at the fifth round, which is faster than the combination of positive clients and negative clients twice as fast. It can be found that malicious behaviors are harmful behaviors for the FL server, which can seriously reduce the accuracy rate of the global model, whereas negative behaviors are more like useless behaviors because negative behaviors neither interfere with the direction of the global model update as much as malicious behaviors, nor do they contribute to the global model update. Furthermore, negative behaviors slow down the convergence of the global model. At the same time, it will cause great financial loss to the FL server if the negative clients are paid based on their false bidding information.

**Figure 4.** The effect of different types of client combinations on the accuracy of FL.

The detection method for malicious clients has been introduced in Section 4.1.1. For negative clients with falsifying data quality, if the clients perform the negative behavior (b) of directly uploading the global model parameters in the next global epoch, it can be detected directly based on the local parameters uploaded by the clients that are the same as the global parameters. If the clients perform the negative behavior (a) and (c), we will design the detection method based on the performances of this type of clients in the FL training process. In order to simulate the real situation where clients falsify data quality, we divide the 15 negative clients with falsified data quality into five clients executing behavior (a) of randomly generating model parameters to upload to the server, three clients executing behavior b of directly uploading global model parameters in the next global epoch and seven clients executing behavior (c) of training with local low-quality data. We compare the performance of the two types of clients with that of the positive clients in the FL training process as shown in Figure 5. The accuracy rate of each class of clients in Figure 5 is taken from the average of all accuracy rates of clients in that class. As can be seen from Figure 5, the local model accuracy rate of the client performing the behavior (a) is always low, below 0.2, because they upload random model updates. The local model accuracy rate of the client performing the behavior (c) increases with the number of global epochs, and the local model accuracy rate is always lower than that of the positive client in the first five epochs, and then greater than that of the positive client in the fifth round and after, but always lower than the global model accuracy rate. The reason why the accuracy rate of the negative client performing the behavior (c) is closer to the global model accuracy rate than the positive client after the fifth round is that the negative client has much less local data than the positive client and has less influence on the global model update direction than the positive client, so the deviation from the global model is smaller.

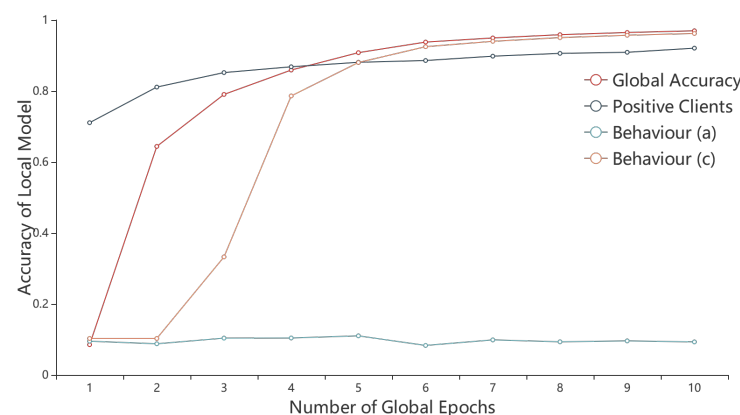


Figure 5. The comparison of local model accuracy between negative clients and positive clients in FL.

The negative clients performing the behavior (a) and (c) are selected for participating in FL only by falsifying the data quality, and it can be seen from Figure 5 that the accuracy rate of the negative clients is much lower than that of the positive clients in the first three epochs compared to the positive clients that are normally selected. Therefore, we can use the K-means clustering algorithm for the detection of negative behavior (a) and (c) in the first three epochs of FL, and cluster the clients in FL by data size, EMD distance, and model accuracy rate, and divide them into positive and negative classes. The effect of cluster detection is shown in Figure 6. The difference between positive and negative clients is obvious. After the FL server detects malicious clients and negative clients, it will refuse to aggregate the model update from the clients, and refuse the clients to continue to participate in FL at the same time.

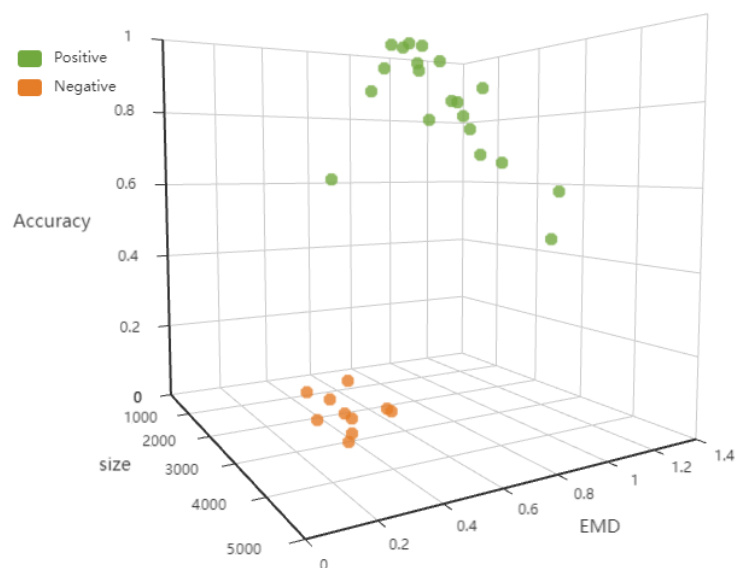


Figure 6. The clustering detection effect diagram of negative behaviors.

6.3. Performances of the Reverse Auction

After selecting reliable clients without malicious clients and negative clients by the reputation mechanism, we will use the D3QN-based reverse auction to select the best set of clients that maximize the social surplus to participate in FL training. To compare with the D3QN-based reverse auction D3QN-auction algorithm proposed in this paper, we also select two algorithms: (i) the greedy-all algorithm, in which all clients selected by the reputation mechanism participate in FL, and (ii) the simple auction algorithm, in which 80% of all reliable clients with lower bid prices are selected to participate in FL. In the simulation experiments, a total of 20 clients are selected by the reputation mechanism to form a set of reliable clients. The D3QN-auction algorithm proposed in this paper selects 14 clients to join the FL training, the greedy-all algorithm selects all 20 clients to join the training, and the simple auction algorithm selects 16 clients to join the training. Figure 7 shows the variation of the accuracy rates of the three algorithms in the FL training. It can be seen from Figure 7 that the accuracy rates of all three algorithms finally stabilize around 0.98, indicating that none of the reliable clients selected by the reputation mechanism has a bad effect on the global model accuracy rate. The D3QN-auction algorithm proposed in this paper reaches the accuracy rate of 0.98 at the earliest and tends to be stable, and the accuracy rate is slightly higher than the other two algorithms, which indicates that the algorithm proposed in this paper can select the minimum set of the reliable clients to achieve the best training effect.

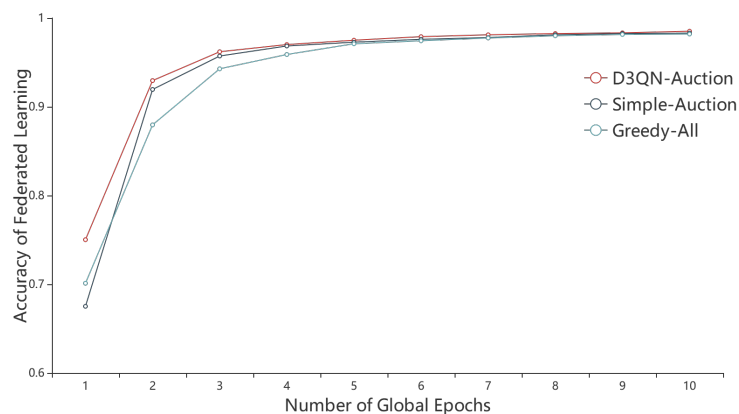


Figure 7. The comparison of the accuracy rates of the three algorithms.

Figure 8 gives the comparison of the three algorithms in terms of social surplus, the average utility of clients and the social surplus per unit cost. It can be seen from Figure 8 that the D3QN-auction algorithm proposed in this paper has the highest social surplus, the average client utility, and the social surplus per unit cost. The algorithm produces much higher economic benefits than both the simple auction and greedy-all algorithms, indicating that the FL server is able to motivate clients to achieve the expected accuracy of FL with less cost. Meanwhile, the highest social surplus and highest social surplus per unit cost indicate that the D3QN-auction algorithm can make the FL service trading market have the highest output ratio, which is 31% higher than the greedy-all algorithm. The clients who obtain the maximum benefit are willing to choose the reverse auction proposed in this paper for payment and settlement, and the increased willingness of clients to participate in FL will make the FL service trading market more active and stable.

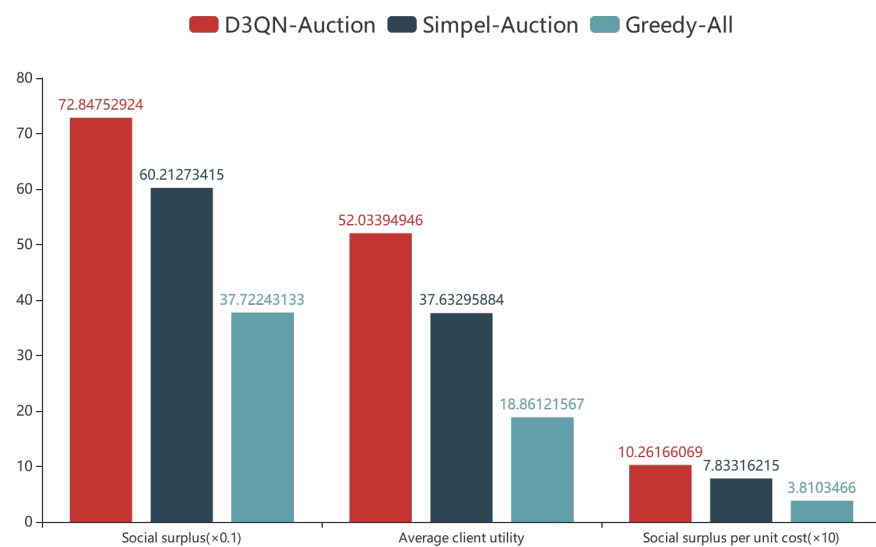


Figure 8. The comparison of economic benefits of the three algorithms.

7. Conclusions

In this paper, we have commercialized the FL service between FL servers and clients in FL, and have considered the FL network as an FL service trading market, focusing on the problem of selection and incentive mechanism of truthful and reliable clients in FL. We have first introduced a reputation mechanism to discriminate and detect the behaviors of clients, and measure the truthfulness and reliability of clients through multiple reputation assessments. And we have designed a scheme for selecting truthful and reliable clients based on the reputation mechanism to ensure that the clients do not have the malicious behaviors of poisoning attacks and negative behaviors of falsifying data quality during the training process. Then, according to the cost and benefit analysis in the FL service trading market, we have designed a reverse auction satisfying IR, IC, and WBB, and have used the DRL algorithm D3QN to select the best set of clients that maximizes the social surplus in the FL service trading market. Finally, experimental simulations have shown that our proposed FL incentive mechanism based on reputation mechanism and reverse auction can achieve a more truthful, reliable, and efficient FL on the basis of ensuring the truthfulness and reliability of participating clients, and at the same time can motivate more clients with high-quality data and high reputation to participate in FL with fewer rewards. The FL incentive mechanism proposed in this paper can significantly improve the economic benefit and the accuracy of FL tasks, which increases the economic benefit by 31% and improves the accuracy from 0.9356 to 0.9813, thus promoting the efficient and stable development of the FL service trading market.

Author Contributions: Conceptualization, A.X. and Y.C.; methodology, A.X. and Y.C.; software, A.X. and Y.C.; validation, A.X. and Y.C.; formal analysis, A.X. and Y.C.; investigation, A.X. and Y.C.; resources, A.X., Y.C., J.C. and S.Y.; data curation, A.X., Y.C. and J.H.; writing—original draft preparation, A.X., Y.C. and H.C.; writing—review and editing, H.C., J.C., S.Y., J.H., Z.L. and S.G.; visualization, A.X. and Y.C.; supervision, H.C., J.C., S.Y., Z.L. and S.G.; project administration, H.C., J.C., S.Y., J.H., Z.L. and S.G.; funding acquisition, H.C., J.H., Z.L. and S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by State Grid Corporation of China Science and Technology Project “Research and application of industry chain finance key technology based on blockchain” (5211DS21NOOU).

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

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

References

1. Zhan, Y.; Li, P.; Guo, S.; Qu, Z. Incentive mechanism design for federated learning: Challenges and opportunities. *IEEE Netw.* **2021**, *35*, 310–317. [CrossRef]
2. Federated Learning: Collaborative Machine Learning Without Centralized Training Data. Available online: <https://ai.googleblog.com/2017/04/federated-learning-collaborative.html> (accessed on 6 April 2017).
3. Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; Ramage, D. Federated learning for mobile keyboard prediction. *arXiv* **2018**, arXiv: 1811.03604.
4. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*; Fort Lauderdale, Florida, USA, 20–22 April 2017; PMLR: New York, NY, USA, 2017; pp. 1273–1282.
5. Zhan, Y.; Zhang, J.; Hong, Z.; Wu, L.; Li, P.; Guo, S. A survey of incentive mechanism design for federated learning. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1035–1044. [CrossRef]
6. Tu, X.; Zhu, K.; Luong, N.C.; Niyato, D.; Zhang, Y.; Li, J. Incentive mechanisms for federated learning: From economic and game theoretic perspective. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1566–1593. [CrossRef]
7. 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.
8. Zeng, R.; Zeng, C.; Wang, X.; Li, B.; Chu, X. A comprehensive survey of incentive mechanism for federated learning. *arXiv* **2021**, arXiv:2106.15406.
9. Feng, S.; Niyato, D.; Wang, P.; Kim, D.I.; Liang, Y.C. Joint service pricing and cooperative relay communication for federated learning. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 815–820.
10. Jiao, Y.; Wang, P.; Niyato, D.; Lin, B.; Kim, D.I. Toward an automated auction framework for wireless federated learning services market. *IEEE Trans. Mob. Comput.* **2020**, *20*, 3034–3048. [CrossRef]
11. Kang, J.; Xiong, Z.; Niyato, D.; Lin, B.; Kim, D.I. 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]
12. Shehadeh, D.; Gawanmeh, A.; Yeun, C.Y.; Zemerly, M.J. Fog-based distributed trust and reputation management system for internet of things. *J. King Saud- Univ.-Comput. Inf. Sci.* **2022**, *34*, 8637–8646. [CrossRef]
13. Din, I.U.; Awan, K.A.; Almogren, A.; Kim, B.S. ShareTrust: Centralized trust management mechanism for trustworthy resource sharing in industrial Internet of Things. *Comput. Electr. Eng.* **2022**, *100*, 108013. [CrossRef]
14. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582.
15. Shayan, M.; Fung, C.; Yoon, C.J.M.; Beschastnikh, I. Biscotti: A ledger for private and secure peer-to-peer machine learning. *arXiv* **2018**, arXiv:1811.09904.
16. Fung, C.; Yoon, C.J.M.; Beschastnikh, I. Mitigating sybils in federated learning poisoning. *arXiv* **2018**, arXiv:1808.04866.
17. Sheugh, L.; Alizadeh, S.H. A note on pearson correlation coefficient as a metric of similarity in recommender system. In Proceedings of the 2015 AI & Robotics (IRANOPEN), Qazvin, Iran, 12 April 2015; pp. 1–6.
18. Yang, M.; Hu, X.X.; Zhang, Q.H.; Wei, J.H.; Liu, W.F. Federated learning scheme for mobile network based on reputation evaluation mechanism and blockchain. *Chin. J. Netw. Inf. Secur.* **2021**, *7*, 99–112.
19. Tran, N.H.; Bao, W.; Zomaya, A.; Nguyen, M.N.; Hong, C.S. Federated learning over wireless networks: Optimization model design and analysis. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1387–1395.

20. Kang, J.; Xiong, Z.; Niyato, D.; Yu, H.; Liang, Y.C.; Kim, D.I. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In Proceedings of the 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), Singapore, 28–30 August 2019; pp. 1–5.
21. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
22. Wang, Z.; Schaul, T.; Hessel, M.; Lanctot, M.; Freitas, N. Dueling network architectures for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; PMLR: New York, NY, USA, 2016; pp. 1995–2003.
23. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
24. Hsu, T.M.H.; Qi, H.; Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv* **2019**, arXiv:1909.06335.

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.