A Contract Theory Based Incentive Mechanism for Federated Learning

Yuan Liu, Mengmeng Tian, Yuxin Chen, Zehui Xiong, Cyril Leung, and Chunyan Miao

Abstract Federated learning (FL) serves as a data privacy-preserved machine learning paradigm, and realizes the collaborative model trained by distributed clients. To accomplish an FL task, the task publisher needs to pay financial incentives to the FL server offloads the task to the contributing FL clients. However, it is challenging to design proper incentives for FL clients because the task is privately trained by the clients. This paper proposes a contract theory-based FL task training model toward minimizing the incentive budget subject to clients being individually rational and incentive compatible in each FL training round. We design a two-dimensional contract model by formally defining two private types of clients, namely data quality and computation effort. To effectively aggregate the trained models, a contract-based aggregator is proposed. We analyze the feasible and optimal contract solutions for the proposed contract model. The experimental results demonstrate that the proposed

Y. Liu (✉)
Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong, China
e-mail: yuanliu@gzhu.edu.cn

M. Tian · Y. Chen
Software College of Northeastern University, Shen Yang, Liao Ning, China
e-mail: tianmm866@163.com

Y. Chen
e-mail: 2001236@stu.neu.edu.cn

Z. Xiong
Pillar of Information Systems Technology and Design, Singapore University of Technology Design, Singapore, Singapore
e-mail: zehui_xiong@sutd.edu.sg

C. Leung
Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly(LILY), Nanyang Technological University, Singapore, Singapore
e-mail: cleung@ece.ubc.ca

C. Miao
School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: ascymiao@ntu.edu.sg

R. Razavi-Far et al. (eds.), *Federated and Transfer Learning*, Adaptation, Learning, and Optimization 27, https://doi.org/10.1007/978-3-031-11748-0_6

framework and contract model can effectively improve the generation accuracy of FL tasks. Moreover, the generalization accuracy of the FL tasks can be improved by the proposed incentive mechanism, the contract-based aggregation is applied.

1 Introduction

With the ubiquitous adoption of internet-connected smart devices and applications, the volumes of private data are increasing at an unprecedented speed. In a traditional data-driven machine learning paradigm, such large data volumes are stored and analyzed on a third-party Cloud server, benefiting from its advantages of computing and storage capacities. However, with the ever-rising data privacy issue in both academics and industry, this centralized paradigm becomes unpractical. In this context, federated learning (FL) was proposed in [2, 13] and has emerged as a potential solution to address this privacy issue, where the private data are stored and used to train a model at end-devices locally.

In a classical FL framework, an FL server posts a target model, referred as an FL task, to be collaboratively trained by distributed FL clients. To attract the FL clients actively participating in training the FL task and compensate their efforts in task execution, the FL server must offer sufficient economic incentives for FL clients [14]. There are Many previous papers have investigated the design of the incentive mechanism for federal learning [4, 6, 11, 12, 21], such as contract theory-based mechanisms [9] with single dimension and [3, 17] with multiple dimensions. However, these studies only considered the effect of the local model and ignored its final rendering effect in the global scope(i.e., on the server).

In the existing contract-based solutions for FL, the data quality toward improving model generalization accuracy [7] is rarely discussed, which is an essential performance metric of a deep learning model in a global context. Moreover, all these contract models have only studied the adverse selection issue by only measuring the value of the local client data(i.e., data size). The FL server offers tasks and contracts to be chosen by clients according to their types. While these papers only considered the maximal value of local data, they ignored the role of the incentive mechanism on the computing behavior of the client. That is, the clients may not always put their efforts into executing FL tasks, resulting in a moral hazard issues. In this study, we aim to design a two-dimensional contract incentive model considering clients' data quality in model generation performance and effort willingness.

The main contributions of this study are summarized as follows.

- A contract theory-based FL procedure is proposed, including eight main processes, to support contract base incentive mechanisms for FL platforms.
- A multi dimensional contract model is designed by formally constructing the two private types of FL clients(i.e., generation type and effort willingness). The measurement of data quality is based on the characteristics of client coverage. In addition, the calculation behavior is optimized by the calculation cost of the client.

- The utilities of clients and the FL server are formalized to solve an optimal contract solution with maximal value of two sides of FL. At the same time, the FL server foucuses on not only economic benefits but also the final aggregation effect of the model. We provide the best aggregate utility based on maximum economic utility.
- A contract-based aggregation scheme is designed to improve the model generalization accuracy. The experimental results based on the the MNIST dataset show that the proposed contract-based incentives and aggregation scheme outperforms other schemes in a single FL training round.

2 Related Work

The design of incentive mechanisms for FL based on game theoretical approaches has been extensively studied [12, 15]. Due to the incomplete information scenario in FL, the FL process is generally modeled as an adverse selection dynamic game model in which FL offers tasks and FL clients choose tasks as they wish. The existing incentive mechanisms can be divided into two main categories: Stackelberg game-based and contract theory-based methods. In the first category, the FL server offers a task associated with a price and clients choose a task and take efforts in training it to achieve the pricing rewards [4, 6, 11]. In [6], a two-stage Stackelberg game was formalized for FL with private data, and a Nash equilibrium (NE) was solved with the optimal privacy budget of clients and the optimal pricing scheme of the server. In [4], the interactions between the model requester and mobile user were formalized as a Stackelberg game to analyze the NE composed of the optimal training data price and data size. In [11], a Stackelberg game-based incentive mechanism was designed for FL clients for strategically setting the local iterations and maximizing task the global accuracy of the FL task. The shortcoming of Stackelberg game-based methods is that they can only consider single-dimension private strategical types.

In the second category, FL servers offer a set of contract items according to the client types, where the contract models can be single-dimensional or multi dimensional depending on the dimensions of the considered type. In [10], the client training quality type was formalized based on computation resources, and an optimal contract solution offered more rewards to clients with higher type values. Considering clients' communication delay and local training cost, [3] introduced a contract-based incentive mechanism to maximize server aggregation accuracy and total payments, where the local training cost was related to clients' network environment. [18] considered both data quality and model computation resources. A recent study [3] investigated a two-dimensional contract model considering data quality in terms of data sizes and communication time types and analyzed the optimal contract solutions in three scenarios: complete information, weakly incomplete information, and strongly incomplete information. There are two limitations in the existing contract-based methods: (1) they neglect to consider the data quality in the aspects of improving model generalization accuracy, which will be studied as an important private type of clients; (2) they assume that the clients take their efforts in executing FL tasks, which bears the designed contracts with the moral hazard issue, where their willingness may hinder the achievement of their optimal solution.

3 System Model

We consider a classical FL platform where an FL task is proposed by a task requester and delegated to a trusted FL server, and the FL server coordinates the FL task distributed and trained by a set of FL clients. The FL clients participate in the FL task training under incomplete information , where they privately train the task model according to their private type, namely their local data, and the FL server cannot observe the clients' behaviors or private types but is aware of the private type distribution through statistics. To effectively incentivize FL clients executing tasks to play the role of generalization of local data as much as possible, we propose a contract theory incentive mechanism-based FL procedure, as shown in Figure 1. It includes eight main processes. Specifically, a task requester posts a model task to the FL server, and the server calculates a set of contracts for the task. The server then publishes the task in the client network, and the clients can choose to sign a contract by registering to the task according to the chosen contract. The clients then take efforts in training the task model based on their private datasets. When a qualified local model is trained, a client can submit the model to the FL server, and the server pays the clients according

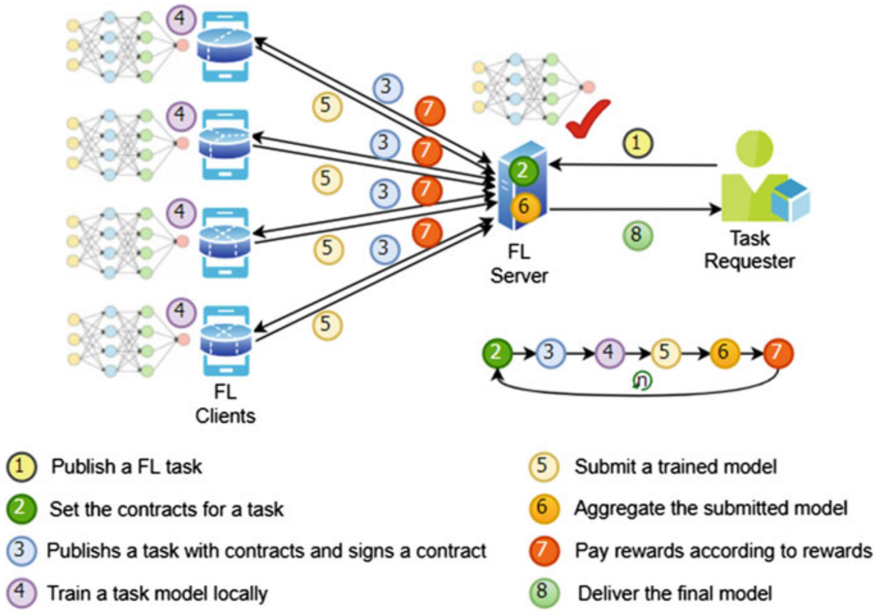

Fig. 1 Overview of the proposed contract theory-based FL procedure

to the corresponding contracts. The server then aggregates the submitted model according to a contract-based aggregation scheme. In the following section, we first formulate the proposed contract, and then specify the utility functions of the clients and the server, and finally introduce the design of contract-based aggregation.

3.1 Contract-Based FL

private information of FL clients:
local data & computational behavior

In the proposed system, FL clients have private information; that is, the local data of the client and the computational behavior on their local data and the FL server cannot predict the client's behavior. We aim to design a contract mechanism to elicit such private information.

The maximum effect that client's local data can exert on FL performance depends on the number of features owned by the local data. We use the coverage of feature quantity to characterize its ability, which is called data coverage quality. Evidently, the client's local data with a larger coverage feature space can exhibit better performance without considering its computational behavior. First, we classify the client according to its data coverage quality, and then design the contract according to its data coverage quality and model performance, namely training willingness.

Suppose there are I types of clients whose data coverage quality types are sorted in an ascending order: $\theta_1 \leq \cdots \leq \theta_i \leq \cdots \theta_I$. For each FL iteration round, the FL server needs to propose a contract set $\Phi = \{\phi_i = (f_i, R_i(f_i))|i \in \{1, \ldots, I\}\}$ to specify the relationship between clients' rewards and registration fee for each client type, where f_i is the registration fee for clients in ith type to accept a task and $R_i(f_i)$ is the corresponding rewards. The design of f_i aims to ensure the participation of rational clients before accepting a contact, and the clients will not participate if they are unable to execute the task. Then, the server broadcasts the contract set among the clients, and each client signs a contract according to its type. The clients then start training the model based on their local data and finally submit the trained model within the required time. The server creates a test about the generalization accuracy of the submitted models. The clients are rewarded with $R_i(f_i)$ if the model passes the test by reaching the corresponding generalization accuracy M_i, and nothing otherwise. For the clients not rewarded, their registration fee will not be returned and used as their penalty for breach of contract.

The model aggregation of the server process can be iterated for n rounds, and the contract set should be set for each round. For a consecutively following round of a task, the generalization accuracy threshold M_i should be set with a marginal increase. In this paper, without loss of generality, we design the contracts for a single aggregation round.

Next, we formalize the types of clients and their utility.

to characterize clients with ability and effort:
data coverage quality and willingness

3.2 Data Coverage Quality with Adverse Selection

In the context of FL, to protect the data privacy of FL clients, the clients are required to provide the trained models to the server instead of directly sharing their local data. Because of the asymmetric information, the quality of the model uploaded by clients cannot be verified, which raises the adverse selection issue [1]. On the other hand, the lower bound of a model's performance in term of accuracy has a positive linear relationship with the local data coverage [7]. Therefore, the local data of a client is closely related to the benefit of the central server, and it can reflect the worst accuracy that the model can achieve on the local data. Thus, we characterize the client type with the generalization ability [7] of the local dataset, which is qualified by the expected feature coverage. We then model the client private type based on the local data quality from the aspect of model generalization capacity [7].

Suppose the feature space with d dimension is denoted by $D = [0, 1]^d$, which is a unit space, and a subspace $\mathcal{A} \in D$ denotes the local data from one of the FL clients. Let $\mu(\mathcal{A})$ be the probability that a random sample in D covered by D and $\mu(\mathcal{A}) = 1$ when $\mathcal{A} = D$. $\mu(D) = 1$ represents the most perfect data that can cover all features. In other words, the model calculated on the total feature space D can be successfully used for any sample judgment as long as it has been fully calculated with maximal training willingness.

For local data of the client $\mathcal{A} = \{x_1, x_2, \ldots, x_n\} \subset D$, we use the concept of neighborhood to describe the feature coverage of the dataset. The relevant definitions are as follows.

Definition 1 (*ϵ-Data Coverage*) With a certain radius ϵ, the coverage of a data set $\mathcal{A}$ consisting of samples $x_1, ...x_n$ is measured by

$$\mu(\mathcal{A}, \epsilon) = D \cap \cup_{x_i \in \mathcal{A}} B(x_i, \epsilon) \tag{1}$$

where $B(x_i, \epsilon)$ is an open ball space centered at x_i with radius ϵ.

Apparently, with the limitation of the range of ϵ, $\mu(\mathcal{A}, \epsilon)$ is a non-decreasing function. Suppose the data space is a Euclidean space, then the range of ϵ is $[0, \sqrt{d}]$. Considering the expected effect of the local dataset, we presents the following definition to show the data quality of the clients.

Definition 2 (*Data Coverage Quality*) The data coverage quality of a local dataset $\mathcal{A}$ is denoted by $\theta(\mathcal{A})$ to be measured as the expected coverage:

$$\theta(\mathcal{A}) = \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu(\mathcal{A}, \epsilon) \mathrm{d}\epsilon. \tag{2}$$

The more features the data cover, the more things the model can learn from the data. Thus the expected feature coverage θ reflects the quality of the local dataset. We consider a concrete set of data coverage quality, denoted by $\Theta = \{\theta_1, ...\theta_I\} \in [0, 1]$, with I types, where the clients with $\theta(\mathcal{A}) \in [\frac{i-1}{I}, \frac{i}{I}]$ belong to type i.

3.3 Training Willingness with Moral Hazard

The client node consumes its local resource to complete an FL task, and the efforts incurred in training are private information, bearing moral hazard issue [5]. We model the second type of client as training willingness characterized by its efforts in training an FL task.

Definition 3 (*Training Willingness*) The training willingness of a client, denoted by $e \in [0, 1]$, is the extent to which the client tasks efforts in a task training.

The training cost of a client is especially determined by its training willingness, which is measured by a convex function [18, 19] as

$$\Phi(e) = \frac{c}{2}e^2 \tag{3}$$

where c represents the unit cost in a given task training environment, such as IoT devices, smart mobile, PC, or server. Without loss of generality, we denote the training willingness of clients in i-th quality type as e_i. The training effort of the client, the consumption of resources, will eventually be manifested in the accuracy of the local model. Therefore, without loss of generally, e is used to represent the accuracy of the local model with $e_i \in [0, 1]$.

3.4 Utility of Client

With a two-dimensional private information θ_i and e_i, the cost of a client in the ith type is

$$C(\theta_i, e_i) = f_i + \frac{c}{2}e_i^2 \tag{4}$$

The client is rewarded with $R_i(f_i)$ if the trained model passed the generalization test benchmark M_i. The probability of passing the test is determined by the data coverage quality and training willingness [20].

Whether the client can get the reward depends on whether the model satisfies the precision benchmark test M_i. For simplicity, let us assume that the client with data quality θ can generate the service fee with θe probability after making efforts e. The selected service node i's utility is given as

$$U_i = \theta_i e_i R_i - f_i - \frac{c}{2}e_i^2. \tag{5}$$

Notice that regardless of whether the client node passes the contract verification or not, he will spend a total expense $C(\theta_i, e_i)$, and whether he can get the reward of the server depends on its local dataset and the cumulative success rate of the local calculation with $\theta_i e_i$.

3.5 Utility of Server

The local model uploaded by the client of type i will generate a revenue for the FL server, denoted by $G(M_i)$, satisfying $G'(M) > 0$ and $G''(M) > 0$. Similar to the client, θ and e determine whether the server aggregates a local model in probability θe. Thus, the utility from one type i of the selected service node is given. Therefore, the utility of the server from enrolling a client in type i is

$$U_s^i = f_i + \theta_i e_i (G(M_i) - R_i) \tag{6}$$

Given the type distribution of clients $\{\beta_i\}$ with $i \in \{1, ...I\}$ and $\sum_{i=1}^{I} \beta_i = 1$, the expected utility of the server is

$$U_s = \sum_{i=1}^{I} \beta_i U_s^i = \sum_{i=1}^{I} \beta_i (f_i + \theta_i e_i (G(M_i) - R_i)) \tag{7}$$

3.6 Contract Optimization Problem

We design contract incentives for the FL server to optimize its utility. At the same time, we also need to meet the benefit maximization constraints (IC) and participation constraints (IR) of the client. The contract optimization problem is formalized as

$$\begin{aligned}
&\max \sum_{i=1}^{I} \beta_i (f_i + \theta_i e_i (G(M_i) - r_i)) \\
&s.t. \\
&(IR) \theta_i e_i R_i - f_i - \frac{c}{2} e_i^2 \geq 0 \\
&(IC) \theta_i e_i R_i - f_i - \frac{c}{2} e_i^2 \geq \theta_i e_i^j R_j - f_j - \frac{c}{2} (e_i^j)^2 \\
&\qquad \forall j \neq i,\ i, j \in \{1, \ldots, n\}
\end{aligned} \tag{8}$$

where e_i^j denotes the effort of type θ_i when selecting contract (f_j, R_j).

The first constraint ensures that each client can achieve non-negative utility, which is also regarded as individual rationality property (IR). The second constraint aims to ensure that each client can achieve their maximal utility by choosing the contract corresponding to their truthful type, which is regarded as incentive compatibility property (IC). Constraints such as IC and IR grantee that the client nodes try their best to complete the FL task unless they do not want to maximize their utility.

3.7 Contract-Based Model Aggregation

In addition to economic benefits, the server should also use economic incentives to achieve the best model aggregation effect. Since the economic incentives selected by the client can show the quality of the local model, we can aggregate the global model according to this information.

3.7.1 Client Model Training

After the selection of contract finishing, the client in type order i will compute model w_i

$$w_i^{local} \leftarrow w_i^{local} - \eta \nabla l(w_i^{local}; b) \tag{9}$$

where η denotes the learning rate and $\nabla l(w_i; b)$ is the gradient of loss function $l(w_i^{local}; b)$ on batch b. Then, the server will aggregate the uploading model if w_i^{local} has passed the test benchmark accuracy M_i of clients in i-th type.

3.7.2 Global Model Aggregation

The server with a set of submitted models should aggregate the models based on their chosen contracts for better model generalization performance. Suppose the total rewards paid by a server in a round is R_{total}, then the weight assigned for a model trained by a client in type i is calculated.

$$W_i^{server} = \frac{R_i}{R_{total}} \tag{10}$$

4 Optimal Contract Solution

In this section, we solve the optimal contract solution to the problem defined in Eq. (8). We first solve the optimal effort willingness made by clients and then calculate the contract solution by maximizing the server utility.

Given the utility of a client in ith type in Eq. (5), we compute the first-order derivative with respect to its effort willingness and obtain

$$\frac{\partial U_i}{\partial e_i} = \theta_i R_i - c e_i \tag{11}$$

A rational client node should always maximize its utility by making the optimal willingness, which is denoted by $\hat{e}_i$ and $\hat{e}_i^j$, in choosing contract $\phi_i = (f_i, R_i(f_i))$ and contract $\phi_j = (f_j, R_j(f_j))$. Thus,

$$\begin{cases} \hat{e}_i = \frac{1}{c}\theta_i R_i \\ \hat{e}_i^j = \frac{1}{c}\theta_i R_j \end{cases}$$

According to the above equations, we can know that a client's willingness is positively determined by the data quality and the chosen contract reward.

Combining the two equations into Eq. (8), the objective function is updated as follows.

$$\begin{aligned} &\max \sum_{i=1}^{n} \beta_i (f_i + \frac{1}{c}\theta_i^2 R_i (G(M_i) - R_i)) \\ &s.t. \\ &(IR)\frac{1}{2c}(\theta_i R_i)^2 - f_i \geq 0, \\ &(IC)\frac{1}{2c}(\theta_i F_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_j)^2 - f_j, \\ &\quad \forall j \neq i,\ i, j \in \{1, \ldots, n\} \end{aligned} \tag{12}$$

The object function (12) reflects the optimal effort the client puts during contract design. Next, we solve the optimal contract solution for each type θ_i. Some important conditions are derived.

Then, we show the relationship between the client type and the corresponding contract item (i.e., the registration fee and the reward).

Lemma 1 (Monotonicity between θ and R) *For any feasible contract (f_i, R_i), $R_i \geq R_j \Leftrightarrow \theta_i \geq \theta_j$.*

Proof of Lemma *For clients of types θ_i and θ_j, the following two IC constrains should be satisfied:*

$$\frac{1}{2c}(\theta_i R_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_j)^2 - f_j \tag{13}$$

$$\frac{1}{2c}(\theta_j R_j)^2 - f_j \geq \frac{1}{2c}(\theta_j R_i)^2 - f_i \tag{14}$$

By adding the above two inequalities, we have

$$(\theta_i^2 - \theta_j^2)(R_i^2 - R_j^2) \geq 0 \tag{15}$$

$$\Rightarrow (\theta_i - \theta_j)(R_i - R_j) \geq 0 \tag{16}$$

for any $\theta_i, \theta_j > 0$ and $R_i, R_j > 0$. □

Lemma 1 implies that a client with a higher type θ is fit for a higher rewards R. Thus, the contract rewards should follow the order $R_1 < \cdots < R_n$ with $\theta_1 < \cdots < \theta_n$. Lemma 1 shows the relationship between the client's ability(i.e., its type) and the reward fee it should get. The client with a higher ability will get more rewards. Similarly, we can get the relationship between the client type and registered fees.

Lemma 2 (Monotonicity between R and f) *For any feasible contract* (f_i, R_i), $R_i \geq R_j \Leftrightarrow f_i \geq f_j$.

Proof of Lemma *The IC constraint holds when a client in type* θ_i *chooses contract* (f_i, R_i) *over* (f_j, R_j).

$$\begin{aligned} &\frac{1}{2c}(\theta_i R_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_j)^2 - f_j \\ &\Rightarrow \begin{cases} f_i - f_j \leq \frac{\theta_i^2}{2c}(R_i^2 - R_j^2) \\ f_j - f_i \geq \frac{\theta_j^2}{2c}(R_j^2 - R_i^2) \end{cases} \end{aligned} \tag{17}$$

According to Eq. (17), *if* $f_i \geq f_j$, *then we have* $R_i \geq R_j$ *and for* $R_j \geq R_i$, *according to inequality* (17), *we can hold* $f_j \geq f_i$.

Lemma 2 shows that R and f have the same trend, namely $R_1 < \cdots < R_n$ with $f_1 < \cdots < f_n$. Similar to Lemma 1, we can see the relationship between transaction fee and registration fee. The higher the registration fee, the higher is the reward.

According to the above lemmas, we can hold the monotonicity between f and θ.

Corollary 1 (Monotonicity between f and θ) *For any feasible contract* (f_i, R_i), $f_i \geq f_j \Leftrightarrow \theta_i \geq \theta_j$.

Proof of Corollary *According to Lemmas 1 and 2, both* θ *and* f *monotonically increase with* R. *Thus, we can derive a positive correlation between* θ *and* f. □

The above lemmas and corollary show the monotonicity properties of feasible contracts. Next, we determine the optimal contract by reducing the IR and IC constraints.

Theorem 1 (IR transitivity) *All IR constraints can be satisfied if the constrain of* θ_1 *is satisfied.*

Proof of Theorem *For any client in type* $i \in \{1, \ldots, I\}$ *and* $i \geq 1$, *we have*

$$\begin{aligned} U_i = \frac{1}{2c}(\theta_i R_i)^2 - f_i &\geq \frac{1}{2c}(\theta_i R_1)^2 - f_1 \\ &\geq \frac{1}{2c}(\theta_1 R_1)^2 - f_1 = U_i \end{aligned} \tag{18}$$

and its utility is monotonous. □

Theorem 2 (Tight IC Constrain) *The following IC constraint is sufficient for the client in type* θ_i *to achieve its maximal utility.*

$$\frac{1}{2c}(\theta_i R_i)^2 - f_i = \frac{1}{2c}(\theta_i R_{i-1})^2 - f_{i-1} \tag{19}$$

where $i \in \{2, ..., I\}$.

Proof of Theorem *The following proof is organized in three parts. First, we reduce the redundant IC constraints in two directions: client in type* θ_i *selects the contracts* ϕ_{i+1} *and* ϕ_{i-1}*, respectively. Then, all redundant constraints will be eliminated, leaving only tight constraints* (19). □

(1) Downward Selection: Selecting a contract with a lower type, we have the following equations:

$$\frac{1}{2c}(\theta_{i+1} R_{i+1})^2 - f_{i+1} \geq \frac{1}{2c}(\theta_{i+1} R_i)^2 - f_i \tag{20}$$

$$\frac{1}{2c}(\theta_i R_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_{i-1})^2 - f_{i-1} \tag{21}$$

Substituting (20) into (21),we have

$$f_{i+1} - f_i \leq \frac{1}{2c}\theta_{i+1}^2(R_{i+1}^2 - R_i^2) \tag{22}$$

$$f_i - f_{i-1} \leq \frac{1}{2c}\theta_i^2(R_i^2 - R_{i-1}^2) \tag{23}$$

Since $\theta_i < \theta_{i+1}$, according to (23), we have

$$\begin{aligned}\frac{1}{2c}\theta_i^2(R_i^2 - R_{i-1}^2) &\leq \frac{1}{2c}\theta_{i+1}^2(R_i^2 - R_{i-1}^2) \\ \Rightarrow f_i - f_{i-1} &\leq \frac{1}{2c}\theta_{i+1}^2(R_i^2 - R_{i-1}^2)\end{aligned} \tag{24}$$

Adding (22) and (24), we have

$$f_{i+1} - f_{i-1} \leq \frac{1}{2c}\theta_{i+1}^2(R_{i+1}^2 - R_{i-1}^2) \tag{25}$$

$$\begin{aligned}\Rightarrow \frac{1}{2c}(\theta_{i+1} R_{i+1})^2 - f_{i+1} &\geq \cdots \\ &\geq \frac{1}{2c}(\theta_{i+1} R_{i-1})^2 - f_{i-1}\end{aligned} \tag{26}$$

Then, we can obtain all downward IC constraints:

$$\begin{aligned}&\frac{(\theta_{i+1} R_{i+1})^2}{2c} - f_{i+1} \geq \frac{(\theta_{i+1} r_{i-1})^2}{2c} - f_{i-1} \geq \cdots \\ &\geq \frac{(\theta_{i+1} r_1)^2}{2c} - f_1\end{aligned}$$

Therefore, all adjacent types of downward IC selections are enough to drive all other downward selection.
(2) Upward Selection: Selecting a contract with a higher type, we have the following equations:

$$\frac{1}{2c}(\theta_{i-1}r_{i-1})^2 - t_{i-1} \geq \frac{1}{2c}(\theta_{i-1}r_i)^2 - t_i \tag{27}$$

$$\frac{1}{2c}(\theta_i r_i)^2 - t_i \geq \frac{1}{2c}(\theta_i r_{i+1})^2 - t_{i+1} \tag{28}$$

Substituting (27) into (28), we have

$$t_i - t_{i-1} \geq \frac{1}{2c}\theta_{i-1}^2(r_i^2 - r_{i-1}^2) \tag{29}$$

$$\begin{aligned} t_{i+1} - t_i &\geq \frac{1}{2c}\theta_i^2(r_{i+1}^2 - r_i^2) \\ &\geq \frac{1}{2c}\theta_{i-1}^2(r_{i+1}^2 - r_i^2) \end{aligned} \tag{30}$$

Adding (29) and (30), we have

$$t_{i+1} - t_{i-1} \geq \frac{1}{2c}\theta_{i-1}^2(r_{i+1}^2 - r_{i-1}^2) \tag{31}$$

$$\Rightarrow \frac{1}{2c}(\theta_{i-1}r_{i-1})^2 - t_{i-1} \geq \frac{1}{2c}(\theta_{i-1}r_{i+1})^2 - t_{i+1} \tag{32}$$

$$\begin{aligned} \frac{1}{2c}(\theta_i R_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_{i+1})^2 - f_{i+1} \geq \ldots \\ \geq \frac{1}{2c}(\theta_i R_n)^2 - f_n \end{aligned}$$

Therefore, we can derive other IC constraints from an adjacent upward type selection.
(3) Tight IC constraints: From the above two steps, the IC constrain of adjacent is left. We further remove the redundant restrictions. The tight IC constraints can be replaced by the adjacent downward IC constrain and monotonicity, as follows.

Proposition 1 *The downward IC constraints can guarantee the upward IC constraints.*

Proof of Proposition *According to Lemmas 1 and 2, we have*

$$\frac{1}{2c}(\theta_i R_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_{i-1})^2 - f_{i-1} \tag{33}$$

$$\frac{1}{2c}(\theta_i R_i)^2 - f_i \geq \frac{1}{2c}(\theta_i R_{i+1})^2 - f_{i+1} \tag{34}$$

According to Eq. (34), we have

$$f_{i+1} \geq \frac{1}{2c}\theta_i^2(R_{i+1}^2 - R_i^2) + t_i \tag{35}$$

$$\Rightarrow f_i \geq \frac{1}{2c}\theta_{i-1}^2(R_i^2 - R_{i-1}^2) + R_{i-1} \tag{36}$$

and according to Eq. (33), we have

$$f_i \leq \frac{1}{2c}\theta_i^2(R_i^2 - R_{i-1}^2) + f_{i-1} \tag{37}$$

Comparing Eqs. (36) and (37), given $\theta_i > \theta_{i-1}$, we can derive that the server will achieve its maximal utility when f_i is assigned its maximal value. So the tight IC constrain is Eq. (19). □

Taking the tight constraints into the objective function in Eq. (12), we can formulate the Lagrange function to solve the final optimal contract:

$$\begin{aligned}\mathcal{L}=&\sum_{i=1}^{I}\left\{\left[\beta_i(f_i + \frac{1}{c}\theta_i^2 r_i(G(M_i) - R_i))\right] + \right.\\ &\left.\lambda_i\left[\frac{1}{2c}(\theta_i R_i)^2 - \frac{1}{2c}(\theta_i R_{i-1})^2 - f_i + f_{i-1}\right]\right\}\\ &+\mu\left[\frac{1}{2c}(\theta_1 R_1)^2 - f_1\right]\end{aligned} \tag{38}$$

where λ_i is the Lagrange multiplier of the IC constraint for θ_i, and μ is that of the IR constraint for θ_1.

Thus, the optimal value of R is

$$R_i = G(M_i), \forall i \in \{1, \ldots, I\}$$

and the optimal value of f can be derived according to IR transitivity and IC transitivity.

$$\begin{aligned}f_1 &= \frac{1}{2c}(\theta_1 R_1)^2\\ f_i &= \frac{1}{2c}(\theta_i R_i)^2 - \frac{1}{2c}(\theta_i R_{i-1})^2 + f_{i-1}, \forall i \in \{2, \ldots, I\}\end{aligned}$$

The above solution is the optimal contract solution obtained after relaxing the constraints. We need to further verify whether the solution satisfies the monotonicity condition of R. If R is not monotonic, the adjustment algorithm [16] can be applied.

5 Experimental Results and Analysis

First, we compare the effect of the incentive mechanism with the same reward and the economic incentive effect of the incentive mechanism based on the contract proposed in this paper. The comparison shows the role of incentive from the perspective of client and server. Then, we observe the role of contract-based incentive mechanism in the global model. We show the effects of different aggregation methods under the same incentive mechanism and contract incentive mechanism.

5.1 Basic Setting

We use MNIST datasets to show the economic benefits of clients and servers. Then, MNIST and CIFARdatasets are used to show the aggregation effect.

There are 10 types of contracts, and 10 types of clients are set up to according to the corresponding contracts. The types of clients follow a uniform distribution(i.e., $\lambda_i = 0.1$). The contract and client settings are shown in Table 1. The data in this paper only adopts the strategy of quantity skew by varying clients' data size [8].

5.2 Economic Experimental Settings

In this part, we show the economic effectiveness of the proposed incentive mechanism from two aspects. First, for 10 different types of clients, 10 different types of contracts designed by changing 10 types of clients into 10 types of clients are selected, respectively.

5.3 Economic Results

The benefits of the client are shown in Fig. 2.

Table 1 Contract and client settings of MNIST task

Type index / Parameters	1	2	3	4	5	6	7	8	9	10
Client type θ_i	0.790	0.795	0.800	0.805	0.810	0.815	0.820	0.825	0.830	0.835
Client data size	1000	1500	2000	2500	3500	5000	6500	8500	12000	16000
Optimal effort e_i	0.279	0.331	0.389	0.451	0.519	0.592	0.670	0.753	0.842	0.936
Test generalization M_i	0.230	0.250	0.270	0.290	0.310	0.330	0.350	0.370	0.390	0.410

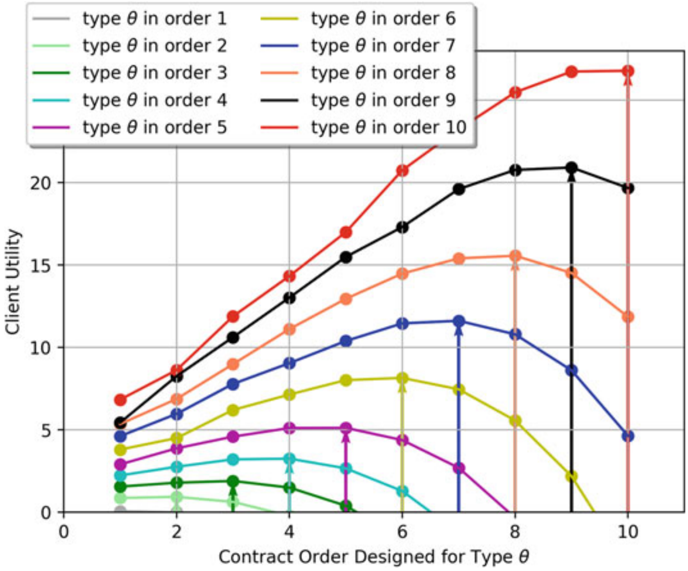

Fig. 2 Client utility with different contract

As shown in Fig. 2, if and only if the client selects a contract suitable for its own type, its benefit can reach the maximum (as shown by the arrow). Therefore, rational clients will choose the contract suitable for their own type. This also inspires clients to actively show their model quality in the case of asymmetric information. In addition, with an increase in the number of client types, the best benefits that can be obtained by clients also increase. This shows the fairness of the proposed incentive mechanism to the client.

In addition, we observe the aggregation effect of the global model reflected by different incentive mechanisms when the client selects the most promising contract. Figure 3 reflects the impact of the default incentive mechanism and the proposed contract based incentive mechanism on the server utility. Higher types of clients will provide better economic utility, such as servers. This conclusion also applies to the performance of social welfare, as shown in Fig. 4.

5.4 Aggregation Experimental Settings

In this section, we evaluate the proposed contract-based incentive mechanism for FL in two classical datasets (i.e. MNIST and CIFAR-10). The proposed model is compared with the other two schemes to demonstrate the effectiveness of our model from the aspect of generalization accuracy.

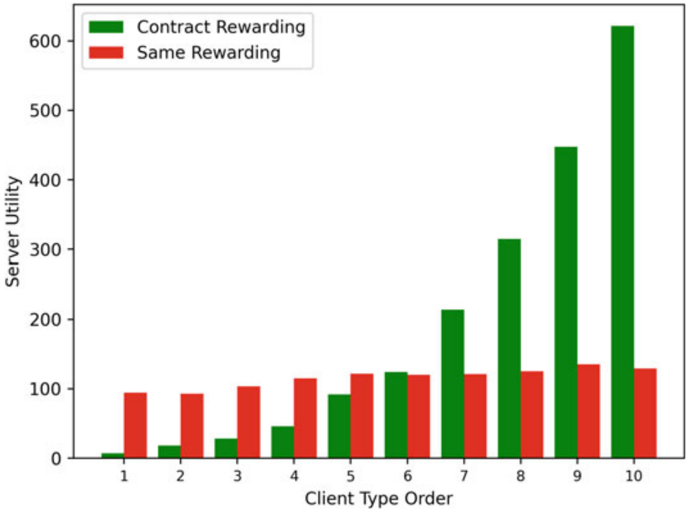

Fig. 3 System utility with optimal contract

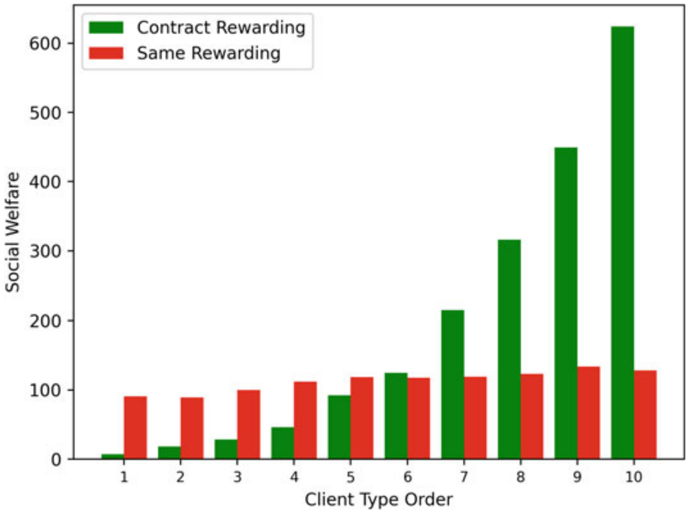

Fig. 4 Social welfare with optimal contract

The experimental environment of this paper is based on the first global iteration, so the target accuracy M is set to a relatively low level. Based on the above settings, we complete steps 1–6 shown in Fig. 1. After the client uploads their local model, the server tests whether the model meets the corresponding test generalization benchmark M for each client according to the chosen contract. Because that the data are heterogeneous and follow non-IID, the data of the server used for testing are part of the whole dataset and chosen randomly in the whole data space, serving as a sampling test for the fraud prevention. Therefore, the testing with the data of the server can effectively show the final generalization effect of the model in the application market. At the same time, through experiments, we find that although our model training in the local environment achieves 93% accuracy, in the server test, only 46% accuracy is achieved. This shows the test error caused by different data coverage between the server and client. Therefore, if the test baseline M is set too high, the test results of the model will deviate considerably. In this experiment, the value of M is relatively low, as shown in Table 1.With an increase in the number of client types, our benchmark requirements increase by 2%.

The following three schemes are compared by setting different rewarding methods and aggregation protocols.

- Scheme-1: The clients are rewarded according to the proposed contract-based solution, and the server aggregates the submitted model according to the contract based aggregation protocol.
- Scheme-2: The clients are rewarded according to the proposed contract-based solution, and the server aggregates the submitted model according to FedAvg protocol [8], where the model aggregation weight is same for all clients.
- Scheme-3: The clients are rewarded equally, and the server aggregates the submitted model according to the FedAvg protocol.

5.5 Aggregation Experimental Results

This generalization accuracy of the three schemes under two different parameter c settings in two datasets is presented in Fig. 5. Here, under the same parameter c, the proposed method (i.e., Scheme-1), shows the highest model generalization accuracy, which is better than Scheme-2; Scheme-3 performs the worst. By using the contract-based incentive mechanism in Scheme-1 and Scheme-2, the clients work harder and consistently perform better than the fixed incentive in Scheme-1. This is because Scheme-1 uses contract-based aggregation, which can set a higher weight for the model trained on a high-quality data source. In addition, comparing the model accuracy under different parameter c, we can observe that the smaller parameter c setting brings the better generalization accuracy, indicating that the clients with lower training cost are more likely to be incentivized to improve the model generalization performance.

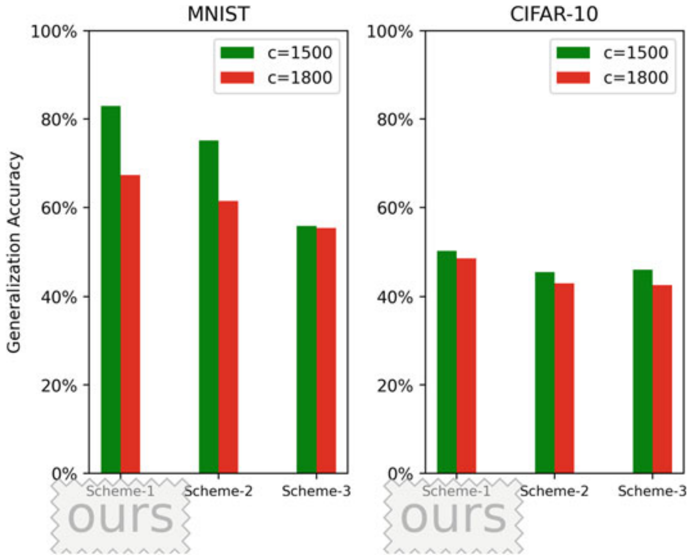

Fig. 5 Generalization accuracy of the proposed scheme compared to the other two schemes

Scheme-3, which is the Federated Averaging Aggregation within the Same Rewarding mechanism, presents the default status about FL rewarding and aggregation. Because the server has no idea about the uploading model of clients, it can only issue the same reward and aggregation model on average. Scheme-2 is the contract Rewarding-based FederatedAveraging Aggregation. It indicates that the client chooses the corresponding contract according to its own type to maximize their own utility, while the server does not fully utilize the model quality information revealed by the selected contract and still uses the default average aggregation strategy. This strategy only shows the incentive effect of the contract on the client, but not the performance effect of the contract on the server. Scheme-1 is the proposed contract rewarding based-weighting aggregation method. It not only fully inspires the client but also makes full use of the information of the selected contract to optimize the aggregation effect. Parameter c represents the unit precision cost of the client and the current task environment of FL. For example, if we want to use the local data of Internet of Things devices to unload federated tasks, the unit precision cost c will be higher because of the smaller power of the devices. In contrast, if we want to use Cloud service providers to unload the federated tasks, c should be a small number. We set the same rewarding scheme with average effort and reward of contract-based rewarding scheme for each client for fairness. In addition, we also adjust parameter c to show the impact of the service environment on model aggregation performance.

6 Conclusion

This paper proposes a contract-based incentive mechanism for FL. A two-dimensional contract model is formally designed, where we consider the client's data coverage quality and effort willingness. We also propose a contract-based FL aggregation protocol. The optimal contract solution is theoretically analyzed. Finally, the proposed incentive mechanism is experimentally evaluated, and the results show that our contract-based scheme achieves higher aggregation accuracy compared to the other two schemes.

Acknowledgements This work is supported in part by Key-Area Research and Development Program of Guangdong Province NO.2020B0101090005; National Natural Science Foundation of China under Grant No.62172085 No.62032013, and No.U20B2046; 111 Project (B16009). Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore.

References

1. Akerlof GA (1970) The market for "Lemons": quality uncertainty and the market mechanism. Quart J Econ 84(3):488–500
2. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Practical Secure Aggregation for Privacy-Preserving Machine Learning. In: ACM SIGSAC conference on computer and communications security, pp 1175–1191
3. Ding N, Fang Z, Huang J (2021) Optimal contract design for efficient federated learning with multi-dimensional private information. IEEE J Sel Areas Commun 39:186–200
4. Feng S, Niyato D, Wang P, Kim DI, Liang Y (2019) Joint service Pricing and cooperative relay communication for federated learning. In: International conference iThings and GreenCom and CPSCom and smartdata, pp 815–820
5. Holmstrom B (1979) Moral hazard and observability. Bell J Econ 10:74–91
6. Hu R, Gong Y (2020) Trading data for learning: incentive mechanism for on-device federated learning. In: IEEE global communications conference, pp 1–6
7. Jin P, Lu L, Tang Y, Karniadakis GE (2020) Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. Neural Netw 130:85–99
8. Kairouz P, McMahan HB et al (2019) Advances and open problems in federated learning. CoRR abs/ arXiv:1912.04977
9. Kang J, Xiong Z, Niyato D, Xie S, Zhang J (2019) Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory. IEEE Internet Things J 6:10700–10714
10. Kang J, Xiong Z, Niyato D, Yu H, Liang Y, Kim DI (2019) incentive design for efficient federated learning in mobile networks: a contract theory approach. In: IEEE VTS Asia pacific wireles communications symposium, pp 1–5
11. Khan LU, Pandey SR, Tran NH, Saad W, Han Z, Nguyen MNH, Hong CS (2020) Federated learning for edge networks: resource optimization and incentive mechanism. IEEE Commun Mag 58:88–93
12. Liu Y, Ai Z, Sun S, Zhang S, Liu Z, Yu H (2020) Fedcoin: a peer-to-peer payment system for federated learning. In: Federated learning - privacy and incentive, vol 12500, pp 125–138
13. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: the 20th international conference on artificial intelligence and statistics, pp 1273–1282

14. Sarikaya Y, Erçetin Ö (2019) Motivating workers in federated learning: a stackelberg game perspective. CoRR abs/ arXiv:1908.03092
15. Wahab OA, Mourad A, Otrok H, Taleb T (2021) Federated machine learning: survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Commun Surv Tutor 23:1342–1397
16. Wang Z, Gao L, Huang J (2020) Multi-cap optimization for wireless data plans with time flexibility. IEEE Trans Mob Comput 19:2145–2159
17. Xiong Z, Kang J, Niyato D, Wang P, Poor HV, Xie S (2020) A multi-dimensional contract approach for data rewarding in mobile networks. IEEE Trans Wireless Commun 19:5779–5793
18. Ye D, Yu R, Pan M, Han Z (2020) Federated learning in vehicular edge computing: a selective model aggregation approach. IEEE Access 8:23920–23935
19. Yu Y, Zhang J, Letaief KB (2016) Joint subcarrier and CPU time allocation for mobile edge computing. In: IEEE global communications conference, pp 1–6
20. Zhang Y, Song L, Pan M, Dawy Z, Han Z (2017) Non-cash auction for spectrum trading in cognitive radio networks: contract theoretical model with joint adverse selection and moral hazard. IEEE J Sel Areas Commun 643–653
21. Zhang Z, Yang T, Liu Y (2020) SABlockFL: a blockchain-based smart agent system architecture and its application in federated learning. Int J Crowd Sci 4:133–147