# Incentive Mechanism for Horizontal Federated Learning Based on Reputation and Reverse Auction

Jingwen Zhang
Sun Yat-sen University
Guangzhou, China

Yuezhou Wu
Sun Yat-sen University
Guangzhou, China

Rong Pan*
Sun Yat-sen University
Guangzhou, China

## ABSTRACT

Current research on federated learning mainly focuses on joint optimization, improving efficiency and effectiveness, and protecting privacy. However, there are relatively few studies on incentive mechanisms. Most studies fail to consider the fact that if there is no profit, participants have no incentive to provide data and training models, and task requesters cannot identify and select reliable participants with high-quality data. Therefore, this paper proposes a federated learning incentive mechanism based on reputation and reverse auction theory. Participants bid for tasks, and reputation indirectly reflects their reliability and data quality. In this federated learning program, we select and reward participants by combining the reputation and bids of the participants under a limited budget. Theoretical analysis proves that the mechanism satisfies computational efficiency, individual rationality, budget feasibility, and truthfulness. The simulation results show the effectiveness of the mechanism.

## CCS CONCEPTS

• **Computing methodologies → Learning paradigms**.

## KEYWORDS

Federated Learning, Incentive Mechanism, Reverse Auction, Reputation

**ACM Reference Format:**
Jingwen Zhang, Yuezhou Wu, and Rong Pan. 2021. Incentive Mechanism for Horizontal Federated Learning Based on Reputation and Reverse Auction. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3449888

## 1 INTRODUCTION

Federated learning is a machine learning framework that meets the requirements of privacy protection, data security, and government laws. All data is kept locally and used to train the local model. Then the center aggregates all local models to obtain a global model [41]. Since there is no need to upload user data, the privacy of user data is prevented from being leaked during uploading, saving, and model training.

*Rong Pan is the corresponding author.


*WWW '21, April 19–23, 2021, Ljubljana, Slovenia*

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449888

Due to the excellent nature of federated learning in machine learning practice, federated learning has become a research hotspot nowadays [43]. Based on federated learning, some applications are developed [11, 29, 42]. Yang et al. [41] classified federated learning based on data distribution. Li et al. [21] discussed the challenges of federated learning, including statistical heterogeneity, expensive communication, and systems heterogeneity.

For data imbalance and non-IID data distribution, McMahan et al. [24] proposed a robust method based on iterative model averaging learning, and Zhao et al. [46] proposed that all clients share a small part of the data globally. Sattler et al. [31] presented a federated multitask learning framework which can group the client population into clusters with jointly trainable data distributions. Communication cost and latency are bottlenecks of federated learning. Yurochkin et al. [44] proposed a Bayesian non-parametric federated learning with only a few communications. Konečnỳ et al. [19] reduces the communication cost of the uplink through structured updates and sketched updates. Caldas et al. [5] limits model size and client participation. Sattler et al. [32] proposed a framework called STC which enables downstream compression. Reisizadeh et al. [30] presented a communication-efficient method with periodic averaging and quantization. To achieve the best balance between reducing communication cost and accelerate convergence, Wang and Joshi [36] designed communication-efficient SGD algorithms. Tran et al. [34] optimized the performance of FL in wireless networks. Privacy preservation is another challenge of federated learning [38]. Chamikara et al. [6] designed a distributed perturbation algorithm. To preserve data privacy, Google proposed a model security aggregation method [4], and Mandal et al. [23] improved this method. Bhowmick et al. [3] designed a new optimal locally differential private mechanism. Geyer et al. [9] and Sun et al. [33] introduced differential privacy into defense against poisoning attacks. Bagdasaryan et al. [2] proved that the model poisoning attack is much more powerful than the data poisoning attack. To accelerate convergence, Liu et al. [22] not only aggregated model updates, but also aggregated momentum terms.

Most of the researches are based on the fact that all parties participate unconditionally and contribute their data honestly. Some studies assume that data is completely accurate. However, due to the inevitable training and communication costs, participants will not serve for free [40]. If there is no reward, participants will not stay in the federated learning system, resulting in the system's unsustainable operation. Besides, participants may submit poor models to improve their utilities even if they are rewarded. Also, the task requester is not aware of the amount and quality of data for each participant. If a data poisoning attack is conducted or the data quality is poor, the performance of the global model may become worse. Therefore, incentive mechanism is necessary for federated learning to motivate workers and select high-quality workers.

In this paper, we propose a horizontal federated learning incentive mechanism called RRAFL, based on reputation and reverse auction theory to motivate all parties to participate actively and allow the requester to select reliable participants with good data quality. Reputation is the requester's evaluation of the participants, which indirectly reflects participant reliability and data quality. Reputation is saved in the blockchain since blockchain cannot be tampered and is open and transparent. We call it the interaction blockchain, and each block holds information about each task. Reverse auction is an auction method in the scenario of a single buyer and multiple sellers. In the reverse auction setting, candidates price their resources, and the requester selects multiple candidates with low price and a good reputation. It is similar to the design of an online auction in advertising space. However, online auction in advertising space uses a forward auction which is an auction method in the scenario of a single seller and multiple buyers.

A good reverse auction method can motivate all parties to actively compete and bid truly. There are typical forms of reverse auction, such as one-dimensional reverse GFP and GSP auctions, which only consider the bid price. However, in federated learning, in addition to the bid price, it is necessary to consider data quality and select an uncertain number of sellers under a limited budget. In addition to the reverse auction formats mentioned above, Che [7] proposed a multi-dimensional reverse auction based on bid price and quality. The true quality is submitted by an honest seller or evaluated by a buyer. However, in federated learning, since participants can lie, and data can not be exposed, and there is no evaluation standard without the final model, it is impossible to obtain the true data quality through the above methods. Therefore in this paper, we use reputation to indirectly reflect the quality and reliability of the candidate, and design a novel reverse auction form that is different from existing ones. By combining reputation and bid price, under the limited budget condition, the requester selects candidates with a good reputation and low bid price. Furthermore, we propose a method for calculating reputation, selecting and paying participants, and measuring contributions. Theoretical analysis proves that RRAFL satisfies computational efficiency, individual rationality, budget feasibility, and truthfulness. Simulation results show its effectiveness. Our contributions are as follows.

- We formalized the FL incentive mechanism problem and proposed a mechanism RRAFL that combines reputation and reverse auction. With a limited budget and considering the quality of participants' data, we designed a participant selection and payment method.
- We introduced a model quality detection method based on marginal contribution and a method of measuring participant contribution. Based on these methods, we proposed a reputation calculation method to indirectly reflect the data quality of participants.
- We proved that our mechanism is computationally efficient, individually rational, budget feasible and truthful. The simulation results show its effectiveness.

The rest of this paper is organized as follows. Section 2 describes the system model and problem definition. Section 3 describes the details of the mechanism RRAFL. Section 4 conducts theoretical analysis. Section 5 gives the simulation results.

**Table 1: Notations and Their Definitions**

| Notations | Definitions |
|---|---|
| $b_j$ | Candidate $j$'s bid price. |
| $c_j$ | True cost of candidate $j$ participating in the task. |
| $p_j$ | Payoff received by participant $j$. |
| $u_j$ | Utility of participant $j$. |
| $Re_j$ | The comprehensive reputation of candidate $j$. |
| $Re_{i,j}$ | Direct reputation of $j$ evaluated by requester $i$. |
| $q_j$ | The unit reputation bid price of candidate $j$. |
| $contrib_j$ | Overall contribution of participant $j$. |
| $contrib_j^{(t)}$ | Contribution of participant $j$ in round $t$. |
| $re_{i,j}$ | Reputation of $j$ evaluated by $i$ in the current task. |
| $Pr_j$ | Pagerank value of candidate $j$. |
| $P_j$ | Transformed Pagerank value of candidate $j$. |
| $\alpha_j$ | The weight of the local model of participant $j$. |
| $\beta$ | The weight of direct reputation. |
| $\varphi_{i,k}$ | Similarity between the requester $i$ and the recommender $k$ in evaluating the common participants' reputation. |
| $\theta_j^{(t)}$ | Angle between the local model update of participant $j$ and the target direction in round $t$. |

## 2 SYSTEM MODEL AND PROBLEM DEFINITION

We outline the overall workflow and give a detailed problem definition. Table 1 lists the main notations and definitions.

### 2.1 System Model

In the original FL setting, the task participants are mobile devices. In our setting, they can be company organizations. We consider a collaborative federated learning scenario in which many individuals form a community. Each individual with data and computing resources in the community can act as a task requester or participant. At a certain moment, an individual with a limited budget, as a task requester, publishes a model training task to the community and recruits others to participate in the task. Interested individuals consider their abilities, weigh the costs and possible rewards, form their strategies, and decide whether to submit their bid price to the task requester. The task requester collects all bids, combines their reputation, selects participants, and finally pays the corresponding rewards. Figure 1 roughly describes the workflow.

- **Step 1:** The task requester broadcasts task information to the community, which describes the budget, required data category and computing resources, etc.
- **Step 2:** Interested Individuals formulate their bidding strategy based on their capabilities, data quantity, and quality, etc., and bid to the task requester according to this strategy.
- **Step 3:** The individuals who bid are called candidates. The task requester downloads the reputation evaluation in all tasks related to the candidate from the interaction blockchain.

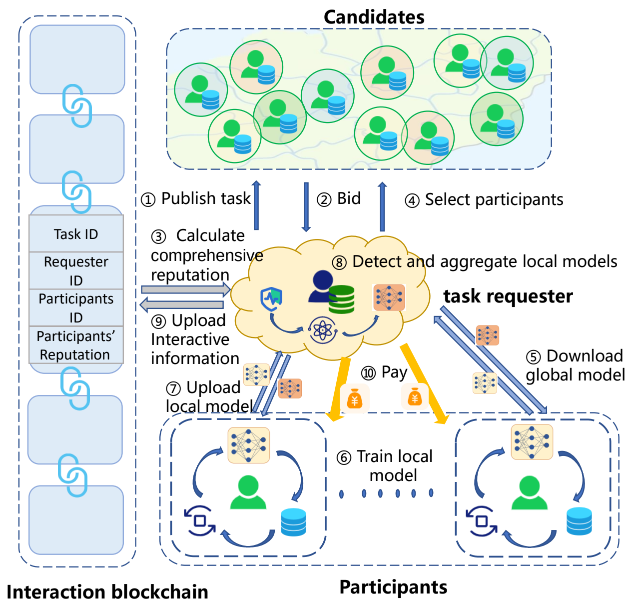


**Figure 1: System Workflow**

Then the requester combines all his historical reputation evaluations on this candidate as the direct reputation, the other requesters' historical reputation evaluation on this candidate as the indirect reputation, and finally combines the direct and the indirect reputation to obtain this candidate's comprehensive reputation.

- **Step 4:** The task requester uses reverse auction to select participants by combining their bids and comprehensive reputations, and then distributes the initial global model.
- **Step 5:** Participants receive the initial global model, iteratively train the local model and send it to the requester.
- **Step 6:** In each round, the requester collects all local models and uses quality detection to filter out bad ones. A new global model is obtained by weighted averaging the remaining local models based on performance in quality detection.
- **Step 7:** After model training, the requester measures the contribution and reputation of each participant in this task.
- **Step 8:** The task requester uploads and saves the interaction information of the task (e.g., the task id, task requester's id, the participants' id, and the participants' reputation in this task) to the interaction blockchain.
- **Step 9:** Finally, the task requester pays the task participants.

## 2.2 Interaction Graph

We build an interaction graph based on the interactions of individuals over a period of time, as shown in Figure 2. Interaction graph $G(V, E)$ is a directed graph, where $V = \{v_1, v_2, ..., v_n\}$ with $v_i$ representing each individual $i$ and $E$ is the set of directed edges. The edge $e_{i,j}$ in $E$ represents the individual $i$ as the task requester has selected the individual $j$ as the participant, and the reputation of individual $j$ evaluated by individual $i$ in a certain task is greater than the threshold $\xi$ in this task. Since the same task requester may choose the same individual in different tasks for a while, an edge may represent multiple interactions between two individuals. The out-degree and in-degree of the $v_i$ can be used to measure the influence of individual $i$ in the community. Influence reflects the reliability of the reputation evaluated by the individual $i$, rather than his own reputation. Section 3.1 describes the role of interaction graph.

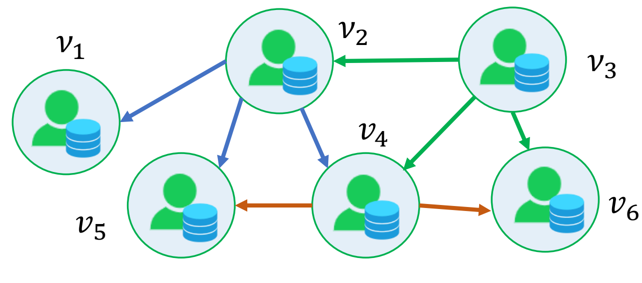


**Figure 2: Interaction Graph between Individuals**

## 2.3 Assumption and Problem Definition

First of all, we make the following assumptions. Our incentive mechanism RRAFL is used for synchronous horizontal federated learning. In a period, multiple model learning tasks have been completed. What we are studying is how to perform new tasks in this context. Every individual can play the role of task requester or participant. Individuals usually collect data related to various tasks, so they have appropriate data before bidding. The data accuracy of each individual may be different, so the data quality is different. Every individual is rational, and only if his utility is non-negative can he participate in the task. The requester has a small amount of relevant data which is not enough to train a good model, so he recruits others to participate in the task. The requester divides his data into validation set and test set, and participants use their data as training set. Besides, when recruiting participants, the requester's budget is likely too low or the candidate bid price is too high, failing to recruit participants. At this time, the requester and candidates can adjust the budget or bid price according to the actual situation. The requester rebroadcasts the task, and the candidate resubmits a new bid.

Based on the above assumptions, the problem is defined below. Suppose a requester $i$ publishes a task with a limited budget $B$. Each candidate $j$ has a private true cost $c_j$ for the task. Candidate $j$ submits bid price $b_j$ to requester $i$ according to his strategy. This process is sealed. The requester $i$ calculates the comprehensive reputation $Re_j$ of each candidate $j$ and selects the candidates to form a participant set $S$ by considering the comprehensive reputation $Re_j$ and bid price $b_j$. Define variable $x_j \in \{0, 1\}$. When candidate $j$ is selected, $x_j = 1$, otherwise $x_j = 0$. Since the budget is limited, Formula (1) should be satisfied.

$$\sum_j x_j p_j \leq B. \tag{1}$$

Each participant should not be paid less than their cost, as

$$p_j \geq x_j c_j. \tag{2}$$

For participant $j$, its utility $u_j$ is computed as follows.

$$u_j = x_j(p_j - c_j). \tag{3}$$

For task requester $i$, his utility $U_i$ is computed as follows.

$$U_i = \sum_j x_j Re_j. \tag{4}$$

In addition to maximizing the utility of the task requester, the following properties need to be met.

- **Computational Efficiency:** Both the selection and payment can be computed in polynomial time.
- **Individual Rationality:** Every participant can get non-negative utility.
- **Budget Feasibility:** The total payment to participants does not exceed the budget of the task requester.
- **Truthfulness:** Only by revealing his true cost in the bid can a candidate maximize his utility. In other words, no matter what others submit, no one can improve his utility by submitting a fake cost.

## 3 INCENTIVE MECHANISM DESIGN

The designed mechanism RRAFL is mainly composed of four parts, including comprehensive reputation calculation, participant selection and payment, model training, and task reputation evaluation.

### 3.1 Comprehensive Reputation Calculation

Reputation indirectly reflects the reliability of the candidates and their data quality. For requester $i$, the panticipant's reputation evaluated by requester $i$ is called *direct reputation*, and partipant's reputation evaluated by other requester $k$ is called *recommended reputation*, and the reputation obtained by combining all recommended reputations is called *indirect reputation*. To generate accurate reputation evaluation, direct reputation and indirect reputation are combined as the *comprehensive reputation*. Other requesters are called *recommender*s, and that whose evaluated reputation is accepted are referred to as *effective recommender*s. Since blockchain cannot be tampered and is transparent, the participant's reputation in each task is preserved in the interaction blockchain. Since the reputation evaluation closer to the current can better reflect the nature of the candidate, requester $i$ performs a exponential moving average of the reputation of candidate $j$ in each task to obtain direct reputation in the $\tau$ th task, as shown in Eq. (5).

$$Re_{i,j}^{(\tau)} = \lambda Re_{i,j}^{(\tau-1)} + (1-\lambda) re_{i,j}^{(\tau)}, \tag{5}$$

where $\lambda$ is the decay factor. $re_{i,j}^{(\tau)}$ represents the reputation of $j$ evaluated by $i$ in the $\tau$-th task and $Re_{i,j}^{(\tau)}$ represents the result of exponential moving average on $\{re_{i,j}^{(\ell)}\}_{\ell=1}^{\tau}$. In the current $\tau$-th task, we use $\{re_{i,j}^{(\ell)}\}_{\ell=1}^{\tau-1}$ to generate $Re_{i,j}^{(\tau-1)}$ for participant selection instead of $Re_{i,j}^{(\tau)}$. After the $\tau$-th task ends, $re_{i,j}^{(\tau)}$ is generated, which can be used for the $(\tau+1)$-th task. In the following, when the discussion is only on a fixed task, the superscript $(\tau)$ in this notation will be omitted for simplicity.

Due to the differences between the recommenders, the recommended reputations should be assigned different weights when combined into an indirect reputation. The weight depends on the direct reputation of the recommender, the Pagerank [28] value of him in the interaction graph, and the similarity between the requester and him in evaluating participant reputation. Reputation can indirectly reflect the reliability of the recommender. If a recommender acts maliciously as a participant, then he is likely to provide a malicious recommended reputation as a recommender. The direct reputation of the recommender evaluated by the requester can be regarded as a partial assessment of his reliability because it only involves the requester's opinion on him.

In the interaction graph formed by the interactions in the latest $N$ tasks, the Pagerank value is used as a global assessment of the recommender's reliability. Pagerank is used here based on two assumptions.

- **Quantity Assumption:** If recommender $k$ has been selected more times and performed better, then it can indirectly reflect that it has a better reputation.
- **Quality Assumption:** If the requester pointing to recommender $k$ is more reliable, then recommender $k$ is more reliable.

Google divides Pagerank into 10 levels, but the levels are not linear. Using the Pagerank value directly will affect the rationality. Therefore, we use Eqs. (6) and (7) to transform the Pagerank value $Pr_k$.

$$Pr_k' = \frac{Pr_k - \mu}{\sigma}, \tag{6}$$

$$P_k = f(Pr_k'), \tag{7}$$

where $\mu = \frac{\sum_{k=1}^{m} Pr_k}{m}$, $\sigma = \sqrt{\frac{\sum_{k=1}^{m}(Pr_k - \mu)^2}{m}}$, $f(\cdot)$ is a sigmoid function and $m$ represents the number of recommenders. $P_k$ is used to calculate the weight of recommended reputation.

The use of similarity here is based on the fact that, if the requester and recommender have multiple common participants and have similar reputation evaluations on the same participant, then the recommended reputation provided by this recommender is reliable. Similarity is measured by a modified cosine formula [15], as in Eqs. (8) and (9). $I$, $K$ are the set of participants who have interacted with requester $i$ and recommender $k$, respectively. $\overline{Re_{i,\cdot}}$ and $\overline{Re_{k,\cdot}}$ are the mean values of the direct reputation of participants evaluated by requester $i$ and recommender $k$, respectively.

$$\varphi_{i,k}' = \frac{\sum_{j \in I \cap K} (Re_{i,j} - \overline{Re_{i,\cdot}})(Re_{k,j} - \overline{Re_{k,\cdot}})}{\sqrt{\sum_{j \in I} (Re_{i,j} - \overline{Re_{i,\cdot}})^2} \sqrt{\sum_{j \in K} (Re_{k,j} - \overline{Re_{k,\cdot}})^2}}, \tag{8}$$

$$\varphi_{i,k} = \max\{0, \varphi_{i,k}'\}. \tag{9}$$

In the end we get the weight $w_k = \frac{Re_{i,k} P_k \varphi_{i,k}}{\sum_k Re_{i,k} P_k \varphi_{i,k}}$ of recommended reputation and indirect reputation $\sum_k w_k Re_{k,j}$.

The greater the number of effective recommenders, the more valuable indirect reputation is. Therefore, the weight $\beta$ of the direct reputation should decrease as the number of effective recommenders $N_{effective}$ increases but has a lower bound. We use the transformed arctan function to calculate the weight of the direct reputation as follows.

$$\beta = -\frac{2(1-\beta_1)}{\pi} \arctan\left(\frac{N_{effective}}{\beta_2 \pi}\right) + 1, \tag{10}$$

where parameter $\beta_1$ controls the lower bound and $\beta_2$ controls how fast the weight of direct reputation decreases as the number of effective recommenders increases. The input is the number of effective recommenders $N_{effective}$, and the output is the weight $\beta$ of direct reputation. The comprehensive reputation $Re_j$ of candidate $j$ is calculated using Eq. (11).

$$Re_j = \beta \cdot Re_{i,j} + (1-\beta)\sum_k w_k Re_{k,j}. \tag{11}$$

## 3.2 Participant Selection and Payment

To obtain a better global model with a limited budget $B$, firstly, participants with low bids should be selected, and secondly, participants with better data quality should be selected. The candidate's unit reputation bid price is defined as

$$q_j = \frac{b_j}{Re_j}. \tag{12}$$

We sort the candidates in a non-descending order of their unit reputation bid price $q_j$.

$$q_1 \leq q_2 \leq \ldots \leq q_n. \tag{13}$$

And $k$ candidates are selected from the ranking from front to back to form a participant set $S$, where $k$ satisfies

$$k = \arg\max_k \{q_{k+1}\sum_{j=1}^{k} Re_j \leq B\}. \tag{14}$$

Then the final payment to participant $j$ is shown in Eq.(15). That is, the payoff is based on the $(k+1)$-th participant in the ranking. Algorithm 1 shows the process of participant selection and payment.

$$p_j = \begin{cases} Re_j q_{k+1} & j \in S, \\ 0 & j \notin S. \end{cases} \tag{15}$$

**Algorithm 1** Participant Selection and Payment

**Input:** Budget B, Candidate Set $S'$,
Comprehensive Reputation $\boldsymbol{Re}$=$\{Re_1, \ldots, Re_j, \ldots, Re_n\}$;
**Output:** Participant Set $\boldsymbol{S}$, Selected Flags $\boldsymbol{X}$, Payment $\boldsymbol{P}$;
$\boldsymbol{P} \leftarrow \{p_1, \ldots, p_j, \ldots p_n\}$, $\boldsymbol{X} \leftarrow \{x_1, \ldots, x_j, \ldots x_n\}$, $\boldsymbol{S} \leftarrow \phi$, $k \leftarrow 1$;
**Selection Phase:**
sort all $j \in S'$ $s.t.$ $\frac{b_1}{Re_1} \leq \ldots \leq \frac{b_j}{Re_j} \leq \ldots \leq \frac{b_n}{Re_n}$;
**while** $\frac{b_{k+1}}{Re_{k+1}}(Re_k + \sum_{j\in S} Re_j) \leq B$ **do**
  $S \leftarrow S \cup \{k\}$; $k \leftarrow k+1$;
**end while**
$k \leftarrow k-1$;
**Payment Phase:**
**for** each $j \in S'$ **do**
  **if** $j \in S$ **then**
    $p_j \leftarrow \frac{b_{k+1}}{Re_{k+1}} Re_j$; $x_j \leftarrow 1$;
  **else**
    $p_j \leftarrow 0$; $x_j \leftarrow 0$;
  **end if**
**end for**
**return** $\boldsymbol{S}, \boldsymbol{X}, \boldsymbol{P}$;

The participant selection and payment problem seems like a classic knapsack problem but it is not. In a knapsack problem, the costs are the input information. In the classic knapsack problem, the costs are public knowledge. However, in our problem, the costs are the participant's private information. The participant can bid different from his true cost strategically. Besides, the payment to participants is not equal to the cost. And the budget constraint applies not to the costs but to the payments that the mechanism uses to support truthfulness. Therefore our problem is not a classic knapsack problem.

## 3.3 Model Training

Although reputation is considered, participants' reliability and data quality cannot be fully guaranteed. Therefore, it is necessary to filter out bad local models through quality detection. We cannot directly filter out bad models by judging whether the local model exceeds the loss threshold, because it is difficult to determine this threshold. Besides, when the local model gets better, the threshold should be reduced accordingly, but it is difficult to determine how much the threshold should be reduced. Thus, We check the quality of a local model using others. In each round, all local models are aggregated to obtain a global model $\mathcal{M}$, and the loss $l$ on the validation set is calculated. Remove a local model $j$ that need to be detected, aggregate the remaining local models to get another global model $\mathcal{M}'$, and calculate the loss $l'_j$ of the model $\mathcal{M}'$. If the difference $\delta_j$ between the loss $l'_j$ and $l$ is not less than the threshold $\delta$, as shown in Formula (16), the local model $j$ is accepted.

$$l'_j - l \geq \delta. \tag{16}$$

It is based on the fact that the loss of the global model should be reduced or maintained if a good local model participates in the aggregation process.

Local models that pass quality detection are used to aggregate into the global model. `FedAvg` [24] determines the weight by the amount of data, but participants may exaggerate the amount of data. Therefore, the weight of each local model that passes the quality detection should be determined based on the performance rather than the amount of data. We define *base score* $s_0$ and *extra score* $s_j$, which determine the weight of the local model of participant $j$. Each local model has the same base score. The extra score $s_j$ is determined by the difference $\delta_j$ between the loss $l'_j$ and $l$ in the quality detection, as in Eq. (17).

$$s_j = \frac{\delta_j - \min_j(\delta_j)}{\sum_j\left(\delta_j - \min_j(\delta_j)\right)}. \tag{17}$$

As shown in Eq. (18), the result derived from the base score $s_0$ and the extra score $s_j$ is used as the weight $\alpha_j$ of each local model $\mathcal{M}_{L_j}^{(t)}$ that passed the detection. It can be seen that the base score controls the influence of the performance on the weight. According to Eq. (19), we can obtain a new global model $\mathcal{M}_G^{(t)}$ in round $t$.

$$\alpha_j = \frac{s_0+s_j}{\sum_j(s_0+s_j)}, \tag{18}$$

$$\mathcal{M}_G^{(t)} = \textstyle\sum_j \alpha_j \mathcal{M}_{L_j}^{(t)}. \tag{19}$$

### 3.4 Measurement of the Contribution and Reputation of Current Task

To measure the contribution of each participant, the requester needs to save all local models and the global model of each round. The parameters of the global model change towards the direction of convergence and are finally fixed. Figure 3 shows how to measure contribution.

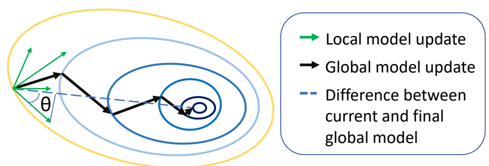


**Figure 3: Contribution Measurement**

First, find the direction vector $\vec{v}$ between the initial global model of the $t$-th round and the final global model. That is, the dotted line in Figure 3. Then, find the angle $\theta_j^{(t)}$ between the local model update $g_j^{(t)}$ of participant $j$ and $\vec{v}$. Then, as shown in Eq. (20), the local model update $g_j^{(t)}$ is projected onto the direction vector $\vec{v}$, and the result is multiplied by the absolute value of the $\cos\theta_j^{(t)}$, and the result is the contribution $contrib_j^{(t)}$ of the participant $j$ in the $t$-th round. Finally, the contribution of participant $j$ in each round is added up as the overall contribution of participant $j$, as shown in Eq. (21).

$$contrib_j^{(t)} = g_j^{(t)} \cos\theta_j^{(t)} |\cos\theta_j^{(t)}|, \tag{20}$$

$$contrib_j = \max\{0, \textstyle\sum_t contrib_j^{(t)}\}. \tag{21}$$

Contribution measurement is based on the following idea. The final global model is the training target. If the local model changes more towards the target, the global model can converge faster, and its contribution should be greater. When the updates of the two local models towards the target direction are equal, the one closer to the target direction should have a larger contribution. The reason is that it is easier to bias the global model if the projection length in the orthogonal direction of the updates away from the target direction is large.

Reputation measurement of the current task is related to the relative contribution and the number of times passing quality detection. The relative contribution value $z_j$ is based on the maximum contribution value and is calculated as follows.

$$z_j = \frac{contrib_j}{\max_j (contrib_j)}. \tag{22}$$

We use the Gompertz function [10] to measure the impact of the quality detection on reputation. Gompertz function is suitable for modeling the concept of trust in individual interactions [12], which is defined as Eq. (23), where $a$, $b$, $c$ are parameters.

$$y = ae^{be^{cx}}. \tag{23}$$

When $a = 1$, $b = -1$, $c = -5.5$, the graph of function is shown in Figure 4. The range of the function is $[0, 1]$.

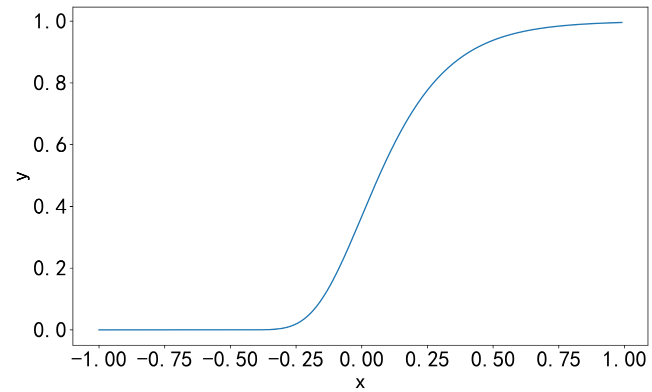


**Figure 4: Graph of Gompertz Function**

The input $\chi_j$ of the Gompertz function is defined as follows.

$$\chi_j = \frac{\omega n_j^{pass} - (1-\omega)\, n_j^{fail}}{\omega n_j^{pass} + (1-\omega)\, n_j^{fail}}, \tag{24}$$

$n_j^{pass}$ and $n_j^{fail}$ are the number of times of passing and failing the quality detection. Compared with passing, failing needs more attention, so the weight $\omega$ of passing the detection ranges from $(0, 0.5]$. $\chi_j$ is the input of the Gompertz function as shown in Eq. (25). Then Eq. (26) shows how requester $i$ measures the reputation of participant $j$ in current task.

$$y_j = ae^{be^{c\chi_j}}, \tag{25}$$

$$re_{i,j} = y_j z_j. \tag{26}$$

## 4 THEORETICAL ANALYSIS

In this section, we prove that RRAFL satisfies computational efficiency, individual rationality, budget feasibility, and truthfulness.

THEOREM 4.1. *RRAFL is computationally efficient.*

PROOF. In Algorithm 1, the time complexity of sorting candidates in ascending order of unit reputation bid price (line 3) is $O(nlogn)$. The time complexity of finding the top $k$ workers in the ranking that meets the selected condition is $O(n)$ (lines 4˜6). The time complexity of determining the payment to all candidates is $O(n)$ (lines 9˜15). Therefore the total time complexity is $O(nlogn + n + n) = O(nlogn)$. □

THEOREM 4.2. *RRAFL is individually rational.*

PROOF. The utility of each candidate $j$ is $u_j = x_j(p_j - c_j)$. If candidate $j$ is not selected, its utility is 0. Otherwise, candidate $j$ is paid $p_j = Re_j \frac{b_{k+1}}{Re_{k+1}}$. Since candidate $j$ is selected, we have $\frac{b_j}{Re_j} \le \frac{b_{k+1}}{Re_{k+1}}$. Hence, $b_j \le Re_j \frac{b_{k+1}}{Re_{k+1}} = p_j$. That is $u_j \ge 0$. □

THEOREM 4.3. *RRAFL is budget feasible.*

PROOF. Our scheme RRAFL selects largest $k$ participants under the condition of $\sum_{j=1}^{k} Re_j q_{k+1} \le B$. Each winning participant

$j$'s reward is $p_j = Re_j \cdot q_{k+1}$, and the loser's reward is 0. Hence, $\sum_{j=1}^{n} p_j = \sum_{j=1}^{k} p_j = \sum_{j=1}^{k} Re_j q_{k+1} \leq B$. □

Lemma 4.4. *[26] An reverse auction mechanism is truthful if and only if:*

- ***Selection rules are monotonous:** If a candidate wins by bid price b, then it can also win when he bids $b' < b$.*
- ***The payment to participants is the critical value:** If the candidate's bid price is greater than this critical value, it is unlikely that it will win the bid.*

Theorem 4.5. *RRAFL is truthful.*

Proof. We first prove that the selection rule of the mechanism is monotonous. In the case where the bids of others remain the same, if a winning candidate $j$ bids a lower price, he will be ahead of the original position or at least stay in the original position in the ranking, and still win. So the selection rule is monotonous.

Then we prove that the payment to participants is the critical value. Assume that the top $k$ candidates in the ranking are selected and the payment to the winning candidate $j$ is $p_j$. If the winning candidate $j$ bids $b'_j < p_j = \frac{b_{k+1}}{Re_{k+1}} Re_j$, meaning that $\frac{b'_j}{Re_j} < \frac{b_{k+1}}{Re_{k+1}}$, then the candidate $j$ is still in the top $k$ positions in the ranking. Since $\frac{b_{k+1}}{Re_{k+1}} \sum_{j=1}^{k} Re_j \leq B$, the top $k$ candidates in the ranking including $j$ are still selected. If the winning candidate $j$ bid $b'_j > p_j = \frac{b_{k+1}}{Re_{k+1}} Re_j$, meaning that $\frac{b'_j}{Re_j} > \frac{b_{k+1}}{Re_{k+1}} Re_j$, then he will be placed after candidate $k+1$ in the ranking. Because of $\frac{b_{k+2}}{Re_{k+2}} \sum_{j=1}^{k+1} Re_j > B$, no matter whether candidate $k+1$ is selected, candidate $j$ will not be selected. Hence the payment to participants is the critical value.

According to Lemma 4.4, the designed scheme RRAFL is truthful. □

Since the candidates are selected and paid according to the unit reputation bid price ranking, the interested individual has a better chance of winning and receives a greater reward if it strives to improve its reputation. Only when a participant provides high-quality data to train the model can it gain a higher reputation. Therefore, the designed scheme can motivate participants to provide high-quality data.

# 5 EXPERIMENT

## 5.1 Experimental Settings

We use the MNIST, Fashion MNIST, and IMDB datasets respectively for experiments. The MNIST is a dataset of handwritten 0-9 digits, which contains 60,000 training samples and 10,000 test samples. The Fashion MNIST is a clothing dataset containing 70,000 samples in 10 categories. The IMDB dataset contains 5000 movie reviews with positive or negative labels. We set up a total of 30 individuals in the community. In the MNIST and Fashion MNIST task, the training set size of each individual is 1000, and in the IMDB task, it is 3000. But the accuracy of the individual data may be different. The number of individuals corresponding to different data accuracy rates is shown in Table 2.

Each requester has a validation set with a size of 2000 and a test set of 2000. Participants only use their data as the training set. Individual data is randomly drawn from the corresponding dataset.

**Table 2: Number of Individuals and Bid Price Range Corresponding to Dfferent Data Accuracy Rates**

| **Data accuracy** | 1.0 | 0.7 | 0.4 | 0.1 |
|---|---|---|---|---|
| **Number of individuals** | 15 | 5 | 5 | 5 |
| **Bid price range** | [4, 6] | [3, 5] | [2, 4] | [1, 3] |

By modifying the label of the sample to others to make the wrong sample, the accuracy of the individual data may be different. For each dataset, respectively, run 100 tasks first, to indicate that the system has been running for a period, and then run another 100 tasks. We use the average of the last 100 tasks to evaluate the mechanism RRAFL. The model consists of two fully connected layers. The input layer has 784 cells, the hidden layer has 50 cells, and the output layer has 10 cells for MNIST and Fashion MNIST. And for IMDB dataset, the number of cells in the three layers is respectively 2000, 16, 2. In each task, according to different probabilities, an individual is randomly selected as the requester, and others are candidates. The candidate randomly generates a bid price from the corresponding bid price range according to a uniform distribution, as shown in Table 2. Participants locally train the model for 1 epoch with a batch size of 100 and a learning rate of 0.05. Table 3 shows other parameter settings.

**Table 3: Parameter Settings**

| **Parameter** | **Setting** |
|---|---|
| Requester's budget | $B = 70$ |
| Base score | $s_0 = 1$ |
| Loss threshold | $\delta = -0.01$ |
| Other parameter settings | $\beta_1 = 0.5, \beta_2 = 4, \lambda = 0.2, \xi = 0.5, \omega = 0.4, N = 30$ |

## 5.2 Experimental Results

Table 4 shows the average contributions, rewards, and reputation of participants with different data accuracy rates. In general, all three decrease as the data accuracy decreases, and the change is non-linear. It shows that our scheme RRAFL can reasonably measure the participant's contribution to a certain extent, and allow participants with good data quality to get more rewards and a higher reputation.

To show the influence of recommenders' fake reputation on comprehensive reputation, we considered two cases. Case 1 is that a different number of recommenders maliciously give a high reputation rating of 0.99 to candidates with a data accuracy rate of 0.1, trying to improve the comprehensive reputation of this bad candidate. Case 2 is that different numbers of recommenders maliciously give all candidates a low reputation which is randomly sampled from the range [0.05, 0.15] according to a uniform distribution, observing the impact on the comprehensive reputation of a candidate with a data accuracy of 1.0. As shown in Table 5, with the number of

**Table 4: Contributions, Rewards, and Reputation of Participants with Different Data Accuracy Rates**

| Dataset | Data accuracy | 1.0 | 0.7 | 0.4 | 0.1 |
|---|---|---|---|---|---|
| MNIST | **Contribution** | 2.041 | 0.386 | 0.209 | 0.022 |
| | **Reward** | 5.641 | 3.953 | 4.107 | 1.109 |
| | **Reputation** | 0.961 | 0.488 | 0.289 | 0.222 |
| Fashion MNIST | **Contribution** | 1.636 | 0.325 | 0.117 | 0.098 |
| | **Reward** | 5.666 | 4.320 | 4.076 | 1.166 |
| | **Reputation** | 0.965 | 0.406 | 0.241 | 0.157 |
| IMDB | **Contribution** | 1.893 | 0.110 | 0.0 | 0.0 |
| | **Reward** | 5.670 | 4.331 | 4.250 | 1.171 |
| | **Reputation** | 0.939 | 0.367 | 0.233 | 0.194 |

malicious recommenders increasing, the comprehensive reputation of Case 1 slightly increases, and that of Case 2 decreases slightly. However, in these two cases, for every 2 additional malicious recommenders, reputation does not change much, no more than 0.0297. This shows that even if the number of fake recommenders increases, the comprehensive reputation of the candidates is still useful.

**Table 5: The impact of the number of malicious recommenders on candidates' comprehensive reputation.**

| Dataset | Number | 0 | 2 | 4 | 6 |
|---|---|---|---|---|---|
| MNIST | **Case 1** | 0.1879 | 0.2093 | 0.2304 | 0.2541 |
| | **Case 2** | 0.9494 | 0.9469 | 0.9465 | 0.9168 |
| Fashion MNIST | **Case 1** | 0.1866 | 0.2092 | 0.2297 | 0.2525 |
| | **Case 2** | 0.8947 | 0.8921 | 0.8913 | 0.8648 |
| IMDB | **Case 1** | 0.1876 | 0.2089 | 0.2320 | 0.2345 |
| | **Case 2** | 0.9774 | 0.9523 | 0.9506 | 0.9471 |

Our scheme without using reputation is a comparison scheme, and we call it RAFL, which only selects the participants with lower bid price and not considers reputation. Suppose the maximum number of participants selected through RRAFL and RAFL is $n_0$. The baseline scheme is to select $n_0$ participants randomly, and we call this scheme vanilla FL. We compared the data quality of participants selected through the three schemes. Table 6 shows that, in our scheme RRAFL, the proportion of participants with data accuracy no less than 0.7 is more than 98%, and the proportion of participants with data accuracy of 1.0 is more than 99%, which is much higher than the other two schemes. It means our scheme RRAFL can help task requester select candidates with good data quality.

Table 7 compares the average loss and accuracy of the final global model of the three schemes in the last 100 tasks. Figure 5 shows the accuracy of the global model obtained by the three schemes in the last 100 tasks for the MNIST dataset. It can be seen that the accuracy of the global model obtained by our scheme RRAFL is higher than that of the other two schemes, and the model loss is lower. For most tasks, the accuracy of the model obtained by our

**Table 6: Proportion of Participants with Good Data Quality Selected by Different Schemes**

| Dataset | Data Accurary | vanilla FL | RRAFL | RAFL |
|---|---|---|---|---|
| MNIST | **1.0** | 0.5047 | **0.9856** | 0.2899 |
| | **≥ 0.7** | 0.6749 | **0.9946** | 0.5165 |
| Fashion MNIST | **1.0** | 0.5047 | **0.9822** | 0.2899 |
| | **≥ 0.7** | 0.6749 | **0.9921** | 0.5165 |
| IMDB | **1.0** | 0.5008 | **0.9898** | 0.3024 |
| | **≥ 0.7** | 0.6653 | **0.9913** | 0.5165 |

**Table 7: Average Loss and Accuracy of the Global Model Obtained by the Three Schemes**

| Dataset | | vanilla FL | RRAFL | RAFL |
|---|---|---|---|---|
| MNIST | **Loss** | 0.8389 | **0.3541** | 0.5557 |
| | **Accuracy** | 0.8690 | **0.9016** | 0.8936 |
| Fashion MNIST | **Loss** | 0.9803 | **0.5728** | 0.6816 |
| | **Accuracy** | 0.7576 | **0.7990** | 0.7924 |
| IMDB | **Loss** | 0.5780 | **0.3305** | 0.5379 |
| | **Accuracy** | 0.8045 | **0.8627** | 0.8053 |

scheme RRAFL is higher than that of the RAFL, and is much better than the vanilla FL. Compared with the other two schemes, our scheme RRAFL can improve the performance of the global model.

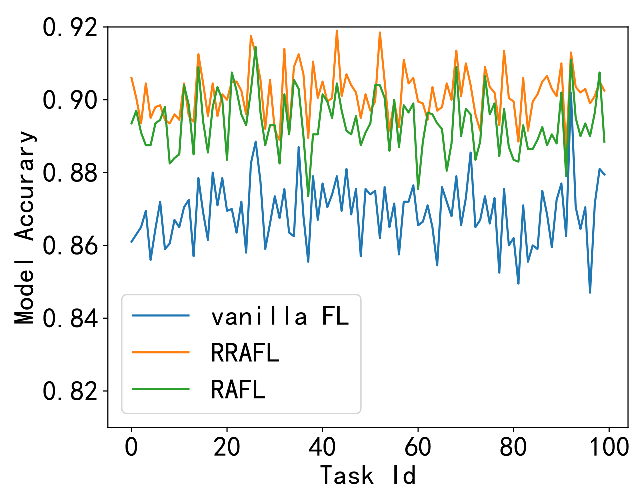


**Figure 5: Accuracy on the test set of the global model of the three schemes in 100 tasks for the MNIST dataset.**

According to Table 6 and Table 7, we can find that the proportion of participants with good data quality selected by RAFL is lower than that of vanilla FL while the performance of the global model is reversed. It is because RAFL has the same quality detection procedure as RRAFL. After quality detection, poor quality models will not participate in model aggregation. Without quality detection, we find that the average losses of the final global model in three different datasets are 0.9559, 1.0757, and 0.6125 respectively, which are all higher than 0.8389, 0.9802, and 0.5780 of vanilla FL. These results confirm the effectiveness of our model quality detection method.

## 6 RELATED WORK

Google first introduced federated learning [18]. McMahan et al. [25] proposed the `FedAvg` algorithm, which determines the local model weight according to the amount of data. However, the client may provide a fake amount or perform a data poisoning attack. Nishio and Yonetani [27] proposed the FedCS protocol to select clients under the condition of limited resources. But only the client's computing resources and communication conditions are considered, not its reliability and data accuracy. Kang et al. [15] proposed a federated learning incentive mechanism combining reputation and contract theory. Wang [35] used Shapley value to reveal detailed data feature importance, but no further incentives are given. Agarwal et al. [1] and Jia et al. [13] used the approximate Shapley value to price the data. Although the compensation can be allocated fairly, it may not be enough to make up for the cost of model training. Kang et al. [16] used contract theory as an incentive mechanism and Khan et al. [17] uses Stackelberg Game. In these settings, the cost of workers is unknown to the center. If the price set by the center does not meet the expectations of workers, the workers may lose their initiative and enthusiasm.

Reverse auction is often used as an incentive mechanism for Crowdsensing. Lee and Hoh [20] first applied reverse auction to crowdsensing. Wang et al. [37] combined reverse auction and reputation as an incentive mechanism for crowdsensing. In federated learning, Xu et al. [39] introduced data quality based on reverse auction. Cong et al. [8] designed a scheme based on VCG to incentivize data owners to truly report its costs and data quality. Because of the lack of measurement standards, there is no way to measure the data quality of the data owner before the model is built. Zeng et al. [45] used multi-dimensional reverse auction as an incentive mechanism to encourage high-quality and low-cost data owner to participate in federated learning. However, there is no guarantee that after workers are selected, they will work according to the agreement of the bidding. Jiao et al. [14] also used a multi-dimensional reverse auction, taking into account the data imbalance and non-IID data distribution measured by EMD. However, before calculating EMD, it needs to obtain the global distribution of all data, which is unrealistic under the setting of federated learning.

## 7 CONCLUSION

To motivate individuals to actively participate in the task and help the task requester select reliable participant with good quality data, we propose an incentive mechanism RRAFL based on reputation and reverse auction. Reputation indirectly reflects individual reliability and data quality. Reverse auction theory encourages individuals to bid truthfully. We provide a method of calculating comprehensive reputation, selecting and paying participants, and measuring contributions. Through theoretical analysis, we have proved that RRAFL satisfies computational efficiency, individual rationality, budget feasibility, and truthfulness. Also, experimental results show the effectiveness of RRAFL. In the future, we can further study how to measure the participants' contributions more reasonably, and how to allocate the rewards when participants join or leave the task dynamically.

## ACKNOWLEDGMENTS

This work was supported by the Special Funds for Central Government Guiding Development of Local Science & Technology (No. 2020B1515310019) and the National Natural Science Foundation of China under Grant U1911201.

## REFERENCES


[1] Anish Agarwal, Munther A Dahleh, and Tuhin Sarkar. 2019. A Marketplace for Data: An Algorithmic Solution. (2019), 701–726.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.

[3] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. 2018. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984* (2018).

[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[5] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).

[6] Mahawaga Arachchige Pathum Chamikara, Peter Bertok, Ibrahim Khalil, Dongxi Liu, and Seyit Camtepe. 2020. Privacy Preserving Distributed Machine Learning with Federated Learning. *arXiv preprint arXiv:2004.12108* (2020).

[7] Yeon-Koo Che. 1993. Design competition through multidimensional auctions. *The RAND Journal of Economics* (1993), 668–680.

[8] Mingshu Cong, Han Yu, Xi Weng, Jiabao Qu, Yang Liu, and Siu Ming Yiu. 2020. A VCG-based Fair Incentive Mechanism for Federated Learning. *arXiv preprint arXiv:2008.06680* (2020).

[9] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[10] Benjamin Gompertz. 1825. XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In a letter to Francis Baily, Esq. FRS &c. *Philosophical transactions of the Royal Society of London* 115 (1825), 513–583.

[11] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).

[12] Kuan Lun Huang, Salil S Kanhere, and Wen Hu. 2014. On the need for a reputation system in mobile phone based sensing. *Ad Hoc Networks* 12 (2014), 130–149.

[13] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nicholas Hynes, Nezihe Merve Gurel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards Efficient Data Valuation Based on the Shapley Value. 89 (2019), 1167–1176.

[14] Yutao Jiao, Ping Wang, Dusit Niyato, Bin Lin, and Dong In Kim. 2020. Toward an Automated Auction Framework for Wireless Federated Learning Services Market. *IEEE Transactions on Mobile Computing* (2020).

[15] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. 2019. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal* 6, 6 (2019), 10700–10714.

[16] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim. 2019. Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. 1–5.

[17] Latif U Khan, Nguyen H Tran, Shashi Raj Pandey, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. 2019. Federated learning for edge networks: Resource optimization and incentive mechanism. *arXiv preprint arXiv:1911.05642* (2019).

[18] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).

[19] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016a. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016a).

[20] Juong-Sik Lee and Baik Hoh. 2010. Sell your experiences: a market mechanism based incentive for participatory sensing. In *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 60–68.

[21] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[22] W. Liu, L. Chen, Y. Chen, and W. Zhang. 2020. Accelerating Federated Learning via Momentum Gradient Descent. *IEEE Transactions on Parallel and Distributed Systems* 31, 8 (2020), 1754–1766.

[23] Kalikinkar Mandal, Guang Gong, and Chuyi Liu. 2018. *Nike-based fast privacy-preserving high-dimensional data aggregation for mobile devices.* Technical Report. CACR Technical Report, CACR2018–10, University of Waterloo, Canada.

[24] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629* (2016).

[25] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Aguera y Arcas. 2016. Federated learning of deep networks using model averaging. (2016).

[26] Roger B Myerson. 1981. Optimal auction design. *Mathematics of operations research* 6, 1 (1981), 58–73.

[27] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.

[28] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The pagerank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[29] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. 2019. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329* (2019).

[30] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*. 2021–2031.

[31] F. Sattler, K. Müller, and W. Samek. 2020. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems* (2020), 1–13.

[32] F. Sattler, S. Wiedemann, K. R. Müller, and W. Samek. 2020. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems* 31, 9 (2020), 3400–3413.

[33] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. 2019. Can You Really Backdoor Federated Learning? *arXiv preprint arXiv:1911.07963* (2019).

[34] Nguyen H Tran, Wei Bao, Albert Zomaya, Nguyen Minh NH, and Choong Seon Hong. 2019. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 1387–1395.

[35] Guan Wang. 2019. Interpret federated learning with shapley values. *arXiv preprint arXiv:1905.04519* (2019).

[36] Jianyu Wang and Gauri Joshi. 2018. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576* (2018).

[37] Yufeng Wang, Xueyu Jia, Qun Jin, and Jianhua Ma. 2016. QuaCentive: a quality-aware incentive mechanism in mobile crowdsourced sensing (MCS). *The Journal of Supercomputing* 72, 8 (2016), 2924–2941.

[38] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2512–2520.

[39] Jia Xu, Weiwei Bao, Huayue Gu, Lijie Xu, and Guoping Jiang. 2018. Improving both quantity and quality: Incentive mechanism for social mobile crowdsensing architecture. *IEEE Access* 6 (2018), 44992–45003.

[40] D. Yang, G. Xue, X. Fang, and J. Tang. 2016. Incentive Mechanisms for Crowdsensing: Crowdsourcing With Smartphones. *IEEE/ACM Transactions on Networking* 24, 3 (2016), 1732–1744.

[41] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[42] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903* (2018).

[43] Hao Yu, Sen Yang, and Shenghuo Zhu. 2019. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5693–5700.

[44] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. *arXiv preprint arXiv:1905.12022* (2019).

[45] Rongfei Zeng, Shixun Zhang, Jiaqi Wang, and Xiaowen Chu. 2020. FMore: An Incentive Scheme of Multi-dimensional Auction for Federated Learning in MEC. *arXiv preprint arXiv:2002.09699* (2020).

[46] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).