香港城市大學
City University of Hong Kong

Department of Computer Science

BSCCS Final Year Project Report
2020-2021

20CS079

Data Valuation for Machine Learning and Federated Learning

(Volume 1 of 1)

Student Name : **CHEN, Jiaqing**

Student No. : **55202577**

Programme Code : **BSCEGU4**

For Official Use Only

Supervisor : **Dr WANG, Cong**

1st Reader : **Dr HOU, Junhui**

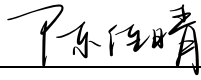
2nd Reader : **Dr CHAN, Mang Tang**

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Data Valuation for Machine Learning and Federated Learning

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: CHEN, Jiaqing Signature: 

Student ID: 55202577 Date: April 12, 2021

Abstract

Recently, federated learning (FL) emerges as a promising framework to collect the dispersed data and train a collaborative machine learning (ML) model with privacy protection. An incentive scheme plays a crucial role in the FL system as they encourage long-term client joining. However, due to information asymmetry between the central server and local users, a key challenge is to evaluate participants' contributions in an objective and efficient manner so as to allocate the payoff fairly. Data valuation in ML context is a systematic study on quantifying the usefulness of a specific data point in a prediction model. It provides a potential solution for FL to measure local client's quality. However, exponential computational complexity and additional communication costs are critical challenges of applying data valuation-based incentive schemes.

In this project, we propose a new round-based data valuation (RDV) approach to serve as a real-time incentive mechanism. It takes advantage of the FL system's unique model aggregation property to increase the valuation efficiency and provide a fine-grained contribution estimation on a per-round basis. It also offers a guideline for the central server to selectively aggregate the local updates to train a better-performing model. We empirically demonstrate the effectiveness of RDV in identifying high-quality participants, the efficiency in allocating payoff, and its potentials in federation optimization.

Acknowledgements

First of all, I would like to express my greatest gratitude and appreciation to my supervisor Prof. Cong Wang. His great vision in the latest emerging research field helps me find a cutting-edge and interesting project subject, and his exceptional enthusiasm in academia profoundly motivates me in problem exploration. I am also grateful for his supportive and patient guidance whenever I run into difficulties during my whole research period. Talking with him is always enjoyable, and I can always learn something new, about about research, study, or even life, from our conversations. To me, he's more of a life advisor than a research supervisor. I hope that my whole research and project will be able to support his research team in some way.

I want to thank Mr. Lei Xu, my senior. I discussed every possible problem in this research topic with him, and he always offered valuable help and actively shared his ideas to clear my problem. I am fortunate to collaborate with him on some research. I also want to thank Mr. Huayi Duan, who has ever provided insightful perspectives on this topic. I have studied a lot from the collaborations with them.

In addition, I appreciate our Computer Science department's prompt supervision and specific instructions on how to complete this report. I also appreciate our IT staff's efforts in providing network support during this special pandemic period.

My gratitude also extends to Miss. Xinlin Li and Miss. Yuran Sun, two of my closest friends. They are always around, company me, inspire me, and bring me a lot of joy during the whole report period as well as the 4-year university life. Especially, I want to thank Mr. Yang Lou for his help in the experiment environment setup and valuable suggestions on all aspects of my project. He also brought a great deal of warmth to my life. Also, I'd like to thank Mr. Leqian Zheng for his advice and help in writing this final report.

The last thank is reserved for my loving parents for their unending and unwavering support, both mentally and physically, during my whole university life. They provided a warm home in which I could grow up and live happily. Their encouragement is what keeps me going.

Table of Contents

1 Introduction

1.1 Background Information

Machine learning (ML) techniques have achieved remarkable results in various domains, and data is an essential ingredient in prediction tasks. However, it is costly and challenging to collect large-scale and high-quality data since the most valuable data disperse among numerous individuals ("data silos"). Directly exchanging them incurs privacy problems. The establishment of regulations like "General Data Protection Regulation (GDPR)" further makes data collection more complicated. Federated learning (FL) is a framework targeted at gathering isolated data while protecting information privacy [1][2][3][4]. It enables dispersed end-users to collaboratively train a global learning model created by the center server and exchange the encrypted model instead of raw data. FL has been applied in a variety of real-world applications. For example, with FL framework, Gboard [5], a virtual keyboard application, provides AI suggestions for end-users' chatting without storing local data in the cloud; UberEATs [6] analyzes traffic information and calculates established time for food delivery without acquiring local traffic sensing data; NVIDIA Clara [7] assists people in deciding the best treatment for patients without directly exchanging patients' sensitive privacy information.

An FL system's main performance relies on long-term local participants, high-quality data inputs, and truthful local training. Naturally, not everyone is willing to share their data truthfully without benefits. Thus, a sustainable federation is that all participating parties provide their high-quality data and receive a corresponding return cyclically [2]. However, a critical problem is how to fairly compensate those local data providers. Fairness entails rewarding those who truthfully put in data resources and training effort [8] while penalizing "free-riders [9]." A fair return encourages participants to contribute high-quality data and perform local training in the long term, while unfair allocation may lead to participant's misbehaviors or permanently leaving. Existing FL platforms, like the Federated AI Technology Enabler (FATE) [1], assume the system has already owned a stable participant group and has no need to attract more data providers. Still, such a precondition cannot be satisfied in practice, especially when the participants are business organizations. Furthermore, since servers typically have little knowledge of local users, the global model can easily be poisoned if the server blindly integrates all local models without evaluating them. Therefore, there is a need to devise an incentive mechanism, which is also a client evaluation scheme, to quantify participants' contributions, fairly allocate the payoff, and identify "free-riders" or malicious individuals.

Early incentive-related studies focus on establishing a contract with participants and determining a reward up front [10][11][12]. Although contract-theory [10][11] and game-theoretical methods [13][14] are used to categorize participants based on their previous performance and offer cor-

responding incentives, their payoffs are not directly correlated with the contributions in the current FL system. Quality-aware data valuation schemes for ML provide another guideline. They quantify how much a data sample can contribute to the performance of a prediction model. Two mainstream valuation approaches are Leave-one-out-based influence function [15] and Shapley value [16][17][18]. Both are capable of evaluating data utility equally and distinguishing high-quality data from noise. They have been applied and extended in FL to share the payoff among participants [19][8][20]. Local users' incentives are therefore explicitly dictated by their contributions in real-time.

1.2 Challenges

Even though data valuation approaches make progress in achieving contribution-related incentive mechanisms and quality-aware model training, several fundamental questions are yet to be solved. First, most data valuation-based incentive schemes determine each clients' payoff at the end of the whole FL training [8] [20]. The participation pool, however, is not constant, and local users may use varying datasets in different rounds [2]. A single valuation result is insufficient to capture the data quality changes in the fly of the training. Also, an effective incentive should allocate the payoff in real-time to encourage client's continuous joining in the subsequent rounds. On the other hand, current data valuation-based schemes suffer a high computational cost [21][18]. To assess individual utility through model output changes, it usually involves several ML model retraining, which takes a long time for deep neural networks. Such a process, additionally, incurs high communication costs in the distributed FL system due to frequent interactions between heterogeneous participants.

1.3 Major works

In this project, we focus on the scenario of a cross-device FL system [3] which has massively distributed local users and try to resolve the above issues. We leverage Shapley value [16] to design a round-based data valuation approach, quantifying each client's contribution per round and using it as a real-time incentive tool. To reduce the computational cost, we take advantage of a specific feature of FL - using the model aggregation process to replace the model retraining for contribution measurement. We theoretically prove its fairness property in each round and during the entire training period. We also propose the corresponding approximation methods to improve the calculation efficiency. Specifically, we apply two traditional sampling-based methods and design a novel clustering-based approach. We conduct a series of experiments to demonstrate their effectiveness in identifying high-quality participants. Furthermore, we make use of the valuation idea to optimize the federated training process. We directly use the Shapley value results to pick positively contributed clients for model aggregation. From the perspective of computational cost, we also suggest a RANSAC-selective strategy to measure relative contributions and select candidate local models. We empirically evaluate the effort of these selective aggregation strategies and find that

they indeed help the global model achieve a better accuracy result in a noisy data environment.

1.4 Organization

The remaining of the report is organized as follows. Section 2 discusses the related work in FL, data valuation, and incentive schemes. Section 3 systematically reviews the process of an FL system and formulates our main problems. Section 4 presents our proposed round-based data valuation method and several approximation processes. We also show how they can be utilized in federation optimization. Section 5 empirically demonstrates their effectiveness through a series of experiments. Section 6 summarizes the project as a whole, identifies limitations, and envisions future developments.

2 Literature Review

2.1 Federated learning

Federated learning (FL) is a framework that allows multiple clients to collaboratively train a universe model so that the dispersed data can be employed while preventing data leakage [5]. FL system can be categorized into cross-device FL and cross-silo FL [3]. Clients in cross-device FL are a group of mobile devices with limited computation capacity, and data is assumed to be partitioned by examples, i.e., vertical partitioned [22]. In contrast, clients in cross-silo FL are reliable organizations, such as business companies. In addition to examples, they partitioned the data by features, i.e., horizontally partitioned [23].

While FL provides an attractive framework for dealing with the "data silos" problem and decomposing an ML task into a collaborative job, it faces several practical challenges. In terms of training effectiveness, data scattered among disparate local participants are typically not independent and identically distributed (non-IID) dataset [2][4], posing additional challenges to the convergence of FL. It is also hard to guarantee each party to act their own roles properly. Lazy participants may incorporate random data in training, and malicious parties may conduct a poisoning attack that inserts noise data or construct poisonous model weights [9]. In terms of privacy, it cannot assure each party to learn nothing more than the information needed to play the roles. Attackers can compromise other participants' data privacy by carefully crafting model updates and inspecting the aggregated model [3].

2.2 Data valuation for machine learning

As the nucleus of the ML, data plays a critical role in pursuing a training task. The key to data valuation in ML context is to find a logical way to quantify data points' contributions toward a prediction model. There are two typical schemes - Leave-one-out and Shapley value.

Leave-one-out. Leave-one-out (LOO) is an intuitive data valuation method that measures a data point's contribution by how much a model's accuracy will lose after removing it [17][24]. It necessitates retraining the entire model with each coming data; thus, its computation cost linearly increases with the amount of training data. Influence function [15] is an asymptotic approximation of LOO. It uses the gradient of the loss function to compute the model parameter change when a specific data point is upweighted, then applies the result as data "influence" and further serves as a data value. This approach eliminates the model retraining process and hence improves computation efficiency. However, empirical experiments show that LOO lacks the relative utility of any two samples [18] and results in undervaluing individuals' contributions. The influence function also inherits this weakness.

Shapley value. The Shapley value (SV), named in honor of Lloyd Shapley, is a classic concept in the cooperative game to distribute the total profits generated by all players' coalition [16]. Ghorbani et al. [17] propose *Data Shapley* to leverage SV in the data valuation problem. The SV of a data sample is the average of all marginal contributions to the model considering all the possible joining orders of samples. Since SV calculation requires listing all permutations of the dataset, several approximation methods have been proposed to reduce the computation complexity, such as Monte-Carlo (MC) Sample [21], Truncated Monte-Carlo (TMC) Sampling [17], group testing [18], and model-customized calculation [24]. However, most of the approximation methods are restricted to particular scenarios or have stringent prerequisites. The efficient calculation of SV for an ML model is still a challenging problem.

2.3 Incentive mechanisms in federated learning

With the strategy of sharing profits with clients, the incentive mechanism is intended to promote client engagement and maintaining long-term collaboration with honest joiners for an FL system. There are two types of schemes. The first is focused on contract theory, which simulates the contract system in real life, assigns work effectively, and pays out according to predetermined contracts. Another is employing the aforementioned data valuation approaches to realize the payoff allocation proportional to the participant contribution.

Contract theory-based mechanisms. There exist information asymmetry issues in FL that the center has no ideas about distributed participants' data quality and available resources, and local users also lack knowledge of the center server [25]. In light of crowdsensing [26][27][28], contract theory is applied to deal with this problem and motivate model trainers to contribute more computation resources [10][11]. It divides data providers into different types according to data quality, designs specific contracts for each types, and offers different resource-reward bundles. In particular, Kang et al. [11] incorporate the reputation system to assess participants' qualifications based on the utility of their previous data and choose the most competent to participate. Lim et al. [25] combine contract theory with Stackelberg games to assign customized contracts and allocates the profit according to participant's marginal contribution. However, due to the limited number of contracts but innumerable kinds of participants, the contract-theoretic approach cannot assign each individual a perfectly matched contract form, and thus cannot achieve complete fairness.

Data valuation-based mechanism. Data valuation aims to quantify the data sample's quality and provide a guideline for an FL system to design reward allocation schemes. Richardson et al. [19] and Ma et al. [29] directly apply the idea of influence function in measuring the contribution, and Ma et al. [29] design specific protocols to perform valuation in the blockchain

environment. More researchers [8][20] [30] [29] utilize SV to ensure fairness. SV enjoys satisfactory properties to perform a fair valuation. However, there are drawbacks in directly using it in an FL system [2][20]: (1) SV calculates the expected marginal contribution by considering the utility changes in every data combination, and therefore the value does not depend on the data order. However, the FL system has an order effect that the federated model depends on the participants' joining sequence. Using SV directly will reduce the order impact. (2) The real data is hidden in local models, so SV computation cannot get direct access to the data itself. (3) Computing SV requires interacting between every possible subset of participants. It is easily realized by a centralized system, but satisfying it bears extra communication cost in the decentralized FL system. Especially, obtaining all possible subsets of the entire training process is impractical when the participant pool is changing.

This project focuses on a data valuation-based incentive scheme and proposes a round-based valuation approach that calculates the SV for each local update per round. It's not a new concept to calculate SV on a per-round basis for FL valuation. Wang et al. [20] estimate the SV in each round and place a premium on participant order. They get the final valuation result by taking the average of round values and distribute it at the end. Wang et al. [8] design a multi-round allocation matrix, assign a particular weight to each turn, and take the weighted average round value. However, the accumulated round SV cannot precisely match the overall SV without a carefully crafted round weight assignation that precisely corresponds a round to its training position in the entire FL (i.e., how much one round has contributed to the final model). The fairness effects of SV results will be harmed by the strategy in [20] that regards each round as the same and the strategy in [8] that assigns an expected smooth weight without careful measurement. In addition, there is not necessary to delay reward payment to the training end. Participants' "regret" is modeled by [31] as the time spent waiting for returns. Though it is hard to quantify different party's cost levels, it is reasonable to take the waiting time into consideration. To diminish the waiting time and improve the incentive effectiveness, we calculate the value on the fly of training and distribute the payoff in real-time after the round value is available in each training cycle. We also suggest a clustering-based approach to approximate the SV value. A most recent work, [29], also proposes applying SV on client groups, whereas the group in the work is based on SV permutation, instead of the quality similarity of different participants, which highly reduces the valuation accuracy.

3 Problem Statement

In this section, we go through the FL system structure, highlighting the position of the model aggregator and the incentive scheme, as well as the objectives we want to achieve. Typically, we choose Shapley value as our primary incentive calculation tool. We will show the corresponding data valuation process and explain why we choose SV here.

3.1 FL system framework

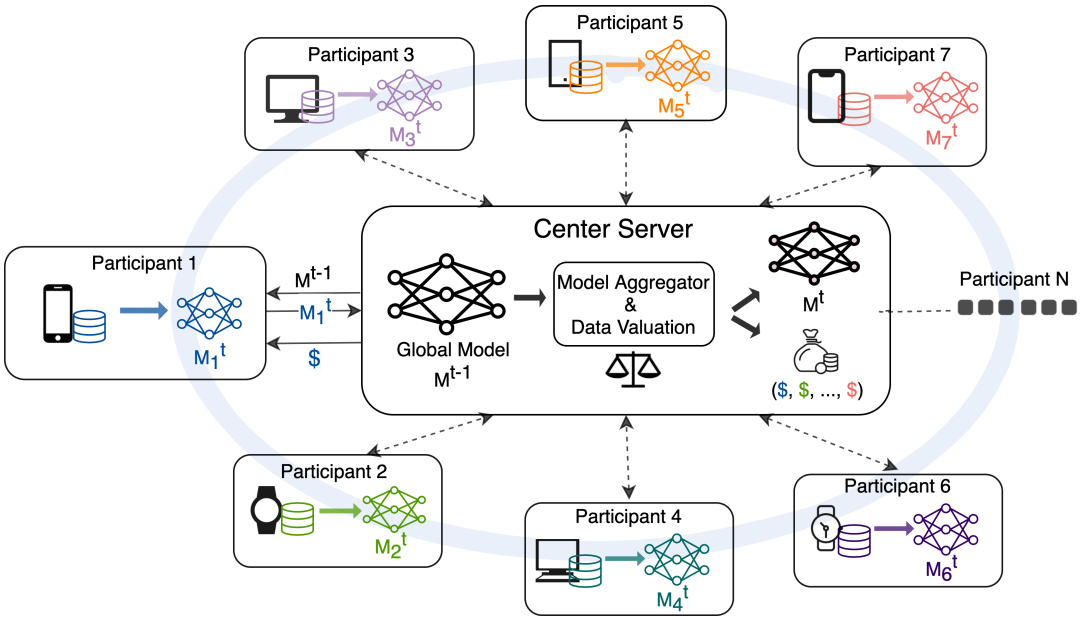


Figure 1: Framework of federated learning in the t^{th} round

As shown in Fig. 1, we consider a cross-device FL system where a large number of heterogeneous participants (also known as local users, clients, data providers) join and perform local training. Suppose there are one central server and N participants, each of whom has a local device (for example, a smartphone) with a dataset. As the FL framework in [5], we formulate the round-t federation process with incentive setting as below:

- *Step 1*: The center server has a global model M^{t-1} and sends it to the local participants.
- *Step 2*: Each client receives the global model parameters and performs *LocUpt*, which takes their owned local data to train the global model and returns the updated one – M_i^t. *LocUpt* varies with the training methods. We assume all participants use stochastic gradient descent in our system.
- *Step 3*: The model aggregator in center server averagely integrates [32] the local updates by $CtlAgg\left(M_1, M_2, \ldots, M_n\right) = \sum_{i=1}^{n} \frac{1}{n} M_i$ to get a new global model M^t for the next round.

- *Step 4*: In order to encourage participants for long-term joining and truthfully training, the incentive calculation part of the server calculates the contribution of each participant and allocates the corresponding amount of reward to participants.

Since the system follows Google's basic FL framework [5], the incentive calculation part and the model aggregator we proposed below can be directly embedded in any existing FL system. The incentive calculation tool takes all local updates in one round as input and outputs a value matrix that records each participant's calculated contribution, assisting the model aggregator in optimizing the FL training process.

3.2 Data valuation-based incentive process

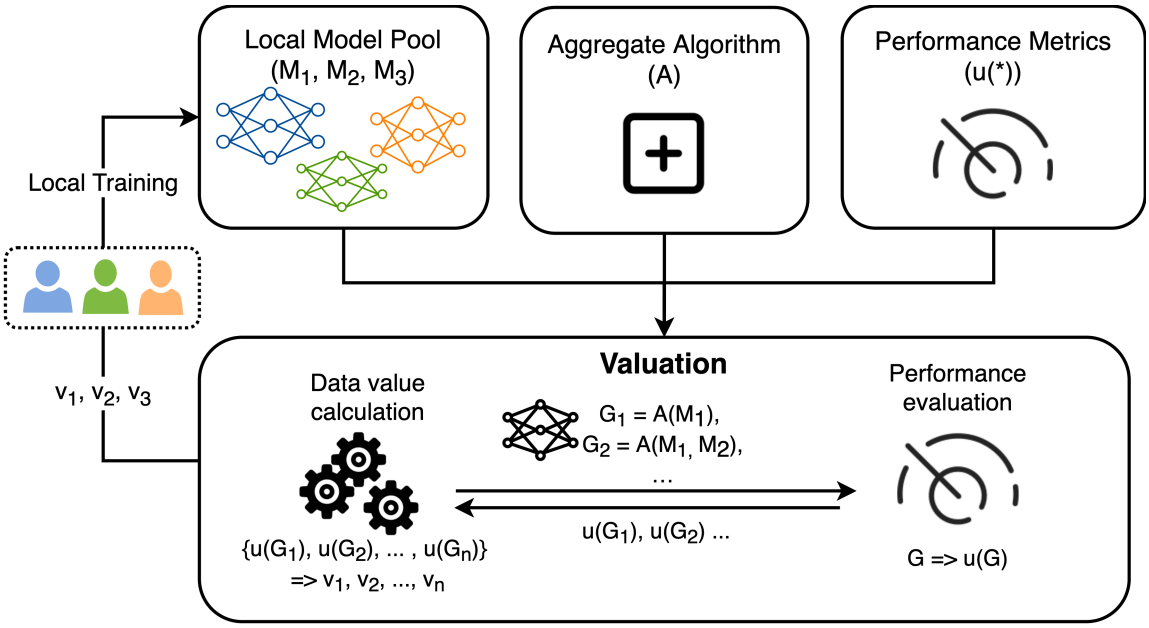


Figure 2: Overview of data valuation framework, in which M demotes the local model, A is the aggregate algorithm, $u(*)$ represents the performance metric, G is the aggregated global model and v is the value assigned to each participant

In the FL context, local data provided can be represented by the local model uploaded, and data valuation refers to the valuation of local models (also the local clients). As shown in Fig. 2, a typical data valuation process for an FL system can be formulated as the following step:

- *Step 1*: The server collects all local models M_i and forms the local model pool.
- *Step 2*: Given an aggregate algorithm A (e.g. *CtrlAgg*) and a predefined performance metric $u(*)$, the server samples different participants, retrains their models and aggregates for a global one $G = A(M)$, together with testing their performance $u(G)$ on a prepared testing set.

- *Step 3*: With returned performance scores, the server calculates each local model's marginal contribution based on a specific data valuation approach and returns the computation result v as the payoff allocated to each participant.

We define the performance metric and participant marginal contribution in our system as below.

Definition 3.1. In the FL system, given a training model M and the testing set D_{test}, the performance of the model is defined by the score function

$$v\left(M\right)=\frac{1}{|D_{\text{test}}\ |}\sum_{i=1}^{N}I\left(\widehat{y_i}=y_i\right) \tag{1}$$

where $I(*)$ is the indicator function and $\widehat{y_i}$ is the predicted result of x_i.

Definition 3.2. In a federation system with transferable utility on the finite local model set N, the marginal contribution (aka quality, utility) of a local participant $i \in N$ with respect to a subset $S \subset N/\{i\}$ is defined as

$$\Delta\left(S,i\right)=v\left(S\cup\{i\}\right)-v\left(S\right) \tag{2}$$

Why Shapley value? The Shapley value is a classic concept in the cooperative game to distribute the total profits generated by all players' coalition. As introduced in Section 2, it takes the average of one sample's all marginal contributions to the model when considering all possible dataset permutations [17][18]. Specifically, the utility of the i-th data sample can be fairly quantified as

$$\phi\left(i\right)=c\sum_{S\subset N/i}\frac{\left[v\left(M_{(S\cup i)}\right)-v\left(M_S\right)\right]}{\binom{N-1}{|S|}} \tag{3}$$

where S is a coalition of samples, N is the size of the dataset, $v\left(*\right)$ denotes a performance matrix to measure the model, and c is a constant. The reason to choose Shapley value comes from the fact that it is the unique value division scheme that satisfies the following desirable properties [17] [18]:

- **Group Rationality**: The value of the entire dataset is wholly distributed among all data samples, i.e. $\phi\left(D\right)=\sum_{i\ \in\ D}\phi\left(i\right)$.
- **Symmetry and Null player**: Two samples with identical contribution will receive the same value, and samples that contribute nothing to the model performance will receive zero payoffs. i.e. $\forall S\subseteq N\backslash\{i,j\},v(S\cup\{i\})=v(S\cup\{j\})$, then $i=j$; $v(S\cup\{i\})=v(S)$,then $i=0$.
- **Additivity**: Summing up the total value gained from multiple utility functions results in the sum of all these utilities, i.e. $\forall i\in N,\phi\left(v_1+v_2,i\right)=\phi\left(v_1,i\right)+\phi\left(v_2,i\right)$ where v_1, v_2 are two utility tests.

3.3 Design Goals

To maximize the participant's involvement willingness, we characterize the valuation requirements into two parts, each with its own definition:

(1) *Fairness*: The value assigned should be proportional to the quality of the local updates as well as the provided dataset. Those who provide high-quality data and perform truthful training should be rewarded handsomely, while those who submit data at random should be paid little to nothing.

(2) *Timely*: The calculation process should be computationally efficient, and the incentive should be distributed promptly, so that the incentive value can represent the participant's contribution in real-time, and participants do not have to wait long for their rewards.

Definition 3.3. A data valuation result of an FL system is said to be fair if it satisfies group rationality, symmetry, null player, and additivity.

Definition 3.4. A calculation process is said to be computationally efficient if it can be computed in a polynomial time.

4 Methodology

4.1 Data valuation-based incentive scheme

Considering a cross-device FL system, we propose a principled round-based data valuation (RDV) scheme. It regards each round of an FL system as an independent ML training process, calculates the Shapley value per round, and distributes the reward on the fly within the intermediate training iterations instead of waiting until all training is over.

Baseline Approach. Actually, SV calculation can be directly embedded in such a round-based valuation framework. The basic idea is to choose different model combinations, ask corresponding participants to retrain their models, and follow the SV calculation flow to compute the result on a per-round basis. This implementation is intuitive, whereas suffering high computational cost. The cost includes the SV computation cost in the center server and the high communication cost among geo-distributed local users. We call this intuitive approach definition-based data valuation (DefDV) and use it as our baseline approach. We further improve it with FL-specific features.

4.1.1 FL-specific round-based data valuation (RDV)

Inspired by the idea in [8], we only ask local users to train the model once per round and directly apply the model aggregation mechanism to measure the participant's marginal contributions instead of repeatedly retraining. This change relieves the communication cost incurred by frequent server-client interactions by only performing calculations inside the server. Since there is no order effect in a particular round, we used the ordinary Shapley value without considering a specific joining order, which completely preserves the fairness property of SV.

Algorithm 1 illustrates the process of utilizing RDV in an entire FL training process. Specifically, in the t-th round, the participant performs *LocUpt* and returns an updated result M_i^t. Before aggregating the updates, the server will: (1) Calculate the gradient of each client. (2) Randomly sample a participant pool S, get the gradient average with $\Delta^t = \frac{1}{n}\sum_{i=0}^{n-1}\Delta_i^t$ for n clients, and apply gradient descent to update the subset global model instead of asking local users for retraining them. (4) Use the reserved testing set to test the performance of the aggregated model, measure the accuracy raise and record it as the subset marginal contribution. (5) Finally, use the Shapley value calculation to obtain the round value of each individual by $\phi_i = c\sum_{S\subset N/i}\frac{\left[v\left(M_{(S\cup i)}\right)-v(M_S)\right]}{\binom{N-1}{|S|}}$. The incentive will be distributed immediately after obtaining this value in each round. We take the sum of the round values to a cumulative value to reflect the overall performance of each client in the entire federated training process.

Although the cumulative round SV still cannot match the overall SV calculated on the whole

Algorithm 1: FL-specific round-based data valuation (RDV)

Input : Participants set N, Round number T, Initial global model M^0, Evaluation metrics $v(.)$, Round weight $(w^1, ..., w^T)$
Output: Cumulative value vector ϕ, Round value vector ϕ^t for $t \in T$
Initialize $(\phi_1, \phi_2, ..., \phi_N) \leftarrow (0, 0, ..., 0)$;
Initialize $t \leftarrow 1$;
while *Convergence criteria not met* **do**
  for $i \in N$ **do**
    $M_i^t \leftarrow$LocUpt(M^t, D_i);
    $\Delta_i^t \leftarrow M_i^t - M^t$
  for $S \subset N/\phi$ **do**
    `// Local update aggregate instead of retraining`
    $\Delta_S^t \leftarrow \sum_{i \in S} \frac{1}{|S|} \cdot \Delta_i^t$;
    $M_S^t \leftarrow M^t + \Delta_S^t$;
  for $i \in N$ **do**
    $\phi_i^t \leftarrow \sum_{S \subset N/i} \frac{v\left(M_{(S \cup i)}^t\right) - v\left(M_S^t\right)}{\binom{N-1}{|S|}}$;
    $\phi_i \leftarrow \phi_i + w^t \phi_i^t$
  $M^{t+1} \leftarrow$ CtlAgg$(M_1^t, ..., M_N^t)$;
  $t \leftarrow t + 1$

FL without a well-calculated round weight allocation, we decompose the fairness requirement of the data valuation in the whole FL process to only one round. The key idea is that we conduct a per-round data valuation distribute the reward without delay, also without being taken to the next round. We can say that if the participant contribution results in each round are fair, the overall cumulative contribution to the whole training process will preserve the equitability.

Theorem 4.1. The proposed principled RDV is fair for all participants in the FL system and satisfy the following properties:

- **Round Group Rationality**: In round t, suppose I^t is participant set in round t, $\phi\left(I^t\right) = \sum_{i \in I} \phi_i^t$.
- **Symmetry and Null player**: Two samples with identical contribution in all rounds will have the same cumulative value at the end, and samples that contribute nothing in all rounds will receive zero payoffs all the time. i.e. $\forall t, S \subseteq I^t \backslash \{i, j\}$, if $v(S \cup \{i\}, t) = v(S \cup \{j\}, t)$, then $\phi_i = \phi_j$; if $v(S \cup \{i\}, t) = v(S, t)$, then $\phi_i = 0$
- **Additivity**: $\forall i \in I, \phi_i\left(v_1 + v_2\right) = \phi_i\left(v_1\right) + \phi_i\left(v_2\right)$.

Proof. In a FL process, the server computes $\{\phi_1^t, \phi_2^t, \ldots, \phi_n^t\}$ based on SV in each round. As SV satisfies group rationality, then the total value for each round is fully distributed. We termed it as round group rationality in RDV. If $v(S \cup \{i\}) = v(S \cup \{j\})$ for every round, then we have $\forall t \in T,\ \phi_i^t = \phi_j^t$, we then can derive that $\phi_i = \sum_{t=1}^T \phi_i^t = \sum_{t=1}^T \phi_j^t = \phi_j$, thus the symmetric is proved. Similarly, suppose $v(S \cup \{i\}) = 0$ for each round, it is clear that $\forall t \in T,\ \phi_i^t = 0$, and $\phi_i = \sum_{t=1}^T \phi_i^t = 0$. Finally, suppose we have $\phi_i^t\left(v_1 + v_2\right) = \phi_i^t\left(v_1\right) + \phi_i^t\left(v_2\right)$, Then,

$$\phi_i(v_1) + \phi_i(v_2) = \sum_{t=1}^{T} \phi_i^t(v_1) + \sum_{t=1}^{T} \phi_i^t(v_2) = \sum_{t=1}^{T} \left(\phi_i^t(v_1) + \phi_i^t(v_2)\right)$$
$$= \sum_{t=1}^{T} \left(\phi_i^t(v_1 + v_2)\right) = \phi_i(v_1 + v_2) + \phi_i(v_2)$$

Therefore, additivity is also proved. □

4.1.2 Estimation 1: Sampling-based RDV approximation

The major problem of SV is its computation complexity, especially with an extensive training pool. Even though RDV eliminates the retraining time cost and the communication cost, it cannot avoid the problem of $\mathcal{O}(2^n)$ permutations when calculating SV. If the number of participants increases a lot, i.e. when n becomes a large number, it is impractical to list all 2^n data combinations and compute all marginal contributions exactly.

In order to improve the calculation efficiency, we utilized the existing SV approximation methods to estimate RDV. Principally, a valuation result can be said as a (ϵ, δ)-approximation to the true Shapley value if $\Pr[\max|\hat{v}_i - v_i| \leq \epsilon] \geq 1 - 1/\delta$ [21]. The permutation sampling-based approximation method is a classical approach, which randomly chooses some permutations and estimates the average term instead of enumerating all permutations and calculating the global mean. It has been demonstrated that the SV cost can be reduced to $\mathcal{O}(NlogN)$ to achieve (ϵ, δ)-approximation where N is the number of data samples (participants). There are two types of sampling-calculation:

- **Stratified Sampling.** Like the idea of *K-subset* approximation in [18], stratified sampling method randomly choose the participant subset in every possible size, strictly record the size-k occurrence time, and reconstruct a participant's SV as his expected contribution to the k-size subsets with random cardinality. As it preserves the stratification architecture of SV, it has high approximation preciseness.

- **Monte-Carlo Sampling [21].** It treats a participant's Shapley value as its expected contribution to any participant subsets before it in a random permutation, such that the only occurrence time that needs to be recorded is the total sampling number. Based on the fact that the change in model performance weakens with more participant's joining, we utilized *Truncated Monte Carlo Sampling (TMC)* [17] to stop the contribution measurement in a sampling set and assign the remaining participant with zero values when the model performance achieves a predefined threshold. This approach only counts the sample occurrence time without considering its joining order in a specific permutation. It loses some approximation accuracy while improving the calculation efficiency.

4.1.3 Estimation 2: Cluster-based RDV approximation (CDV)

We propose a cluster-based round data valuation (CDV) scheme to further approximate the RDV. In CDV, we apply *K-means* clustering method to group participants based on the similarity of their updated models, perform the SV calculation in the unit of clusters, and assign the cluster value to each cluster member equally. We use cosine distance here to measure the similarity of local updates: Suppose M_1, M_2 are two local models and m_1, m_2 are their parameter set, the cosine distance of M_1, M_2 is defined as the following equation.

$$||M_1 - M_2||_{\cos} = 1 - < m_1, m_2 > / \sqrt{|m_1|\,|m_2|} \tag{4}$$

Algorithm 2: Cluster-based round data valuation (CDV)

Input : Participants set N, Round number T, Cluster number K, Initial global model M^0, Evaluation metrics $v(.)$, Round weight $(w^1, ..., w^T)$
Output: Cumulative value vector ϕ, Round value vector ϕ^t for $t \in T$
Initialize $(\phi_1, \phi_2, ..., \phi_N) \leftarrow (0, 0, ..., 0)$;
Initialize $t \leftarrow 1$;
while *Convergence criteria not met* **do**
  for $i \in N$ **do**
    $M_i^t \leftarrow$LocUpt(M^t, D_i);
    $p_i^t \leftarrow$ Predict(M_i^t, D_{test})
  $(C_1^t, ..., C_K^t) \leftarrow$ Clustering$(p_i^t, ..., p_N^t)$;
  for $c \in K$ **do**
    $M_c^t \leftarrow$ CtlAgg$(M_i^t$ for $i \in C_c^t)$;
    $\Delta_c^t \leftarrow M_c^t - M^t$;
  for $S \subset N/\phi$ **do**
    `// Local update aggregate instead of retraining`
    $\Delta_S^t \leftarrow \sum_{c \in S} \frac{1}{|S|} \cdot \Delta_c^t$;
    $M_S^t \leftarrow M^t + \Delta_S^t$;
  for $i \in N$, $c \in K$ **do**
    $\phi_c^t \leftarrow \sum_{S \subset K/c} \frac{v\left(M^t_{(S \cup C_c^t)}\right) - v\left(M_S^t\right)}{\binom{K-1}{|S|}}$;
    $\phi_i^t \leftarrow \frac{1}{|C_c^t|}\phi_c^t$;
    $\phi_i \leftarrow \phi_i + w^t \phi_i^t$
  $M^{t+1} \leftarrow$ CtlAgg$(M_1^t, ..., M_N^t)$;
  $t \leftarrow t + 1$

As shown in algorithm 2, after receiving the local updates $M_0^t, M_1^t, \dots M_{n-1}^t$, the server will test each M_i^t for $i \in 0, 1, 2, \dots, n-1$ on a pre-reserved testing set D_{test} and obtain a prediction result vector $p_i^t = p_0^t, p_1^t, \dots, p_{(n-1)}^t$. With the assumption that the similar prediction results imply the similar underlying dataset, n participants will be divided into k clusters $C_1, C_2, \dots, C_k$ based on the cosine distance of their prediction results. Benefited from the K-means approach, the partition satisfies that: (1)$\bigcup_{i=1}^{k} C_i = N$; (2) $C_i \cap C_j = \phi$ for any $i, j \in 1, 2 \dots K$; (3) $arg\ min_C \sum_{i=1}^{K} \sum_{p_i \in C_i} ||p_i - \bar{p_i}||_{cos}$, where $\bar{p_i}$ is the mean of cluster C_i. After partitioning parti-

cipants into clusters, the SV will be calculated in the unit of the cluster. Finally, we assume that the participants in the same cluster share a similar quality of local data and contribute nearly the same in training, so every participant in C_i receives $\phi_i = \frac{\phi_{C_i}}{|C_i|}$ as the round value. In this way, $\mathcal{O}(2^n)$ permutations will be reduced to $\mathcal{O}\big(2^k\big)$ where k is a predefined limited number, and hence the computational complexity is substantially degraded. However, it is worth noting that the efficiency improvement and the valuation accuracy are highly dependent on the choice of k, and there is a tradeoff between the efficiency and valuation accuracy.

4.1.4 Summary

A summary of the round-based Shapley value, including the baseline approach - DefDV, our proposed approach - RDV, and its estimations is listed below.

Approaches	**Computational Complexity**	**Approximation**
DefDV	$\mathcal{O}(2^n) * \mathcal{O}(E)$[1]	Exact SV
RDV	$\mathcal{O}(2^n)$	Exact SV
K-subset DV	$\mathcal{O}(n \log n)$	(ϵ, δ)-approximation
TMC-DV	$\mathcal{O}(n \log n)$	(ϵ, δ)-approximation
CDV	$\mathcal{O}\big(2^k\big)$ with predefined k	Depends on k

[1] $\mathcal{O}(E)$ represents the computatonal complexity of multiple model retraining and cmmunication costs.

Table 1: A summary of round-based data valuation schemes

4.2 Data quality-aware Federated training optimization

After receiving local updates, a primary way is to averagely aggregate the updates to get a new global model [32]. However, if a noisy dataset is incorporated or a poisonous update is uploaded, the global model's performance will be hurt, and the convergence speed will be substantially slowed down. Therefore, with the idea of measuring each participant's contribution, timely removing the hostile contributed clients and only choosing those high-quality users is a rational way to create a better training environment.

4.2.1 Random sample consensus selective aggregation

Inspired by the Random sample consensus (RANSAC) algorithm [33], we try to find the optimal participant set to update the global model in each round by iteratively sampling different combinations and testing the results. Instead of purely iteratively sampling sets, we record each test result to the selected participant as their sampling values following the idea of measuring user quality by contributions. Clearly, a higher-quality participant set will receive a higher test result, and each member in the set will gain a higher sampling value. Finally, the top-n participants with

the highest average sampling value will be chosen to aggregate for the global model.

We make some assumptions here: (1) The local model trained by a higher-quality dataset will always positively contribute to the global model's performance in any combination set, and the low-quality model vice versa. (2) It is possible that low-quality participants would gain a high sampling value if they are always chosen with high-quality, but this probability will be low with sufficient iterative round. With these assumptions, we can say that we obtain the relative quality values of each participant, and we only use the models with high-quality datasets to perform federated aggregation. The specific algorithm is listed in Algorithm 3.

Algorithm 3: RANSAC-selective model aggregation

Input : Participants set N, Round number T, Initial global model M^0, Evaluation metrics $v(.)$, Round weight $(w^1, ..., w^T)$, Iteration time k, Selection number n
Output: Updated global model M^T
Initialize $t \leftarrow 1$;
while *Convergence criteria not met* **do**
  Initialize $(\phi_1, \phi_2, ..., \phi_N) \leftarrow (0, 0, ..., 0)$;
  for $i \in N$ **do**
    $M_i^t \leftarrow$LocUpt(M^t, D_i);
    $\Delta_i^t \leftarrow M_i^t - M^t$
  for k *times* **do**
    $S \leftarrow$ Random Sample n participants
    $\Delta_S^t \leftarrow \sum_{i \in S} \frac{1}{|S|} \cdot \Delta_i^t$;
    $M_S^t \leftarrow M^t + \Delta_S^t$;
    `// Average distribute the model accuracy`
    $\phi_i \leftarrow \phi_i + \frac{v(M_S^t)}{n}$ for $i \in S$;
  $sort(\phi)$
  $G^t \leftarrow \{i | i \in N, \phi_i$ is one of top-n average values$\}$
  $M^{(t+1)} \leftarrow \sum_{i \in G^t} \frac{1}{|G^t|} M_i^t$
  $t \leftarrow t + 1$

4.2.2 Valuation-based selective aggregation

Compared with obtaining relative quality by iterative testing, data valuation here is a more effective tool to distinguish malicious people from high-quality participants. By exploiting our SV results, we propose two aggregation strategies, *Positive-Only* and *Positive-Weighted*, to optimize the federated training process:

- **Positive-Only Strategy.** In each round, after calculating the RDV of each participant, we select the local updates with positive SV into the aggregation group and perform average aggregation on only positively contributed updates. Namely, the server updates the global model as $M^{(t+1)} = \sum_{i \in N, \phi_i > 0} \frac{1}{m} M_i^t$ where m is the number of participants with positive SV.
- **Positive-Weighted Strategy.** As the value assigned is proportional to the quality of local updates, we can assume that the local model with a higher SV is more useful and should

be weighted more in the new global model. Therefore, we take the positively contributed participants and perform weighted average on their local updates. The weight is set to be $w_i = \frac{\phi_i}{\sum_{j \in N, \phi_j > 0} \phi_j}$, and $M^{(t+1)} = \sum_{i \in N, \phi_i > 0} w_i M_i^t$.

Algorithm 4: Valuation-based selective aggregation

Input : Participants set N, Round number T, Initial global model M^0, Evaluation metrics $v(.)$, Round weight $(w^1, ..., w^T)$
Output: Updated global model M^T
Initialize $(\phi_1, \phi_2, ..., \phi_N) \leftarrow (0, 0, ..., 0)$;
Initialize $t \leftarrow 1$;
while *Convergence criteria not met* **do**
  for $i \in N$ **do**
    $M_i^t \leftarrow$LocUpt(M^t, D_i);
    $\Delta_i^t \leftarrow M_i^t - M^t$
  for $S \subset N/\phi$ **do**
    $\Delta_S^t \leftarrow \sum_{i \in S} \frac{1}{|S|} \cdot \Delta_i^t$;
    $M_S^t \leftarrow M^t + \Delta_S^t$;
  for $i \in N$ **do**
    $\phi_i^t \leftarrow \sum_{S \subset N/i} \frac{v\left(M_{(S \cup i)}^t\right) - v\left(M_S^t\right)}{\binom{N-1}{|S|}}$
  `// Conclude positive contributed participants`
  $G^t \leftarrow \{i | i \in N, \phi_i^t > 0\}$;
  if *Positive-Only Strategy* **then**
    $M^{(t+1)} \leftarrow \sum_{i \in G^t} \frac{1}{|\mathrm{G^t}|} M_i^t$
  if *Positive-Weighted Strategy* **then**
    $M^{(t+1)} \leftarrow \sum_{i \in G^t} \frac{\phi_i^t}{\sum_{j \in G^t} \phi_j^t} M_i^t$
  $t \leftarrow t + 1$

5 Experiments

We design a series of experiments to test the correctness and performance of the introduced schemes from both effectiveness and efficiency perspectives. We focus on the main properties of a cross-device FL system - non-IID, unbalanced, massively distributed local training parties [32] - with the aim of demonstrating the proposed schemes' capacity and robustness in dealing with these specific FL problems.

5.1 Experiment Setup

Datasets. The experiments are conducted on the MNIST dataset [34] to perform an image-classification work in a simulated FL system. The MNIST dataset contains 60000 training images and 10000 testing images. Each image is a 28*28 matrix corresponding to a hand-written digit from 0 to 9. Among the training images, the number of images with different labels (digits) ranges from 5421 to 6742. Hence, the training set is not entirely independent and identically distributed (IID) but almost balanced.

We process the training set and distribute it to different participants into the following cases in order to simulate noisy, unbalanced, non-IID dataset problems in the FL system.

- **Standard original distribution (OD).** Based on the original shuffled training collection, we assign an equal number of data samples to each participant at random, ensuring that the data distributed are of the same size and obey the IID setting.
- **Noise Insertion (ND).** We create noise samples based on the OD dataset and distribute them to various participants in a predetermined proportion.
- **Unbalanced distribution (UD).** We randomly allocate each participant some data with different sizes.
- **Non-IID distribution.** To simulate a non-IID data environment, we first sort the data by digit label, divide it into ten parts according to the digit label, and assign each client with only two parts. For example, participant 1 has only hand-written images with labels "0" and "1", and participant 2 has only data with labels "2" and "3".
- **Dynamic situation.** We set all clients with OD dataset at first, while in the intermediate round (a) some noise is inserted; (b) some participants' training sizes are adjusted.

Environment. The codes are implemented on Python 3.7.9 with TensorFlow 2.3.1 and TensorFlow Federated 0.17.0, and the experiment is conducted on MacOS 10.15.7 with Intel Core i7 CPU @ 2.8GHz and Ubuntu 20.04 with GeForce RTX 2080 Ti.

Compared Algorithm. In the data valuation part, to verify the effectiveness, RDV and its estimations are tested to identify high-quality datasets in noisy data environments and unbalanced data cases. They are compared to the valuation results obtained using the OD data collection. To measure efficiency, we use DefDV as a baseline algorithm for calculation time comparison. The average aggregation process (namely, "Normal-strategy") is served as the benchmark in the federated optimization part.

Evaluation Principles and Metrics. A data valuation scheme can be regarded as effective if the values assigned corresponds to the dataset quality. It is said to be more efficient if it requires less valuation time per round. A selective federation approach is effective if it improves the model accuracy more in a training model. Lastly, it can be considered as Non-IID robust if the model accuracy can increase steadily in a non-IID data environment.

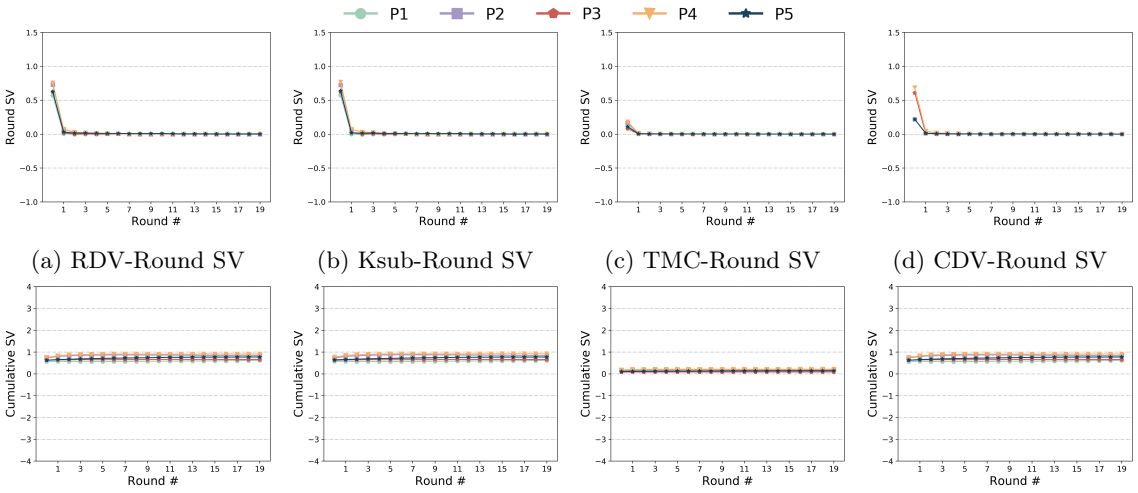
5.2 Experiment Results

5.2.1 Experiment evaluation on Round-based Data Valuation

(a) Effectiveness measurement.

In terms of effectiveness, we run RDV, K-subset DV, TMC-DV, and CDV (k=0.6*client amount) scheme on OD, ND, UD, and dynamic situation data cases. To make testing results clear, we only set 5 participants, each with 1000 local data, as the basic OD case. We perform the training in 20 rounds, recording the calculated round SV as well as a cumulative SV for each participant.

Ordinary dataset. Fig. 3 shows the round SV and cumulative SV change with the number of rounds under OD data case. It is clear that, with the same local data size and distribution, clients are assigned nearly the same SV under all approaches.

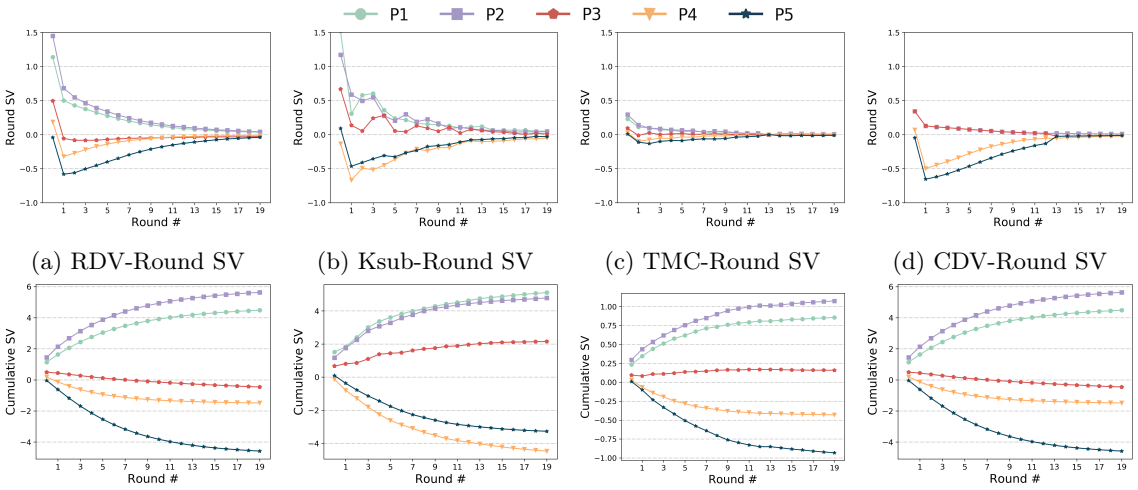


(a) RDV-Round SV (b) Ksub-Round SV (c) TMC-Round SV (d) CDV-Round SV

(e) RDV-Cumulative SV (f) Ksub-Cumulative SV (g) TMC-Cumulative SV (h) CDV-Cumulative SV

Figure 3: Effectiveness evaluation on ordinary dataset

Noisy dataset. We insert the noise in the proportion of (0.35, 0.65, 1) to the last three clients in this case. It is expected that the clean data providers will be given higher values than noisy data providers. As shown in Fig. 4, although participants' round values tend to be similar in the later rounds, they are assigned quite different values in the early period under all approaches, especially in RDV and K-subset DV (ksub) (which calculate the values more precisely). Cumulative SV results demonstrate that the more noise in the dataset, the lower the value assigned, which aligns with the expectations.

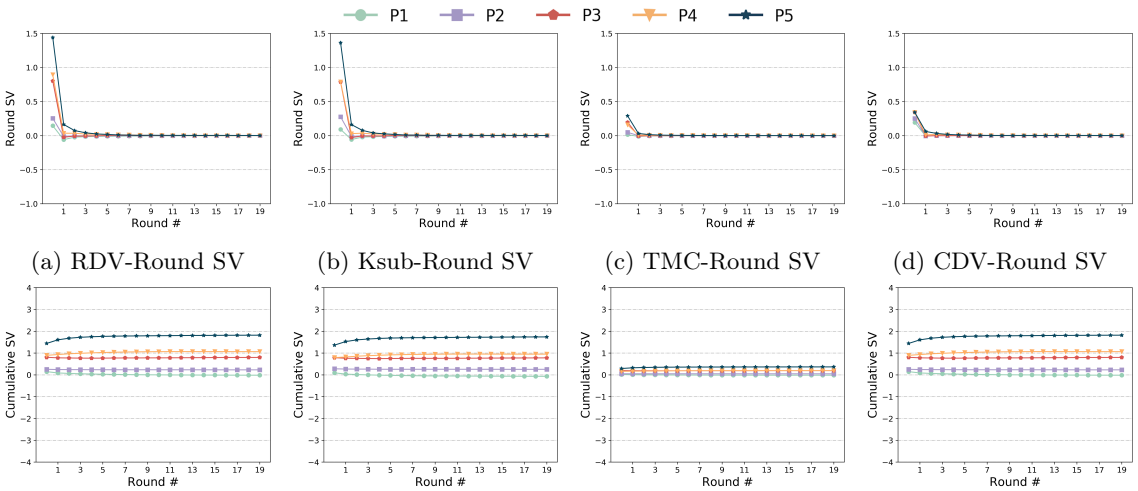


(a) RDV-Round SV (b) Ksub-Round SV (c) TMC-Round SV (d) CDV-Round SV

(e) RDV-Cumulative SV (f) Ksub-Cumulative SV (g) TMC-Cumulative SV (h) CDV-Cumulative SV

Figure 4: Effectiveness evaluation on noisy dataset

Unbalanced dataset. We allocate the data with size (100, 100, 1000, 1000, 10000) to five clients in this case. When offering the same high-quality data, those who contribute more data are supposed to receive higher returns. The results demonstrated in fig. 5 shows that the values obtained in all approaches also meet this expectation.



(a) RDV-Round SV (b) Ksub-Round SV (c) TMC-Round SV (d) CDV-Round SV

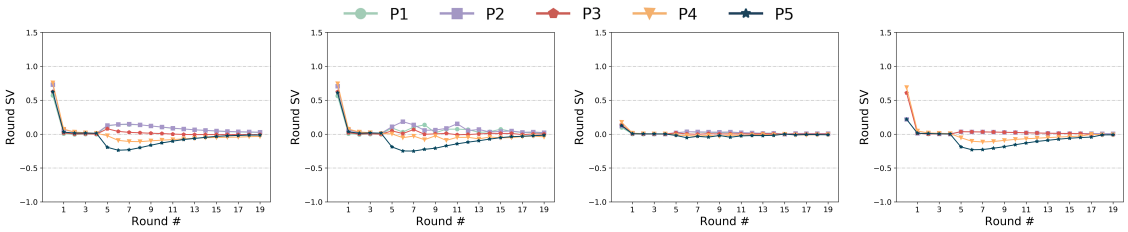
(e) RDV-Cumulative SV (f) Ksub-Cumulative SV (g) TMC-Cumulative SV (h) CDV-Cumulative SV

Figure 5: Effectiveness evaluation on unbalanced dataset

	P1	**P2**	**P3**	**P4**	**P5**
Round 1	0.5692	0.7300	0.6252	0.7569	0.6236
Round 5	-0.0002	0.0100	-0.002	0.0121	0.0123
Round 6	***0.1213***	***0.1267***	***0.0785***	***-0.0231***	***0.1946***
Round 15	0.0563	0.0465	-0.020	-0.0240	-0.0648
Cumulative (20 rounds)	0.1768	2.0962	0.7739	-0.1099	-0.8638

Table 2: Overview of RDV valuation results when adding noise in round 6

Dynamic situation. In the first case, the noise data is introduced into the last three participants with ND ratio in round 6. According to Fig. 6 and Table 2, the valuation result has a sudden shift after noise insertion, and the values are inversely proportional to the noise ratio. The cumulative Shapley value also satisfies this principle even if we insert the noise in the intermediate of the training. (we omit the cumulative Shapley value graph here.)

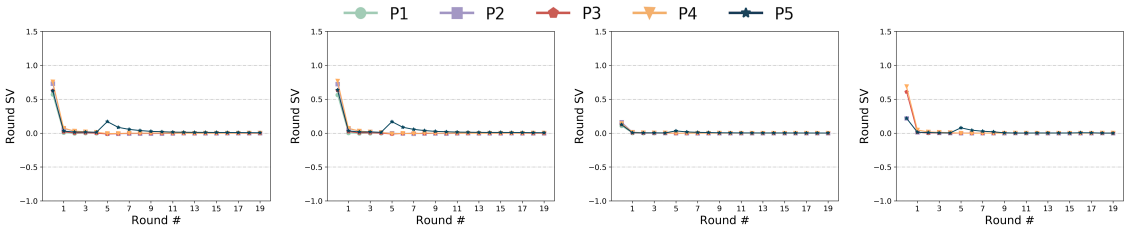


(a) RDV (b) Ksub (c) TMC (d) CDV

Figure 6: Round SV when adding noise in round 6

In the second case, the 5^{th} participant (P5) incorporates more local training data in round 6 (the data size changes from 1000 to 30000). As shown in Fig. 7, the round SV for P5 has a peak value at the size change round. According to Table 3, the SVs for all other participants are below zero in this round, mainly owing to the changed neighborhood data environment. The cumulative SV of P5 is also the highest one after 20 rounds.

These experimental results demonstrate that RDV and its estimations are able to adapt to a dynamic training pool, detect the change in the local training set timely, and reflect it to the valuation results.



(a) RDV (b) Ksub (c) TMC (d) CDV

Figure 7: Round SV when P5 adding more data in round 6

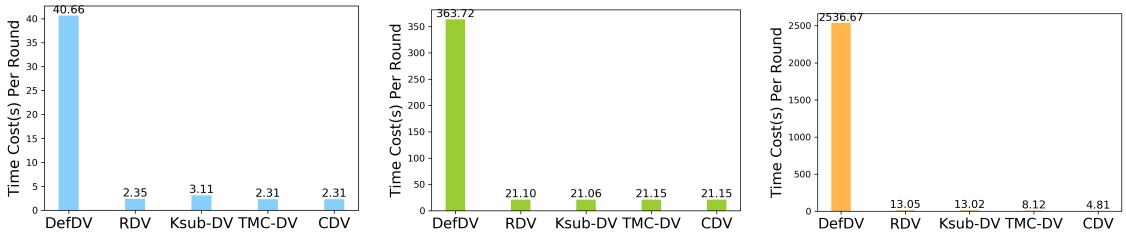
(b) Efficiency measurement.

We assess valuation efficiency from two aspects: (1) when enlarging the size of each local data,

	P1	**P2**	**P3**	**P4**	**P5**
Round 1	0.5692	0.7300	0.6252	0.7569	0.6236
Round 5	-0.0002	0.0100	-0.002	0.0121	0.0123
Round 6	***-0.0112***	***-0.0078***	***-0.0136***	***-0.0058***	***0.1696***
Round 15	-0.002	-0.0045	-0.001	0.0019	0.0148
Cumulative (20 rounds)	0.5246	0.7862	0.5967	0.8798	1.7773

Table 3: Overview of RDV valuation results when P5 adding more data in round 6

(2) when increasing the number of participants. Correspondingly, we establish "5 participants + each with 1000 data samples" as a baseline case, "5 participants + each with 10000 samples" to satisfy the first case, "10 participants + each with 1000 samples" for the second one. We apply all four approaches in all three data sets, keep track of the total time spent on each round (including training and valuation time), and compare the average terms.



(a) 5 p + 1000 local data (b) 5 p + 10000 local data (c) 10 p + 1000 local data

Figure 8: Time cost per round for DefDV, RDV, K-subset DV, TMC-DV and CDV($k = 0.6n$) under different participant settings

As reported in Fig. 8, the time cost of DefDV is much higher than other methods in all three situations. Raising the local data size and scaling up the FL system by adding more participants will both significantly increase the time costs of DefDV. In contrast, they increase our methods' time costs in a manageable way, particularly in the 10-participant environment. Especially, K-subset DV, TMC-DV, and CDV make no significant advancements in dealing with more local data scenarios (Fig. 8 (b)), but they both function for large participant situations (Fig. 8 (c)), especially CDV, which meets the massively distributed requirement of a cross-device FL system.

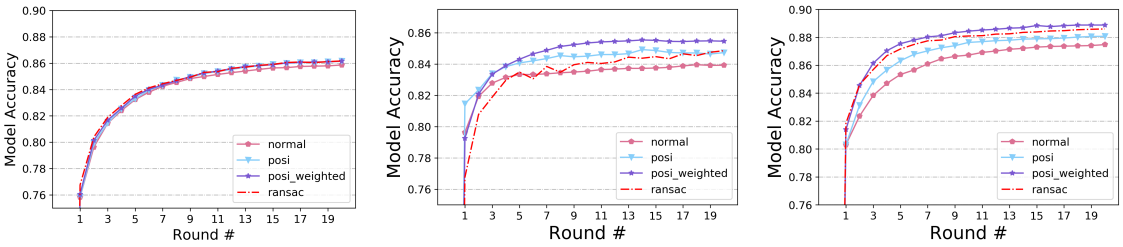
5.2.2 Experiment evaluation on federated training optimization

The goal of FL is to gain a well-performed global model after several rounds' training. We formulate the effectiveness of a federated aggregation algorithm as how much it improves the model accuracy in a given number of training rounds. We use the average aggregation *CtlAgg* scheme as the baseline and test the effectiveness of RANSAC-selective aggregation and data valuation-based Positive-Only/Weighted strategies. We record the accuracy of the global model in each round.

(a) Effectiveness on noisy data environment

Identifying high-quality local updates is the primary goal of selecting local updates for aggregation.

We construct a noisy data environment by arbitrarily adding noise or assigning data sizes to local users at random. Fig. 9 (a) shows that all aggregation strategies have the same effects in a high-quality and balanced dataset. However, in both noise data environment (Fig. 9 (b)) and unbalanced data setting (Fig. 9 (c)), RANSAC-selective and data valuation-based selective aggregation all display better performance. The model trained by these methods gains better performance after 20-round training. Among them, Positive-weighted works the best.



(a) OD environemnt (b) ND environment (c) UD environment

Figure 9: Model accuracy changes trained on OD, ND and UD data environment with average aggregation *CtlAgg* ('normal'), Positive-Only ('posi'), Positive-Weighted ('posi-weighted'), RANSAC-selective ('ransac' with iteration number k=20 and chosen model number n=4) in 20 training rounds

(b) Robustness in the non-IID data environment

As aforementioned, non-IID is a primary feature of an FL system. Due to a mismatch in the data distribution in the testing set and a specific client's local set, the same honest participants and useful model updates would possibly obtain different SVs. Therefore, it is still a question whether these selective aggregation strategies are robust to the non-IID dataset.

We allocate the clients with non-IID distribution case here. The training results shown in Fig. 10 indeed demonstrate the robustness of these three aggregation strategies in the Non-IID dataset. Compared with the average aggregation, they can achieve nearly the same model performance after several rounds' training, even if the accuracy improvement is not stable (mainly due to they remove the participants whose data distribution is largely different from the testing set in some rounds). The positive-weighted strategy is influenced by the non-IID distribution the most. For RANSAC-selective approach, we experiment two parameter settings: $n = 4$ and $n = 6$ (choose 4 or 6 participants for training per round). It can be discovered that increasing n will improve its non-IID robustness.

5.3 Experiment Summary

- **Round-based data valuation strategies.** RDV, K-subset DV, TMC-DV, and CDV are all able to identify the high-quality data, and they have substantial efficiency improvement compared with the DefDV. RDV's performance is stable and satisfactory in all data settings. K-subset DV, TMC-DV, and CDV enable the FL system to be scaled out and applicable to

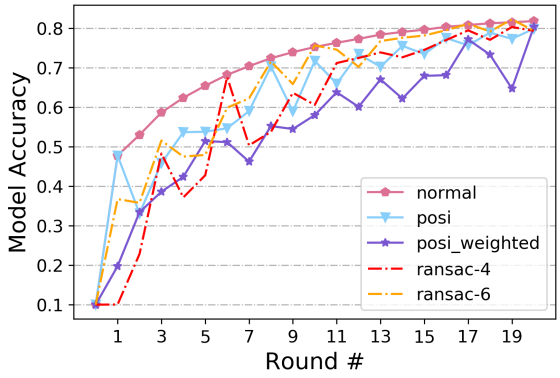


Figure 10: Accuracy change trained on Non-IID dataset with average aggregator *CtlAgg* ('normal'), Positive-Only ('posi'), Positive- Weighted ('posi-weighted') and RANSAC-selective with n=4 ('ransac-4') and n=6 ('ransac-6') in 20 training rounds

real-life cases with many participants. Considering the approximation precision, it is clear that TMC-DV has the weakest approximation performance than K-subset DV, mainly due to its unstratified estimation and truncated calculation. Our proposed CDV performs well when we set the cluster number to be 0.6 of the total participant number. However, there is, in general, a trade-off between the valuation accuracy and efficiency. A smaller number of clusters will boom the efficiency significantly while incurring less accurate valuation results simultaneously.

- **FL training optimization.** The RANSAC-selective and data valuation selective aggregation strategies are useful in FL training in a noisy data environment. They are also robust to the Non-IID dataset after some rounds' training. Specifically, the data valuation-based strategies perform better, but calculating the exact value for each round is also time-consuming, especially when the number of participants is large. The positive-weighted approach also involves individual weights calculation, which further poses computational costs. RANSAC-selective requires less computational time, but its performance depends on the parameter setting. Running more iterations and selecting more local updates will produce a better model outcome but also incur more computational costs. Therefore, the choice between different aggregation strategies and their specific working parameters can be determined based on the specific FL training context.

6 Conclusion

6.1 Summary of achievements

In this project, we propose a data valuation-based per-round incentive scheme for FL to measure the participant's contribution to the training in real-time. We theoretically prove its fairness during each round and the whole training period. It provides a guideline to allocate the payoffs proportional to each participant's contribution to the federation system for long-term joining. It also offers an idea to optimize the model aggregation process by selecting only positively contributed local updates. For efficiency consideration, we apply sampling-based approaches and devises a clustering-based method to estimate the Shapley value in a reasonable approximation bound. They make it possible to expand the FL system to a large number of local users. The feasibility and efficiency of our proposed schemes have been demonstrated by experimental tests on a real-world dataset.

6.2 Discussions and future work

As data valuation in FL is a new topic, our project only serves as a starting point and a guideline for applying data valuation in incentive and also aggregation strategies. There are still many problems from all aspects remaining to be solved in the future to make it practical in the real-world.

ML-related problems. It is clear that an ML model usually gains high performance improvement at the early stage of training, whereas this improvement fades away as the model becomes "perfect." Such a general trend has a direct impact on our accuracy-based Shapley value. Participants will begin with a large payout and end with nearly zero round rewards. Such a valuation result is particularly unfavorable to those who enter in the middle, as they will always benefit little as a result of their late entry. Although some researchers [20] argue that it is in line with the "first come, many get" principle of market economy, a more balanced payoff is still expected. Also, we assume the dataset without noise is the truthful and high-quality one that should compensate more. However, the noise sometimes benefits an ML model to make it generalize well, and those who insert noise in the local dataset may gain even more.

Valuation-related problems. In an ideal case, all honest contributors with a clean dataset will receive a positive reward in each round. Otherwise, the negative payoff would discourage the participants' involvement. However, experiment results show that the individual gain highly depends on the entire data environment. If the scale of the datasets is relatively small, honest clients can still be assigned negative values. In addition, although we theoretically prove the fairness of RDV over the entire training period, cumulative round SV is still not equivalent to

the SV calculated in the whole FL process. Designing a round-weight assign scheme, in which each client's round return is determined by both round values and round weights, is one possible solution. We have incorporated this design into the algorithm, but each round is currently given equal weight. A finely designed round-weight allocation scheme should take all factors that affect the valuation results and the training time into consideration, such as the global model complexity, the number of participants, and the size of data utilized in total.

Participant-related problems. The current incentive scheme considers a static set of participants. In real-life situations, however, the participant pool will be dynamic with people's leaving and new one's joining. Based on the principle of paying for data utility without transferring ownership, after some participants quit, the data or model shared by them cannot be reused in subsequent rounds. As a consequence, the model distributed in the following is hard to define, and the contribution of the newcomers is also difficult to evaluate.

Privacy-related problems. Data valuation-based incentive scheme also induces privacy issues: (1) Different payoff values can be room to disclose related participant information. For instance, the local updates with similar values possibly share similar features, making it possible to conduct educated guesses about the contents of the local dataset. (2) Model-specific valuation may lead to the disclosure of the global model used. Therefore, it is a challenge to protect data and model privacy both before and after valuation work, which should be concerned by both industry and academics. Furthermore, the payoff allocation requires a currency exchange, the process of which should be handled with caution.

We hope these problems and limitations can be resolved in the near future, and such a data valuation-based incentive scheme can be deployed in real life soon.

7 References

[1] *Fedai home*. [Online]. Available: https://www.fedai.org.

[2] Q. Yang, Y. Liu, T. Chen and Y. Tong, 'Federated machine learning: Concept and applications,' *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, 'Advances and open problems in federated learning,' *arXiv preprint arXiv:1912.04977*, 2019.

[4] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang and M. Guizani, 'Reliable federated learning for mobile networks,' *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[5] G. A. Blog, *Federated learning: Collaborative machine learning without centralized training dat*. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[6] H. Jeremy and D. B. Mike, *Meet michelangelo: Uber's machine learning platform*. [Online]. Available: https://eng.uber.com/michelangelo-machine-learning-platform/.

[7] K. Powell, *Nvidia build gold standard for ai infrastructure in the clinic*. [Online]. Available: https://blogs.nvidia.com/blog/2018/10/10/kings-college-london-nvidia-clara/.

[8] T. Song, Y. Tong and S. Wei, 'Profit allocation for federated learning,' in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 2577–2586.

[9] J. Lin, M. Du and J. Liu, 'Free-riders in federated learning: Attacks and defenses,' *arXiv preprint arXiv:1911.12560*, 2019.

[10] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang and D. I. Kim, 'Incentive design for efficient federated learning in mobile networks: A contract theory approach,' in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, IEEE, 2019, pp. 1–5.

[11] J. Kang, Z. Xiong, D. Niyato, S. Xie and J. Zhang, 'Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,' *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[12] H. Cai, D. Rueckert and J. Passerat-Palmbach, '2cp: Decentralized protocols to transparently evaluate contributivity in blockchain federated learning environments,' *arXiv preprint arXiv:2011.07516*, 2020.

[13] J. Nie, J. Luo, Z. Xiong, D. Niyato and P. Wang, 'A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing,' *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2018.

[14] J. Nie, J. Luo, Z. Xiong, D. Niyato, P. Wang and M. Guizani, 'An incentive mechanism design for socially aware crowdsensing services with incomplete information,' *IEEE Communications Magazine*, vol. 57, no. 4, pp. 74–80, 2019.

[15] P. W. Koh and P. Liang, 'Understanding black-box predictions via influence functions,' in *International Conference on Machine Learning*, PMLR, 2017, pp. 1885–1894.

[16] L. S. Shapley, 'A value for n-person games,' *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[17] A. Ghorbani and J. Zou, 'Data shapley: Equitable valuation of data for machine learning,' in *International Conference on Machine Learning*, PMLR, 2019, pp. 2242–2251.

[18] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song and C. J. Spanos, 'Towards efficient data valuation based on the shapley value,' in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 1167–1176.

[19] A. Richardson, A. Filos-Ratsikas and B. Faltings, 'Rewarding high-quality data via influence functions,' *arXiv preprint arXiv:1908.11598*, 2019.

[20] T. Wang, J. Rausch, C. Zhang, R. Jia and D. Song, 'A principled approach to data valuation for federated learning,' in *Federated Learning*, Springer, 2020, pp. 153–167.

[21] J. Castro, D. Gómez and J. Tejada, 'Polynomial calculation of the shapley value based on sampling,' *Computers & Operations Research*, vol. 36, no. 5, pp. 1726–1730, 2009.

[22] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato and C. Miao, 'Federated learning in mobile edge networks: A comprehensive survey,' *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[23] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečn, S. Mazzocchi, H. B. McMahan *et al.*, 'Towards federated learning at scale: System design,' *arXiv preprint arXiv:1902.01046*, 2019.

[24] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. J. Spanos and D. Song, 'Efficient task-specific data valuation for nearest neighbor algorithms,' *VLDB Endow.*, vol. 12, no. 11, pp. 1610–1623, 2019.

[25] W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung and H. V. Poor, 'Hierarchical incentive mechanism design for federated machine learning in mobile networks,' *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9575–9588, 2020.

[26] R. K. Ganti, F. Ye and H. Lei, 'Mobile crowdsensing: Current state and future challenges,' *IEEE communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[27] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing and X. Mao, 'Incentives for mobile crowd sensing: A survey,' *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2015.

[28] L. G. Jaimes, I. J. Vergara-Laurens and A. Raij, 'A survey of incentive techniques for mobile crowd sensing,' *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 370–380, 2015.

[29] S. Ma, Y. Cao and L. Xiong, 'Transparent contribution evaluation for secure federated learning on blockchain,' *arXiv preprint arXiv:2101.10572*, 2021.

[30] G. Wang, C. X. Dang and Z. Zhou, 'Measure contribution of participants in federated learning,' in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 2597–2604.

[31] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato and Q. Yang, 'A fairness-aware incentive scheme for federated learning,' in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, 'Communication-efficient learning of deep networks from decentralized data,' in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 1273–1282.

[33] M. A. Fischler and R. C. Bolles, 'Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,' *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] L. Yann, C. Corinna and J. B. Christopher, *The mnist database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/.

8 Appendix - Monthly Logs

October 2020

- Work completed: Completed the final revision of the survey. Specifically, the whole framework of the survey was changed, and some newest development, "Federated Shapley", was added into the content.
- Work in progress: (1) Literature review of data valuation framework, incentive schemes in Cross-Silo Federated Learning. (2) Identify major unsolved problems and potential proposed solutions.
- Major problems: (1) The difference between "value" and "price": Some proposed frameworks focused mainly on currency allocation while some are concerned with accurately quantifying the utility without a real paying process. The major different settings and requirements in these two scenarios need to be further researched. (2) The problem of dynamic pricing in federated learning has not been completely figured out currently. What the major challenges here? Can it be solved with some game theory strategies? Have some possible solutions been put forward?
- Plan for next month: (1) Keep on literature review regarding FL and crowdsensing to fully understand the problem of dynamic pricing and other problems as well, try to find more unsolved issues. (2) Study the knowledge related to Stackelberg Game, which could be a possible incentive mechanism designed for a non-cooperative FL scenario. (3) Start resolving the problems and designing.

November 2020

- Work completed: (1) Reviewed the related work in crowdsourcing, studied the application of Stackelberg Game. (2) Completed the interim report I. (3) Discussed with the supervisor and planned the initial design.
- Work in progress: (1) Literature review of data valuation framework, incentive schemes in Cross-Silo Federated Learning. (2) Initial design.
- Major problems: (1) How to design the two-stage valuation method to solve the problem of dynamic participants? (2) What is the major communication cost in federated learning? Has it been solved in previous work?
- Plan for next month: (1) Keep on literature review to explore the communication cost problem in federated learning. (2) Complete the initial design.

December 2020

- Work completed: (1) Reviewed the related work in communication cost problems and related security issues in federated learning. (2) Completed the preliminary design. (3) Prepared the experiment.
- Work in progress: (1) Experiment environment set up and initial test. (2) Data valuation scheme design and experiment design. (3) Follow the latest literature work in incentive scheme design.
- Major problems: (1) Experiment environment setup problems. (2) Will a cumulative stratified valuation mechanism share the same fairness with the original valuation method?
- Plan for next month: (1) Completed the experiment part to test the design. (2) Further develop the design scheme.

January 2021

- Work completed: (1) Completed the experiment design. (2) Completed the preliminary experiment part and get the experiment results. (3) Completed the interim report II.
- Work in progress: (1) Literature review on the related experiment part of some works on incentive scheme design. (2) Experiment. (3) Refine the payoff allocation algorithm design according to experiment test results. (4) Explore more problems that remain to be tackled.
- Major problems: (1) Will a cumulative stratified valuation mechanism share the same fairness with the original valuation method? (2) How to ensure that participants who contribute truthful data always get a positive Shapley value? (3) How to deal with the situation where the Shapley value of most participants tends to zero at the later period of the training?
- Plan for next month: (1) Conduct a more complex and complete experiment to test the algorithm result. (2) Further refine and develop the design scheme with the objective of solving the above problems.

February 2021

- Work completed: (1) Completed the interim report II. (2) Literature review on the latest data valuation works to refine the designed scheme. (3) Discovered several remaining problems.
- Work in progress: (1) Conduct a more complete experiment to test the algorithm result. (2) Further refine and develop the design scheme. (3) Continue literature review with the objective of resolving remaining problems.

- Major problems: (1) How to ensure stable positive for truthful contributed clients? (2) How to deal with the situation where the Shapley value of most participants tends to zero at the end of the training? (3) Theoretically prove the fairness of round-based Shapley value.
- Plan for next month: (1) Further refine and develop the design scheme with the objective of solving the above problems. (2) Theoretically prove the properties of round-base Shapley value. (3) Connect the design with existing FL / ML frameworks. (4) Start the final report.

March 2021

- Work completed: (1) Theoretically proved the properties of round-base Shapley value. (2) Used approximation approach to estimate RDV to improve efficiency. (3) Conducted a more completed experiment to test the results.
- Work in progress: (1) Write the final report. (2) Complete other theoretical proofs.
- Plan for next month: The deadlines of the final report and the presentation are approaching. In the remaining time, I will finish the last part of FYP, completing the report and preparing the presentation.

April 2021

- Check all experiment results.
- Complete and review the final report.
- Make the presentation slide and the demo.