

History-Aware and Dynamic Client Contribution in Federated Learning

Bishwamittra Ghosh^{a,*}, Debabrota Basu^b, Huazhu Fu^c, Yuan Wang^c, Renuga Kanagavelu^c, Jin Peng Jiang^d, Yong Liu^c, Rick Siow Mong Goh^c and Qingsong Wei^{c,**}

^cInstitute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore

^aMax Planck Institute for Software Systems, Germany

^bUniv. Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRISTAL, France

^dEVYD Technology, Singapore

Abstract. Federated Learning (FL) is a collaborative machine learning (ML) approach, where multiple clients participate in training an ML model without exposing their private data. *Fair and accurate assessment of client contributions* facilitates incentive allocation in FL and encourages diverse clients to participate in a unified model training. Existing methods for contribution assessment adopts a co-operative game-theoretic concept, called Shapley value, but under *restricted assumptions*, e.g., all clients' participating in all epochs or at least in one epoch of FL.

We propose a history-aware client contribution assessment framework, called FLContrib, where client-participation is dynamic, i.e., a subset of clients participates in each epoch. The theoretical underpinning of FLContrib is based on the Markovian training process of FL. Under this setting, we directly apply the linearity property of Shapley value and compute a historical timeline of client contributions. Considering the possibility of a limited computational budget, we propose a *two-sided fairness criteria* to schedule Shapley value computation in a subset of epochs. Empirically, FLContrib is efficient and consistently accurate in estimating contribution across multiple utility functions. As a practical application, we apply FLContrib to detect dishonest clients in FL based on historical Shapley values.

1 Introduction

Over the last decade, Federated Learning (FL) [12, 14, 36] has emerged as the de facto standard for collaborative Machine Learning (ML) without exposing private data. FL facilitates the involvement of multiple data owners or local clients in training a global ML model without the need to share any raw data [4, 26, 33], and thus, it is aligned with the standard data protection policies like GDPR [29]. As the success of FL depends on the participation of a diverse pool of data contributing clients, a *fair and accurate assessment of client contribution* is important to facilitate incentive allocation [34, 16], resolve free rider issues [8, 38], and encourage diverse clients to participate in FL training [23, 19]. Thus, assessing client contribution in FL has become a question of growing interest [30, 31, 32].

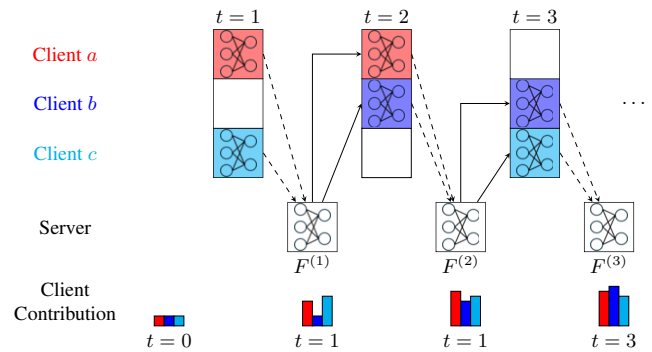


Figure 1: An illustration of FLContrib, where clients participate in FL dynamically (empty square indicates non-participation). The theory behind contribution assessment is the Markovian training process of FL, where the global model update at epoch t depends on local model updates by participating clients at epoch t , upon receiving the global model from epoch $t - 1$. Thus, at any epoch, the *incremental* contribution is distributed over participating clients, and non-participating clients receive null contribution. Their *total* contribution is aggregated over the history of contributions of all past epochs.

Client Contribution in FL. The existing client contribution assessment techniques in FL can be categorized in two ways: individual assessment and coalition assessment. In individual assessment, a local client is evaluated based on its similarity with the global model, the local relative accuracy, or the reputation of the client [21, 23]. In coalition assessment, the marginal effect of a local client is evaluated when it joins a set of other clients in the training process [9, 18, 30]. Coalition assessment is known to be more effective than individual assessment in the heterogeneous data setting, where a client only has a partial coverage of the feature space but the global model needs to cover the whole feature space [31]. But a coalition-based assessment can depend on the order in which the clients join in the training and are being evaluated.

Shapley Value as Client Contribution. To address the client-ordering issue in coalition assessment, concepts from co-operative game theory, such as Shapley value [24], are proposed for assessing client contribution in FL [13, 18, 30, 31], where Shapley value deter-

* Corresponding Author: bghosh@mpi-sws.org.

** Corresponding Author: wei_qingsong@ihpc.a-star.edu.sg

mines how much a client contributes to the total utility of a coalition of clients. However, majority of the methods consider *restricted assumptions or rely on heuristic definitions of client contribution using Shapley value*. For example, Liu et al. [18], Song et al. [25], Wei et al. [32] considered that all clients participate in all training epochs to perform Shapley value computation, failing to simulate a real-world setting where only a subset of clients participates in training at any given epoch [6, 31]. To address this issue, Fan et al. [6] considered a non-participating-client setting, but their method of applying low rank matrix factorization incurs poor scalability and accuracy, and relies on restricted assumptions, such as Lipschitz continuous model loss and all clients mandatorily participating together in at least one epoch. In a similar direction, Wang et al. [31] defined the contribution of non-participating clients as a heuristic, without providing any theoretical justification. Therefore, our research question considers a practical FL scenario: *Can we scalably and accurately assess client contribution in federated learning, and provide a theoretical justification, when a subset of clients participates in a training epoch?*

Proposed Framework. As an affirmative answer, we propose a Shapley value based framework, called FLContrib, for a history-aware and dynamic contribution assessment of participating and non-participating clients across multiple utility functions, such as cross-entropy loss or accuracy. In particular, we resort to the Markovian training process of FL (Figure 1), where the global model update at epoch t (a new state) depends on local model updates by participating clients at epoch t , upon receiving the global model from epoch $t - 1$ (previous state). As such, the contribution of the current epoch is distributed to participating clients in that epoch, and non-participating clients are given null contribution. Furthermore, the Markovian training process allows to directly apply the linearity property of Shapley value, where the Shapley value of a client in all epochs is the sum of incremental Shapley values of the client in individual epochs. In the following, we discuss our **three-fold contributions**:

- **History-aware and Dynamic Client Contribution.** Due to Markovian FL training process and the application of the linearity property of Shapley value, we theoretically demonstrate that the total Shapley value of a client is the sum of incremental Shapley values in all epochs, where client-participation is dynamic. Specifically, at any epoch, the runtime complexity for computing Shapley value is dictated by participating clients only. In Lemma 3.6, the exact Shapley value computation is exponential with participating clients and linear with non-participating clients. FLContrib offers a modular design, allowing us to apply off-the-shelf efficient approximation algorithms for computing Shapley value, such as via Monte Carlo sampling [3] and its variants [13, 28, 30], and also complementary contribution analysis [37]. Furthermore, the separation between contribution assessment of participating and non-participating clients in FLContrib allows us to adapt existing approaches such as Liu et al. [18], Wei et al. [32] focusing solely on participating clients to facilitate non-participating clients' contribution assessment.
- **Accuracy-efficiency Trade-off.** Computing Shapley value in all epochs leads to a more accurate assessment of client contribution, but at the cost of computational inefficiency. We simulate a trade-off between accuracy and efficiency of contribution assessment by scheduling Shapley value computation over a *desirable* subset of epochs. Unlike existing methods ignoring the history of client participation [18, 25], we consider a two-sided fairness objectives for an optimal scheduling. Server-side fairness prioritizes epochs with higher incremental utility and higher client exposition, i.e.,

allowing Shapley value computation in epochs when a client participates. Client-side fairness, on the other hand, aims to minimize the pair-wise difference of exposition probability of a client with other clients. In both cases, the incremental utility of an epoch is used to prioritize Shapley value computation when global model has more impact. We express the scheduling problem as a linear program and show that FLContrib achieves an improved trade-off between accuracy and efficiency of contribution assessment.

- **Application: Detecting Dishonest Clients.** FLContrib yields historical client contribution over epochs as a time series of Shapley values. In a controlled experiment, we leverage such data to analyze client intention in FL training, such as identifying dishonest clients who intentionally poison their local data during training. Empirically, Shapley values via FLContrib identify the poisonous window where the client is dishonest and separate honest clients from dishonest ones.

Motivation and Extended Related Work

Several approaches have applied Shapley value to assess client contribution in FL. Notably, Song et al. [25] utilized the gradients of local clients and proposed an one-epoch evaluation and a multi-epoch evaluation (MR) for client contribution. Wei et al. [32] extended MR and proposed a truncated multi-epoch (TMR) evaluation to eliminate an entire epoch based on a pre-defined threshold. Liu et al. [18] further improved computation by eliminating unnecessary epochs based on the incremental utility and proposed an improved permutation ordering of clients during assessment. *In all these methods, all clients must participate in each training epoch, which we relax in our study.*

To our knowledge, two works have considered a non-participating-client setting like ours. Without any formal basis, Wang et al. [31] defined the Shapley value of participating and non-participating clients at each epoch: Shapley value is initialized to 0, and then at every epoch, Shapley values of participating clients is computed using the whole history of their participations. In contrast, our analysis derives the Shapley values of all the clients exactly over a Markovian FL training and also provides the flexibility to use different initializations if one performs a warm start. To address non-participating clients, Fan et al. [6] proposed to guess the missing utility of a client-combination when a subset of clients participates in training. They relied on low rank matrix factorization to approximate the incomplete utility matrix. Their method has several limitations: low rank matrix factorization requires a suitable characteristic of the utility function, such as Lipschitz continuous model loss, and a restricted assumption on client participation, where all clients must participate together in at least one epoch. Furthermore, matrix factorization is a one-shot approach, and thus, assessing contributions in a new epoch requires re-computation of the whole matrix, which is infeasible in practice. As such, Fan et al. [6] incurred high computational time and estimation error. *In our study, we aim to achieve both improved accuracy and efficiency in assessing client contributions with dynamic client participation.*

2 Preliminaries: FL and Shapley Value

Federated Learning (FL). FL is a collaborative ML framework that allows the training of multiple clients with local private data [12]. We consider a centralized single server-based FL setting, where the server is trustworthy and performs both model aggregation and contribution assessment (Figure 1).

Let I denote the set of all clients with cardinality $m \triangleq |I|$. F denotes the global model, and T denotes the total number of training epochs. In each epoch t , $1 \leq t \leq T$, a subset of clients $I^{(t)} \subseteq I$ is selected for training. Each client $i \in I^{(t)}$ receives the global model $F^{(t-1)}$ from the last epoch, trains a local model $F_i^{(t)}$ on the local data D_i , and sends the local gradient $\Delta_i^{(t)} \triangleq F_i^{(t)} - F^{(t-1)}$ to the server for aggregation. $\Delta_i^{(t)}$ may include multi-step gradients performed by the client. FLContrib is agnostic to the gradient updating algorithm.

The server preforms a federated aggregation on received local gradients $\{\Delta_i^{(t)}\}_{i \in I^{(t)}}$ to derive a global model in each epoch. For example, the FedAvg algorithm performs the aggregation by weighing the local gradients with the relative local data size [20]. In fact, FedAvg aggregates local models instead of local gradients, which is mathematically equivalent.

$$F^{(t)} = F^{(t-1)} + \sum_{i \in I^{(t)}} \frac{|D_i|}{\sum_{i' \in I^{(t)}} |D_{i'}|} \Delta_i^{(t)} \quad (1)$$

We consider a *utility function* $v(F, D_{\text{val}}) \in \mathbb{R}$ to evaluate F on a hold-out validation dataset D_{val} . The utility function can be multi-dimensional, such as model loss, accuracy, performative-fairness [35], demographic-fairness [22], or their combination. We use $v(F)$ to denote the utility when it is clear from the context.

Shapley Value. In co-operative game theory, Shapley value [24] computes a unique distribution of the total utility in a coalition of m players. Thus, Shapley value provides the premise to assess client contributions in FL, where the total utility is distributed based on the marginal contributions of clients in all possible coalitions.

To compute Shapley value, the utility function needs to be evaluated on a FL sub-model, which is aggregated over a subset of clients' models – our goal is to find the marginal utility when a client's model is added to the sub-model. We denote the sub-model as F_S and the utility as $v(F_S, D_{\text{val}})$, where $S \subseteq I$. Avoiding notational clutter, we denote $u(S) \equiv v(F_S, D_{\text{val}})$ as the utility of the respective subset of clients, where $u : 2^I \rightarrow \mathbb{R}$.

Therefore, Shapley value $\varphi_i(u)$ of a client i is defined as

$$\varphi_i(u) \triangleq \frac{1}{m} \sum_{S \subseteq I \setminus \{i\}} \frac{1}{\binom{m-1}{|S|}} (u(S \cup \{i\}) - u(S)), \quad (2)$$

where $u(S \cup \{i\}) - u(S)$ is the marginal utility of client i for including i to the subset of clients $S \setminus \{i\}$. Informally, Shapley value computes the average marginal utility of the client i in all possible coalitions. Shapley value satisfies a set of desirable properties [24].

- **Decomposability (or efficiency):** The sum of Shapley values of all clients is the total utility of the global model, $\sum_{i \in I} \varphi_i(u) = u(I) = v(F, D_{\text{val}})$.
- **Symmetry:** Two clients $i, i' \in I$ contribute equally $\varphi_i(u) = \varphi_{i'}(u)$, if their marginal utilities are equal $u(S \cup \{i\}) = u(S \cup \{i'\})$ for every subset of clients $S \subseteq I \setminus \{i, i'\}$ not containing i, i' .
- **Null Client:** A null client does not contribute. The Shapley value of a null client is zero, $\varphi_i(u) = 0$. This happens when the marginal utility of the client is zero, $u(S \cup \{i\}) - u(S) = 0$, for every subset of clients $S \subseteq I \setminus \{i\}$ not containing i .
- **Linearity:** Let the utility function be a linear combination of multiple utility functions. Given $u_1, u_2 : 2^I \rightarrow \mathbb{R}$, if $u = u_1 + u_2$, then the Shapley value of a client on the combined utility is a linear combination of Shapley values on individual utilities, $\varphi_i(u) = \varphi_i(u_1 + u_2) = \varphi_i(u_1) + \varphi_i(u_2)$.

In the paper, we apply the *decomposability and null-client properties* to assess contributions of participating and non-participating

clients, respectively. In addition, we apply the *linearity property* to account for the Markovian training process in FL, enabling us to perform epoch-wise contribution assessment.

3 Methodology

We present FLContrib, a game-theoretic framework to assess history-aware client contributions in a single-server federated learning with dynamic client participation. We first formalize the problem statement and present a baseline algorithm to exactly assess client contribution with a non-participating-client setting (Section 3.1). Then, we discuss an efficient scheduling procedure for a faster client contribution assessment (Section 3.2).

Problem Statement. Our objective is to assess contributions of clients by computing their Shapley values. Given (i) the initial global model $F^{(0)}$, (ii) the gradients of participating clients in T training epochs $\{\{\Delta_i^{(t)}\}_{i \in I^{(t)}}\}_{t=1}^T$, and (iii) an utility function v , we compute the Shapley value $\varphi_i(v)$ of all clients $i \in I$. Intuitively, between two clients $i, i' \in I$, if i has higher Shapley value than i' , $\varphi_i(v) > \varphi_{i'}(v)$, then i contributes more to the utility than i' .

3.1 Exact Client Contribution over Epochs

To access contribution exactly, we divide key concepts into three: per-epoch Shapley value computation due to Markovian FL training, an extension of FL model update to incorporate non-participating clients, and an adaptive sub-model reconstruction to compute marginal utility. We conclude this subsection by discussing runtime complexity and error bounds.

Shapley Value on Incremental Utility. FL undergoes a Markovian training process: In each epoch, the global model depends on participating clients' local models, independent of how the global model is updated in earlier epochs. As such, we directly apply the linearity of Shapley value by decomposing the total utility of the global model in all epochs as the sum of incremental utilities between consecutive epochs – analogously, the Shapley value on the total utility is the sum of incremental Shapley values in every epoch. Most existing methods proposed to compute incremental Shapley value like ours [18, 25, 31], but none made the precise connection between Markovian FL training and the linearity of Shapley value. Next, we define the incremental utility between epoch t and $t - 1$ as the difference of the model utility in consecutive epochs.

$$\delta v(F^{(t)}, F^{(t-1)}) \triangleq v(F^{(t)}) - v(F^{(t-1)}). \quad (3)$$

Lemma 3.1 (Utility Decomposition). *The total utility in a multi-epoch FL training is the sum of incremental utilities in all the training epochs and the utility of the initial FL model.*

$$v(F^{(t)}) = \sum_{t=1}^T \delta v(F^{(t)}, F^{(t-1)}) + v(F^{(0)}), \quad (4)$$

where $v(F^{(0)})$ denotes the utility of the initial model.

Applying the linearity of Shapley value, the total Shapley value of a client after all epochs is the sum of Shapley values computed on the incremental utility in each epoch.

Lemma 3.2 (Contribution Decomposition). *For a client i , let $\varphi_i(v)$ be the Shapley value after T epochs, and let $\varphi_i^{(t)}(\delta v)$ be the Shapley value on the incremental utility at epoch t . Since v is a linear sum of*

δv , we apply the linearity of Shapley value: the Shapley value w.r.t. v is the sum of Shapley values w.r.t. δv between epoch 1 to T and the Shapley value of the initial model.

$$\varphi_i(v) = \sum_{t=1}^T \varphi_i^{(t)}(\delta v) + \varphi_i^{(0)}(v) \quad (5)$$

Participation-aware Model Update. To assess contributions of both participating and non-participating clients, we express the FL model update at an epoch using all clients.

$$\begin{aligned} F^{(t)} &= F^{(t-1)} + \sum_{i \in I^{(t)}} w(i, I^{(t)}) \Delta_i^{(t)} \\ &= F^{(t-1)} + \sum_{i \in I} \mathbb{1}(i \in I^{(t)}) w(i, I^{(t)}) \Delta_i^{(t)}, \end{aligned}$$

The indicator function $\mathbb{1}(\cdot) \in \{0, 1\}$ returns 1 for a true argument, such as when client i participates at epoch t , and 0 otherwise. To simplify notation, we define a *participation-aware weight function* λ by multiplying indicator $\mathbb{1}$ and local weight w , and use λ to extend Eq. (1) to all clients. Thus, λ is equal to w when client i participates, and 0 otherwise.

$$\lambda(i, S) = \mathbb{1}(i \in S) w(i, S) = \begin{cases} w(i, S) & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$F^{(t)} = F^{(t-1)} + \sum_{i \in I} \lambda(i, I^{(t)}) \Delta_i^{(t)} \quad (7)$$

Example 1. In Figure 1, let three clients be $I = \{a, b, c\}$ with $|D_a| = |D_b| = |D_c|$. In epoch $t = 1$, $I^{(1)} = \{a, c\}$. Hence, $\lambda(a, I^{(1)}) = \lambda(c, I^{(1)}) = \frac{1}{2}$ and $\lambda(b, I^{(1)}) = 0$.

Adaptive Sub-model Reconstruction. An intermediate step for computing Shapley value is to compute the marginal utility of a client w.r.t. its presence and absence in a subset of other clients. Instead of retraining on all possible subsets, we *reconstruct* a sub-model by storing and utilizing local gradients of participating clients at the FL server, similarly as Liu et al. [18], Song et al. [25]. Importantly, we consider an *adaptive weighting* of local gradients to ensure that each sub-model reconstruction mimics the full global model training. In particular, the weight is defined as $w(i, S) \triangleq |D_i| / (\sum_{i' \in S} |D_{i'}|)$, which is adaptive w.r.t. S . The adaptive weighting is overlooked in earlier methods by Lin et al. [17], where $w(i, S) = |D_i| / (\sum_{i' \in I} |D_{i'}|)$, ignoring the effect of S , and naively normalizing for all clients I . We incorporate this weight into the participation-aware weight function λ for reconstructing the sub-model $F_S^{(t)}$.

$$F_S^{(t)} = F^{(t-1)} + \sum_{i \in S} \lambda(i, S \cap I^{(t)}) \Delta_i^{(t)} \quad (8)$$

Here, $\lambda(i, S \cap I^{(t)})$ ensures that only participating clients in $S \cap I^{(t)}$ receive non-zero weights, i.e., their Shapley value is computed in the traditional way, since they explicitly update the global model at the current epoch. In contrast, non-participating clients do not influence sub-model reconstruction (Lemma 3.3) and thereby receive a deterministic Shapley value (Lemma 3.4).

Lemma 3.3 (Influence of Non-participating clients). *Let $F_S^{(t)}$ be a sub-model consisting of a subset of clients $S \subseteq I$ and $i \notin I^{(t)}$ be a non-participating client in epoch t . Non-participating clients do not influence sub-model reconstruction, formally $F_{S \cup \{i\}}^{(t)} = F_S^{(t)}$.*

Algorithm 1 Assessing Client Contribution

```

1: Input: Initial global model  $F^{(0)}$ , clients  $I$ , utility  $v$ .
2:  $\varphi_i^{(0)} = \frac{v(F^{(0)})}{|I|}, \forall i \in I$ 
3: for  $t = 1$  to  $T$  do
4:    $I^{(t)} \leftarrow \text{ClientSelection}(I)$ 
5:    $\Delta_i^{(t)} \leftarrow \text{ClientTraining}(F^{(t-1)}, D_i), \forall i \in I^{(t)}$ 
6:    $F^{(t)} \leftarrow \text{FedAggregate}(F^{(t-1)}, \{\Delta_i^{(t)}\}_{i \in I^{(t)}})$ 
7:    $\{\varphi_i^{(t)}\} \leftarrow \text{FLContrib}(\delta v, \{\Delta_i^{(t)}\}, F^{(t-1)})$ 
8: end for
9: return  $\{\sum_{t=0}^T \varphi_i^{(t)}\}$ 

```

Example 2 (Continuing Example 1). Since $I^{(1)} = \{a, c\}$, then client b does not participate in epoch $t = 1$. Hence, $F_{\{a,b,c\}}^{(1)} = F_{\{a,c\}}^{(1)} = F^{(0)} + \frac{1}{2} \Delta_a^{(1)} + \frac{1}{2} \Delta_c^{(1)}$, $F_{\{a,b\}}^{(1)} = F_{\{a\}}^{(1)} = F^{(0)} + \Delta_a^{(1)}$, $F_{\{b,c\}}^{(1)} = F_{\{c\}}^{(1)} = F^{(0)} + \Delta_c^{(1)}$ and $F_{\emptyset}^{(1)} = F^{(0)}$.

Lemma 3.4 (Deterministic Contribution). *At epoch t , the Shapley value of a non-participating client $i \notin I^{(t)}$ with respect to the incremental utility δv is zero, $\varphi_i^{(t)}(\delta v) = 0$. Therefore, the non-participating client is a null client.*

Wang et al. [31] *heuristically* defined the same deterministic Shapley value of non-participating clients, whereas our paper provides necessary theoretical justification.

Remark 3.5 (Initial FL model and Shapley Value). We explicate the need to compute Shapley value of the initial FL model, which can be a randomly initialized model or a model after warm start. Hence, the utility of the initial FL model $F^{(0)}$ is not necessarily zero, rather the utility of $F^{(0)}$ on the validation dataset D_{val} . Thus, a reasonable initial Shapley value is the per-client utility of the initial model, $\varphi_i^{(0)}(v) = v(F^{(0)})/|I|$.

Runtime Complexity For Exact Computation. Due to the deterministic Shapley value of non-participating clients, the runtime complexity is linear for non-participating clients and exponential for participating clients. Non-participating clients incur a linear complexity since their contribution is recorded as a history for later use. The exponential complexity of participating clients is a known result for an exact Shapley value computation.

Lemma 3.6 (Runtime Complexity). *Let m be the total number of clients and $\frac{1}{\tau} \in [0, 1]$ be the ratio of participating to non-participating clients. In an epoch, the runtime complexity of exactly computing Shapley value is $\mathcal{O}(2^{\frac{m}{\tau}} + (1 - \frac{1}{\tau})m)$. In T epochs, the total running time is $\mathcal{O}(2^{\frac{m}{\tau}} T + (1 - \frac{1}{\tau})mT)$.*

Modular Design and Approximation Error. FLContrib has a modular design, where a wide variety of Shapley value algorithms as well as existing contribution methods in FL can be applied (ref. Algorithm 1). The exact computation can be replaced with approximation algorithms, such as Monte Carlo sampling [9] or complementary contribution analysis [37]. Also, existing methods by Liu et al. [18], Wei et al. [32] can be applied, where non-participating-client setting is ignored: In each epoch, we apply their Shapley value computation algorithm to participating clients and compute a deterministic Shapley value for non-participating clients. Below, we provide the error bound of FLContrib as a meta approach invoking existing Shapley value algorithms.

Lemma 3.7 (Error Bound). *If we apply an ϵ -approximation algorithm to compute the incremental Shapley values at each epoch, the*

total estimation error in the global Shapley value is $\mathcal{O}(\frac{T\epsilon}{\tau})$, which is of the same order as existing algorithms.

3.2 A Scheduler for Faster Computation

We discuss a scheduling procedure to assess client contribution efficiently without sacrificing accuracy. We consider a computational budget, such as the maximum number of epochs to compute Shapley value, and derive an optimal subset of epochs by solving a constrained optimization problem.

We propose two-fold objectives in the optimization problem to incorporate the history of client participation. **Server-sided fairness:** The server aims to increase utility coverage by computing Shapley value in a desirable subset of epochs where the sum of incremental utilities is higher. Also, the server prioritizes higher exposure of clients: informally, a client is said to be exposed if it participates in training in an epoch and Shapley value is computed in the same epoch. **Client-sided fairness:** Each client aims to minimize the pair-wise difference of exposure probabilities with other clients. Now, we discuss two mixed integer linear programs (MILP) to achieve (i) only server-sided fairness and (ii) two-sided fairness [5].

An ILP for One-sided Fairness. Let $z^{(t)} \in \{0, 1\}$ be a binary variable indicating whether Shapley value is computed in epoch t . The epoch weight $p^{(t)}$ is defined as normalized absolute incremental utility, $p^{(t)} = \frac{|\delta v(F^{(t)}, F^{(t-1)})|}{\sum_{t=1}^T |\delta v(F^{(t)}, F^{(t-1)})| + |v(F^{(0)})|}$, and $\sum_{t=1}^T p^{(t)} = 1$. For each client, we first compute per-epoch participation rate, $x_i^{(t)} = \mathbb{1}(i \in I^{(t)}) / \sum_{t=1}^T \mathbb{1}(i \in I^{(t)})$. Hence, the exposure probability of a client at epoch t is defined as $x_i^{(t)} z^{(t)}$. Let $k \leq T$ denote the maximum number of epochs to compute Shapley value. The following ILP program satisfies server-sided fairness.

$$\begin{aligned} \max_{\mathbf{z}} \quad & \sum_{t=1}^T p^{(t)} z^{(t)} + \gamma \sum_{i \in I} \sum_{t=1}^T x_i^{(t)} z^{(t)} \quad (9) \\ = \quad & \sum_{t=1}^T \left(p^{(t)} + \gamma \sum_{i \in I} x_i^{(t)} \right) z^{(t)} \\ \text{such that} \quad & \sum_{t=1}^T z^{(t)} \leq k \end{aligned}$$

In the maximization problem, we jointly maximize the utility coverage of epochs (the first term) and the exposure probability of clients (the second term), both of which are linear with $z^{(t)}$. We put a constraint that the sum of $z^{(t)}$ is at most k to meet the budget constraint. In addition, we consider a hyper-parameter $\gamma \in \mathbb{R}^{\geq 0}$ to prioritize between utility coverage and client exposition. An off-the-shelf ILP solver can output an optimal solution of $\mathbf{z}^{(t)*}$ such that we compute Shapley value in an epoch t if and only if $z^{(t)*} = 1$.

An MILP for Two-sided Fairness. To achieve server-client two-sided fairness, we consider a joint optimization problem: we maximize the utility coverage of epochs where Shapley value is computed and minimize the pair-wise absolute exposure probability of clients in Eq. (10).

$$\begin{aligned} \max_{\mathbf{z}} \quad & \sum_{t=1}^T p^{(t)} z^{(t)} - \gamma \sum_{i, i' \in I} \left| \sum_{t=1}^T (x_i^{(t)} - x_{i'}^{(t)}) z^{(t)} \right| \quad (10) \\ \text{such that} \quad & \sum_{t=1}^T z^{(t)} \leq k \end{aligned}$$

Since the objective function has absolute terms, a direct reduction to an MILP program requires defining $\mathcal{O}(m^2)$ auxiliary real-valued variables and linear constraints.

An alternate approach is to maximize the lower bound of the objective function in Eq. (10), which asks for an ILP solution with respect to the variable $z^{(t)}$.

$$\begin{aligned} & \sum_{t=1}^T p^{(t)} z^{(t)} - \gamma \sum_{i, i' \in I} \left| \sum_{t=1}^T (x_i^{(t)} - x_{i'}^{(t)}) z^{(t)} \right| \\ & \geq \sum_{t=1}^T \left(p^{(t)} - \gamma \sum_{i, i' \in I} |(x_i^{(t)} - x_{i'}^{(t)})| \right) z^{(t)} \quad (11) \end{aligned}$$

Since in the alternate approach we maximize the lower bound of the objective function, the approach is more restrictive than the former approach. Here, we refer to it as **Two-sided Fair LB** scheduling.

Remark 3.8. We explain cases when one-sided versus two-sided fairness is preferred. Let us consider two kinds of non-uniformity: (1) some epochs have more participating clients than other epochs, i.e., non-uniform client-participation across epochs, (2) some clients participate more than others, i.e., non-uniform participation across clients. For (1), one-sided fairness is preferred where epochs with higher participation are selected to increase client exposure, i.e., when more clients participate, Shapley value is computed for a granular contribution assessment. For (2), two-sided fairness is preferred where the pair-wise difference of client exposure is minimized, i.e., one client is not favored more than another client to compute Shapley value. Finally, for the trivial case when all clients participate uniformly across epochs, both one-sided and two-sided fairness may act equivalently, where one can simply rank epochs by corresponding incremental utilities, and choose the top k epochs to compute Shapley value without needing to solve an ILP program.

Lemma 3.9 (Continuing Lemma 3.6). *For $k \leq T$ denoting the maximum number of epochs for Shapley value computation, the runtime complexity is $\mathcal{O}(2^{\frac{m}{\tau}} k + (1 - \frac{1}{\tau}) m k)$, which is $\frac{k}{T}$ fraction of the total runtime complexity without scheduling.*

4 Empirical Performance Evaluation

We conduct an empirical evaluation of FLContrib. In the following, we discuss objectives of the experiments, experimental setup, and experimental results. **Our objectives of experiments** are two-fold¹.

- **Comparative Performance:** How does FLContrib compare with existing methods in computational time and estimation error of client contribution assessment?
- **Ablation Study:** How do different scheduling procedures and associated parameters in FLContrib impact the computational time and estimation error of client contribution assessment?

Experimental Setup. We implement a prototype of FLContrib in Python 3.8. For Shapley value computation, we consider an approximation algorithm based on complementary contribution [37]. For a faster scheduling (Section 3.2), we consider model loss as the epoch-weight $p^{(t)}$, while assessing client contribution to multiple utility functions such as loss and accuracy. In addition, we normalize each term in the objective functions in Eq. (9), (10), (11)

¹ Theoretical proofs, details on experimental setup such as FL training, and additional experimental results including model aggregation using Shapley value, the impact of different approximation algorithms for Shapley value etc. are in the Appendix of the supplementary material [10].

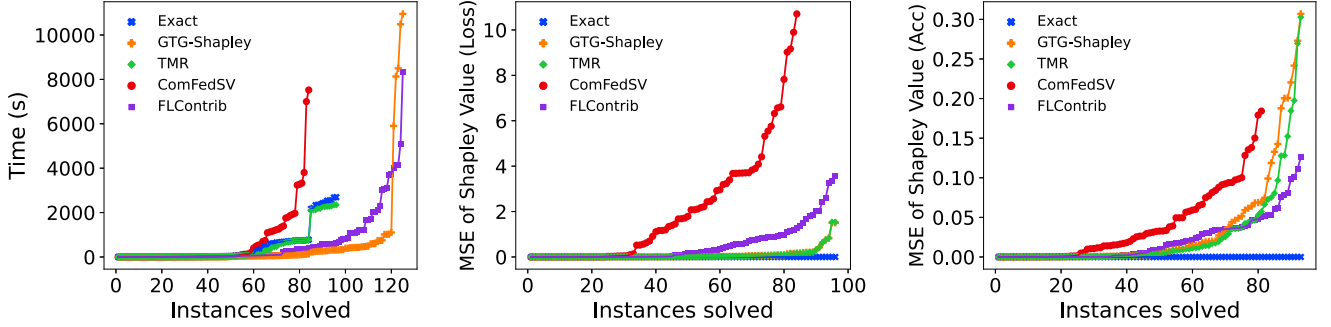


Figure 2: Comparison of different methods for assessing client contribution in FL based on computational time (left), and mean squared error or MSE of Shapley value on model loss (middle) and model accuracy (right). In each cactus plot, a point (x, y) denotes that a method can assess contribution in x many instances within y seconds (left), or within y MSE (middle and right). Therefore, any method reaching the bottom-right part of the plot has the best performance. FLContrib is the most efficient method in computational time, while being consistently accurate in contribution assessment across multiple utility functions: model loss and accuracy.

and consider $\gamma = 1$ for an equal priority. We compare FLContrib with existing methods, namely ComFedSV [6], GTG-Shapley [18], and TMR [32]. Among them, we *adapt* GTG-Shapley and TMR to the *non-participating-client* setting. We consider three datasets: Adult [2], COMPAS [1], and CIFAR10 [15], where the first two datasets are on tabular data and the last one is on image data. We experiment with two types of neural networks: a fully connected MLP model for tabular dataset and a CNN model for image classification. In comparative evaluation, we consider 132 benchmark instances for assessing client contribution by varying the number of clients in $\{4, 8, 16, 32, 64\}$, the number of training epochs in $\{12, 25, 37, 50\}$, and three random seeds. We consider a cut-off time for contribution assessment to 12K seconds. In the following, we discuss our results.

Results of Computational Efficiency: FLContrib is the Most Efficient Method. We compare the computational time of different methods in Figure 2 (left). We consider 132 benchmark instances, where each instance implies an FL training on three datasets with different number of clients, epochs, and repeated runs. Firstly, the exact approach (based on Section 3.1) and TMR solve 96 instances each and time out in rest of the instances. Secondly, ComFedSV, the state-of-the-art for contribution assessment in the non-participating client setting, solves only 84 instances. Finally, FLContrib with two-sided fairness and GTG-Shapley solve 125 instances each, with FLContrib taking a maximum of 8.3K seconds vs. 10.9K seconds by GTG-Shapley. *Therefore, FLContrib is the most computationally efficient method in client contribution assessment than the existing methods.*

Results of Estimation Error: FLContrib is Consistently Accurate across Multiple Utility Functions. We compare the mean squared error (MSE) of Shapley value estimated by different methods and report results in Figure 2 (middle and right). We consider 96 instances where the exact method computes Shapley value within the cut-off time. Firstly, ComFedSV performs poorly by incurring higher MSE than others. Secondly, GTG-Shapley and TMR are tailored for model loss as the utility function and achieve lower MSE than FLContrib (Figure 2 middle) – both approaches incur similar estimation error and their curves overlap. However, when considering model accuracy as utility, they incur higher MSE than FLContrib (Figure 2 right). Importantly, FLContrib yields consistent estimation error across different utility functions. *Therefore, FLContrib achieves well-balanced performance in estimation error than all competitive methods.*

Ablation Study. We analyze the impact of different scheduling procedures in FLContrib while varying the number of clients in Fig-

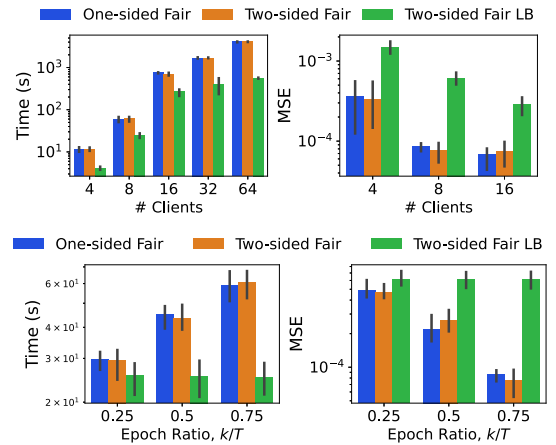


Figure 3: Effect of the number of clients (top) and the maximum epochs parameter in different scheduling procedures (bottom) on computation time and MSE of Shapley values. Figure 3 (top) and maximum epoch for Shapley value computation in Figure 3 (bottom).

In Figure 3 (top), increasing clients results in higher computational time and less estimation error of Shapley value by different scheduling procedures in FLContrib. While it is expected that with more clients, computational time will increase, the decrease in error of Shapley value is less intuitive, and observed across different scheduling procedures and parameter settings. In addition, FLContrib with one-sided and two-sided fairness demonstrate a similar performance in both computational time and MSE. However, FLContrib with two-sided fair LB solves a more restrictive objective function and results in lower computational time but higher MSE than others.

In Figure 3 (bottom), we vary the ratio $\frac{k}{T}$ – a higher ratio denotes computing Shapley value in more epochs. As a result, the computation time increases with an increase in the ratio, resulting in lower MSE – the pattern is observed across different scheduling procedures. Therefore, the parameter k effectively controls the trade-off between computational time and estimation error in FLContrib.

Summary of Results. FLContrib demonstrates an efficient performance in computation time while achieving a consistent estimation error of Shapley value in multiple utility functions. In particular, out of 132 instances, FLContrib solves 125 instances within the cut-off time and yields the lowest estimation error of Shapley value w.r.t.

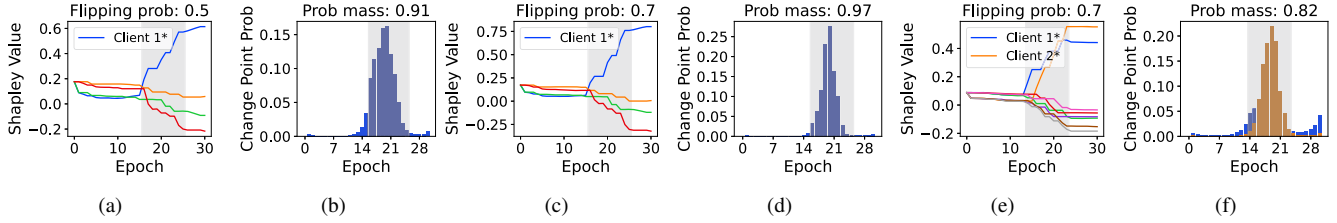


Figure 4: Cumulative Shapley value of individual clients computed on global loss over multiple epochs (Figure 4a, 4c, and 4e). We simulate a poisonous window when a subset of clients (marked with *) acts dishonestly by poisoning their local data and adversely impacting global loss – *historical Shapley value can detect this event*, where Shapley value of dishonest clients increases suddenly and substantially than honest clients. Consequently, we compute the change point detection probability of Shapley values in Figure 4b, 4d, and 4f, respectively, identifying the poisonous window shown in the gray color.

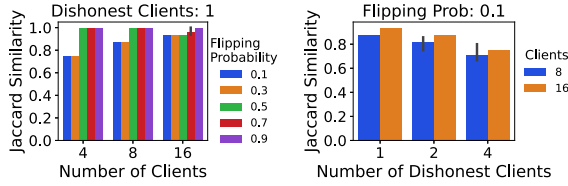


Figure 5: Effectiveness of separating honest clients from dishonest ones based on historical client contribution – a higher Jaccard similarity denotes a higher separation. Separation success increases when the flipping probability or poisonous intention of dishonest clients increases, and when total clients increases while keeping dishonest clients fixed. In contrast, separation success decreases as more clients become dishonest w.r.t. fixed total clients.

model accuracy. In ablation study, different parameters and scheduling procedures of FLContrib precisely control the trade-off between accuracy and efficiency of contribution assessment in FL.

5 Applications of FLContrib

We demonstrate the application of FLContrib in analyzing client intention based on historical client contributions over multiple epochs. Unlike [6, 31, 18, 25], we consider the history of client contributions rather than a single contribution value post training, since historical contribution is more informative to detect sudden behavioral change in individual clients. We consider a controlled experimental setup: each client possesses a uniform data distribution, except a subset of *dishonest clients* who intentionally poison their data within a window of several epochs – it is unknown beforehand when clients switch their intention. For illustration, a dishonest client chooses to flip the data label with a certain flipping probability inside the window.

Objectives. We consider two objectives: (i) Can we detect the window in which a dishonest client poisons its data? (ii) Can we separate honest clients from dishonest ones? In both cases, we rely on the cumulative Shapley value of clients over epochs and answer affirmatively to the two questions.

Detecting Poisonous Window. In Figure 4a, we consider 1 out of 4 clients as dishonest, where the cumulative Shapley value on model loss of the dishonest client diverges from honest clients within the poisonous window (gray colored region). In Figure 4b, we apply a Bayesian inference based change point detection algorithm [7] with the goal of identifying the poisonous window – the probability of change point of Shapley values has significantly higher mass within the window (average probability mass is 0.91), showing the potential

of applying any threshold-based classifier for the purpose of detection. Also, increasing the flipping probability from 0.5 in Figure 4a to 0.7 in Figure 4c, i.e., intensifying the level of dishonestly, the probability mass increases from 0.91 to 0.97 – a greater chance of detection. In another dimension, we increase total clients to 8 and dishonest clients to 2 in Figure 4e, where the change point probability becomes 0.82, which is still evident of poisonous intention. *All these evidences support our claim that historical client contribution via Shapley value has the potential of being an effective identification of poisonous client intention in FL.*

Separating Honest Clients. Historical contribution of clients capture their intention during FL training. As such, to separate honest clients from dishonest ones, we apply K-means time-series clustering algorithm [27] on the cumulative Shapley value of clients. Our hypothesis is that majority of the honest clients should be in one cluster due to their uniform data distribution, while placing dishonest clients in separate clusters. Subsequently, we measure the Jaccard similarity index [11] between the set of honest clients and the set of all clients that are in a cluster containing at least one honest client – higher similarity (value closer to 1) implies that all honest clients are plausibly in one cluster and hence a higher separation between honest and dishonest clients.

In Figure 5 (left), while increasing the flipping probability, Jaccard similarity increases, eventually reaching 1. Also, by keeping dishonest clients fixed at 1, the similarity index increases with increasing clients in Figure 5 (left). Also, while increasing dishonest clients in Figure 5 (right), the similarity decreases, implying hardness of separation with more dishonest clients. *Therefore, Shapley values demonstrate the potential in separating out honest clients.*

6 Conclusion

We study the assessment of client contribution in centralized single-server federated learning using Shapley values. We propose FLContrib to assess client contribution when a subset of clients participate in each training epoch. We discuss a scheduling procedure satisfying two-sided fairness between server and clients for a faster contribution assessment. Empirically, FLContrib is the most efficient method while being consistently accurate in estimating contribution across multiple utility functions. In addition, historic client contributions enable us to analyzing dishonest clients. In future, we extend FLContrib to assess client contribution in a decentralized federated setting. Our framework opens up further research on client selection and federated aggregation based on historic client contributions.

Acknowledgements

This Research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Project (IAF-ICP) (Award No: I2301E0020) and Japan-Singapore Joint Call: Japan Science and Technology Agency (JST) and Agency for Science, Technology and Research (A*STAR) 2024 (Award No: R24I6IR141), administered by A*STAR. D. Basu acknowledges the ANR JCJC project REPUBLIC (ANR-22-CE23-0003-01), and the PEPR project FOUNDRY (ANR23-PEIA-0003) for partially supporting this work.

References

- [1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias risk assessments in criminal sentencing. *ProPublica*, May, 23, 2016.
- [2] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: <https://doi.org/10.24432/C5XW20>.
- [3] J. Castro Cantalejo, D. Gómez González, and J. A. Tejada Cazorla. Polynomial calculation of the shapley value based on sampling. 2009.
- [4] E. Cyffers, A. Bellet, and D. Basu. From noisy fixed-point iterations to private admm for centralized and federated learning. *arXiv preprint arXiv:2302.12559*, 2023.
- [5] V. Do, S. Corbett-Davies, J. Atif, and N. Usunier. Two-sided fairness in rankings via lorenz dominance. *Advances in Neural Information Processing Systems*, 34:8596–8608, 2021.
- [6] Z. Fan, H. Fang, Z. Zhou, J. Pei, M. P. Friedlander, C. Liu, and Y. Zhang. Improving fairness for data valuation in horizontal federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2440–2453. IEEE, 2022.
- [7] P. Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and computing*, 16:203–213, 2006.
- [8] Y. Fraboni, R. Vidal, and M. Lorenzi. Free-rider attacks on model aggregation in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1846–1854. PMLR, 2021.
- [9] A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.
- [10] B. Ghosh, D. Basu, F. Huazhu, W. Yuan, R. Kanagavelu, J. J. Peng, L. Yong, G. S. M. Rick, and W. Qingsong. History-aware and dynamic client contribution in federated learning. *arXiv preprint arXiv:2403.07151*, 2025. URL <https://arxiv.org/pdf/2403.07151>.
- [11] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.
- [12] 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. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.
- [13] 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*, 6(6):10700–10714, 2019.
- [14] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799, 2021.
- [15] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- [16] 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*, 7(10):9575–9588, 2020.
- [17] X. Lin, X. Xu, S.-K. Ng, C.-S. Foo, and B. K. H. Low. Fair yet asymptotically equal collaborative learning. *arXiv preprint arXiv:2306.05764*, 2023.
- [18] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–21, 2022.
- [19] L. Lyu, X. Xu, Q. Wang, and H. Yu. Collaborative fairness in federated learning. *Federated Learning: Privacy and Incentive*, pages 189–204, 2020.
- [20] 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*, pages 1273–1282. PMLR, 2017.
- [21] T. Nishio, R. Shinkuma, and N. B. Mandayam. Estimation of individual device contributions for incentivizing federated learning. In *2020 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2020.
- [22] A. Papadaki, N. Martinez, M. Bertran, G. Sapiro, and M. Rodrigues. Minimax demographic group fairness in federated learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 142–159, 2022.
- [23] A. Richardson, A. Filos-Ratsikas, and B. Faltings. Rewarding high-quality data via influence functions. *arXiv preprint arXiv:1908.11598*, 2019.
- [24] L. S. Shapley et al. A value for n-person games. 1953.
- [25] T. Song, Y. Tong, and S. Wei. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2577–2586. IEEE, 2019.
- [26] S. Tavana, A. Schliep, and D. Basu. Federated learning of oligonucleotide drug molecule thermodynamics with differentially private admm-based svm. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 459–467. Springer, 2021.
- [27] R. Tavenard, J. Faozi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020. URL <http://jmlr.org/papers/v21/20-091.html>.
- [28] T. van Campen, H. Hamers, B. Husslage, and R. Lindelauf. A new approximation method for the shapley value applied to the wtc 9/11 terrorist attack. *Social Network Analysis and Mining*, 8:1–12, 2018.
- [29] P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed.*, Cham: Springer International Publishing, 10(3152676):10–5555, 2017.
- [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)*, pages 2597–2604. IEEE, 2019.
- [31] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song. A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive*, pages 153–167, 2020.
- [32] S. Wei, Y. Tong, Z. Zhou, and T. Song. Efficient and fair data valuation for horizontal federated learning. *Federated Learning: Privacy and Incentive*, pages 139–152, 2020.
- [33] X. Yin, Y. Zhu, and J. Hu. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021.
- [34] 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*, pages 393–399, 2020.
- [35] S. Zezulka and K. Genin. Performativity and prospective fairness. *arXiv preprint arXiv:2310.08349*, 2023.
- [36] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.
- [37] J. Zhang, Q. Sun, J. Liu, L. Xiong, J. Pei, and K. Ren. Efficient sampling approaches to shapley value approximation. *Proceedings of the ACM on Management of Data*, 1(1):1–24, 2023.
- [38] Z. Zhu, J. Shu, X. Zou, and X. Jia. Advanced free-rider attacks in federated learning. In *the 1st NeurIPS Workshop on New Frontiers in Federated Learning Privacy, Fairness, Robustness, Personalization and Data Ownership*, 2021.