Contributions Estimation in Federated Learning: A Comprehensive Experimental Evaluation

Yiwei Chen
Tsinghua University
chen-yw20@mails.tsinghua.edu.cn

Kaiyu Li
Tsinghua University
ky-li18@mails.tsinghua.org.cn

Guoliang Li
Tsinghua University, Zhongguancun Lab
liguoliang@tsinghua.edu.cn

Yong Wang
Tsinghua University
wangy18@mails.tsinghua.edu.cn

ABSTRACT

Federated Learning (FL) provides a privacy-preserving and decentralized approach to collaborative machine learning for multiple FL clients. The contribution estimation mechanism in FL is extensively studied within the database community, which aims to compute fair and reasonable contribution scores as incentives to motivate FL clients. However, designing such methods involves challenges in three aspects: effectiveness, robustness, and efficiency. Firstly, contribution estimation methods should utilize the data utility information of various client coalitions rather than that of individual clients to ensure effectiveness. Secondly, we should beware of adverse clients who may exploit tactics like data replication or label flipping. Thirdly, estimating contribution in FL can be time-consuming due to enumerating various client coalitions.

Despite numerous proposed methods to address these challenges, each possesses distinct advantages and limitations based on specific settings. However, existing methods have yet to be thoroughly evaluated and compared in the same experimental framework. Therefore, a unified and comprehensive evaluation framework is necessary to compare these methods under the same experimental settings. This paper conducts an extensive survey of contribution estimation methods in FL and introduces a comprehensive framework to evaluate their effectiveness, robustness, and efficiency. Through empirical results, we present extensive observations, valuable discoveries, and an adaptable testing framework that can facilitate future research in designing and evaluating contribution estimation methods in FL.

PVLDB Reference Format:
Yiwei Chen, Kaiyu Li, Guoliang Li, and Yong Wang. Contributions Estimation in Federated Learning: A Comprehensive Experimental Evaluation. PVLDB, 17(8): 2077 - 2090, 2024. doi:10.14778/3659437.3659459
PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/veevang/flce.

Kaiyu Li and Guoliang Li are the corresponding authors.
This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 17, No. 8 ISSN 2150-8097.
doi:10.14778/3659437.3659459

1 INTRODUCTION

Federated Learning (FL) provides a collaborative machine learning paradigm with low communication overhead while preserving data privacy [30, 36, 41, 47, 71]. In FL, clients (i.e., participants) train machine learning models locally and send model updates to a central coordinator. The coordinator aggregates these updates and shares improved global model parameters with clients, allowing them to refine their local models without sharing raw data.

However, clients may exhibit reluctance to share their data in the absence of incentives or if they perceive the revenue allocation as unfair [14, 16]. For instance, if an enterprise's transaction data from January and the first quarter are considered equally contributive to the FL task, the enterprise might be disinclined to share its transaction data from February to March, as it would not yield any additional benefits. However, in many scenarios, the transaction data from February to March could potentially lead to further improvement in the global model. Hence, the study of **F**ederated **L**earning **C**ontribution **E**stimation (FLCE) emerged to develop fair and reasonable methods for incentivizing clients' participation in FL and ensure the usability and sustainability of the FL ecosystem [30].

Challenges. It is crucial to consider the following three challenges when designing such contribution estimation methods.

- *Effectiveness.* Effectiveness ensures that a client's estimated contribution aligns with its importance in the grand coalition. Assessing contributions solely based on individual data utility might not accurately represent a client's true significance in FL, as it fails to consider a client's marginal contribution when combined with other clients. For instance, a client with a large amount of data on general cases may have high utility based on quantity alone, while a client with scarce but complementary data may be assigned a lower utility. However, even a small number of complementary data records can yield a significant performance increase to the global model when combined with sufficient general data [69].

- *Robustness.* In FL, the presence of strategic clients and malicious clients is a potential concern [23, 38, 40]. Certain clients may seek to gain advantages through manipulated data. For instance, they may replicate their data to inflate their rewards at approximately zero marginal cost. Alternatively, some clients might aim to compromise the performance of the global model by manipulating the values of their data, e.g., using the trick of flipping labels to poison the performance of the global model.

- *Efficiency.* As mentioned above, calculating a client's contribution may involve enumerating different client coalitions and measuring

their respective data utilities [19, 78], which may result in prohibitively high computational costs when the number of clients is large. For example, when utilizing the original ShapleyValue method [19], it is necessary to enumerate all the coalitions of n clients and train a model for each of these client coalitions. The method, therefore, has a time complexity of $O(2^n)$.

Multiple techniques are proposed to solve the above concerns. For the first challenge, methods based on ShapleyValue [19] and LeastCore [78] are proposed, utilizing the data utility information of all the 2^n coalitions. For the second challenge, algorithms have been proposed to defend against malicious users. For example, the RobustVolume metric is proposed, which is immune to data replication [77]. For the third challenge, several optimization techniques have been proposed to reduce the computational complexity of the FLCE process, e.g., sampling [42, 68] and gradients reusing [70].

Motivations. Despite numerous methods proposed to address these challenges, each has distinctive advantages and limitations within specific settings. Certain issues may hinder researchers from attaining a comprehensive, unbiased, and systematic understanding of these methods. Firstly, existing contribution estimation methods have not been evaluated within the same experimental settings, leading to unfair and incomplete comparison results. Secondly, previous papers lack comprehensive evaluation indicators. None of them consider evaluating the effectiveness, robustness, and efficiency of FLCE methods simultaneously. Thirdly, the absence of a standardized and inclusive testing framework hinders the investigation and testing of new FLCE methods in practical implementations. For example, [27] demonstrated that ShapleyValue can outperform the LeaveOneOut regarding scalability and usability but did not assess the robustness of these methods against various adverse behaviors. [23] only studied methods that can detect label flipping while ignoring other adverse behaviors, such as data replication.

Contributions. In this paper, we make the following contributions:
- *An in-depth survey.* Through an extensive survey, we examined a wide range of FLCE methods, providing valuable insights for FLCE and potentially inspiring new methods. We break down the FLCE problem into three progressive sub-problems: data utility metrics, contribution estimation schemes, and optimization techniques, enabling an in-depth examination of each sub-problem independently.
- *A comprehensive evaluation.* We extensively evaluated state-of-the-art FLCE methods, encompassing various datasets, data distributions, federation settings, and adverse behaviors, and observed their effectiveness, robustness, and efficiency. To the best of our knowledge, we are the first to comprehensively evaluate all types of FLCE methods within a unified experimental framework.
- *An extensive set of findings.* Based on experimental observations in various FL scenarios, we have gained deep insights and concluded the advantages and limitations of different data utility metrics, contribution estimation schemes, and optimization techniques into summarized findings. These valuable findings may contribute to the development of new FLCE methods.
- *An extensible testing framework.* Our team developed a flexible testing framework capable of accommodating multiple implemented methods. This framework is designed to support both existing and newly proposed contribution estimation methods, serving as a potential benchmark for evaluating performance in this field.

The rest of this paper is organized as follows. We give the formal definition of FLCE in Section 2, review the data utility metrics in Section 3, the FLCE schemes in Section 4, and the state-of-the-art optimization techniques in Section 5. We conduct comprehensive experiments, discuss the empirical results in Section 6, provide prospects of FLCE in Section 7, and conclude our study in Section 8.

2 PRELIMINARIES

2.1 Federated Learning

Federated Learning (FL) is a distributed machine learning framework that enables training models on decentralized data sources without transferring the raw data to a central server [47]. Each round of FL training consists of three phases. First, each client independently trains the model using its local data, ensuring data privacy by only transmitting local updates such as model parameters or gradients to the server rather than sharing its raw data. Second, the central server collects and aggregates the clients' updates, incorporating them with the current model to generate an enhanced global model [47]. Third, the central server distributes the updated parameters of the global model to the clients, allowing each client to generate their respective enhanced local models [30]. This iterative process of local training, aggregation, and broadcast continues for multiple global rounds until specific criteria are met, such as achieving a predetermined model performance or reaching a threshold number of iterations.

Note. This paper focuses on the survey and evaluation of HFL [79], which involves clients with distinct samples in the same feature space. The exploration of VFL, where clients possess different features of the same samples, is deferred for future research [14, 21, 60]. Furthermore, since FL concentrates more on supervised learning rather than unsupervised learning, we restrict our work to the tasks of classification and regression in the survey parts of this paper.

2.2 Federated Learning Contribution Estimation

To motivate data holders to engage in FL, it is essential to estimate their contributions and offer incentives accordingly. Specifically, when considering a grand coalition consisting of a set of clients $\mathcal{N} = \{1, 2, \ldots, n\}$, an FLCE method calculates a vector $\Phi = (\phi_1, \phi_2, \ldots, \phi_n)$, where ϕ_i denotes the contribution of client i. An ideal FLCE method is expected to have the following properties:

(1) **Effectiveness.** The estimated contribution of a client should align with its significance in the cooperation within the grand coalition. However, in practice, obtaining the ground truth of clients' contributions is not feasible. Therefore, in this paper, we utilize a commonly employed client removal indicator [61] (see Section 6), which solely relies on the estimated contributions to evaluate the effectiveness of FLCE methods.

(2) **Robustness.** In FL, some clients may attempt to gain advantages or reduce the performance of the global model intentionally by conducting strategic or malicious behaviors. The FLCE robustness of a client can be determined by examining the relative contribution change $\frac{\phi_{\text{adv}} - \phi_{\text{orig}}}{|\phi_{\text{orig}}|}$. ϕ_{orig} represents the **orig**inal contribution of a client, which is estimated using a certain FLCE method in a grand coalition without adverse data; whereas ϕ_{adv} refers to the

contribution of the same client estimated using the same FLCE method in the same grand coalition, with the only difference being that this client has introduced its **adv**erse data. An FLCE method is deemed more robust if it shows negative changes instead of positive changes resulting from adverse behaviors.

(3) **Efficiency.** Considering the potential computationally demanding nature of contribution estimation, especially when dealing with a large number of clients [19, 78], it is crucial to assess the running time as a practical evaluation criterion for FLCE methods. In this paper, we focus on the execution time of FLCE algorithms, ignoring the training time of FL models.

Note. Many FLCE methods share similarities with Data Valuation and Pricing (DVP) algorithms [11, 53, 83]. However, FLCE methods are tailored to practical contexts that involve data privacy, client-server communication costs, and the integration of FLCE with the model training process, while DVP typically focuses on data value computation from a theoretical perspective.

2.3 Research Question Breakdown

This paper makes a significant contribution by breaking down the FLCE problem into three progressive sub-problems, allowing for a more focused and in-depth examination of each sub-problem independently. Firstly, given any coalition $\mathcal{S} \subseteq \mathcal{N}$, we aim to utilize a utility metric $v(\mathcal{S})$ to accurately capture the data's usefulness for the FL task, effectively reflecting the utility of $\mathcal{S}$ itself. Secondly, the contribution ϕ_i is employed to quantify the extent to which client i contributes to the FL task, distinct from $v(\{i\})$. Finally, to expedite the computation of ϕ_i, optimization techniques can be employed to improve the computational efficiency of FLCE.

(1) **Data Utility Metrics.** The data utility of a coalition $\mathcal{S} \subseteq \mathcal{N}$ is measured using a data utility metric $v(\cdot)$. Formally, we define the function $v(\cdot) : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ with $v(\emptyset) = 0$, where $2^{\mathcal{N}}$ represents the power set of $\mathcal{N}$ [6]. For example, if the utility metric corresponds to the Accuracy achieved on a test set, then $v(\mathcal{S})$ denotes the test accuracy of the model trained on data from coalition $\mathcal{S}$. Further details regarding the data utility metrics can be found in Section 3.

(2) **Contribution Estimation Schemes.** Based on data utility metric $v(\cdot)$, a contribution estimation scheme ϕ_i can be devised to assess the significance that client i contributes to the FL task. It is essential to clarify that ϕ_i is distinct from $v(\{i\})$, and the data utility of a coalition does not generally equate to the sum of the contributions of its clients, i.e., $v(\mathcal{S}) \neq \sum_{i \in \mathcal{S}} \phi_i$. Section 4 introduces and compares various contribution estimation schemes.

(3) **Optimization Techniques.** In many FLCE methods, a significant amount of client coalition enumeration and retraining is required. As a result, there is a need for optimization techniques that strike a balance between efficiency, effectiveness, and robustness. Building upon the FLCE schemes in Section 4, FLCE optimization techniques (e.g., coalitions sampling, gradient reusing and computation truncation) are discussed in Section 5.

3 DATA UTILITY METRICS

Data utility metrics can be divided into two types: test-set-dependent metrics (when the test set is available), and test-set-independent metrics (when the test set is not accessible).

3.1 Test-set-dependent Metrics

The data utility of a non-empty coalition $\mathcal{S} \subseteq \mathcal{N}$ can be assessed by training a model using $\mathcal{D}_{\mathcal{S}}$, the data from $\mathcal{S}$, and measuring its performance (ModelPerformance) on the test set $\mathcal{D}_t$ held by the central server. Accuracy and R^2 are two standardized test-set-dependent metrics [7], which are widely adopted to reflect ModelPerformance. They can be formally represented as:

$$v(\mathcal{S}) = \begin{cases} \dfrac{\sum_{(x,y)\in\mathcal{D}_t} \mathbb{I}[y = \hat{y}]}{|\mathcal{D}_t|} & \text{(Accuracy)} \quad \text{classification task} \\ 1 - \dfrac{\sum_{(x,y)\in\mathcal{D}_t}(y-\hat{y})^2}{\sum_{(x,y)\in\mathcal{D}_t}(y-\bar{y})^2} & (R^2) \quad \text{regression task} \end{cases}$$

where $|\cdot|$ denotes the cardinality of a set, and (x, y) represents a testing pair in the test set $\mathcal{D}_t$, where y is the true label of data record x. The predicted label of input x obtained from the model trained on $\mathcal{S}$ is denoted as $\hat{y}$. Furthermore, $\bar{y}$ stands for the average of y, given by $\bar{y} = \frac{\sum_{(x,y)\in\mathcal{D}_t} y}{|\mathcal{D}_t|}$. $\mathbb{I}[y = \hat{y}]$ equals 1 if $y = \hat{y}$ and 0 otherwise. A higher Accuracy or R^2 value of the model represents a higher data utility, and it reaches its maximum value when $v(\mathcal{S}) = 1$.

The selected metric should accurately capture the data utility based on the specific task at hand. The mentioned metrics can be customized and tailored to align with particular applications and meet specific requirements. For example, in classification tasks, Macro F1 Score [8] is preferable over Accuracy when dealing with datasets that have extremely imbalanced classes. This is because Macro F1 Score calculates the F1 score independently for each class and takes the unweighted average of them, providing a more robust evaluation [24]. In regression tasks, R^2 can be replaced with other metrics such as root mean square error (RMSE) or mean absolute error (MAE) to suit the needs of the analysis. More insights and different perspectives on metric selection and normalization can be found in [9, 73]. Besides, due to the limitation of space, uncommon data utility metrics designed for specific scenarios rather than general FLCE are omitted [10, 25].

3.2 Test-set-independent Metrics

In practical scenarios, obtaining a comprehensive test set can be challenging due to limited knowledge of the application context. Consequently, there is a need to develop data utility metrics that are not reliant on a specific test set.

3.2.1 Statistical Metrics. In an ideal scenario, the utility of a client's dataset would ideally increase with the number of records it contains (DataQuantity) [15, 81], assuming that the datasets from various clients are independent and identically distributed (i.i.d.). In other words, greater quantity would indicate higher data utility. However, due to the non-i.i.d. distribution of the data across different clients in most practical scenarios, it is inappropriate to make such an assumption [82]. Besides, existing FLCE schemes do not adopt general statistical metrics like mean and standard deviation values due to their limited ability to accurately reflect the true data value in most cases. Moreover, practical challenges arise from the lack of knowledge about the specific application context, making it impractical to design customized statistical metrics for FLCE. Additionally, since each client only has a limited number of records

in cross-device FL [22], directly uploading statistical metric values to the central server can pose privacy risks to individuals [28].

In a more general context, the diversity of data can serve as a reflection of its utility. The greater the diversity of the data, the higher its utility [20]. In a FL task with clients holding datasets with d data features, each sample in the coalition $\mathcal{S}$ can be represented as a vector in a d-dimensional Euclidean space. The degree to which these vectors are spread out in the feature space indicates the diversity. Formally, the data diversity of $\mathcal{S}$ can be denoted as $v(\mathcal{S}) = \sqrt{\det\left(X_{\mathcal{S}}^{\intercal} X_{\mathcal{S}}\right)}$ based on Gram determinant (Volume) as is proposed in [77], where $X_{\mathcal{S}} \in \mathbb{R}^{m \times d}$ is a matrix composed of the feature values of the data in the coalition $\mathcal{S}$, with m representing the number of records in the dataset of coalition $\mathcal{S}$. This metric is considered to achieve good performance when there is no inclusion of adverse behaviors, and when the data distributions exhibit uniform or normal patterns. However, a significant drawback of this metric is that clients can artificially inflate their data utilities by duplicating their own data.

To address this issue, RobustVolume is proposed [77] as an alternative. This metric involves discretizing the feature space into a collection of data cubes, compressing each cube into a single vector, and constructing a data matrix $\widetilde{X_{\mathcal{S}}}$ that approximates the original feature value matrix $X_{\mathcal{S}}$ using these compressed vectors. The utility function can be formally denoted as:

$$v(\mathcal{S}) = \sqrt{\det\left(\widetilde{X_{\mathcal{S}}}^{\intercal} \widetilde{X_{\mathcal{S}}}\right)} \times \prod_{j \in \Phi} \rho_j \tag{1}$$

where $\rho_j = \sum_{p=0}^{\varphi_j} \alpha^p$, $\alpha \in [0, 1]$, and Φ represents the collection of all cubes. The coefficient ρ_j is determined by the number of samples within cube j, denoted as φ_j, and can reflect the importance of the cube. Since the summation of a geometric progression is bounded when the common ratio α is within $[0, 1)$, the influence of data replication will be restricted.

3.2.2 Model Parameter Metrics. In FL, the global model is expected to be the optimal model compared to models trained on any other coalitions, assuming positive contributions from each client and the absence of adverse behavior. Therefore, we can assess the similarity between the global model trained on the grand coalition and the model trained on a specific coalition $\mathcal{S}$ as $v(\mathcal{S})$. A straightforward approach is to gauge data utility by computing the inverse of the *L2* distance between the parameters of the two models [5]. However, this method may not consistently capture the actual similarity between models in each training round due to the inherent randomness in the training process, especially when specific techniques like dropout are applied [65]. The following two promising alternatives are often preferred as parameter-based data utility metrics.

- Gradient Similarity. From the perspective of optimization, the objective of FL is to find an approximate solution to optimize the cost function by iteratively updating the model's parameters using gradients. The gradient vectors in each iteration reflect how the model changes in this round of training. Therefore, a higher similarity between the gradient update of the model trained on coalition $\mathcal{S}$ and that of the global model trained on the grand coalition $\mathcal{N}$ [76] indicates a greater data utility for $\mathcal{S}$. As a result, a utility metric, which is referred to as CosineGradient, is defined as:

$$v(\mathcal{S}) = \beta^{T-1} \cos\left(\boldsymbol{u}_{\mathcal{S}}^{1}, \boldsymbol{u}_{\mathcal{N}}^{1}\right) + (1-\beta) \sum_{t=2}^{T} \beta^{T-t} \cos\left(\boldsymbol{u}_{\mathcal{S}}^{t}, \boldsymbol{u}_{\mathcal{N}}^{t}\right) \tag{2}$$

where, the gradient of the model trained on coalition $\mathcal{S}$ and that on the grand coalition $\mathcal{N}$ in the t^{th} round of training are represented as $\boldsymbol{u}_{\mathcal{S}}^{t}$ and $\boldsymbol{u}_{\mathcal{N}}^{t}$, respectively. T represents the total number of training rounds and β is a predetermined weight.

Nevertheless, this metric might not accurately capture the true utility of data in certain rounds due to the stochastic nature of gradient descent. There are instances where this metric could even yield negative values for high-contribution coalitions, especially in cases involving non-convex loss functions [76].

- Model Uncertainty. Under the assumption that all clients positively contribute to the grand coalition, the utility of a coalition can also be measured based on how much it reduces the uncertainty of the global model parameters upon its introduction. This utility can be quantified using the entropy changes (InformationGain) in the model parameters [62], represented formally as $v(\mathcal{S}) = \mathbb{H}(\boldsymbol{\theta}) - \mathbb{H}(\boldsymbol{\theta} \mid \mathcal{D}_{\mathcal{S}})$, where θ represents the vector of model parameters, $\mathbb{H}(\theta)$ signifies the prior information entropy (i.e., uncertainty) of the model parameters, and $\mathbb{H}(\theta \mid \mathcal{D}_{\mathcal{S}})$ stands for the posterior information entropy of the model parameters after trained with the data $\mathcal{D}_{\mathcal{S}}$. The difference between these two information entropies represents the uncertainty reduction or information gain on the model parameters θ resulting from the introduction of coalition $\mathcal{S}$.

In contrast to gradient similarity, this metric offers several advantages. It ensures stability by excluding randomness and doesn't require the optimal global model as the reference. Furthermore, it relaxes the requirement that all clients must contribute positively to some extent. Even with a few malicious clients, it remains effective in measuring data utility, as the presence of poisoned data from malicious clients may increase the uncertainty of model parameters within their coalitions. However, this metric relies on the distribution of model parameters, which typically requires the global model to contain a Bayesian layer [4], and may additionally introduce challenges in reaching a consensus on specific learning settings among different clients [62, 72].

3.3 Insights of Choosing Data Utility Metrics

We can summarize the following conclusions regarding the selection of utility metrics. First, when a test set is available, ModelPerformance accurately reflects the data utility. Second, when a test set is not available, but the datasets from different clients are i.i.d. without strategic clients, DataQuantity can be used to measure utility. Third, in cases where no test set is available, and the data distribution is not i.i.d. without strategic clients and the data distribution is simple, the data distribution diversity (Volume) can be employed. In the presence of data replication, RobustVolume is a better alternative to Volume. Fourth, when no test set is available, and the data distribution is not i.i.d., but the data distribution is simple, we can employ model uncertainty (InformationGain) and gradient similarity (CosineGradient) to measure data utility even in the presence of adverse behaviors. Finally, when there are a moderate number of malicious clients or the data distribution is too complex, no existing metrics can reflect the data utility well.

4 CONTRIBUTION ESTIMATION SCHEMES

This section presents four schemes for estimating each client's contribution to the FL task (denoted as ϕ_i), given any data utility metric $v(\cdot)$ discussed in Section 3.

4.1 Individual

The Individual scheme determines the contribution of a client in FL by its data utility, i.e., $\phi_i = v(\{i\})$. This approach is commonly employed in practice due to its simplicity and efficiency [10, 31]. However, employing the Individual scheme for estimating a client's contribution may neglect the significance of other clients' data. For instance, a client with a limited amount of data that complements the data of other clients can significantly improve the performance of the global model. Unfortunately, since this client lacks the typical data found in common test scenarios that most clients possess, it may be assigned a low contribution.

This absence of constraints makes it challenging to quantify clients' contributions to model performance [57], creating unfair revenue allocation, as each client's contribution may not accurately represent their proportionate share of the grand coalition's utility, despite normalizing contributions values based on their sum [84].

Note. Another idea in FLCE is to utilize the local datasets and require clients to self-report their utility results [31, 52] as their contributions. However, in practice, the local datasets of different clients come from heterogeneous sources, rendering the assumption of i.i.d. invalid, and the self-reported utility may be inaccurate.

4.2 LeaveOneOut

The LeaveOneOut scheme, commonly utilized in machine learning for cross-validation [32], serves as the basis for determining a client's contribution by considering the marginal data utility loss when that client is excluded [27, 34, 55]. Formally, the contribution of a specific client can be expressed as $\phi_i = v(\mathcal{N}) - v(\mathcal{N}\backslash\{i\})$.

When estimating the contribution of a client i, the LeaveOneOut scheme assumes that it is the last addition to the grand coalition $\mathcal{N}$, which may raise fairness issues. For example, consider a client coalition $\mathcal{S}$ where each client i possesses similar but high-utility data. From the perspective of LeaveOneOut, removing any client $i \in \mathcal{S}$ would not result in a significant reduction in data utility, leading to the estimation of the clients in $\mathcal{S}$ as low-contribution clients. However, from an alternative viewpoint, removing all clients $i \in \mathcal{S}$ simultaneously may cause a substantial reduction in data utility. Additionally, similar to the Individual scheme, the sum of all clients' contributions is not restricted to a specific value, posing challenges to revenue allocation.

4.3 ShapleyValue

ShapleyValue originated from cooperative game theory [59] and has since gained recognition as a promising scheme for estimating clients' contributions [19, 68]. The ShapleyValue of client i can be interpreted as the expected value of the marginal utility increase that results from the inclusion of client i in all permutations, denoted as $\phi_i = \mathbb{E}_{\pi\in\Pi}\left[v(\mathcal{S}_\pi^i \cup \{i\}) - v(\mathcal{S}_\pi^i)\right]$, where Π is the set of all permutations of clients, $\mathcal{S}_\pi^i$ is the coalition of clients that precede client i in the permutation π. However, in practice, a simplified formula for calculating ShapleyValue can be expressed as follows:

$$\phi_i = \frac{1}{n}\sum_{\mathcal{S}\subseteq\mathcal{N}\backslash\{i\}} \frac{1}{\binom{n-1}{|\mathcal{S}|}}[v(\mathcal{S}\cup\{i\}) - v(\mathcal{S})] \tag{3}$$

where $|\cdot|$ denotes the cardinality of a set, and the term $\frac{1}{n}$ represents the probability that client i appears after coalition $\mathcal{S}$ and before the remaining clients in a permutation of $\mathcal{N}$. Equation 3 simplifies the estimation of client i's contribution by considering all coalitions that do not include client i, and computing the weighted average of the marginal increase in utility resulting from introducing client i. The ShapleyValue scheme offers greater fairness and reasonableness compared to Individual and LeaveOneOut as it considers all possible permutations of clients, satisfying a set of properties:

(1) **Group Rationality.** The utility of the grand coalition equals the sum of all clients' contributions [56]. Formally, $v(\mathcal{N}) = \sum_{k=1}^{n}\phi_k$.

(2) **Symmetry.** If the marginal contributions of any two clients are the same across all possible client coalitions, their contributions estimated by ShapleyValue will also be the same [56, 59]. Formally, $\phi_i = \phi_j$, if $v(\mathcal{S}\cup\{i\}) = v(\mathcal{S}\cup\{j\}), \forall\mathcal{S}\subseteq\mathcal{N}\backslash\{i,j\}$.

(3) **Zero Element.** When the marginal contribution of a client is consistently zero across all possible client coalitions, this signifies that the client's contribution is zero; that is, the client is a null player [56]. Formally, $\phi_i = 0$, if $v(\mathcal{S}\cup\{i\}) = v(\mathcal{S}), \forall\mathcal{S}\subseteq\mathcal{N}\backslash\{i\}$.

(4) **Additivity.** If the utility function is the sum of multiple metrics, the final contribution is obtained by summing all the partial contributions based on each metric separately. Formally, $\phi_i[u+v] = \phi_i[u] + \phi_i[v]$, where u and v represent two different utility metrics [56, 59].

The symmetry and zero element properties play a crucial role in ensuring fairness during contribution estimation [26]. The symmetry property guarantees that each client's contribution is evaluated without bias or favoritism, irrespective of the order in which clients are considered. The zero-element property promises that if a client is not helpful to any coalition, it will not be considered as a contributor. In addition, the group rationality property ensures that the contribution represents the allocation of the grand coalition's data utility [59]. Furthermore, the additivity property enables the efficient introduction of new data utility metrics, as new contributions can be added to the existing ones to obtain the final value, eliminating the need to recalculate data utilities of original metrics.

However, a limitation of the ShapleyValue scheme is its requirement to enumerate all possible coalitions and calculate their data utilities, which becomes impractical when dealing with a large number of clients in real-world FL systems. In Section 5, we will explore optimization techniques to reduce the computational cost of the ShapleyValue scheme.

4.4 LeastCore

Another well-known cooperative gaming solution concept for FLCE is the LeastCore scheme [54, 78], which is based on the core theory that the sum of all clients' contributions within a coalition should be equal to or greater than the utility of the coalition [58]. However, achieving this exact goal may not be feasible in practice. Therefore, in the context of FLCE, our objective is to minimize the maximum

Table 1: Comparison of contribution estimation schemes.

Scheme	Individual [45]	LeaveOneOut [27]	ShapleyValue [69]	LeastCore [78]
Group Rationality	×	×	✓	✓
Symmetry	✓	✓	✓	×
Zero Element	✓	✓	✓	✓
Additivity	✓	✓	✓	×
Stability	×	×	×	✓
Complexity	$O(n)$	$O(n)$	$O(2^n)$	$O(2^n)$
Fairness	×	×	individual fairness	group fairness

deficit, which represents the gap between the summation of all contributions in a coalition and the data utility of the coalition. LeastCore can be formulated as:

$$\min\ e \quad \text{s.t.} \begin{cases} \sum_{i\in\mathcal{N}} \phi_i = v(\mathcal{N}) \\ \sum_{i\in\mathcal{S}} \phi_i + e \geq v(\mathcal{S}) \quad \forall \mathcal{S} \subseteq \mathcal{N} \end{cases} \tag{4}$$

The LeastCore scheme involves a total of 2^n linear inequality and one equality constraint, making it a linear programming problem. In practical implementations, we can compute a solution of LeastCore using the simplex or the interior-point method [74]. Consistent with ShapleyValue, LeastCore also satisfies the group rationality and zero element property. Besides, the LeastCore scheme satisfies the stability property, which ensures that the contribution sum of any coalition is maximized compared to its overall utility. For instance, in specific scenarios where $e = 0$, each coalition is guaranteed a reward no smaller than its utility value, a property known as coalitional rationality [63]. This provision ensures that no client experiences a loss, discouraging them from leaving the grand coalition, and thus can maintain the stability of the grand coalition in FL. However, like ShapleyValue, the LeastCore scheme needs to enumerate all possible coalitions, which may raise feasibility issues in scenarios with a large number of clients.

4.5 Comparison of FLCE Schemes

Table 1 provides a comparison of FLCE schemes. Generally, Individual and LeaveOneOut are simple and efficient schemes for estimating contributions. They have a linear increase in time complexity with the number of clients and are commonly used due to their intuitiveness and ease of implementation. Nonetheless, these schemes can not ensure fairness or reasonableness without considering various client coalitions. On the other hand, the ShapleyValue and LeastCore schemes satisfy the group rationality and zero element properties [51], ensuring fairness and reasonableness in contribution estimation. In particular, ShapleyValue ensures fairness at the individual level while LeastCore emphasizes fairness at the coalition level with group fairness [78]. Thus, they are applicable in a broader range of scenarios, though achieving these desirable properties requires exponential computation cost.

The relationship between contribution estimation schemes and data utility metrics reveals that the Individual scheme is highly sensitive to the selected metrics, unlike the LeaveOneOut, ShapleyValue, and LeastCore schemes which are more robust due to their complex calculations. Furthermore, a key distinction between ShapleyValue and LeastCore is that ShapleyValue supports symmetry and additivity, whereas LeastCore supports stability. Additivity allows ShapleyValue to integrate new utility metrics efficiently, eliminating recalculation of currently used utilities.

5 OPTIMIZATION TECHNIQUES

The efficiency of FLCE can be influenced by two primary factors. First, certain schemes like ShapleyValue [19] and LeastCore [78] requires the enumeration of 2^n coalitions. Second, estimating the contributions of clients necessitates training the model of multiple coalitions. To tackle the first factor, we can compute the approximate FLCE results by sampling a subset of the permutations (for ShapleyValue) or inequalities (for LeastCore). Concerning the second factor, we can reuse the gradients from the FL global model training, thereby avoiding redundant computations. Besides, certain computations with negligible impact on FLCE can be truncated.

5.1 Sampling

When calculating the ShapleyValue of a client, a naive optimization approach involves randomly choosing samples from the entire set of $n!$ permutations of $\mathcal{N}$ and calculates the marginal data utility gains attributed to the inclusion of this client. Subsequently, the mean of these marginal data utility gains serves as an approximation of the client's contribution to $\mathcal{N}$ [19]. Yet, in real-world applications, relying on random sampling can lead to inaccurate estimates of contributions. This inaccuracy stems from the unbalanced number of times different clients appear in a given position in the permutations. Specifically, a client's estimated marginal benefit will likely be higher if it is in the front position in a permutation. To ensure fairness in FLCE, every client should appear equally at specific positions in the selected permutations.

5.1.1 Structured sampling. An effective alternative is structured sampling. Specifically, when approximating the contribution of a client, structured sampling divides the randomly sampled permutations into n equally sized groups. For the i-th group, the method swaps the estimated client with the client at the i-th position in the permutation. Then, it calculates the average marginal contribution of the client being assessed across these n groups. This equalizes the number of samples for each position, ensuring a relatively fair result for the client being estimated [68].

5.1.2 Guided sampling. In the computation of ShapleyValue, where data utility metrics like Accuracy are utilized, it is observed that the marginal contributions of the initial clients tend to be larger than those of the later clients. Building on this insight, it becomes apparent that focusing on the clients appearing at the beginning of the permutations is important. This led to the proposal of the guided sampling technique. In contrast to structured sampling, guided sampling works by ensuring that the first n' positions of the sampled permutations encompass all possible permutations of $P(n, n')$, where $n' \ll n$. Afterward, a random ordering of the remaining $n - n'$ clients is generated. For instance, when $n' = 1$, the clients must appear an equal number of times in the first position, while no such requirement exists for the last $n - 1$ positions [42].

5.1.3 Subsampling. The above two methods perform well when the number of clients is not excessively large, e.g., cross-silo FL [86]. However, in scenarios where each data record is considered a separate client in a cross-device FL setting, these techniques still necessitate sampling a substantial number of permutations, leading to significant computational demands. To enhance computational efficiency, a subsampling technique tailored for cross-device FL can

be implemented. This involves selecting a small subset of representative clients from the entire coalition $\mathcal{N}$, thereby forming a smaller coalition $\mathcal{N}_p$. FLCE processes are then applied to clients in $\mathcal{N}_p$ using the above strategies. Following this, a regression model is trained using the estimated contributions of clients in $\mathcal{N}_p$ to predict the contributions of the remaining clients [18]. Nevertheless, given the extensive number of clients in cross-device FL, maintaining a sufficiently large sample size for $\mathcal{N}_p$ to accurately represent the full spectrum of clients in $\mathcal{N}$ can be challenging and time-intensive.

5.2 Gradients Reusing

Most FLCE methods typically require extensive model retraining, leading to high overhead. A practical solution to reduce this cost is reusing gradient updates from the FL global model training process.

5.2.1 One round reusing. A straightforward approach involves storing the clients' gradient updates from the FL global model training, then reusing these gradient updates in the coalition model training processes, and finally calculating the data utility of coalitions after a certain number of rounds. While this method circumvents the need to recalculate gradients and thereby reduce training costs, the reused gradients may not represent the actual gradients in non-global model training processes. This discrepancy can result in a considerable cumulative error over multiple training rounds, thereby diminishing the accuracy of the estimated data utility [64].

5.2.2 Multiple rounds reusing. An alternative approach involves estimating the contributions of each client in each training round and then aggregating these contributions over multiple rounds to obtain the overall client contributions [42, 64, 66]. In round t, the central server utilizes the updates from all the n clients to compute the gradient update $\boldsymbol{u}^t_{\mathcal{S}}$ for each coalition $\mathcal{S}$. Based on current global model $\mathcal{M}^t$, and the gradient update $\boldsymbol{u}^t_{\mathcal{S}}$, the server derives the model $\mathcal{M}^{t+1}_{\mathcal{S}}$, and the data utility of coalition $\mathcal{S}$. Then, a client's contribution in the t-th round can be derived based on the utilities of associated coalitions. Ultimately, it aggregates all the contribution scores of a specific client over multiple rounds to derive its overall contribution. Instead of training 2^n models and computing gradient updates in each model training process, this reusing technique reduces the model training complexity from exponential (2^n) to constant (1). This method can be further optimized through the implementation of truncation techniques [42] (see Section 5.3).

5.3 Truncation

In FLCE, certain computations have a negligible impact on the contribution estimation results. Therefore, eliminating these non-essential computations, a process known as truncation, can achieve a trade-off between the efficiency and other properties of FLCE methods. Note that this truncation strategy can be combined with sampling and gradient reuse methods when appropriate. Permutation truncation is a prevalent method for streamlining the computation of ShapleyValue, which is based on the observation that the marginal utility gain from each subsequent client decreases over the order of permutation. Therefore, once the utility difference between an initial coalition of clients and the grand coalition becomes negligible, further calculation of marginal contributions is considered redundant and can be skipped [19].

Table 2: Characteristics of datasets. "#Domain" is calculated by multiplying the distinct value counts of all the columns.

Dataset	Application	#Features	#Samples	#Domain
Tic-Tac-Toe	Board Game	9	958	10^3
Adult	Social Science	14	30 162	10^{21}
Bank	Finance and Marketing	16	45 211	10^{21}
Dota2	E-sports	116	102 944	10^{20}
Credit-Card	Finance (Fraud Detection)	29	568630	10^{166}

Note. Many other optimization techniques are not illustrated in this paper, as our primary focus lies on the fundamental FLCE optimization techniques. These methods can either be considered as variations of the algorithms discussed in this paper [13, 14, 43, 46, 70, 75, 76] or are not widely studied [1, 19, 25, 31, 37, 44, 61, 69, 80, 85].

6 EXPERIMENT

6.1 Setup

We conducted simulations within an 8-client HFL configuration, where clients possessed different samples within the same feature space, all intended for classification tasks. It is important to note that when the number of clients, n, becomes excessively large, the excessive computational complexity associated with ShapleyValue and LeastCore, which is 2^n, renders our experiments unfeasible.

6.1.1 Datasets. The datasets utilized in this study were sourced from the UCI Machine Learning Repository [33] or Kaggle [29]. The characteristics of these datasets are shown in Table 2.

(1) **Tic-Tac-Toe Dataset [2].** It comprises all possible board layouts at the ends of the games. The label indicates whether the player who initiates the game and takes the 'x' symbols is the winner.

(2) **Adult Dataset [3, 35].** Sourced from the US Census database, this dataset comprises attributes concerning personal information such as age, workclass, and educational levels. The label variable denotes whether an individual's annual earnings surpass $50,000.

(3) **Bank Dataset [48–50].** The Bank dataset was sourced from a Portuguese banking institution, which includes attributes such as the age, occupation, and marital status of clients. The classification target is whether a client will subscribe to a term deposit.

(4) **Dota2 Dataset [67].** This dataset encompasses extensive game-specific information from numerous rounds of Dota2 matches, e.g., the game mode, type, and the selected heroes for both teams. The classification target is whether team 1 emerges victorious.

(5) **Credit-Card Dataset [12]**. This dataset contains anonymized credit card transactions made by European cardholders in 2023. The objective of the concerned classification is to determine whether a transaction is fraudulent. We utilized this dataset to evaluate the effectiveness of FLCE methods in scenarios on large-scale data.

6.1.2 Method Selection. Our experiment covers most of the surveyed data utility metrics, contribution estimation schemes, and optimization techniques. Nonetheless, a specific few algorithms were omitted from our experiments due to their incompatibility with our primary experimental framework, which is designed to ensure the convincingness and broad applicability of our experiment results. Specifically, we excluded one-round gradient reuse

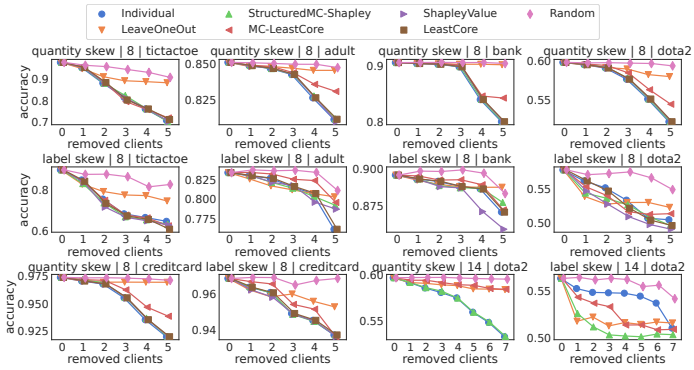

Figure 1: Client removal line chart used to assess the effectiveness of FLCE methods. A more effective FLCE method is expected to decrease faster. The captions are formatted as x|y|z, where x denotes the data partitioning strategy, y denotes the number of clients, and z specifies the dataset name.

from our experiments due to its potential to degrade model training quality, attributed to the accumulation of errors from gradient reuse, as detailed in Section 5.2. Furthermore, the implementation of InformationGain, which requires a customized machine learning model, was excluded to maintain equitable comparisons across FLCE methods, as discussed in Section 3.2.2. Subsampling, tailored for cross-device FL scenarios where each client holds a single data instance, does not align with our primary focus in this paper, where clients have multiple data instances.

6.1.3 Metrics. We evaluated the effectiveness, robustness, and efficiency of various FLCE methods, as elaborated in Section 2.2. Notably, for comparing method effectiveness, we utilized the client removal line plot [19, 61], as obtaining the ground truth of clients' contributions was not feasible in our experiments. In this approach, after estimating client contributions, clients are removed in descending order of contribution, and the data utility of the remaining coalition of clients is measured and marked on the figure. As each iteration removes the top-contributing client, a notable decrease in the utility of the remaining coalition is anticipated. Rather than comparing the estimated contribution scores with the ground truth, the philosophy of the client removal line plot is to visually contrast any two different FLCE methods, aiming to offer insights into their relative effectiveness.

6.1.4 Data Preprocessing. The initial datasets underwent encoding, with categorical attributes being transformed using a one-hot encoder and numerical attributes being converted to their corresponding z-scores. We did not consider string attributes in our experiment. Following this, a random splitting was made, allocating 10% of the datasets as test sets while the remaining 90% were designated as training sets.

Subsequently, the training set was divided and allocated to 8 clients. In practical scenarios, clients are less likely to possess i.i.d. samples in equal proportions. Thus, we simulated the following two specific scenarios. The first scenario employed the Dirichlet distribution [39] to partition the dataset into 8 segments with varying sample quantities while maintaining identical data distributions across these segments (Quantity Skewed). By following the approach suggested in [17, 39], the second scenario divided a dataset into multiple subgroups based on their labels and then used the Dirichlet distribution to randomly distribute the samples within each subgroup across the 8 clients (Label Skewed). This approach resulted in varying data distributions and quantities across clients.

6.1.5 Settings. The parameter α for the Dirichlet distribution was varied from 0.3 to 0.8. The size of the hidden layer in the 2-layer MLP model was adjusted between 4 and 24, while the number of global rounds was set between 3 and 80. The learning rates spanned from 0.001 to 0.01, and the batch sizes varied from 16 to 256. The selection of these parameter values was contingent on the inherent characteristics of the datasets. We repeated the effectiveness and robustness experiments 10 times and the efficiency experiment 3 times and computed the average outcomes to mitigate the influence of random variations.

6.1.6 Machine. All experiments were conducted on an Ubuntu server with 2 Intel CPUs running at 3.10 GHz, 256 GB of RAM, and 4 RTX 3090 graphics cards, each boasting 24 GB of memory, and utilizing CUDA 12.1. The implementation of all methods was carried out using Python 3.9.7 along with PyTorch 1.10.1.

6.2 Effectiveness Experiment

In this section, we assessed the effectiveness of Individual, LeaveOneOut, ShapleyValue, and LeastCore. Additionally, two optimized FLCE methods, StructuredMC-Shapley and MC-LeastCore, tailored for accelerating ShapleyValue and LeastCore respectively, were also evaluated, and a baseline method involving random client removal, Random, was employed. The results are shown in Figure 1.

We have the following observations. Firstly, ShapleyValue exhibited the most promising performance on various datasets and FL scenarios compared to other FLCE methods. This is attributed to the extensive enumeration within ShapleyValue, which aids in identifying the marginal contributions of clients within coalitions. Secondly, LeastCore was less effective than ShapleyValue with equivalent complexity, because LeastCore is not specifically designed to identify high-contributing individual clients but to maintain grand coalition stability. Thirdly, both sampling methods, MC-LeastCore and StructuredMC-Shapley, exhibit lower performance compared to the non-optimized methods, suggesting a loss of effectiveness due to sampling. Fourthly, Individual nearly matches ShapleyValue performance in quantity-skewed FL, and maintains a moderate level of performance in label-skewed FL, because of the positive correlation between data quantity and contributions under i.i.d. FL setting. However, under label-skewed FL, the heterogeneity of data samples across clients makes the performance of a client's model an inadequate reflection of its contribution. Fifthly, LeaveOneOut is the least effective in quantity-skewed FL, only slightly outperforming Random, but has moderate effectiveness in label-skewed FL. In quantity-skewed FL, this ineffectiveness mainly stems from the lack of data distinctiveness among clients, which leads to only slight performance degradation when a single client is removed. Besides, the performance of a machine learning task is not strictly positively related to the quantity of the records. In label-skewed FL, due to the heterogeneity of data across clients,

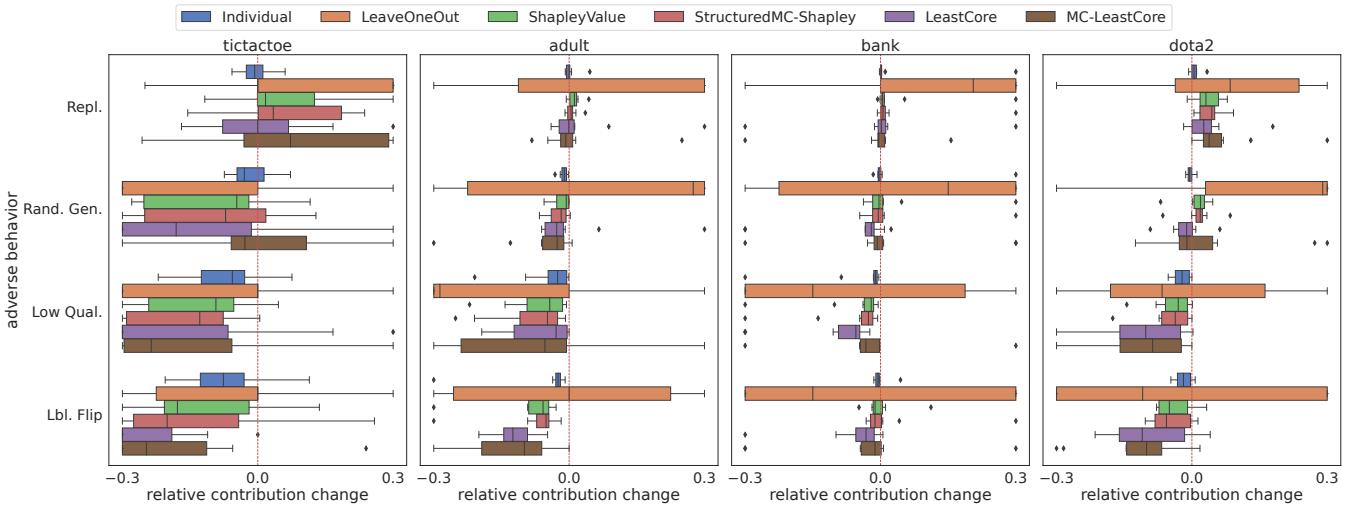

Figure 2: Robustness evaluation of FLCE methods. When considering data replication and random data generation behaviors, a smaller relative increase in contribution indicates better method performance. Conversely, for scenarios involving low-quality data and label flips, a greater relative decrease in contribution corresponds to superior method performance.

removing a client can cause sensible model performance decline, making LeaveOneOut moderately effective.

It's crucial to note that Individual and LeastCore seem to outperform ShapleyValue in the Adult dataset when the client removal rate exceeds 50%. This indicates their ability to detect low-contributing clients, resulting in reduced Accuracy towards the curve's end. However, as the client removal chart primarily evaluates an FLCE method's capacity to identify high-contribution clients, the middle and end segments of the curve cannot validate Individual and LeastCore's superiority over ShapleyValue.

Additionally, we simulated a scenario involving 14 clients participating using the Dota2 dataset. Due to the exponential complexity of ShapleyValue and LeastCore, we exclusively deployed their optimized versions, namely StructuredMC-Shapley and MC-LeastCore. Under a quantity-skew setting, the experimental outcomes resembled those in the 8-client setting, leaving LeaveOneOut and MC-LeastCore less effective. In the label-skew scenario, StructuredMC-Shapley outperforms other methods and Individual exhibits reduced effectiveness, also consistent with observations in the 8-client configuration.

Finding 1. *Most FLCE methods demonstrate effective performance under the quantity-skew scenario because the data across clients is homogeneous, where a client's contribution approximately correlates with its data volume. For label skewed FL, the ShapleyValue method evaluates a client's importance by aggregating its marginal contributions across all possible coalitions, which is likely the key to addressing FLCE under more challenging settings.*

6.3 Robustness Experiment

We simulated four commonly encountered adverse behaviors, comprising two strategic actions aimed at increasing individual contributions (i.e., data replication and random data generation), and two malicious actions designed to harm the global model (i.e., low-quality data and label flipping). In the context of data replication and random data generation, a client might duplicate a fraction of the initial samples and generate new samples, respectively. For generating low-quality data, we shuffled the labels of partial client data samples. In the case of label flipping, a portion of the labels were inverted to their opposite values. For the sake of simplicity and without loss of generality, we considered a single adverse client within the FL framework, responsible for altering 30% of the original dataset. We assessed the relative change in contribution caused by these adverse behaviors using the formula in Section 2.2.

Figure 2 demonstrates the robustness of these methods by showing the average and dispersion of relative contribution changes. The following observations can be made. First, Individual shows the highest robustness across all adverse behavior, where the averaged relative changes are always 0 or negative. This is because training models based on different clients' samples separately can accurately reflect the impact of adverse behaviors on the data quality of individual clients. Second, most averaged relative changes of ShapleyValue and StructuredMC-Shapley are around 0 for data replication and random data generation and tend to be negative for low-quality data and label flip, indicating their robustness. This occurs because strategic behaviors tend to maintain or slightly decrease the marginal contribution of a client, and malicious behaviors can reduce the marginal contribution of a specific client. Third, despite having theoretically similar complexity, LeastCore and MC-LeastCore show increased dispersion compared to ShapleyValue and StructuredMC-Shapley, showing moderate performance regarding adverse behaviors even though they exhibit negative relative changes. This occurs because adverse behaviors result in alterations to the linear programming constraints, which, in turn, lead to substantial fluctuations in the optimal solution to the contribution estimation problem. MC-LeastCore exhibits notably diminished robustness in comparison to LeastCore, implying that

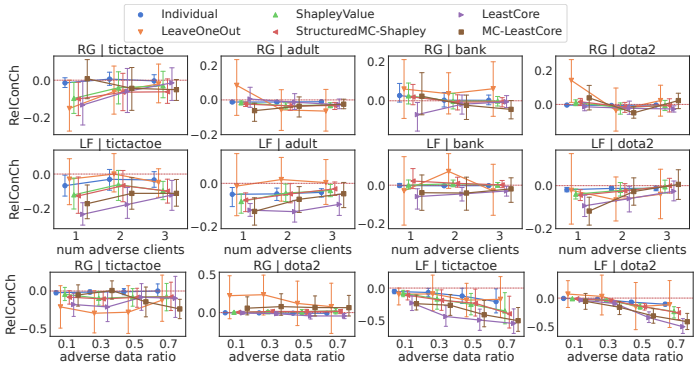

Figure 3: Robustness evaluation of FLCE methods with varying number of adverse clients (Row 1-2) and adverse data ratio (Row 3). We use "RelConCh" to represent relative contribution change.

MC-LeastCore possesses only a moderate capacity to replicate the robustness and stability of LeastCore. This discrepancy arises from the fact that sampling leads to the loss of linear programming constraints, resulting in a change in the optimal point. Last, the changes in LeaveOneOut contributions exhibit high dispersion and many of the changes are positive, indicating low robustness of the results.

We also studied the correlation between the robustness and the number of adverse clients. The results are shown in Row 1-2 of Figure 3. We observed that the relative contribution change of Individual is not sensitive to the number of adverse clients, because Individual assesses a client's contribution without considering other clients. Meanwhile, relative contribution changes for ShapleyValue and StructuredMC-Shapley remain constant or slightly rise when the number of adverse clients increases, attributed to the diminishing data utility of coalitions that include other adverse clients. Besides, we assessed how robustness correlates with adverse data ratio in Row 3 of Figure 3. Generally, the relative contribution changes of ShapleyValue and StructuredMC-Shapley decline with the increase of adverse data ratio, as these methods can reflect the reduction of the clients' data quality by considering marginal contributions. Besides, the relative contribution decrease of Individual is significantly lower than that of the other methods, since Individual assigns a relatively high contribution to adverse clients.

Finding 2. *FLCE methods that overlook clients' cooperation, such as Individual, can reflect the impact of adverse behaviors on data. FLCE methods that evaluate a wide range of coalitions, like ShapleyValue and StructuredMC-Shapley, demonstrate robustness against adverse behaviors. FLCE methods that only consider few limited client collaborations, such as LeaveOneOut, show reduced robustness to adverse actions.*

6.4 Efficiency Experiment

We measured and compared the computation time overhead of various FLCE methods, considering both CPU and GPU overhead required to complete these methods. In addition to the original methods and their sampling optimized variants, MultiRounds (MR), a ShapleyValue-based method optimized by the "gradient reuse technique", was also evaluated. Note that we only evaluate the FLCE efficiency, other than the summation of FL training time and FLCE time. We implemented a cache mechanism to store all previously computed data utility metrics (if applicable), which eliminates the need for redundant re-evaluation of coalitions, ensuring that the implementation aligns with the theoretical computational complexity. Figure 4 shows the experimental results.

ShapleyValue and LeastCore exhibited the highest computational costs, while StructuredMC-Shapley and MC-LeastCore had relatively high computational requirements. Individual, LeaveOneOut, and MR were found to be most efficient. It is important to note that the time expenditure for MR was lower than that of StructuredMC-Shapley and ShapleyValue because MR only necessitates the training of a single model, whereas StructuredMC-Shapley and ShapleyValue involve the training of multiple models.

Even though, compared with Individual, MR has a smaller complexity and LeaveOneOut theoretically does not have a greater complexity, their practical time requirements are notably higher than those of Individual. The primary reason is that MR, a method with the same amount of training data and number of rounds as Individual, necessitates the evaluation of a considerable number of models (i.e., $O(2^n)$) during each training round. Conversely, during training, Individual does not require such a large number of model evaluations in each round. In contrast, LeaveOneOut needs to utilize the combined data of coalitions consisting of all or $(n-1)$ clients to train a model, significantly increasing the volume of training data and consequently increasing computation time. Additionally, despite having the same theoretical time complexity, StructuredMC-Shapley demonstrates slightly lower computational costs than MC-LeastCore. This discrepancy arises because the structured sampling technique employed by StructuredMC-Shapley enables the reusing of previously calculated utility metric values. In contrast, MC-LeastCore samples exactly $n^2 \log n$ distinct coalitions in each round, resulting in longer computation times compared to StructuredMC-Shapley.

Finding 3. *In general, the practical running time of FLCE methods exhibits a positive correlation with their theoretical complexity, which is determined by the number of evaluated coalitions. Besides, the computational cost is also influenced by the quantity of training data and the selection of optimization techniques.*

6.5 Evaluating Various Utility Metrics

In this section, we evaluated the most representative utility metrics, namely Accuracy, DataQuantity, CosineGradient, and RobustVolume. We chose ShapleyValue as the contribution estimation scheme to integrate with these metrics, due to its superior ability to effectively estimate contributions.

6.5.1 Effectiveness. Figure 5 demonstrates the effectiveness of ShapleyValue with various data utility metrics under quantity skew and label skew distributions without adverse behaviors. We exclusively assess the RobustVolume algorithm using the Tic-Tac-Toe dataset due to its susceptibility to numerical instability, which restricts its applicability. We do not include an evaluation of Volume in this paper because the metric exhibits low robustness against data replication [77] and high computational complexity. In practical scenarios, obtaining a high-quality test set managed by the central

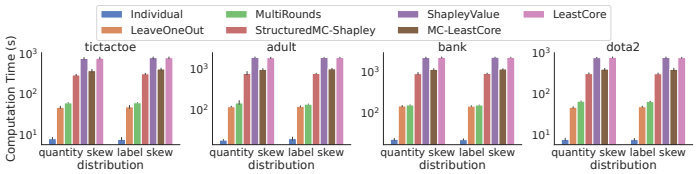

Figure 4: Efficiency of FLCE methods.

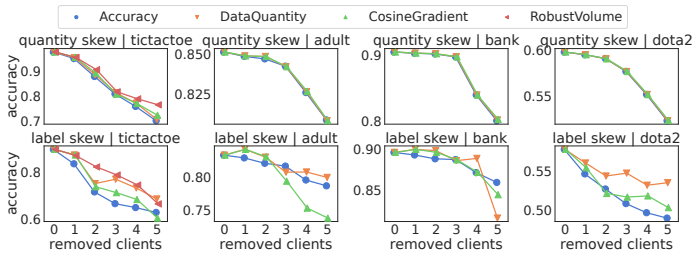

Figure 5: Effects of various utility metrics.

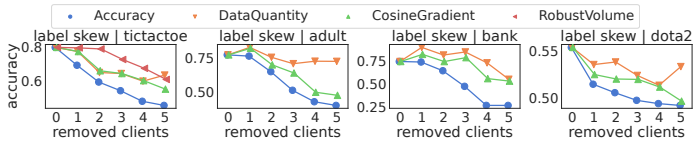

Figure 6: Effects of utility metrics on label flipped datasets.

server to calculate Accuracy can be challenging, making it important to find alternatives. With this consideration, if a utility metric, when combined with ShapleyValue, exhibits a consistent downward trend in the client removal chart with Accuracy as its vertical coordinate, it can be regarded as a viable alternative to Accuracy.

We have the following observations. Firstly, CosineGradient is a strong alternative for Accuracy across various distributions. This is because the similarity between a coalition's model gradient and the global model's gradient effectively reflects the coalition's utility when the global model is considered optimal, without accounting for malicious clients. Secondly, DataQuantity demonstrates solid substitutability for Accuracy in scenarios with quantity-skewed data distribution, but its substitutability is only moderate when dealing with label-skewed distribution. This difference stems from the data homogeneity inherent in quantity-skewed distribution, whereas label-skewed distribution lacks this characteristic. Lastly, RobustVolume exhibits a moderate level of substitutability for Accuracy across all distribution types. This is because data diversity does not necessarily align with the proportion of data within the dataset that contributes to models.

We also assess the utility metrics' performance in the Collaborative Adverse Behavior scenario, where two adverse clients alter 80% of the original datasets, focusing on one of the key adverse behaviors, label flipping, due to space constraints. Figure 6 illustrates the effectiveness of utility metrics in the presence of label-flipping behavior within the Collaborative Adverse Behavior scenario. Our results reveal that CosineGradient, while demonstrating some effectiveness, has limited substitutability for Accuracy. This limitation arises when the global model gradient inaccurately represents the global optimal gradient due to label flip occurrences under label

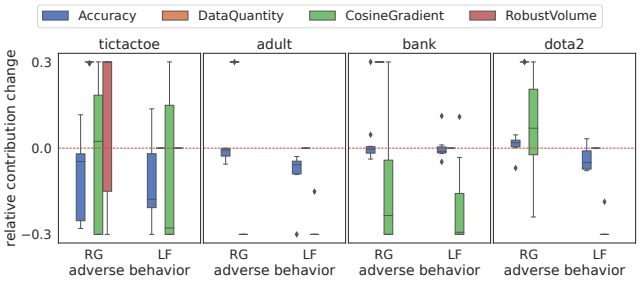

Figure 7: Robustness evaluation of utility metrics. Random data generation (RG) and label flipping (LF) are selected as examples of strategic and malicious behaviors, respectively.

skew, leading to misjudgments of contributions. Both DataQuantity and RobustVolume exhibit reduced substitutability for Accuracy as they do not leverage label information, rendering them unable to identify malicious clients.

> Finding 4. *In the absence of adverse behavior, we can use DataQuantity and CosineGradient as substitutes for Accuracy in quantity-skewed datasets and CosineGradient in label-skewed datasets. However, no existing metrics can completely replace Accuracy in cases of heavy adversity like significant label flipping.*

6.5.2 Robustness and Efficiency. Figure 7 reflects the robustness of distinct data utility metrics in scenarios where one client alters 30% of its data. First, DataQuantity does not possess the capability to identify random generation and label-flipping behaviors, because it does not consider data quality. Second, CosineGradient can identify most label-flipping behaviors, but its robustness against random generation is low. Third, RobustVolume demonstrates weak robustness against random generation and label flipping because it does not depend on labels and is unable to identify the quality of labels, and the term "robust" in its algorithm specifically refers to its resilience against replication [77]. Note that RobustVolume exhibited cases where some relative contribution changes become negative in the random data generation scenario, which is inconsistent with the expected results. This is attributed to the numerical instability. Additionally, the time cost of various data utility metrics combined with the ShapleyValue scheme can be found in Table 3, aligning with their theoretical complexity.

6.6 Evaluating Optimization Techniques

We assessed the performance of sampling and gradient reuse techniques to optimize the FLCE process. Truncation is applied in conjunction with these techniques when it is deemed suitable. We exclusively showcase the results of the effectiveness experiment due to space constraints. The experimental results demonstrating the robustness and efficiency of typical optimized FLCE methods can be found in Section 6.3 and Section 6.4.

6.6.1 Sampling. We evaluated three techniques, namely random sampling, structured sampling, and guided sampling, and the standard ShapleyValue without sampling is selected as a baseline. We denoted three ShapleyValue variants as TMC-Shapley, StructuredMC-Shapley, and GuidedTMC-Shapley, where T stands for Truncation

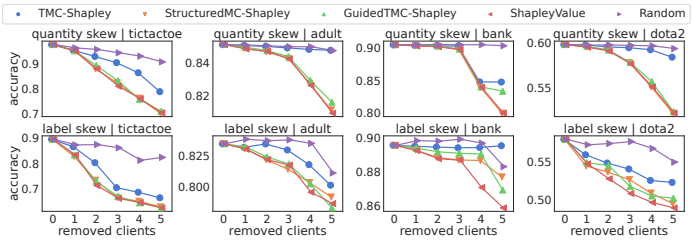

Figure 8: Effectiveness evaluation of ShapleyValue using various sampling techniques.

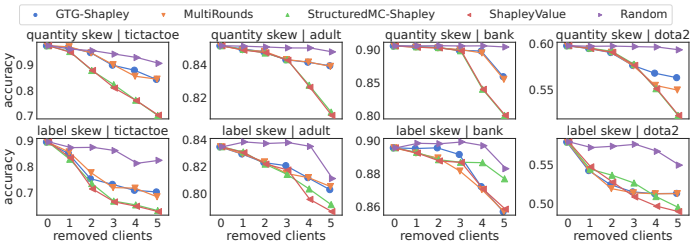

Figure 9: Effectiveness evaluation of ShapleyValue using different gradient reuse techniques.

and MC stands for MonteCarlo. In Figure 8, we can draw the following observations. First, the performance drop caused by TMC-Shapley is only a little sharper than Random in quantity-skewed FL. In label-skewed FL, TMC-Shapley has certain effectiveness, but it is still the worst due to the absence of systematic sampling. Conversely, the performance drop in StructuredMC-Shapley and GuidedTMC-Shapley falls between that of TMC-Shapley and ShapleyValue, suggesting that the systematic sampling technique can improve the effectiveness of methods.

6.6.2 Gradient Reuse. We conducted an evaluation of two contribution estimation methods based on gradient reuse: MR and Guided Truncation Gradient Shapley (GTG-Shapley). The results, depicted in Figure 9, yield the following observations. First, in cases of quantity skew distribution, both GTG-Shapley and MR exhibit smaller decreases in Accuracy compared to ShapleyValue and StructuredMC-Shapley. Second, in the context of label skew, both GTG-Shapley and MR show moderate decreases overall. Both observations suggest a relatively limited ability of gradient reuse methods to identify highly contributing clients. This is primarily due to GTG-Shapley and MR relying on the global model's performance, which may not accurately reflect the model's performance when trained solely on specific coalitions. Last, the magnitudes of the decreases identified by GTG-Shapley and MR are similar in most scenarios, implying that GTG-Shapley serves as an effective approximation and improvement over MR.

7 PROSPECT

Following our comprehensive review of the literature and analysis of experimental results, we have observed that research efforts in the field of FLCE have only begun to emerge in recent years, with many proposed methods still distant from practical implementation in FL systems. In light of these findings, we suggest the following prospects that may serve as inspiration for future work in FLCE.

Table 3: Efficiency evaluation of utility metrics (in seconds).

Dataset	Accuracy	DataQuantity	CosineGradient	RobustVolume
Tic-Tac-Toe	8.26×10^{2}	1.09×10^{-2}	1.67×10^{1}	2.01×10^{1}
Adult	1.96×10^{3}	3.93×10^{-2}	1.92×10^{1}	/
Bank	2.34×10^{3}	3.73×10^{-2}	2.09×10^{1}	/
Dota2	7.88×10^{2}	7.13×10^{-1}	6.94	/

(1) **Utility Metric.** A metric is needed to identify adverse behaviors efficiently with minimal computational expense, without relying solely on data quantity or diversity to avoid unjustly rewarding strategic clients. Ideally, this metric would align with the mathematical principles of cooperative game theory, such as additivity [6].

(2) **FLCE Scheme.** No contribution estimation scheme exists that can fulfill the criteria of effectiveness, robustness, and efficiency simultaneously. It would be beneficial to either explore a more balanced trade-off among these three aspects or prove that it is not feasible to simultaneously meet all three requirements in FLCE.

(3) **Optimization Technique.** The current sampling techniques have been tailored for ShapleyValue, and there has been no exploration of sampling techniques, such as systematic sampling, specifically designed for LeastCore. Besides, while gradient reuse techniques theoretically tackle the problem of high estimation costs compared to model training costs, their practical effectiveness and efficiency remain sub-optimal, necessitating further investigation.

(4) **Benchmarking.** Our paper focused on binary classification datasets with structural data due to space constraints. Future research could explore multi-classification and regression datasets, as well as image datasets. Furthermore, due to the lack of widely recognized real-world FL datasets, there is a need to create realistic partitioned FLCE datasets and establish a ground truth for clients' contributions.

8 CONCLUSION

In this paper, we have undertaken an extensive examination of contribution estimation methods in FL and have introduced a comprehensive evaluation framework. Our analysis encompasses data utility metrics, contribution estimation schemes, and optimization techniques, offering valuable insights for prospective research endeavors. Our comprehensive evaluation encompasses a wide range of scenarios and datasets, thereby serving as a benchmark for evaluating the performance of contribution estimation methods in the context of FL. Through rigorous analysis, we have identified the strengths and limitations inherent in various methods, thereby contributing to the advancement of more effective contribution estimation methods for FL. Our adaptable testing framework accommodates both existing and forthcoming methods, simplifying the evaluation process in this ever-evolving domain. In summary, this research not only advances the theoretical aspects of FL contribution estimation but also enhances our understanding of the practical implementation of FLCE methods.

ACKNOWLEDGMENTS

This paper was supported by National Key R&D Program of China under Grant Number 2023YFB4503600, NSF of China (61925205, 62232009, 62102215), and Zhongguancun Lab.

REFERENCES

[1] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. 2018. cpSGD: Communication-efficient and differentially-private distributed SGD. *Advances in Neural Information Processing Systems* 31 (2018).

[2] David Aha. 1991. Tic-Tac-Toe Endgame. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5688J.

[3] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

[4] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

[5] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *SIAM review* 60, 2 (2018), 223–311.

[6] Rodica Branzei, Dinko Dimitrov, and Stef Tijs. 2008. *Models in cooperative game theory*. Vol. 556. Springer Science & Business Media.

[7] Alessandro Di Bucchianico. 2014. Coefficient of Determination (R 2). *Wiley StatsRef: Statistics Reference Online* (2014).

[8] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832.

[9] Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development* 7, 3 (2014), 1247–1250.

[10] Yiqiang Chen, Xiaodong Yang, Xin Qin, Han Yu, Piu Chan, and Zhiqi Shen. 2020. Dealing with label quality disparity in federated learning. *Federated Learning: Privacy and Incentive* (2020), 108–121.

[11] Zicun Cong, Xuan Luo, Pei Jian, Feida Zhu, and Yong Zhang. 2021. Data Pricing in Machine Learning Pipelines. *arXiv preprint arXiv:2108.07915* (2021).

[12] Nidula Elgiriyewithana. 2023. *Credit Card Fraud Detection Dataset 2023, Version 1*. Kaggle. Retrieved February 11, 2024 from https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023/version/1

[13] Zhenan Fan, Huang Fang, Zirui Zhou, Jian Pei, Michael P Friedlander, Changxin Liu, and Yong Zhang. 2021. Improving Fairness for Data Valuation in Horizontal Federated Learning. *arXiv preprint arXiv:2109.09046* (2021).

[14] Zhenan Fan, Huang Fang, Zirui Zhou, Jian Pei, Michael P Friedlander, and Yong Zhang. 2022. Fair and efficient contribution valuation for vertical federated learning. *arXiv preprint arXiv:2201.02658* (2022).

[15] Shaohan Feng, Dusit Niyato, Ping Wang, Dong In Kim, and Ying-Chang Liang. 2019. Joint service pricing and cooperative relay communication for federated learning. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 815–820.

[16] Raul Castro Fernandez, Pranav Subramaniam, and Michael J. Franklin. 2020. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proc. VLDB Endow.* 13, 11 (2020), 1933–1947. http://www.vldb.org/pvldb/vol13/p1933-fernandez.pdf

[17] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10112–10121.

[18] Amirata Ghorbani, Michael Kim, and James Zou. 2020. A distributional framework for data valuation. In *International Conference on Machine Learning*. PMLR, 3535–3544.

[19] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*. PMLR, 2242–2251.

[20] Bin Guo, Huihui Chen, Qi Han, Zhiwen Yu, Daqing Zhang, and Yu Wang. 2016. Worker-contributed data utility measurement for visual crowdsensing systems. *IEEE Transactions on Mobile Computing* 16, 8 (2016), 2379–2391.

[21] Xiao Han, Leye Wang, and Junjie Wu. 2021. Data valuation for vertical federated learning: An information-theoretic approach. *arXiv preprint arXiv:2112.08364* (2021).

[22] Chao Huang, Ming Tang, Qian Ma, Jianwei Huang, and Xin Liu. 2023. Promoting Collaborations in Cross-Silo Federated Learning: Challenges and Opportunities. *IEEE Communications Magazine* (2023).

[23] Jiyue Huang, Rania Talbi, Zilong Zhao, Sara Boucchenak, Lydia Y Chen, and Stefanie Roos. 2020. An exploratory analysis on users' contributions in federated learning. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 20–29.

[24] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. 2013. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 245–251.

[25] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. 2019. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619* (2019).

[26] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1167–1176.

[27] Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. 2021. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8239–8247.

[28] Noah Johnson, Joseph P Near, and Dawn Song. 2018. Towards practical differential privacy for SQL queries. *Proceedings of the VLDB Endowment* 11, 5 (2018), 526–539.

[29] Kaggle [n.d.]. *Kaggle: Your Machine Learning and Data Science Community*. Retrieved February 14, 2024 from https://www.kaggle.com

[30] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

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

[32] Michael Kearns and Dana Ron. 1999. Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. *Neural Computation* 11, 6 (1999), 1427–1453.

[33] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. [n.d.]. *The UCI Machine Learning Repository*. Retrieved Oct 1, 2023 from https://archive.ics.uci.edu

[34] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*. PMLR, 1885–1894.

[35] Ron Kohavi et al. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.. In *Kdd*, Vol. 96. 202–207.

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

[37] Yongchan Kwon, Manuel A Rivas, and James Zou. 2021. Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 793–801.

[38] Junqing Le, Di Zhang, Xinyu Lei, Long Jiao, Kai Zeng, and Xiaofeng Liao. 2023. Privacy-Preserving Federated Learning With Malicious Clients and Honest-but-Curious Servers. *IEEE Trans. Inf. Forensics Secur.* 18 (2023), 4329–4344. https://doi.org/10.1109/TIFS.2023.3295949

[39] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 965–978.

[40] Suyi Li, Yong Cheng, Yang Liu, Wei Wang, and Tianjian Chen. 2019. Abnormal Client Behavior Detection in Federated Learning. *CoRR* abs/1910.09933 (2019). arXiv:1910.09933 http://arxiv.org/abs/1910.09933

[41] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine* 37, 3 (2020), 50–60.

[42] Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. 2021. GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning. *arXiv preprint arXiv:2109.02053* (2021).

[43] Xuan Luo, Jian Pei, Zicun Cong, and Cheng Xu. 2022. On shapley value in data assemblage under independent utility. *arXiv preprint arXiv:2208.01163* (2022).

[44] Hongtao Lv, Zhenzhe Zheng, Tie Luo, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, and Chengfei Lv. 2021. Data-Free Evaluation of User Contributions in Federated Learning. In *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. 1–8. https://doi.org/10.23919/WiOpt52861.2021.9589136

[45] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. 2020. Collaborative fairness in federated learning. *Federated Learning: Privacy and Incentive* (2020), 189–204.

[46] Shuaicheng Ma, Yang Cao, and Li Xiong. 2021. Transparent contribution evaluation for secure federated learning on blockchain. In *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 88–91.

[47] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[48] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.

[49] Sergio Moro, Raul Laureano, and Paulo Cortez. 2011. Using data mining for bank direct marketing: An application of the crisp-dm methodology. (2011).

[50] S. Moro, P. Rita, and P. Cortez. 2012. Bank Marketing. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K306.

[51] Tri-Dung Nguyen. 2015. The fairest core in cooperative games with transferable utilities. *Operations Research Letters* 43, 1 (2015), 34–39.

[52] Shashi Raj Pandey, Nguyen H Tran, Mehdi Bennis, Yan Kyaw Tun, Aunas Manzoor, and Choong Seon Hong. 2020. A crowdsourcing framework for on-device

federated learning. *IEEE Transactions on Wireless Communications* 19, 5 (2020), 3241–3256.

[53] Jian Pei. 2020. A survey on data pricing: from economics to data science. *IEEE Transactions on knowledge and Data Engineering* 34, 10 (2020), 4586–4608.

[54] Bezalel Peleg and Peter Sudhölter. 2007. *Introduction to the theory of cooperative games*. Vol. 34. Springer Science & Business Media.

[55] Adam Richardson, Aris Filos-Ratsikas, and Boi Faltings. 2019. Rewarding high-quality data via influence functions. *arXiv preprint arXiv:1908.11598* (2019).

[56] Alvin E Roth. 1988. Introduction to the Shapley value. *The Shapley value* (1988), 1–27.

[57] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. 2022. The shapley value in machine learning. *arXiv preprint arXiv:2202.05594* (2022).

[58] David Schmeidler. 1969. The nucleolus of a characteristic function game. *SIAM Journal on applied mathematics* 17, 6 (1969), 1163–1170.

[59] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).

[60] Yuxin Shi, Han Yu, and Cyril Leung. 2023. Towards fairness-aware federated learning. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[61] Sung Kuk Shyn, Donghee Kim, and Kwangsu Kim. 2021. Fedccea: A practical approach of client contribution evaluation for federated learning. *arXiv preprint arXiv:2106.02310* (2021).

[62] Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. 2020. Collaborative machine learning with incentive-aware model rewards. In *International conference on machine learning*. PMLR, 8927–8936.

[63] Tamás Solymosi and TES Raghavan. 2001. Assignment games with stable core. *International Journal of Game Theory* 30 (2001), 177–185.

[64] Tianshu Song, Yongxin Tong, and Shuyue Wei. 2019. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2577–2586.

[65] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[66] Qiheng Sun, Xiang Li, Jiayao Zhang, Li Xiong, Weiran Liu, Jinfei Liu, Zhan Qin, and Kui Ren. 2023. Shapleyfl: Robust federated learning based on shapley value. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2096–2108.

[67] Stephen Tridgell. 2016. Dota2 Games Results. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5W593.

[68] Tjeerd van Campen, Herbert Hamers, Bart Husslage, and Roy Lindelauf. 2018. A new approximation method for the Shapley value applied to the WTC 9/11 terrorist attack. *Social Network Analysis and Mining* 8 (2018), 1–12.

[69] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. 2019. Measure contribution of participants in federated learning. In *2019 IEEE international conference on big data (Big Data)*. IEEE, 2597–2604.

[70] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. 2020. A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive* (2020), 153–167.

[71] Yong Wang, Kaiyu Li, Guoliang Li, Yunyan Guo, and Zhuo Wan. 2024. Fast, Robust and Interpretable Participant Contribution Estimation for Federated Learning. In *ICDE*.

[72] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[73] Cort J Willmott and Kenji Matsuura. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research* 30, 1 (2005), 79–82.

[74] Margaret Wright. 2005. The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bulletin of the American mathematical society* 42, 1 (2005), 39–56.

[75] Haocheng Xia, Jinfei Liu, Jian Lou, Zhan Qin, Kui Ren, Yang Cao, and Li Xiong. 2023. Equitable Data Valuation Meets the Right to Be Forgotten in Model Markets. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3349–3362.

[76] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. 2021. Gradient driven rewards to guarantee fairness in collaborative machine learning. *Advances in Neural Information Processing Systems* 34 (2021), 16104–16117.

[77] Xinyi Xu, Zhaoxuan Wu, Chuan Sheng Foo, and Bryan Kian Hsiang Low. 2021. Validation free and replication robust volume-based data valuation. *Advances in Neural Information Processing Systems* 34 (2021), 10837–10848.

[78] Tom Yan and Ariel D Procaccia. 2021. If you like shapley then you'll love the core. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5751–5759.

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

[80] Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data valuation using reinforcement learning. In *International Conference on Machine Learning*. PMLR, 10842–10851.

[81] Rongfei Zeng, Chao Zeng, Xingwei Wang, Bo Li, and Xiaowen Chu. 2021. A comprehensive survey of incentive mechanism for federated learning. *arXiv preprint arXiv:2106.15406* (2021).

[82] Yufeng Zhan, Peng Li, Kun Wang, Song Guo, and Yuanqing Xia. 2020. Big data analytics by crowdlearning: Architecture and mechanism design. *IEEE Network* 34, 3 (2020), 143–147.

[83] Mengxiao Zhang and Fernando Beltrán. 2020. A survey of data pricing methods. *Available at SSRN 3609120* (2020).

[84] Bowen Zhao, Ximeng Liu, and Wei-neng Chen. 2021. When crowdsensing meets federated learning: Privacy-preserving mobile crowdsensing system. *arXiv preprint arXiv:2102.10109* (2021).

[85] Jie Zhao, Xinghua Zhu, Jianzong Wang, and Jing Xiao. 2021. Efficient client contribution evaluation for horizontal federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3060–3064.

[86] Shuyuan Zheng, Yang Cao, and Masatoshi Yoshikawa. 2023. Secure Shapley Value for Cross-Silo Federated Learning. *Proceedings of the VLDB Endowment* 16, 7 (2023), 1657–1670.