Towards Fair, Robust and Efficient Client Contribution Evaluation in Federated Learning

Meiying Zhang, Huan Zhao, Sheldon Ebron and Kan Yang
Dept. of Computer Science, University of Memphis, USA.
{mzhang6, hzhao2, sebron, kan.yang}@memphis.edu

***Abstract*—Federated Learning (FL) is widely applied in communication networks. The performance of clients in FL can vary due to various reasons. Assessing the contributions of each client is crucial for client selection and compensation. It is challenging because clients often have non-independent and identically distributed (non-iid) data, leading to potentially noisy or divergent updates. The risk of malicious clients amplifies the challenge especially when there's no access to clients' local data or a benchmark root dataset. In this paper, we introduce a novel method called Fair, Robust, and Efficient Client Assessment (FRECA) for quantifying client contributions in FL. FRECA employs a framework called FedTruth to estimate the global model's ground truth update, balancing contributions from all clients while filtering out impacts from malicious ones. This approach is robust against Byzantine attacks and incorporates a Byzantine-resilient aggregation algorithm. FRECA is also efficient, as it operates solely on local model updates and requires no validation operations or datasets. Our experimental results show that FRECA can accurately and efficiently quantify client contributions in a robust manner.**

***Index Terms*—FRECA, Client Assessment, Contribution Evaluation, Fairness, Robustness, Efficiency, Federated Learning**

I. Introduction

Federated learning (FL) has various applications in communications, such as mobile network optimization [1], quality of service improvement [2], and security and anomaly detection [3]. Participants or clients in FL actively contribute to the training of a global model by providing local models trained on their own data. It is important to rigorously quantify the individual contributions of clients, which is an essential step for efficient client selection, fair allocation of profit earned through the FL process, and design of incentive mechanisms aimed at attracting high-valued participants. The assessment of client contributions presents a notable challenge, as data is indirectly conveyed through locally trained models utilizing the global model as a foundation.

Traditional data valuation or pricing methods [4]–[6] are thus not applicable. The degree of contribution is intricately influenced not only by the size and distribution of a client's data but also by factors such as the specific FL task, the initial global model serving as the training basis, the iteration/round of training in which the client participates, and the collective composition of clients participating in the same round. Consequently, there is a compelling need for a precise measurement of the contribution made by a client to the global model in each training round, specifically quantifying the impact of each local model or update on the aggregated global model.

Due to various qualities of data and trained local model, it is unfair to treat all the clients equally [7]–[9] or evaluate client contributions based on the size of the training dataset [10]. A dishonest client may train the local model over partial datasets or claim a large size of training dataset for more rewards. Some existing client contribution evaluation methods only focus on whether the client has submitted the model updates or whether the norms of model updates are within a threshold [11]. In [12], a deletion-based approach is proposed to evaluate the contribution of an individual client by comparing the accuracy of the global model with and without this client.

More accurate Shapley value approaches [13], [14] model FL as a cooperative game and compute the contribution of each player as the marginal impact on the overall reward which is the accuracy achieved by the global model. However, the Shapley value approach has two significant shortcomings: 1) It imposes *intensive computational demand*, which stems from the need to reconstruct and evaluate a variety of sub-models. To address this, techniques such as random permutation sampling and group testing have been introduced [15], [16]. However, these methods only partially mitigate the computational intensity, which becomes particularly burdensome as the number of clients increases; and 2) The framework requires an *auxiliary validation dataset* to assess the performance of all the sub-models. However, such a validation dataset may not be feasible in many FL applications due to privacy and regulation constraints.

Another line of approach is distance-based methods [17], [18] which assess each client's contribution per FL round based on the distance between the client's local model and the prior global model. Unlike Shapley value approaches, distance-based methods do not require extra validation datasets but they face two critical challenges: 1) *These methods primarily focus on the gap between the recent global model and local updates,* ***lacking a solid ground truth*** *for comparison*. FLTrust [19] suggests a benign root dataset as a standard, but data privacy issues often hinder its adoption; and 2)*No defense strategies on the aggregator side are taken into account*. The influence of different Byzantine-resilient strategies during the aggregation process can significantly impact the evaluation of each client's contribution. For instance, it may be pertinent to consider factors like the aggregation weights, which could include the ratio of local to the total data samples in algorithms

This work was supported in part by NSF DGE 2146427.

like FedAvg. Consequently, this presents an essential question: should the evaluation of a client's contribution rely solely on the distance between their local model and the aggregated model, or should it be a more nuanced measure that incorporates these aggregation weights?

To address these issues, we introduce a novel method called Fair, Robust, and Efficient Client Assessment (FRECA) for quantifying client contributions in FL. The contributions of this paper are summarized as follows:

- We propose a novel method FRECA for quantifying client contributions in FL. FRECA employs a framework called FedTruth to estimate the global model's ground truth update, balancing contributions from all clients while mitigating impacts from malicious ones.
- To the best of our knowledge, this is the first contribution evaluation method to incorporate defense mechanism against malicious clients. This approach is robust against Byzantine attacks and also efficient, as it operates solely on local model updates and requires no validation operations or datasets.
- Our experimental results show that FRECA can accurately and efficiently quantify client contributions in a robust manner.

The remainder of this paper is organized as follows: In Section II, we present the related work of client contribution evaluation and Byzantine-resilient aggregation algorithms in FL. Section III describes the problem formulation of federated learning, existing client contribution assessment methods, and FedTruth framework to estimate ground truth of the global model update. In Section IV, we present our method FRECA, Section V provides experimental evaluation, and Section VI concludes the paper.

II. Related Work

Many methods have been proposed to measure the contribution of a client in FL, which usually fall in two directions: *Shapley Value Approaches* and *Distance-based Approaches*.

A. Shapley Value Approach

Shapley value [13] serves as an equitable framework for gauging contribution. It calculates the marginal contribution, delineating the variance in overall rewards when a participant either engages in or refrains from a particular activity. Methodologies for efficient computation were delineated by Jia *et al.*, such as using Locality Sensitive Hashing in KNN scenarios [20] and leveraging Shapley value sparsity [21]. In the federated learning context, researchers envisioned each client as a 'player', examining their influence on model performance. For instance, [14] presented the Contribution Index (CI) echoing the principles of the Shapley value. In a parallel vein, [15] introduced the Federated Shapley value (Federated SV) that uniquely considers the chronological order of client participation. Both CI and Federated SV not only adhere to the fairness principles of the Shapley value but also offer feasible computational methods through approximation algorithms.

B. Distance-based Approaches

In [17], the contribution of each client is determined by utilizing the 'attention weight' (effectively the aggregation weight) which is discerned based on the divergence between a client's local model and the global model from the previous round. A presumption of this strategy is the belief that the larger the influence a client exerts on the global model, the more significant their contribution, which might be challenged by real-world complexities. Similarly, [18] measures contribution by inspecting the angular difference between local and global loss function gradients, postulating that a smaller angle signifies a more pronounced contribution to the global model update. Distance-based methodologies *eliminate the need to assess model performance using supplementary validation datasets*. However, they do face a notable challenge: with emphasis being on determining the distance between the global and local models, *the lack of a definitive "ground truth" for the global model*, which would otherwise serve as a standard for distance measurements. One proposed solution FLTrust [19] involves using a benign root dataset as this standard, but it often proves untenable in FL settings due to prevailing data privacy and regulatory hurdles.

C. Byzantine-Resilient Aggregation Algorithms

Byzantine attack is a common attack in FL that aims to make the global model converged to a sub-optimal model by arbitrarily altering local model updates. Types of Byzantine attack include model-boosting attack [22], Gaussian noise attack [23], and constraint-and-scaling attack [8]. Several aggregation methods are proposed [19], [23], [24] to defend against this attack. *Krum* [23] selects the local model from one 'best' client as the global model for each round, thus ignoring contributions from other clients. *Trimmed Mean* [24] tries to remove malicious clients by trimming outliers from local models, but in this way, benign models trained on underrepresented data may also be removed. In *FLTrust* [19], aggregation weights are estimated based on the similarity between each model update with a *ground-truth model update* which is trained by the aggregator using a benign root dataset. However, this benign root dataset may not be practical in many applications.

Table I lists the comparison between our proposed FRECA and the existing approaches.

TABLE I: Comparison of Client Assessment Goals

Scheme	**No Validation Dataset**	**Efficient**	**Byzantine-resilient**	**Attack Detection**
SV [14]	✗	✗	✗	✗
LOO [12]	✗	✓	✗	✗
Distance-based [17]	✓	✓	✗	✗
Our FRECA	✓	✓	✓	✓

III. Preliminaries

A. Federated Learning

A general FL system consists of an aggregator and a set of clients S. Let $\mathcal{D}_k$ be the local dataset held by the client k ($k \in S$). The typical FL goal [10] is to learn a model collaboratively without sharing local datasets by solving

$$\begin{aligned} &\min_{w} F(w) = \sum_{k \in S} p_k \cdot F_k(w), \\ &s.t. \sum_{k \in S} p_k = 1 \ (p_k \geq 0), \end{aligned} \tag{1}$$

where

$$F_k(w) = \frac{1}{n_k} \sum_{j_k=1}^{n_k} f_{j_k}(w; x^{(j_k)}, y^{(j_k)})$$

is the local objective function for a client k with $n_k = |\mathcal{D}_k|$ available samples. p_k is usually set as $p_k = n_k / \sum_{k \in S} n_k$ (e.g., FedAvg [10]). The FL training process usually contains multiple rounds, and a typical FL round consists of the following steps:

1) *client selection and model update*: a subset of clients S_t is selected, each of which retrieves the current global model w_t from the aggregator.
2) *local training*: each client k trains an updated model $w_t^{(k)}$ with the local dataset $\mathcal{D}_k$ and shares the model update $\Delta_t^{(k)} = w_t - w_t^{(k)}$ to the aggregator.
3) *model aggregation*: the aggregator computes the global model updates as $\Delta_t = \sum_{k \in S_t} p_k \Delta_t^{(k)}$ and update the global model as $w_{t+1} = w_t - \eta \Delta_t$, where η is the server learning rate.

FedAvg [10] is the original aggregation rule, which generates a representative global model after receiving the local models from trustworthy (i.e., benign) clients. This algorithm averages all local model weights selected based on the number of samples the clients used. FedAvg has been shown to work well when all the clients are benign, but is vulnerable to model poisoning attacks such as Byzantine attack.

B. Client Assessment

Shapley value is the most commonly used state-of-the-art method for client contribution assessment in federated learning. In game theory, a player's Shapley value is a weighted sum of marginal contributions of all possible coalitions (group of players), where marginal contribution is the difference in total rewards between the player joining and not joining the coalition [13], [25]. In a FL setting, Shapley value-based contribution [14] is defined as follows.

$$SV_t(k) = C \cdot \sum_{S \subseteq S_t \setminus \{k\}} \frac{U(M_{S \cup \{k\}}) - U(M_S)}{\binom{|S_t|-1}{S}} \tag{2}$$

where t denotes a FL round and k denotes a client, C is a constant, and $U(M_S)$ is a utility function of a model M trained on a group of clients S. The utility function can be the accuracy of the model evaluated on a validation dataset.

Another way to measure the contribution of a client is the leave-one-out (LOO) method [12], which calculates the change in model performance when the client is removed from the client group participating in the same round. Using the same notation as in Eqn. 2, LOO contribution can be expressed as

$$LOO_t(k) = U(M_{S_t}) - U(M_{S_t \setminus \{k\}}) \tag{3}$$

C. FedTruth

Inspired by truth discovery mechanisms [26]–[28], in our previous work [29], we propose a new model aggregation algorithm, namely FedTruth, which enables the aggregator to uncover the truth among all the received local model updates. The *ground-truth model update* is computed as the weighted average of all the local model updates with aggregation weights dynamically chosen based on the distances between the estimated truth and local model updates.

Suppose the aggregator receives n_t different model updates $\Delta_t^{(1)}, \cdots, \Delta_t^{(n_t)}$ in FL round t. To find the global update Δ_t^*, we formulate an optimization problem aiming at minimizing the total distance between all the model updates and the estimated global update:

$$\begin{aligned} &\min_{\Delta_t^*, \mathbf{p_t}} D(\Delta_t^*, \mathbf{p_t}) = \sum_{k=1}^{n_t} g(p_t^{(k)}) \cdot d(\Delta_t^*, \Delta_t^{(k)}) \\ &s.t. \quad \sum_{k=1}^{n_t} p_t^{(k)} = 1 \end{aligned} \tag{4}$$

where $d(\cdot)$ is the distance function and $g(\cdot)$ is a non-negative coefficient function. $p_t^{(k)}$ is the performance of the local model $\Delta_t^{(k)}$ which is calculated based on the distance.

To solve this optimization problem, FedTruth iteratively computes the estimated truth Δ_t^* and the performance values $\mathbf{p^t}$ using coordinate descent approach [30].

Updating Aggregation Weights: Once the truth Δ_t^* is fixed, FedTruth first calculates the performance of each model update $\{p_t^{(k)}\}(k = 1, \cdots, n_t)$ as

$$p_t^{(k)} = d(\Delta_t^*, \Delta_t^{(k)}) / \sum_{k'=1}^{n_t} d(\Delta_t^*, \Delta_t^{(k')}). \tag{5}$$

Then, the aggregation weights can be updated as

$$a_t^{(k)} = \frac{g(p_t^{(k)})}{\sum_{k=1}^{n_t} g(p_t^{(k)})}. \tag{6}$$

Updating the Truth: Based on the new aggregation weights $\{a_t^{(1)}, \cdots, a_t^{(n_t)}\}$, the truth can be estimated as

$$\Delta_t^* = \sum_{k=1}^{n_t} a_t^{(k)} \cdot \Delta_t^{(k)} \tag{7}$$

The global model update and aggregation weights will be updated iteratively until convergence criteria are met. It is easy to see that the longer the distance between the local model update and the estimated truth, the smaller aggregation weight will be assigned in calculating the truth. This principle can eliminate the impacts of malicious model updates and keep certain contributions from a benign outlier model update.

IV. FRECA: A Fair, Robust and Efficient Client Assessment Method

To ensure fair client evaluation in FL, we propose two key metrics: the Client Performance Metric, which measures the discrepancy between a client's model outputs and the ground truth, and the Net Contribution Metric, which quantifies the extent of each client's contribution to the global model.

A. Client Performance Evaluation Metric

In the FedTruth aggregation algorithm, larger aggregation weights will be assigned to the model updates closer to the global model update, so the aggregation weight can somehow reflect the reliability of the clients. We will use this aggregation weight (AW) to evaluate the client performance.

According to Eqn. 5, the performance of each model is calculated based on the distance between the local model and the estimated truth of the global model. For example, the distance function $d(\cdot)$ can be expressed as:

- Euclidean distance:

$$d_l(\Delta_t^*, \Delta_t^{(k)}) = ||\Delta_t^* - \Delta_t^{(k)}|| \tag{8}$$

- Angular distance:

$$d_a(\Delta_t^*, \Delta_t^{(k)}) = arccos(S_c(\Delta_t^* - \Delta_t^{(k)}))/\pi \tag{9}$$

  where S_c is the cosine similarity.

- Hybrid distance:

$$d(\Delta_t^*, \Delta_t^{(k)}) = \alpha \cdot d_l(\Delta_t^*, \Delta_t^{(k)}) + (1-\alpha) \cdot d_a(\Delta_t^*, \Delta_t^{(k)})$$

  where $\alpha \in [0, 1]$ is a combination weight.

The aggregation weight is calculated based on the regulation function $g(\cdot)$ as in Eqn. 6. In order to guarantee the convergence of FedTruth and comply with the principle of truth discovery, the authors have shown that this regulation function should be a decreasing function, monotonous and differentiable in the aggregation weight domain. Some simple but effective coefficient functions are as follows:

$$g(p_t^{(k)}) = 1/p_t^{(k)} \quad \text{or} \quad g(p_t^{(k)}) = -\log(p_t^{(k)}). \tag{10}$$

Therefore, *the client performance can be quantified as the aggregation weight of the converged iteration of FedTruth*. In our experiments, we choose the Euclidean distance function and the $g(p_t^{(k)}) = 1/p_t^{(k)}$ as the regulation function.

B. Net Contribution Evaluation Metric

If the aggregation is the simple average of all the local models, the net contribution is the same as the client performance (i.e., aggregation weights). However, in FedTruth, the aggregation weights are dynamically calculated during the estimation of the truth of the global model in each FL round. To answer the question that "should the evaluation of a client's contribution rely solely on the distance between their local model and the aggregated model, or should it be a more nuanced measure that incorporates these aggregation weights?". We propose a novel net contribution evaluation metric that counts both the aggregation weights and the model distance contributing to client contribution.

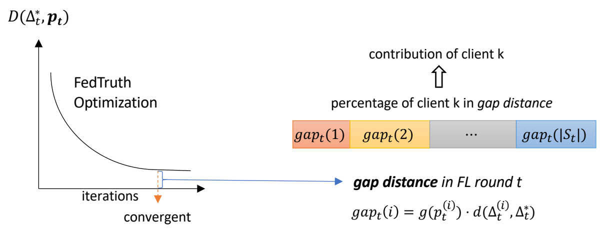
$D(\Delta_t^*, \mathbf{p}_t)$
FedTruth Optimization
iterations
convergent
contribution of client k
⇧
percentage of client k in *gap distance*
$gap_t(1)$
$gap_t(2)$
···
$gap_t(|S_t|)$
gap distance *in FL round t*
$gap_t(i) = g(p_t^{(i)}) \cdot d(\Delta_t^{(i)}, \Delta_t^*)$

Fig. 1: Gap Distance between all model updates and the estimated global model update

As shown in Fig. 1, to evaluate the net contribution of each client in each FL round t, we will first define a ***gap distance*** of this round, which measures the gap distance between all the model updates and the converged ground truth of the global model update:

$$gap_t(S_t) = \sum_{i \in S_t} gap_t(i) = \sum_{i \in S_t} g(p_t^{(i)}) \cdot d(\Delta_t^{(i)}, \Delta_t^*) \tag{11}$$

where S_t is the set of participating clients in round t, and Δ_t^* is the converged global model update.

We further compute how much percentage of a client k contributes to this gap distance by considering both the aggregation weights and the distance between its local model updates and the estimated global model update:

$$\ell_t^{(k)} = \frac{g(p_t^{(k)}) \cdot d(\Delta_t^{(k)}, \Delta_t^*)}{\sum_{i \in S_t} g(p_t^{(i)}) \cdot d(\Delta_t^{(i)}, \Delta_t^*)}, \tag{12}$$

With no access to any individual/global dataset that can be used to evaluate the accuracy of the global model, we define **net contributions based on the percentage in the gap distance**, i.e., *a client contributes more to the global model if it has less percentage in the gap distance.*

Specifically, given a set of percentages $\{\ell_t^{(k)}\}_{k \in S_t}$ where $\sum_{k \in S_t} \ell_t^{(k)} = 1$, client contributions $\{\mathcal{C}_t^{(k)}\}_{k \in S_t}$ can be calculated by solving the following linear equation:

$$\sum_{i \in S_t} \mathcal{C}_t^{(k)} = 1 \quad \text{and} \quad \frac{\ell_t^{(i)}}{\ell_t^{(k)}} = \frac{\mathcal{C}_t^{(k)}}{\mathcal{C}_t^{(i)}}, \forall i, k \in S_t \tag{13}$$

V. Experimental Results

A. Settings

We implement FL with FedTruth algorithm on 8 clients using MNIST, CIFAR-10 and FashionMNIST datasets. For each, we trained a CNN model for 10-30 FL rounds, with 10 local epochs, a batch size of 64 and a learning rate of 0.001. The models were implemented using PyTorch framework, and the experiments were run on the Google Colab platform using GPU back-end resources with 51.0GB System RAM, 15.0GB GPU RAM, and 166.8GB Disk.

For each client in each round, we computed Net Contribution (FRECA Net) (Equation 13) and Aggregation Weight (AW) (Equation 6) as client performance metric (FRECA

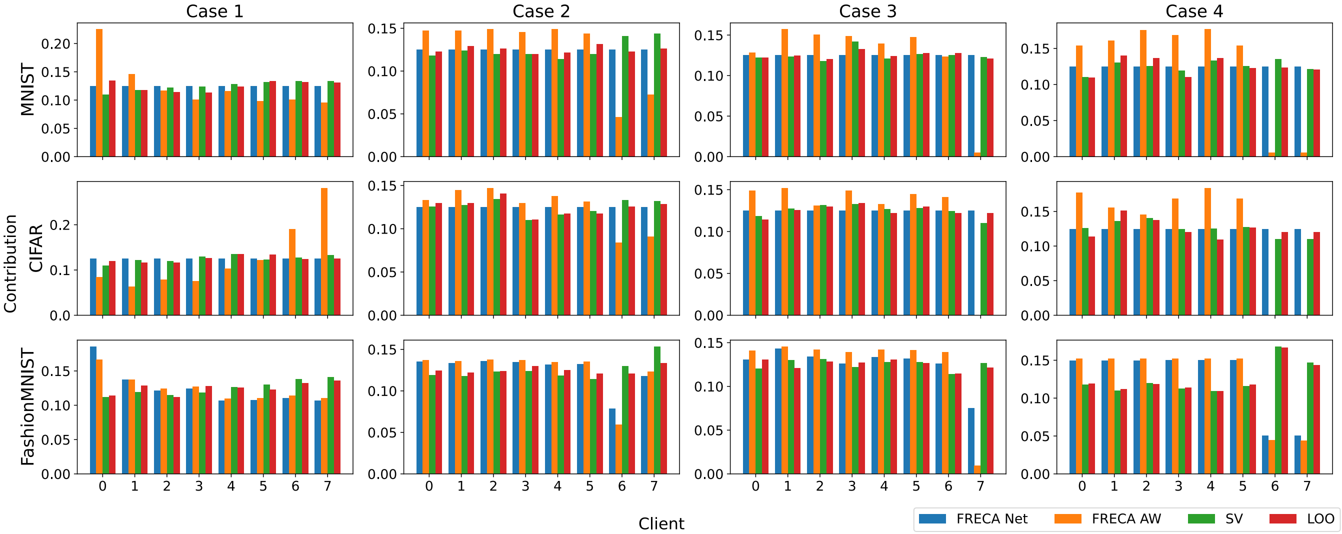

Fig. 2: Contribution for Case 1-4

AW). As baselines, we computed Shapley Value (SV) (Equation 2) and Leave-One-Out (LOO) (Equation 3), scaling the values to the range of 0 to 1 using min-max scaling and Softmax function. We averaged these metrics across rounds to obtain final contribution metrics for each client.

We present our client assessment results for 4 cases regarding client data distribution and Byzantine attack scenarios:

- Case 1: non-iid setting, each client having a different number of labels in their data
- Case 2: non-iid setting, each client having 1 or 2 labels in their data
- Case 3: iid setting, 1 attacker among clients
- Case 4: iid setting, 2 attackers among clients

Non-iid means each client does not have data samples for all labels, iid means each client have samples for all labels and the samples are distributed uniformly across all labels.

B. Case 1: non-iid setting, each client with different # labels

The number of labels is 1, 2, 3, 4, 6, 8, 9, 10 for Clients 0 to 7, i.e., Client 0 has data samples with 1 label, Client 1 has 2 labels, etc.. The total data size is the same for each client. Fig. 2 shows 4 contribution metrics for each client with 3 datasets. Notice that, in Case 1, FRECA Net (blue) is similar to SV (green) or LOO (red) in most cases, indicating our method computes the same assessment as SV/LOO in much less time (see Fig. 3). FRECA AW (orange) mostly aligns with the net contribution with a few exceptions which can be partially explained by the composition of different labeled data within all 8 clients. For example, Client 0 in MNIST case provides data samples with one label that takes up about 70% of total samples for this label, which may have led to a higher aggregation weight.

C. Case 2: non-iid setting, each client with 1-2 labels

6 clients have data with 1 label, 2 clients have data with 2 labels, the sample size per label being the same across clients. Labels are assigned to clients in a non-replacement manner such that all 10 labels are covered. Case 2 in Fig. 2 shows roughly similar values for net contribution, SV and LOO indicating similar contributions from different clients. The AW values are relatively low for the last 2 clients, recognizing the outliers among the clients. This outlier-identifying ability of AW can be utilized to detect malicious clients as detailed in Case 3.

D. Case 3: iid setting, 1 attacker among clients

Each client has the same sample size, distributed uniformly across all labels. One client (Client 7) commits a attack to the global model by amplifying its local model parameters by a factor (e.g., 10). It is obvious from the 3rd column of Fig. 2 that this malicious client is successfully identified by FRECA AW assigning near-zero values to this client. This significantly small AW diluted the impact of the malicious amplified model, resulting in a net contribution similar to other clients. The SV and LOO, on the other hand, were not able to detect the malicious client.

E. Case 4: iid setting, 2 attackers among clients

Client data settings are the same as in Case 3, but this time, there are two attackers: Client 6 and 7. Similar to Case 3, we see a stark difference in AW between attackers and normal clients, successfully identifying the attacks with MNIST and CIFAR. With FashionMNIST dataset, SV and LOO assign higher values to the attackers which can be disastrous, while on the contrary, FRECA Net and AW both give higher values to non-attackers and much lower values to attackers. With FRECA Net and AW combined, we can be sure that the last two clients have significantly lower contribution than others.

F. Time Efficiency

The time taken to compute both FRECA Net and FRECA AW, SV, and LOO is depicted in Fig. 3, for 10 FL rounds with 8 clients. The time complexity is $O(2^n)$ for SV and $O(n)$

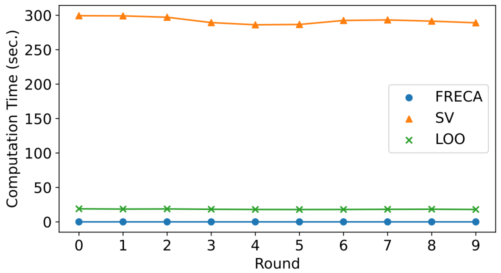

Fig. 3: Computation Time Comparison

for LOO and FRECA, with n being the number of clients. As expected, the computation time for SV is the highest, averaging to 292 s/round due to the exhaustive evaluation of aggregated models on the validation dataset for all possible combinations of clients. The average time for LOO and FRECA is 18s and 1.5s, respectively. With the same theoretical time complexity $O(n)$, LOO is still much slower than our method because of the evaluation on the validation dataset.

VI. Conclusion

We introduce a novel method FRECA to quantify client contributions in FL, employing FedTruth framework to estimate the global model's ground truth update, which can balance contributions from all clients while filtering out impacts from malicious ones. This approach is robust against Byzantine attacks as it incorporates a Byzantine-resilient aggregation algorithm, and efficient as it operates solely on local model updates and requires no validation datasets. We show through our experimental results that FRECA can accurately and efficiently quantify client contributions in a robust manner.

References

[1] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[2] A. Hammoud, H. Otrok, A. Mourad, and Z. Dziong, "On demand fog federations for horizontal federated learning in iov," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3062–3075, 2022.

[3] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021.

[4] L. Chen, P. Koutris, and A. Kumar, "Towards model-based pricing for machine learning in a data marketplace," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 1535–1552.

[5] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[6] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, "Collaborative machine learning with incentive-aware model rewards," in *International conference on machine learning*. PMLR, 2020, pp. 8927–8936.

[7] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, S. Zeitouni *et al.*, "Flame: Taming backdoors in federated learning," in *Proceedings of 31st USENIX Security Symposium*, 2022, p. to appear.

[8] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[9] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016, pp. 508–519.

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

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

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

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

[14] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2577–2586.

[15] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, "A principled approach to data valuation for federated learning," *Federated Learning: Privacy and Incentive*, pp. 153–167, 2020.

[16] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, "Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–21, 2022.

[17] B. Yan, B. Liu, L. Wang, Y. Zhou, Z. Liang, M. Liu, and C.-Z. Xu, "Fedcm: A real-time contribution measurement method for participants in federated learning," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[18] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1078–1088, 2021.

[19] X. Cao, M. Fang, J. Liu, and N. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," in *Proceedings of NDSS*, 2021.

[20] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. J. Spanos, and D. Song, "Efficient task-specific data valuation for nearest neighbor algorithms," *arXiv preprint arXiv:1908.08619*, 2019.

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

[22] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*. PMLR, 2019, pp. 634–643.

[23] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 118–128.

[24] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[25] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International conference on machine learning*. PMLR, 2019, pp. 2242–2251.

[26] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, "Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 1986–1999, 2016.

[27] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[28] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, "On the discovery of evolving truth," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 675–684.

[29] S. C. Ebron Jr and K. Yang, "Fedtruth: Byzantine-robust and backdoor-resilient federated learning framework," *arXiv preprint arXiv:2311.10248*, 2023.

[30] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.