Robust Client-level Contribution Assessment in Horizontal Federated Learning

Project Student: Vincent Cloutier
Supervisor: Zhuan Shi
Professor: Boi Faltings
EPFL Artificial Intelligence Lab (LIA)

ABSTRACT

Federated Learning (FL) enables participants to collaborate on model training without sharing raw data, but its effectiveness depends heavily on the quality of each participant's local dataset. We propose the Federated Banzhaf Value (FBV), a robust and scalable data valuation framework that accurately identifies both high- and low-quality clients. Our evaluation shows that the FBV strongly correlates with the true Shapley value, remains more stable under severe data perturbations than leading Shapley value- and influence-based estimators, and is significantly more efficient than SV estimators. Leveraging these properties, our FBV-based debugging mechanism selectively removes problematic clients, thereby improving test accuracy and lowering test loss across heterogeneous, mislabeled, and poisoned scenarios—regardless of the severity of data corruption or the number of bad clients.

1 INTRODUCTION

Federated Learning (FL) has recently become a popular machine learning paradigm as it enables several clients (e.g., devices or organizations) to jointly train a model in a decentralized manner without sharing their local data [1]. However, the data these clients hold can vary substantially in quality and distribution. Specifically, it has been noted that some participants may possess heterogeneous or even adversarial data, which can negatively affect the performance of the resulting model [2–4]. Since FL is a privacy-preserving learning paradigm, identifying these bad clients is inherently nontrivial, which poses a significant obstacle to ensuring a trustworthy model.

This challenge has motivated research on participant valuation in FL in an attempt to solve two important questions:

1. *How can we measure a client's contribution to the global model?*
2. *How can we leverage these valuations to improve model performance?*

The current approaches to federated data valuation and model debugging provide only a limited solution and are often computationally intensive [5]. Some of these approaches reduce the costs by estimating the contributions of clients using influence functions [6–10]; however, such approaches might not be able to capture the interdependence between the participants' datasets or may not be able to counteract

sophisticated adversarial threats. Thus, the SV has become the gold standard for the valuation of clients and extensive research has focused on its approximation [11–16], as computing it directly requires 2^N retrainings. Nevertheless, some of these estimators provide unstable client valuations and, therefore, may not be suitable for real-world FL systems [5].

Ensuring lower run-to-run variance in these valuations is, therefore, our primary concern, as poor robustness can significantly undermine the effectiveness of any debugging framework that relies on them, leading to suboptimal model performance and reducing client incentives to participate in the FL process. The main idea behind our approach is to use the Banzhaf Value (BV), which has been demonstrated to be exponentially more robust than the Shapley Value in the centralized learning context [17].

First, we define the Federated Banzhaf Value (FBV) by extending the BV to the federated setting. Then, we introduce a linear-time estimator for the FBV. In our experiments, the FBV proves more robust than both an influence-function-based alternative [6, 18] and Federated Shapley [13] with average Spearman Rank Correlations of 0.92, 0.35, and 0.80 respectively, on CIFAR-10. Further, the FBV achieves this robustness improvement while running orders of magnitude faster than Federated Shapley [13], taking an average time of 83.26s vs. 28390.73s on CIFAR-10. Based on the FBV, we also propose a model debugging framework to improve the performance of the model even in the presence of poor partial data.

1. *Data Valuation*: Our estimator achieves a Pearson's Correlation Coefficient (PCC) of 0.9738 on CIFAR-10 and 0.9385 on Fashion-MNIST when compared to the true SV, demonstrating its accuracy.
2. *FL Debugging*: Our FBV-based debugging framework consistently improves model performance across heterogeneous, mislabeled, and poisoned data scenarios. In the vast majority of examined settings, retraining after excluding low-quality clients identified by the FBV leads to higher test accuracy and lower test loss than the initial model. Notably, our debugging framework remains effective even when per-client data corruption is minimal. Furthermore, unlike other debugging frameworks that assume a majority of clients are either good or bad, [6, 7, 9, 11], our approach does not rely on such assumptions and is effective in both scenarios, illustrating its generalizability.

In the remainder of this paper, we present the technical details of our proposed data valuation method and debugging framework and evaluate them thoroughly across a wide range of settings.

2 Related Works

2.1 Federated Learning

Federated Learning (FL) is a machine learning field that allows for the training of models in a distributed environment where multiple clients participate in the learning process without having to transfer raw data to a central server [1]. This learning framework allows for data privacy and protection as well and may enhance the model's generalization ability [1]. Traditional FL algorithms, like FedAvg, leverage a central server to average the model parameters that each client has computed in order to develop a global model that is refined over a number of communication rounds [19]. Variants and extensions of traditional FL aim to solve the problems of data heterogeneity, limited communication, and varying device resources [20–24].

2.2 Robustness in FL

Although FL offers the possibility of achieving privacy-preserving and efficient model training, it should be noted that FL models are usually less accurate than their centrally trained counterparts, especially when

the data distribution is non-IID [2]. Also, FL frameworks are susceptible to attacks from malicious clients' [25].

These attacks can be classified based on the point of manipulation: data or model. Data poisoning attacks are aimed at changing the local training data and can be implemented, for instance, by adding noise with the intention of affecting the global model performance [26]. They may also involve the creation of backdoors in some samples with the aim of embedding hidden behaviours in a specific set of inputs while preserving the accuracy of the model on most inputs [27]. Model poisoning attacks, however, are those that seek to tamper with local updates with the aim of impeding the global model's convergence or worsening it's performance [3].

Most defenses depend on robust aggregators (AGRs) to counteract the effects of malicious updates, as described in [25]. Recently suggested AGRs that employ clipping and filtering heuristics are more effective than early AGRs. However, there are still ways for adversarial attacks to compromise these defenses since advanced techniques can be developed to overcome them [4].

2.3 Shapley Value in FL

Shapley Value (SV)-based methods have been introduced as a useful tool for data valuation in the context of centralized learning [28]. However, in the case of federated learning (FL), a participant's contribution may not be fixed: late client contributions often have less influence on the final model than early contributions. To incorporate this temporal aspect, recent work [12, 13] adapts the classical SV to the FL setting, where the value of participants' data is computed based on their contribution as training evolves.

However, calculating the SV in FL is still challenging since it entails retraining the model for every coalition of clients. To overcome this challenge, several approximation techniques have been put forward in [11–16]. All these approximations can be divided into two main groups: those that try to avoid model retraining entirely with gradient-based approximations and those that only seek to reduce the number of model retrainings. The latter may also involve gradient-based estimators [14] or, more commonly, randomized permutations of the training order such as [13, 15, 16], thus estimating each participant's expected contribution. Although random sampling may not capture some specific interactions between the participants [5], these approaches have been shown to be effective in practice. For instance, the Federated Shapley Value (FSV) has been shown to provide better results in identifying data quality compared to a simpler approach like Leave-One-Out [13].

2.4 Influence Functions in FL

There has been an increasing interest in using influence function-based approaches [29] for data valuation in federated settings as an alternative to the SV-based methods. Instead of computing complex SVs, these methods employ second-order optimization techniques in order to determine the importance of each client or sample in the overall model. For example, Fed-Influence [10] calculates the sum of sample-level influence scores to estimate the impact of deleting training examples on the ultimate model. Still, this approach is based on Hessian-based computations, which can be both computationally and communicationally expensive. To reduce these costs, subsequent methods [6–9] that employ Hessian-vector products for computing influence values more efficiently have taken hold.

However, directly computing sample-level influence for all data points can still be intractable in large-scale settings. To address this, hierarchical methods first identify negatively influential clients before computing per-sample influence within these subsets [6, 7, 9]. These methods assume that most clients are not corrupted and that model training stabilizes over later training rounds to use training logs for bad client identification. For example, [6] measures each client C_k's deviation from the global model

as: $D_k = \frac{1}{N(k)} \sum_{t=T/2}^{T} s_t^k \|\theta_t^k - \theta_t\|$, where $s_t^k = 1$ if C_k participates in round t, and 0 otherwise, and $N(k)$ is the number of such rounds. Then, if $\frac{D_k}{\text{median}(\{D_l | l \in [K]\})} > \delta_T$, they consider C_k to be negatively influential. After bad clients are identified, the server collaborates with each negatively influential client to compute the influence $I_f(z_{k,i})$ of each training sample $z_{k,i}$ on the final loss.

Thus, while certain influence-based frameworks require less computation than SV-based methods, they often rely on assumptions—such as the majority of clients being well-behaved [6, 7, 9]—and may be less robust overall [30].

3 Preliminaries and Problem Definition

In this work, we consider the problem of quantifying client participation in FL training in a robust manner, all the while without compromising privacy. We first formalize a canonical federated learning system: Here we have a server S and a number of clients $C = \{1, 2, \ldots, N\}$ where each client k has a local dataset $D_k = \{z_{k,1}, z_{k,2}, \ldots, z_{k,m_k}\}$ of size m_k. Now, given a loss function $l(\cdot; \theta)$, we define the client loss as $L_k(\theta) = \frac{1}{m_k} \sum_{i=1}^{m_k} l(z_{k,i}; \theta)$ where θ is the parameter vector of the model. The server collaborates with the clients with the goal of learning the global model $\hat{\theta}$:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{k \sim \text{Uniform}\{1,\ldots,N\}}[L_k(\theta)] = \arg\min_{\theta} \frac{1}{N} \sum_{k=1}^{N} L_k(\theta),$$

where Uniform$\{1, \ldots, N\}$ is the probability distribution over the clients as we are using Fed-Average [19] as our underlying FL aggregation method.

To solve this minimization problem, the server first sets the initial values of the global model's weights to be θ_0. Then, during each global communication round, $t \in \{1, 2, \ldots, T\}$, the server chooses a set of clients C_t to participate in the training process and sends them the current model parameters θ_{t-1}. Then, every client k in C_t simultaneously trains their local model as follows:

$$\theta_t^k = \theta_{t-1} - \delta_t^k,$$

where $\delta_t^k = \eta \nabla L_k(\theta_{t-1})$ is the client update. Each client then uploads θ_t^k to the server which aggregates the updates with:

$$\theta_t = \frac{1}{|C_t|} \sum_{k \in C_t} \theta_t^k$$

or equivalently, we can write $\theta_t = \theta_{t-1} - G_t$ where G_t is the global update $G_t = \frac{1}{|C_t|} \sum_{k \in C_t} \delta_t^k$.

We assume that the server and all clients are semi-honest (they follow the training process but are curious about others' data), which aligns with common practice [6, 8, 31, 32]. Furthermore, we assume the server holds a reliable validation/test dataset that is error-free, while certain clients may possess corrupted data (e.g., mislabeled samples).

Our primary objective is to present an efficient model-debugging framework for FL that identifies high-quality clients and prioritizes them to enhance model performance in the presence of this corrupted data. Also, our design should not introduce any unnecessary privacy risks while measuring contributions, as privacy preservation is the most attractive feature of FL. Finally, and most importantly, our design should ensure that client valuations remain stable across runs and are resilient to sample noise, thereby

guaranteeing fairness and preventing the valuation mechanism from being unduly influenced by small perturbations.

4 METHODOLOGY

In this section, we first briefly review the basic concepts of data valuation and the properties a data valuation method should meet. Then, we extend these concepts to the federated setting and explain how to estimate the utility and compute the Banzhaf Value in FL. Finally, we describe how to use such valuations for model debugging in the context of FL.

4.1 CLIENT-LEVEL DATA VALUATION

Let $C = \{1, \ldots, N\}$ denote a set of clients. Data valuation aims to quantify how useful each client in C is to model training. To accomplish this aim, we first define a model-dependent utility function $\nu : 2^N \to \mathbb{R}$, which gives a score for any subset $S \subseteq C$ which is the performance (for instance validation loss) of a model trained on S. We denote the valuation of a client $i \in C$ with respect to ν by $\phi(i, \nu)$. The most common data valuation schemes in the literature, seek to preserve some or all of the following properties:

1. *Symmetry*: If the clients i and j contribute equally to the utility of every coalition, then i and j should have the same valuation. More specifically, if for all $S \subseteq C \setminus \{i, j\}$, we have $\nu(S \cup \{i\}) = \nu(S \cup \{j\}$, then $\phi\left(i, \nu\right) = \phi\left(j, \nu\right)$.
2. *Null Player*: If a client i does contribute to the utility of any coalition, then the valuation of i should be zero. More specifically, if for all $S \subseteq C \setminus \{i\}$, we have $\nu\left(S \cup \{i\}\right) = \nu\left(S\right)$ then $\phi\left(i, \nu\right) = 0$.
3. *Linearity*: The sum of the valuations under multiple utility functions should equal the valuation under the sum of those functions. That is, $\phi(i, \nu_1) + \phi(i, \nu_2) = \phi(i, \nu_1 + \nu_2)$ for all $i \in C$.
4. *Efficiency*: The value of the whole client set should be equal to the sum of the values of all the clients in that set. That is, $\nu\left(I\right) \leq \sum_{i \in C} \phi\left(i, \nu\right)$.

The only data valuation scheme that satisfies all four aforementioned properties is the Shapley Value (SV) [33]. As a result of this unicity, the Shapley Value is a widely used data valuation scheme. It is defined as the average marginal contribution of a client i to all possible coalitions:

$$\phi^{SV}(i, \nu) = \frac{1}{N} \sum_{S \subseteq C} \binom{N-1}{|S|}^{-1} \left[\nu(S) - \nu(S \setminus \{i\})\right]. \tag{1}$$

An alternative to the Shapley value is the Banzhaf Value (BV), which computes the average marginal contribution of i but gives equal importance to every coalition.

$$\phi^{BV}(i, \nu) = \frac{1}{2^{N-1}} \sum_{S \subseteq C} \left[\nu(S) - \nu(S \setminus \{i\})\right] \tag{2}$$

It should be noted that while not adjusting for the size of S implies the Banzhaf Value does not satisfy efficiency, the Banzhaf Value does satisfy symmetry, null player, and linearity. Also, since many ML data valuation applications focus on ranking rather than raw scores, the efficiency property is often not essential [17]. Moreover, the Banzhaf Value has been shown to provide significantly more ranking stability than the Shapley Value in centralized learning [17]. It is for this reason that it is the focus of our framework.

4.2 FEDERATED BANZHAF VALUE

We propose the Federated Banzhaf Value (FBV) to extend the Banzhaf Value to the FL setting. Consider a federated learning process divided into T rounds. In round t, a set of clients C_t participates. The Federated Banzhaf Value (FBV) of a client i is defined as:

$$\phi^{FBV}(i,\nu) = \sum_{t=1}^{T} \phi_t^{FBV}(i,\nu), \tag{3}$$

where

$$\phi_t^{FBV}(i,\nu) = \frac{1}{2^{|C_t|-1}} \sum_{S \subseteq C_t} \left[\nu(S) - \nu(S \setminus \{i\})\right]. \tag{4}$$

Here, $\phi_{\text{FBV}}^t(i,\nu)$ represents the contribution of client i in round t. The function ν assesses the utility associated with each subset of participating clients in that round. Although this definition directly aligns with the definition of the classical Banzhaf Value, directly computing it is computationally expensive. Evaluating $\nu(S \cup \{i\})$ and $\nu(S)$ for all subsets S would retraining the model 2^N times, making this approach infeasible for large N.

4.2.1 APPROXIMATION

To make the FBV computable in practice, we must derive an approximation that avoids the exponential complexity of evaluating every subset's utility and, preferably, avoids any additional model retraining. First, we define the utility function as $\nu(S) = L(\theta(\emptyset)) - L(\theta(S))$, where $L(\theta)$ is the loss on the server-held validation set, and $\theta(S)$ are the parameters obtained by training only on the clients in S. Thus, the marginal contribution of a client k is:

$$\nu(S) - \nu(S \setminus \{k\}) = L(\theta(S \setminus \{k\})) - L(\theta(S)). \tag{5}$$

Intuitively, the marginal contribution measures how much worse the model's performance would be if we had not included client k in the model training. This quantity is easier to approximate than the full utility function because changes in model performance stem from changes in model parameters, and so, we can approximate the difference in loss using gradient information available from the training process. By using a first-order Taylor expansion of $L(\theta)$ around the global parameters θ_{t-1} from the previous round, we have:

$$L(\theta(S \setminus \{k\})) - L(\theta(S)) \approx \nabla L(\theta_{t-1}) \Delta G_t^{-k}, \tag{6}$$

where $\nabla L(\theta_{t-1})$ is the gradient of the validation loss at θ_{t-1}, and ΔG_t^{-k} represents the change in the global update at round t if client k were removed. Now the problem becomes about estimating ΔG_t^{-k}, as direct computation of ΔG_t^{-k} would still require model retraining. First recall from Section 3 that the global update at round t is $G_t = \frac{1}{|C_t|} \sum_{j \in C_t} \delta_t^j$, where δ_t^j is the local update from client j. Thus we see that removing client k changes this to $G_t^{-k} = \frac{1}{|C_t|-1} \sum_{j \in C_t \setminus \{k\}} \delta_t^j$, and so $\Delta G_t^{-k} = G_t^{-k} - G_t$.

Now, we use the approach from [11] and related work in federated learning (FL) that use gradient-based approximations to estimate a client's influence [8, 10]. The key idea is that removing a client k is equivalent to upweighting k's contribution to the global update throughout the training process by $-\frac{1}{n}$, since this would reduce k's effective contribution from $\frac{1}{n}$ to 0. Specifically, we have:

$$G_t^{-k} = \frac{1}{n} \sum_{i=1}^{n} \bar{\delta}_t^i - \frac{1}{n} \bar{\delta}_t^k, \tag{7}$$

where $\bar{\delta}_t^i$ represents the local update from participant i after upweighting k by $-\frac{1}{n}$. Using the chain rule and a first-order approximation, we get:

$$G_t^{-k} = \eta \frac{1}{n} \sum_{i=1}^{n} \nabla L_i(\bar{\theta}_{t-1}) - \frac{1}{n} \eta \nabla L_k(\bar{\theta}_{t-1}), \tag{8}$$

where $\bar{\theta}_{t-1}$ is the global model after applying the upweighting of k.

Expanding this result and assuming the loss function is twice differentiable, we employ a second-order approximation with the Hessian $H_{\theta_{t-1}}$ at θ_{t-1}:

$$\Delta G_t^{-k} = -\frac{1}{|C_t|} \delta_t^k + \eta \Omega_t^k, \tag{9}$$

where $\Omega_t^k = H_{\theta_{t-1}} \left(\sum_{j=1}^{t-1} \Delta G_j^{-k} \right)$, and η is a learning rate factor.

In summary, the change in gradients due to removing k can be approximated by considering the immediate reduction in k's contribution and a Hessian-corrected term that accounts for how previous updates would have changed without k.

Combining (5), (6), and (9), we can approximate the FBV for client k during a given round t as:

$$\begin{aligned}
\hat{\phi}_t^{FBV}(i) &= \frac{1}{2^{|C_t|-1}} \sum_{S \subseteq C_t \setminus \{k\}} [-\nabla L(\theta_{t-1}) \Delta G_t^{-k}] \\
&= \frac{1}{2^{|C_t|-1}} 2^{|C_t|-1} [-\nabla L(\theta_{t-1}) \Delta G_t^{-k}] \\
&= -\nabla L(\theta_{t-1}) \Delta G_t^{-k} \\
&= \frac{1}{|C_t|} \delta_t^k \nabla L(\theta_{t-1}) - \eta \Omega_t^k \nabla L(\theta_{t-1}),
\end{aligned} \tag{10}$$

and so following from (3), we can approximate the cumulative FBV for client k as:

$$\hat{\phi}^{FBV}(i) = \sum_{t=1}^{T} \hat{\phi}_t^{FBV}(i). \tag{11}$$

With the need for exponential retraining eliminated by our approximation, we now turn to its associated complexity and privacy considerations.

4.2.2 Complexity and Privacy Analysis

Computing the first-order term $\delta_t^k \nabla L(\theta_{t-1})$ for all clients and rounds requires $O(pT|C_t|)$ operations at the server, where p is the number of model parameters and T is the total number of rounds. No additional client computation beyond what is needed for standard FL is required, as the local updates δ_t^k are collected at the server, and the server already computes $\nabla L(\theta_{t-1})$ from its validation set.

In contrast, directly computing the second-order term $\Omega_t^k \nabla L(\theta_{t-1})$ for all clients and rounds involves forming and multiplying by the Hessian $H_{\theta_{t-1}}$, a $p \times p$ matrix. Naïvely doing so would cost the server $O(p^2 T|C_t|)$ operations, which is prohibitively expensive for large models, and would also require additional client computation and communication of order $O(p^2 T)$ since they must compute and transmit second-order gradient information at each round.

To mitigate this computational burden, Hessian-Vector Products (HVPs) are used to avoid the explicit construction of the Hessian. This approach reduces the overhead but still involves higher complexity than the first-order case and can become substantial for large models.

If the complexity still proves too high, we can omit the second-order terms entirely. In doing so, the total FBV formula remains unchanged, but we replace (10) with the following approximation:

$$\hat{\phi}_t^{FBV}(i) = \frac{1}{|C_t|}\delta_{t,i}\nabla L(\theta_{t-1}), \tag{12}$$

which provides a lower-cost way to measure the per-round FBV.

It should also be noted that the first-order estimation adheres to the same privacy standard as the standard FL setup, since it relies solely on the training-essential communication of local model updates and global parameters. In contrast, the second-order estimation, while still not sharing any raw training data, introduces the need to exchange additional intermediate gradient or Hessian information. This non-essential information, potentially reveals more details about local data distributions and therefore offers a weaker privacy guarantee than standard FL.

In practice, these approximations allow us to balance accuracy, complexity, and privacy. When second-order accuracy is not critical, the simpler first-order approximation provides a practical and privacy-preserving solution. However, if bad clients are harder to detect and the infrastructure can support the additional complexity, the second-order information can be incorporated with an acceptable relaxation of privacy constraints.

4.3 Federated Model Debugging

With the FBV in hand, we can build a FL debugging framework to reduce the impact of problematic clients on model performance. The key insight is that clients with low FBVs are those that either fail to improve the global model or actively degrade it. By identifying these low-FBV clients, the server can take corrective measures to enhance overall model performance.

We propose a simple two-step federated model debugging framework grounded in the FBV (Fig 1):

1. *Bad Client Identification:* After T rounds of standard federated training, the server computes the FBV for each participating client using the approximations discussed in Section 4.2. Clients with low total FBVs are deemed "bad" clients.

2. *FBV-Driven Model Updating:* Once bad clients are identified, the server can take action in subsequent rounds of training. Specifically, it may remove bad clients entirely or, if complete removal is not desirable, adjust their selection probabilities. This involves increasing the probability of selecting high-FBV clients while decreasing that of low-FBV clients.

   After these adjustments, the server restarts federated training using the refined participant set or reweighted probabilities.

Algorithm 1 outlines the debugging process used in this paper. We begin with a standard FL procedure (e.g., FedAvg) and augment it with FBV computation as well as client removal.

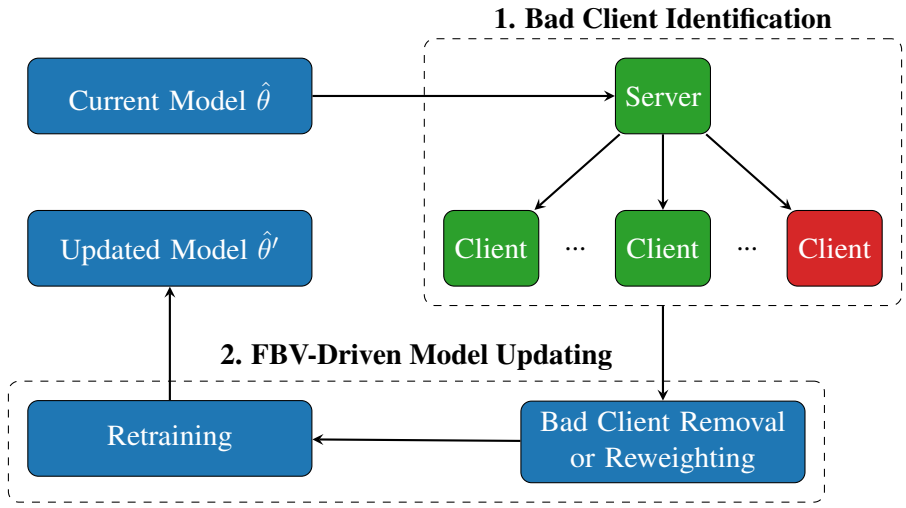

FIGURE 1
Overview of FBV-Debug.

Algorithm 1: FBV-Debug

Input: C: clients, F: fraction of clients selected per round, T: number of rounds
Output: Refined global model $\hat{\theta}'$ and per-client FBV estimates $(\hat{\phi}^{FBV}(1), \ldots, \hat{\phi}^{FBV}(N))$
Initialize global model θ_0;
Initialize $\hat{\phi}^{FBV}(i, \nu) \leftarrow 0$ for all $i \in \{1, \ldots, N\}$;
for $t = 1, 2, \ldots, T$ **do**
  $m \leftarrow \max(\lfloor F \cdot |C| \rfloor, 1)$;
  Sample a set $C_t \subseteq C = \{1, \ldots, N\}$ of size m;
  Compute $\nabla L(\theta_{t-1})$ using the server's validation set;
  for *each* $k \in C_t$ *in parallel* **do**
    $\theta_t^k \leftarrow \texttt{LocalUpdate}(\theta_{t-1})$;
    Approximate $\hat{\phi}_t^{FBV}(k)$ using (12) or (10);
    Update $\hat{\phi}^{FBV}(k) \leftarrow \hat{\phi}^{FBV}(k) + \hat{\phi}_t^{FBV}(k)$;
  end
  $\theta_t \leftarrow \frac{1}{m} \sum_{k \in C_t} \theta_t^k$;
end
Identify bad clients based on $\hat{\phi}^{FBV}(i, \nu)$ (e.g., those with negative FBV);
Restart training without the bad clients;
return $\hat{\theta}'$, $(\hat{\phi}^{FBV}(1), \ldots, \hat{\phi}^{FBV}(N))$

It should be noted that Algorithm 1 is only the approach that we are proposing in this paper and that the framework behind our algorithm is very flexible. For instance, instead of retraining, we could dynamically modify the client selection process (as in [8, 11]). Furthermore, the framework could use thresholding or clustering strategies to identify bad clients instead of simply selecting those with negative valuations. Still, as demonstrated in Section 5.4, even this simple approach enhances model performance across a wide range of scenarios, emphasizing the ease of interpreting the FBV for data valuation in FL.

5 EVALUATION

In this section, we assess the efficacy of FBV-Debug. First, we compare the FBV against the classical Shapley value. Then we compare the robustness of the FBV relative to existing federated data valuation

schemes, namely, the Federated Shapley Value (FSV) [13] and an influence-function-based approach [6]. Finally, we show that employing the FBV for identifying and avoiding the low-quality clients during training enhances both model performance and loss.

5.1 Shared Experimental Configuration

5.1.1 Datasets

In order to guarantee our results are widely applicable we perform the experiments on two popular datasets.

1. *CIFAR-10* [34]: This dataset comprises of 60,000 colour images with a resolution of 32×32. These images are also divided into 10 classes with $50,000$ training and $10,000$ test images.
2. *Fashion-MNIST* [35]: This dataset comprises of 70,000 black and white images with a resolution of 28×28. These images are also divided into 10 classes with $60,000$ training and $10,000$ test images.

In each experiment, we set aside 10% of the training set as a validation set for the central server. The rest of the training images are distributed amongst N clients in one of the following ways:

1. *Homogenous:* Each client receives an equally sized, random subset of the remaining training data.
2. *Heterogenous:* First, each client receives an equally sized, random subset of the remaining training data. Then, we select m random clients and restrict their data to k random classes. This simulates how client datasets may not be representative of the dataset as a whole.
3. *Mislabeled:* First, each client receives an equally sized, random subset of the remaining training data. Then, we select m random clients and randomly mislabel $k\%$ of their samples. This simulates label noise, reflecting real-world situations [36] where data could be inconsistently annotated due to human error or adversarial behavior.
4. *Poisoned:* First, each client receives an equally sized, random subset of the remaining training data. Then we select m random clients and alter $k\%$ of their samples into poison samples [26]. More specifically, we use a weighted combination of some base image and a target image to generate a noisy sample z like so $z = \gamma b + (1 - \gamma)t$, and then annotate it with the label of the target image as described in [6]. We use $\gamma = 0.9$ for CIFAR-10 and $\gamma = 0.5$ for Fashion-MNIST. This simulates how certain FL scenarios may be vulnerable to malicious modifications of client data.

These distribution methods enable us to evaluate how well the FBV scheme identifies low-quality clients and test the robustness of our approach under realistic conditions.

5.1.2 Training Protocol

For all our experiments, we employ standard convolutional neural network (CNN) architectures (LeNet & ResNet) [37, 38]. All models are trained using the Federated Averaging (FedAvg) protocol [19]. We conduct training for a maximum of 50 global rounds. However, training may terminate earlier if the model achieves predefined accuracy thresholds—specifically, 80% test accuracy on CIFAR-10 or 85% test accuracy on Fashion-MNIST. Unless otherwise specified, in each global round, we randomly select 60% of the available clients without replacement to participate in that round. Each selected client computes their local update using a batch size of 128 and ADAM for optimization [39]. Finally, model performance is assessed using test accuracy and cross-entropy loss on the respective test sets.

5.2 Correlation with True Shapley Value

5.2.1 Experiment Setup

In this experiment, we aim to validate the Federated Banzhaf Value (FBV) against the true Shapley value, a well-established ground truth. Computing the true Shapley value requires retraining the model 2^N times—once for every subset of the N clients. For computational feasibility, we set $N = 5$.

Each retraining follows the procedure described in Section 5.1.2. However, as some subsets represent a small portion of the total training set, each global round draws from every client that is present in that subset to mitigate the variance introduced to the computed Shapley values. We then compute the true Shapley values using Equation 1, defining the utility function $\nu(S)$ as the test accuracy of the model trained on subset S of the training data. Also, when we train on the subset that includes all N clients, we compute the FBV as described in Section 4.2.

We repeat this experiment three times for each of the data settings described in Section 5.1.1. In the bad data settings, 20% of the clients are "bad," having either 60% corrupted data or, in the heterogeneous scenario, data restricted to four distinct categories.

Finally, we measure the Pearson Correlation Coefficient (PCC) between the FBV and the true Shapley values across all twelve runs. This multiple-run evaluation ensures that our findings are statistically significant and not impacted by random fluctuations in model training.

5.2.2 Results

The Pearson Correlation Coefficient (PCC) between the FBV and the true Shapley values is 0.9738 on CIFAR-10, indicating a nearly perfect linear relationship. On Fashion-MNIST, the PCC is 0.9385, also demonstrating a very strong positive correlation.

5.2.3 Analysis

The slightly lower correlation scores for Fashion-MNIST may be attributable to the dataset's more easily distinguishable class boundaries, which enable the model to converge more rapidly on small subsets. As a result, subtle differences in client contributions may become more difficult to discern. Despite this, both correlations are sufficiently high to suggest that the FBV provides accurate data valuations.

5.3 Robustness

5.3.1 Experiment Setup

To assess the robustness of the Federated Banzhaf Value (FBV), we examine how the rankings associated with the FBV, the Federated Shapley Value (FSV) [13], and an influence-function-based method [6] change across multiple runs.

Specifically, we perform five runs of federated training with $N = 10$ clients following the procedure described in Section 5.1.2. However, to isolate the stochastic effects of model training, we keep all client properties constant and select the same clients in each round. In every run, we compute the FBV (using both the first- and second-order approximations from Equations 12 and 10), the FSV, and the influence-function-based values. We then construct a median "true" ranking across the five runs and compute the Spearman Rank Correlation (SRC) between this median ranking and each individual run's ranking, and average these SRCs to assess overall stability.

We repeat this experiment once for each of the data settings described in Section 5.1.1. In the bad data settings, 20% of the clients are "bad," having either 60% corrupted data or, in the heterogeneous scenario,

data restricted to four distinct categories.

5.3.2 Results

Figures 2 and 3 summarize both the average SRCs and runtimes for CIFAR-10 and Fashion-MNIST, respectively.

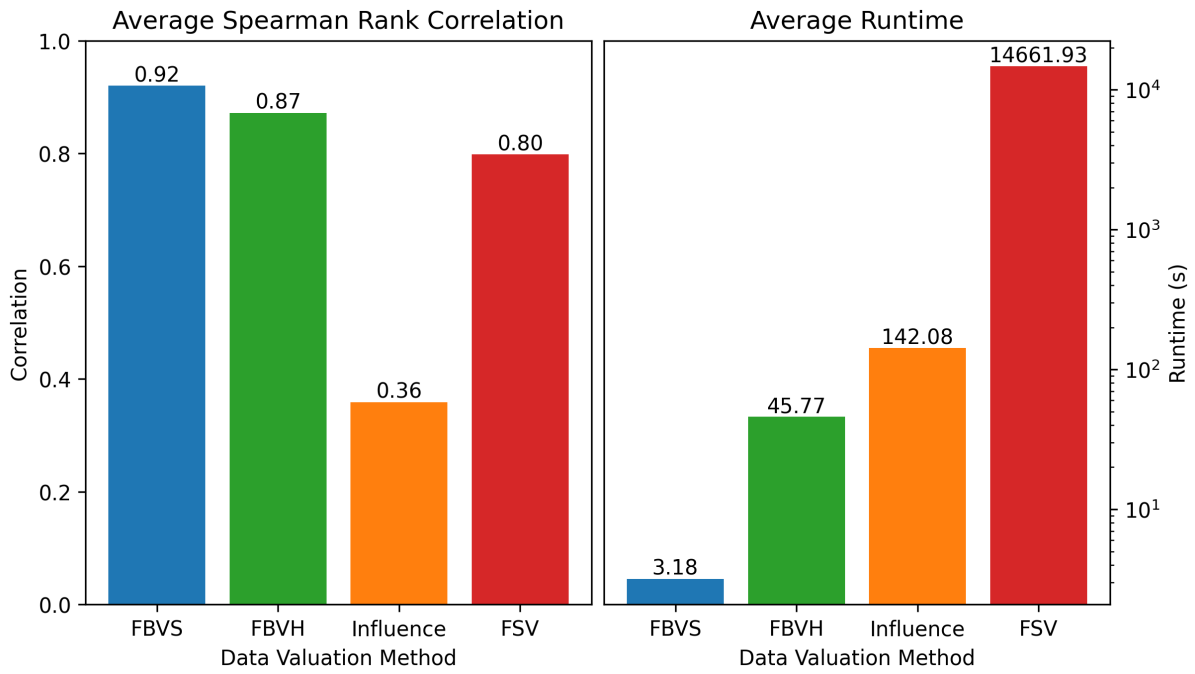

Figure 2
Average SRC and Runtime on CIFAR-10 Across All Settings From Section 5.1.1.

5.3.3 Analysis

First, both the first-order (FBVS) and the second-order (FBVH) Federated Banzhaf Values preserve a high average Spearman Rank Correlation (SRC), implying that rankings are quite stable across multiple runs. The slightly higher variance we observe in the FBVH likely stems from its reliance on Hessian information, which makes it sensitive to the curvature of the loss. In the case of Fashion-MNIST, however, this second-order information seems to exert less downward pressure on robustness because of the dataset's more easily distinguishable class boundaries.

Second, while the score of the Federated Shapley Value (FSV) is quite stable, it remains lower than the FBV methods on average. Also, it is worth noting that its runtime is exponentially larger, as shown in Figure 2.

Finally, while the influence function offers a compromise in terms of runtime, it has the lowest average SRC. This lower robustness is likely because the influence function is the only tested method that works at the sample level. Computing sample level valuations necessarily requires a higher sensitivity to perturbations made to individual samples, which likely contributes to its reduced robustness.

These findings highlight the practical advantages of the FBV approach in real-world FL scenarios, where efficiency and robustness are critical. Crucially, the stability and consistency of FBV-derived rankings mean that clients cannot credibly repudiate the fairness of their valuations based on stochastic fluctuations.

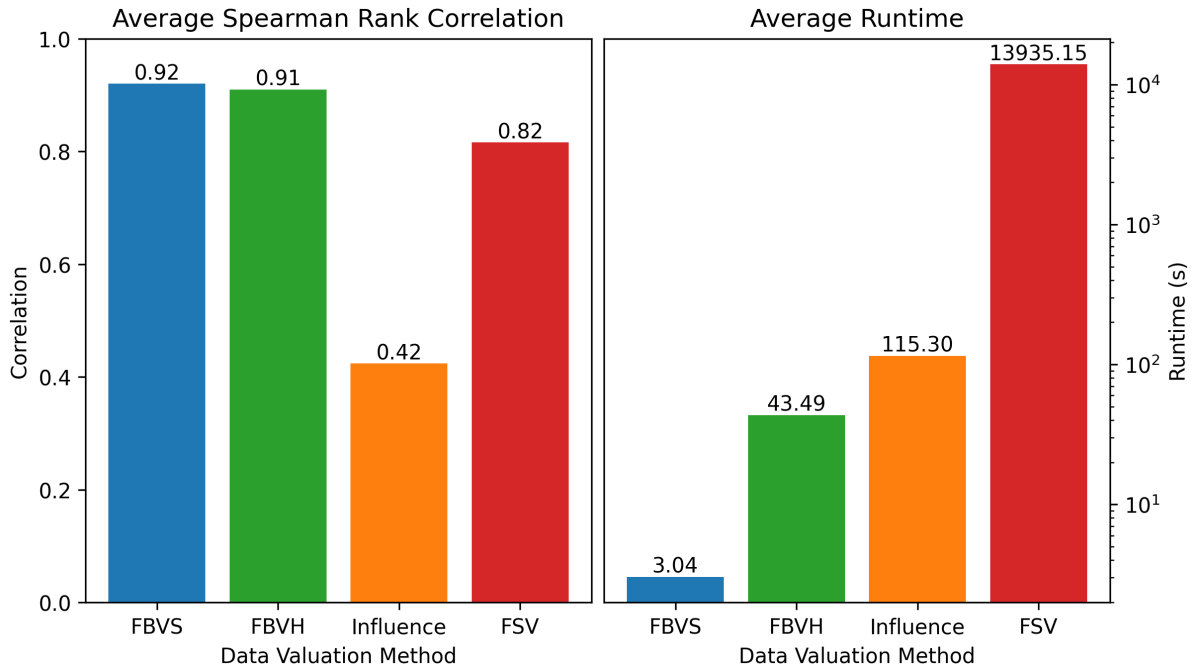

FIGURE 3
Average SRC and Runtime on Fashion-MNIST Across All Settings From Section 5.1.1.

5.4 RETRAINING

5.4.1 EXPERIMENT SETUP

To assess the effectiveness of our debugging framework in improving the final global model performance, we conduct a series of retraining experiments following the protocol in Section 5.1.2. We start by randomly initializing a global model and performing standard federated training with $N = 10$ clients, evaluating each client's FBV as described in Section 4.2. After this initial training, we record the global model's test accuracy and loss.

We then repeat the training process from scratch under the same configuration, but this time excluding low-quality clients—those identified with negative FBVs as described in Section 4.3. After this second training phase, we measure the global model's test accuracy and loss again, comparing these results against the baseline values obtained before debugging.

We repeat this retraining procedure for each experimental setting described in Section 5.1.1. In the heterogeneous case, we vary both the number of bad clients $m \in 2, 4, 6, 8$ and the number of categories per bad client $k \in 4, 6, 8$ to simulate increasingly skewed data scenarios. For the mislabeled and poisoned settings, we similarly vary m and adjust the proportion of bad data per bad client $k \in 20\%, 40\%, 60\%$. In the poisoned Fashion-MNIST setting, we further increase these proportions ($k = 70\%, 80\%, 90\%$) because the model did not experience a noticeable performance degradation at lower levels.

5.4.2 RESULTS

Figures 4, 5, and 6 summarize the retraining results on CIFAR-10 for the heterogeneous, mislabeled, and poisoned settings, respectively. Similarly, Figures 7, 8, and 9 summarize the retraining results on Fashion-MNIST.

5.4.3 Analysis

In every setting, the test accuracy of the retrained model is better than the initial accuracy, proving the effectiveness of this FBV-based debugging framework. While test loss generally decreases as well, there is one specific mislabeled scenario (20% mislabeled data with 8 bad clients in CIFAR-10) where loss does not improve. In this edge case, the test loss is already low (compared to the other settings) as each client brings in too little mislabeled data to raise the initial loss. Thus, when we completely remove 8 out of 10 clients, it harms the model more than it helps the model. This scenario shows that if a large majority of clients are only slightly harmful, a reweighting framework may be more effective than a complete removal framework.

Despite this exception, the overall pattern is clear: even when accuracy improvements are modest, substantial reductions in test loss indicate that our debugging approach yields a more stable and accurate predictive distribution. By demonstrating its effectiveness across various types and levels of data degradation, we show its potential to enhance model trustworthiness and reliability. These findings highlight the practical value of our FBV-based approach in real-world FL scenarios without guaranteed data quality.

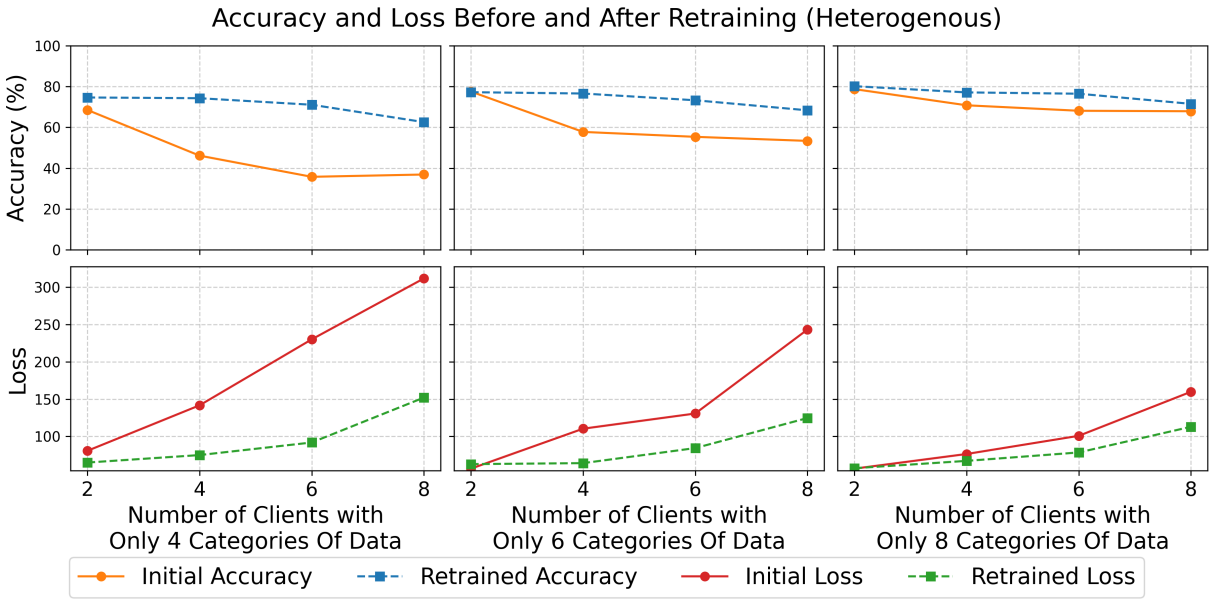

FIGURE 4
Retraining Performance on Heterogenous CIFAR-10 with Varying Numbers of Bad Clients and Categories.

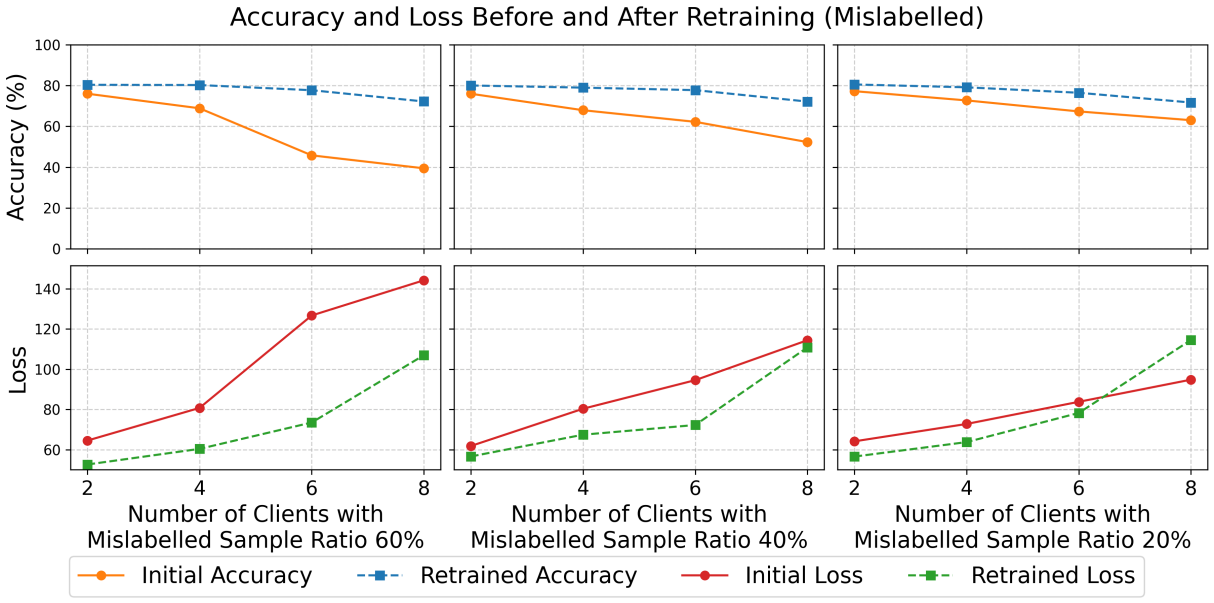

FIGURE 5
Retraining Performance on Mislabeled CIFAR-10 with Varying Numbers of Bad Clients and Mislabeled Samples.

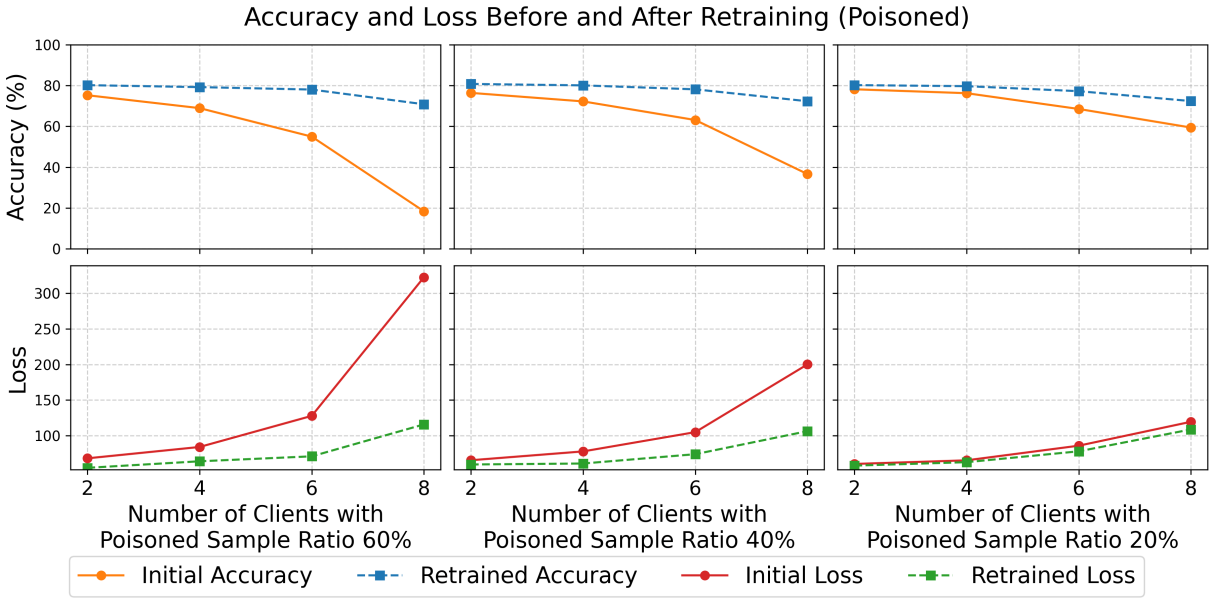

FIGURE 6
Retraining Performance on Poisoned CIFAR-10 with Varying Numbers of Bad Clients and Poisoned Samples.

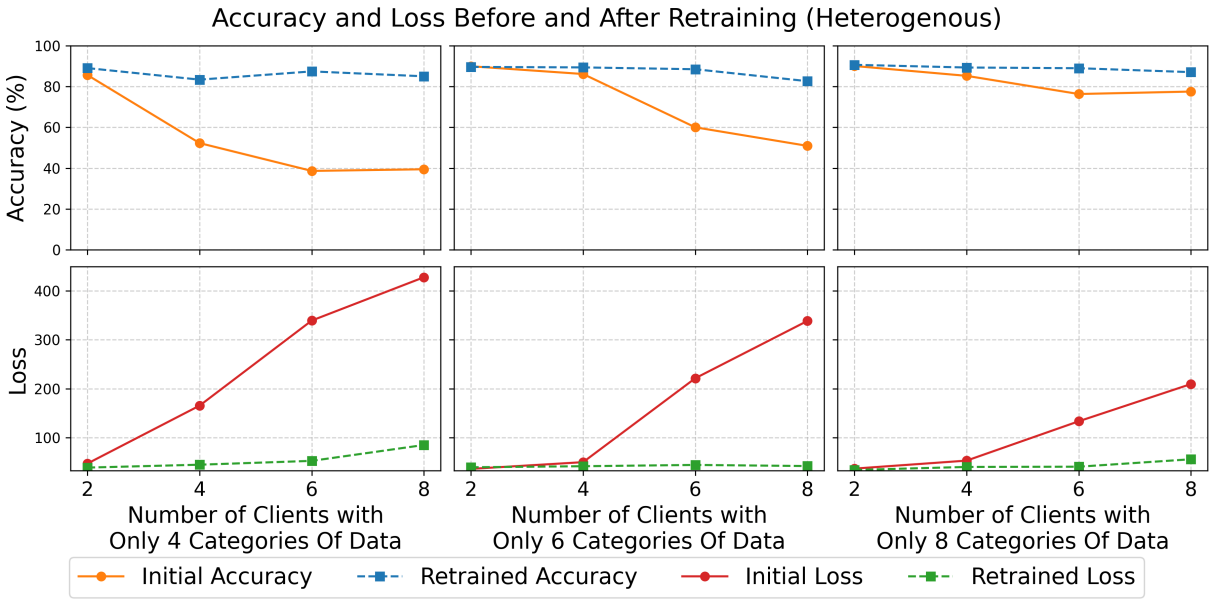

FIGURE 7
Retraining Performance on Heterogenous Fashion-MNIST with Varying Numbers of Bad Clients and Categories.

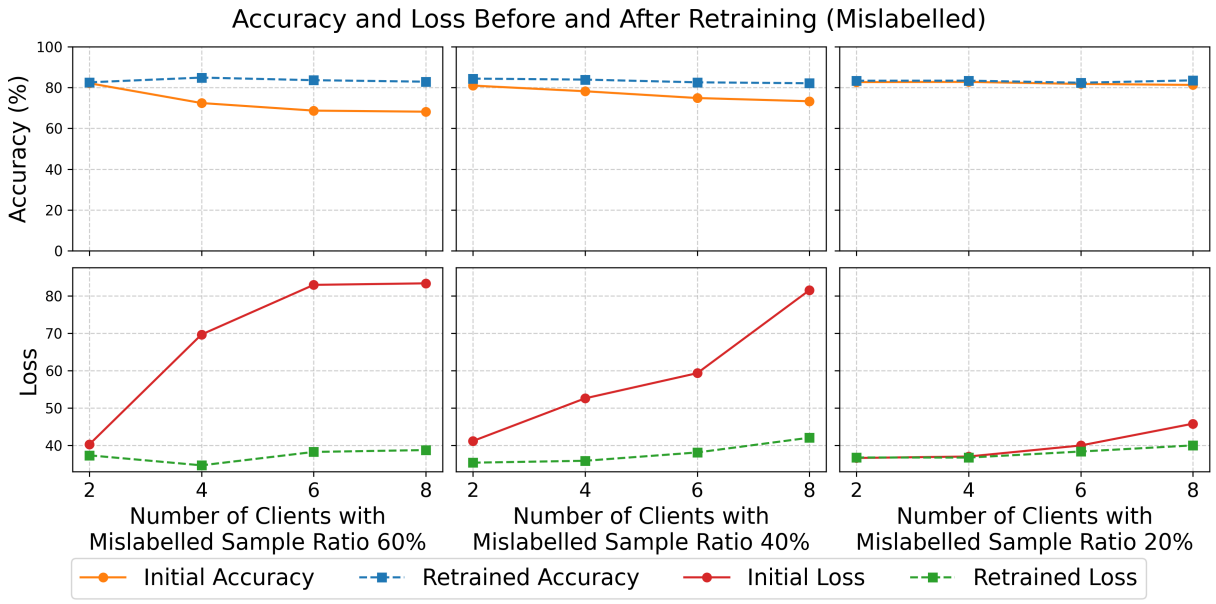

FIGURE 8
Retraining Performance on Mislabeled Fashion-MNIST with Varying Numbers of Bad Clients and Mislabeled Samples.

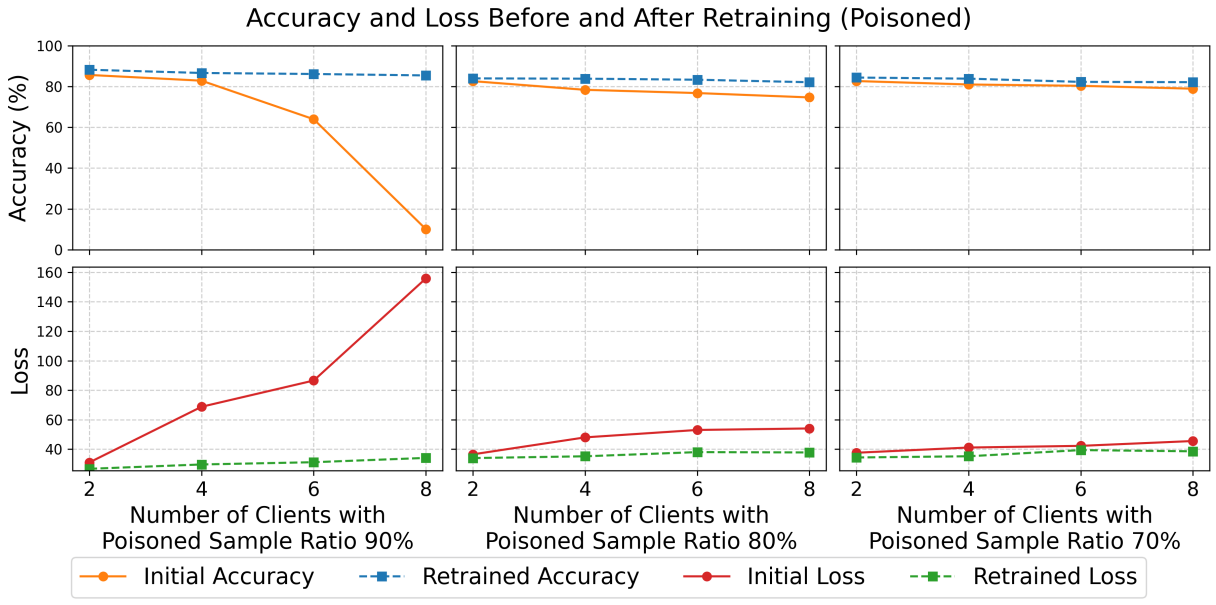

FIGURE 9
Retraining Performance on Poisoned Fashion-MNIST with Varying Numbers of Bad Clients and Poisoned Samples.

6 CONCLUSION

We proposed the Federated Banzhaf Value (FBV), a new data valuation approach for FL, and demonstrated that it can identify and handle low-quality or malicious clients effectively. Based on the FBV, the debugging approach presented in this paper is capable of effectively identifying faulty participants and filtering them out, thus mitigating their detrimental effects on the model performance in heterogenous and adversarial data environments. In addition, the experimental results demonstrate that the FBV has a high correlation with the SV and is also more computationally efficient and less sensitive to data perturbations compared to the influence-based methods and the SV estimators.

6.1 LIMITATIONS AND FUTURE WORK

While our results are promising, several potential avenues for future research could build upon our findings. First, our current experiments were limited to basic datasets such as CIFAR-10 and Fashion-MNIST as well as particular neural architectures–LeNet and ResNet. Expanding our experiments to a larger number of datasets and more complicated models would ensure our findings are widely applicable. Second, the identified edge case in Section 5.4.3 indicates that further exploration is needed to determine when reweighting problematic clients is preferable to complete removal. Investigating adaptive strategies that balance client inclusion and exclusion depending on the degree and nature of data corruption would be a productive direction for future work.

REFERENCES

[1] Tian Li et al. 'Federated Learning: Challenges, Methods, and Future Directions'. In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60. DOI: 10.1109/MSP.2020.2975749.

[2] Hangyu Zhu et al. *Federated Learning on Non-IID Data: A Survey*. 2021. arXiv: 2106.06843 [cs.LG]. URL: https://arxiv.org/abs/2106.06843.

[3] Arjun Nitin Bhagoji et al. *Analyzing Federated Learning through an Adversarial Lens*. 2019. arXiv: 1811.12470 [cs.LG]. URL: https://arxiv.org/abs/1811.12470.

[4] Yuxin Yang et al. *A Learning-Based Attack Framework to Break SOTA Poisoning Defenses in Federated Learning*. 2024. arXiv: 2407.15267 [cs.CR]. URL: https://arxiv.org/abs/2407.15267.

[5] Chao Ren et al. *Advances and Open Challenges in Federated Foundation Models*. 2024. arXiv: 2404.15381 [cs.LG]. URL: https://arxiv.org/abs/2404.15381.

[6] Anran Li et al. 'Efficient Federated-Learning Model Debugging'. In: *IEEE 37th International Conference on Data Engineering (ICDE)*. 2021, pp. 372–383.

[7] Anran Li et al. 'Sample-level Data Selection for Federated Learning'. In: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. 2021, pp. 1–10. DOI: 10.1109/INFOCOM42981.2021.9488723.

[8] Anran Li et al. 'FedCSS: Joint Client-and-Sample Selection for Hard Sample-Aware Noise-Robust Federated Learning'. In: *Proc. ACM Manag. Data* 1.3 (Nov. 2023). DOI: 10.1145/3617332. URL: https://doi.org/10.1145/3617332.

[9] Anran Li et al. 'Privacy-Preserving Efficient Federated-Learning Model Debugging'. In: *IEEE Transactions on Parallel and Distributed Systems* 33.10 (2022), pp. 2291–2303. DOI: 10.1109/TPDS.2021.3137321.

[10] Yihao Xue et al. *Toward Understanding the Influence of Individual Clients in Federated Learning*. 2021. arXiv: 2012.10936 [cs.LG]. URL: https://arxiv.org/abs/2012.10936.

[11] Junhao Wang et al. 'Efficient Participant Contribution Evaluation for Horizontal and Vertical Federated Learning'. In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 2022, pp. 911–923. DOI: 10.1109/ICDE53745.2022.00073.

[12] Nurbek Tastan et al. *Redefining Contributions: Shapley-Driven Federated Learning*. 2024. arXiv: 2406.00569 [cs.LG]. URL: https://arxiv.org/abs/2406.00569.

[13] Tianhao Wang et al. 'A Principled Approach to Data Valuation for Federated Learning'. In: *Federated Learning: Privacy and Incentive*. 2020, pp. 153–167.

[14] Tianshu Song, Yongxin Tong and Shuyue Wei. 'Profit Allocation for Federated Learning'. In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 2577–2586. DOI: 10.1109/BigData47090.2019.9006327.

[15] Zelei Liu et al. *GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning*. 2021. arXiv: 2109.02053 [cs.AI]. URL: https://arxiv.org/abs/2109.02053.

[16] Khaoula Otmani, Rachid El-Azouzi and Vincent Labatut. 'FedSV: Byzantine-Robust Federated Learning via Shapley Value'. In: *ICC 2024 - IEEE International Conference on Communications*. 2024, pp. 4620–4625. DOI: 10.1109/ICC51166.2024.10622175.

[17] Jiachen T. Wang and Ruoxi Jia. 'Data Banzhaf: A Robust Data Valuation Framework for Machine Learning'. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. 2023, pp. 6388–6421.

[18] Juhan Bae et al. *If Influence Functions are the Answer, Then What is the Question?* 2022. arXiv: 2209.05364 [cs.LG]. URL: https://arxiv.org/abs/2209.05364.

[19] H. Brendan McMahan et al. *Communication-Efficient Learning of Deep Networks from Decentralized Data*. 2023. arXiv: 1602.05629 [cs.LG]. URL: https://arxiv.org/abs/1602.05629.

[20] Krishna Pillutla, Sham M. Kakade and Zaid Harchaoui. 'Robust Aggregation for Federated Learning'. In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 1142–1154. DOI: 10.1109/TSP.2022.3153135.

[21] Samuel Horvath et al. *FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout*. 2022. arXiv: 2102.13451 [cs.LG]. URL: https://arxiv.org/abs/2102.13451.

[22] Amirhossein Reisizadeh et al. *FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization*. 2020. arXiv: 1909.13014 [cs.LG]. URL: https://arxiv.org/abs/1909.13014.

[23] Enmao Diao, Jie Ding and Vahid Tarokh. *HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients*. 2021. arXiv: 2010.01264 [cs.LG]. URL: https://arxiv.org/abs/2010.01264.

[24] Sai Praneeth Karimireddy et al. 'SCAFFOLD: Stochastic Controlled Averaging for Federated Learning'. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 5132–5143. URL: https://proceedings.mlr.press/v119/karimireddy20a.html.

[25] Peng Liu, Xiangru Xu and Wen Wang. 'Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives'. In: *Cybersecurity* 5 (2022). URL: https://api.semanticscholar.org/CorpusID:246447197.

[26] Ali Shafahi et al. *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks*. 2018. arXiv: 1804.00792 [cs.LG]. URL: https://arxiv.org/abs/1804.00792.

[27] Thuy Dung Nguyen et al. *Backdoor Attacks and Defenses in Federated Learning: Survey, Challenges and Future Research Directions*. 2023. arXiv: 2303.02213 [cs.LG]. URL: https://arxiv.org/abs/2303.02213.

[28] Amirata Ghorbani and James Zou. *Data Shapley: Equitable Valuation of Data for Machine Learning*. 2019. arXiv: 1904.02868 [stat.ML]. URL: https://arxiv.org/abs/1904.02868.

[29] Pang Wei Koh and Percy Liang. *Understanding Black-box Predictions via Influence Functions*. 2020. arXiv: 1703.04730 [stat.ML]. URL: https://arxiv.org/abs/1703.04730.

[30] Samyadeep Basu, Philip Pope and Soheil Feizi. *Influence Functions in Deep Learning Are Fragile*. 2021. arXiv: 2006.14651 [cs.LG]. URL: https://arxiv.org/abs/2006.14651.

[31] Stacey Truex et al. 'A Hybrid Approach to Privacy-Preserving Federated Learning'. In: *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–11. ISBN: 9781450368339. DOI: 10.1145/3338501.3357370. URL: https://doi.org/10.1145/3338501.3357370.

[32] Yijing Li et al. 'Privacy-Preserved Federated Learning for Autonomous Driving'. In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2022), pp. 8423–8434. DOI: 10.1109/TITS.2021.3081560.

[33] Abraham Neyman. 'Uniqueness of the Shapley value'. In: *Games and Economic Behavior* 1.1 (1989), pp. 116–118. DOI: https://doi.org/10.1016/0899-8256(89)90008-0. URL: https://www.sciencedirect.com/science/article/pii/0899825689900080.

[34] Alex Krizhevsky, Vinod Nair and Geoffrey Hinton. 'CIFAR-10 (Canadian Institute for Advanced Research)'. In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[35] Han Xiao, Kashif Rasul and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747 [cs.LG]. URL: https://arxiv.org/abs/1708.07747.

[36] Curtis G. Northcutt, Anish Athalye and Jonas Mueller. *Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks*. 2021. arXiv: 2103.14749 [stat.ML]. URL: https://arxiv.org/abs/2103.14749.

[37] Y. Lecun et al. 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[38] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[39] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[40] Ashwin Jadhav. *GitHub - Federated-Learning-PyTorch: Implementation of Communication-Efficient Learning of Deep Networks from Decentralized Data*. [Accessed 17-12-2024]. 2020. URL: https://github.com/AshwinRJ/Federated-Learning-PyTorch.

[41] Alston Lo. *GitHub - torch-influence: A simple PyTorch implementation of influence functions*. [Accessed 17-12-2024]. 2023. URL: https://github.com/alstonlo/torch-influence.

[42] Modassir Afzal. *GitHub - ResNet-9: Designed a smaller architecture implemented from the paper "Deep Residual Learning for Image Recognition" and achieved 93.65% accuracy*. https://github.com/Moddy2024/ResNet-9?tab=readme-ov-file. [Accessed 17-12-2024]. 2023.