ShapleyFL: Robust Federated Learning Based on Shapley Value

Qiheng Sun
Zhejiang University
qiheng_sun@zju.edu.cn

Xiang Li
Zhejiang University
lixiangzx@zju.edu.cn

Jiayao Zhang
Zhejiang University
jiayaozhang@zju.edu.cn

Li Xiong
Emory University
lxiong@emory.edu

Weiran Liu
Alibaba Group
weiran.lwr@alibaba-inc.com

Jinfei Liu*
Zhejiang University
ZJU-Hangzhou Global Scientific and
Technological Innovation Center
jinfeiliu@zju.edu.cn

Zhan Qin
Zhejiang University
qinzhan@zju.edu.cn

Kui Ren
Zhejiang University
kuiren@zju.edu.cn

ABSTRACT

Federated Learning (FL) allows clients to form a consortium to train a global model under the orchestration of a central server while keeping data on the local client without sharing it, thus mitigating data privacy issues. However, training a robust global model is challenging since the local data is invisible to the server. The local data of clients are naturally heterogeneous, while some clients can use corrupted data or send malicious updates to interfere with the training process artificially. Meanwhile, communication and computation costs are inevitable challenges in designing a practical FL algorithm. In this paper, to improve the robustness of FL, we propose a Shapley value-inspired adaptive weighting mechanism, which regards the FL training as sequential cooperative games and adjusts clients' weights according to their contributions. We also develop a client sampling strategy based on importance sampling, which can reduce the communication cost by optimizing the variance of the global updates according to the weights of clients. Furthermore, to diminish the computation cost of the server, we propose a weight calculation method by estimating differences between the Shapley value of clients. Our experimental results on several real data sets demonstrate the effectiveness of our approaches.

CCS CONCEPTS

• **Computing methodologies** → *Distributed algorithms.*

KEYWORDS

Federated Learning; Shapley Value

ACM Reference Format:
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 (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599500

1 INTRODUCTION

Federated Learning (FL) [25, 32] is a distributed machine learning paradigm that enables local clients to collectively train a central model under the coordination of a server. Specifically, the server organizes the collaborative training process through model parameter interaction while the raw data of each client is stored locally. Therefore, it preserves the privacy of clients by preventing the exposure of raw data. FL has shown extensive applications in many fields, such as medical care [37], finance [27], and data markets [44]. For example, for medical care, several hospitals can collectively train a disease classifier without sharing the raw data of patients in the FL paradigm, thus protecting their privacy.

Motivation. Despite the promising progress in mitigating problems of privacy [1, 47], Non-IID data [24], and communication cost [28], FL still suffers from a great vulnerability in robustness. For example, the data of different clients naturally tends to be heterogeneous [34]; some malicious clients may use corrupted data or even send noised parameters to manipulate the model [35]. The performance of the central model can be dramatically degraded in such complex yet practical scenarios. Thus, robustness is a crucial desideratum for an FL algorithm. However, it is challenging to develop a robust FL algorithm since the server has little prior knowledge of clients due to privacy concerns.

Standard FL treats all clients indiscriminately, and weights clients either uniformly or proportionally to the size of their local data sets when aggregating the local model updates, which lacks robustness since both the value of clients and data are unequal in reality. To address the problem, some works consider that valuable clients should be more similar to each other, such as Krum [2] and coordinate-wise median-based algorithms [50]. These works require all clients to participate in each round for their statistical error rate guarantee, incurring intractable communication cost.

*the corresponding author.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599500

Majority-based and geometric median-based robust aggregation methods [3, 15, 18, 35, 46] have been further proposed, which allow partial client participation. However, they cannot maintain effectiveness when malicious clients account for a large proportion and submit similar gradients. In addition, some valuable clients whose local data sets and gradients differ from others may be filtered out by majority-based and similarity-based methods. Therefore, the above methods often lead to suboptimal performance since they do not consider a fair valuation of the influence of each client.

Shapley Value (SV) is a concept to measure each player's contribution in the cooperative game theory [39]. It has been proven as the unique way that satisfies four desired properties for contribution allocation: balance, symmetry, additivity, and zero element. Specifically, the Shapley value of one player is the weighted average of all its marginal contributions, which is the utility differences between player sets with and without the player. Shapley value captures all possible cooperation scenarios of each player and hence can distinguish valuable players from malicious players. Naturally, Shapley value can be employed to evaluate the contribution of clients. The Shapley value of clients can be computed on the server by evaluating the utilities based on gradients without accessing the raw data. In addition, Shapley value is a model-agnostic solution, i.e., it can provide a fair evaluation for each client regardless of the type of training models and attack methods. *To this end, can we design a robust FL algorithm based on Shapley value of the clients?*

Challenges. Despite the appealing properties of Shapley value, there are several challenges in adopting Shapley value to evaluate the contributions of clients and further design a robust FL algorithm. First, the clients participating in the training process vary from round to round, making it hard to use Shapley value to evaluate the relative contributions between all clients. Second, the communication cost is one of the bottlenecks in implementing FL [13, 20]. It is necessary to ensure the convergence and reduce the communication cost when dynamically adjusting the weights of clients. Third, computing the exact Shapley value for the clients under FL requires retraining the models for different subsets of clients and the computation is known to be a #P-hard problem due to the enumeration of subsets [10]. When there are a large number of clients, a huge computation cost on the server is inevitable, even with Shapley value approximation methods. It is challenging to design an efficient Shapley value-based weighting method.

Contributions. To improve the robustness of FL, we design an adaptive federated learning algorithm, which adapts the weights of clients for aggregating the local model updates according to their historical contributions. Each client's weight is set to be proportional to its surrogate federated Shapley value (Definition 4.3), which combines the marginal contributions of selected clients in all rounds and provides an effective approximation of the Shapley values by the clients' overall contributions to the ongoing training process. In addition, we give a thorough analysis of the convergence and stability of our proposed method.

We also put effort into improving the convergence and reducing the communication cost of adaptive federated learning. Our analysis shows that the weights of clients significantly impact the variance of the global update estimator. To reduce the variance of the estimator of global update, the clients with higher weights should be selected with higher probability. Thus, we develop a client sampling method according to importance sampling, which adjusts client selection probability dynamically in the training process.

Furthermore, we propose a more efficient Shapley value approximation method to reduce the computation cost in adaptive federated learning. Inspired by the fact that the sample of differences between Shapley values of clients has smaller variances than that of Shapley values themselves, we estimate the differences between Shapley values of clients first and then derive surrogate federated Shapley values from the differences, rather than directly calculating surrogate federated Shapley values as existing inefficient sampling approaches do.

The main novelty of the paper is that we treat the FL training as sequential cooperative games for the first time to enhance its robustness, and the server can use weighting and sampling strategies based on Shapley value of clients. Concretely, we summarize our contribution as follows.

- **Significance or broad impact**. We focus on addressing the fundamental problem in federated learning: improving the robustness to heterogeneous data challenges and poisoning attacks.
- **Proposed method**. We propose an adaptive weighting method based on the surrogate federated Shapley value for robustness. We further develop a client-importance sampling strategy to enhance communication efficiency and a surrogate federated Shapley value approximation method to save computation cost.
- **Theoretical guarantee**. We provide a formal convergence analysis showing that the proposed algorithm can achieve the same convergence rate as state-of-the-art. Besides, we give the stability analysis of adaptive federated learning on the upper bound of the loss change in consecutive rounds.
- **Experimental validation**. We perform a comprehensive evaluation of a range of vulnerable scenarios on different real datasets. The results demonstrate that our proposed algorithm significantly improves the robustness of federated learning over the baselines in different settings, e.g., achieving 13.8% improvement in accuracy compared to the best-performing baseline algorithm RFL [35] on realistic healthcare dataset Fed-ISIC2019 [11].

2 RELATED WORK

In this section, we discuss related work on robust federated learning and Shapley value.

2.1 Federated Learning

Federated learning has attracted widespread attention since it allows collaborative model optimization without exposing the local data. Many efforts [20, 25, 51] have been made on various aspects of FL, such as user privacy requirements, Non-IID data challenges, and communication issues.

Ensuring robustness of FL faces many complicated scenarios, such as 1) heterogeneous/imbalanced data, 2) irrelevant data, and 3) poisoned data or model updates. Training on heterogeneous and imbalanced data among clients, which is pervasive in real-world applications, can result in biased models [38]. Thus, Shuai et al. [40] proposed a personalized federated learning framework that can simultaneously address local and global data imbalance. Dishonest clients may participate in model training using irrelevant data for improper remuneration [33], which can lead to catastrophic failure of models. To address the issue, Cho et al. [8] proposed biased client selection, which allows clients with higher local loss to have more opportunities to participate in the training. Even worse, malicious clients may use corrupted data to attack the training process [23, 49]. To address the problem, Han and Zhang [16] assumed that some clients are trusted so that they can evaluate the credibility of other clients by predicting results on trusted items. Tahmasebian et al. [41] proposed a robust aggregation algorithm inspired by the truth inference methods via incorporating the client's reliability in the aggregation against the poisoning attacks. However, these methods fail to deal with collusion attacks since they cannot evaluate the contributions of individual clients fairly. Besides, they are designed for specific robust issues, which are not flexible for all scenarios.

There are many works focusing on the communication optimization of FL. Luo et al. [30, 31] proposed an efficient FL algorithm that optimally chooses control variables, e.g., communication interval, to reduce the communication rounds and an adaptive client sampling method to tackle system and statistic heterogeneity to minimize wall-clock time. Recently, several adaptive optimization approaches in FL have been proposed to improve convergence, such as the decomposition of ordinary differential equations of corresponding centralized optimizers [19]. However, how to reduce the communication cost while ensuring the robustness of FL is not considered by the above methods.

2.2 Shapley Value

Shapley value is widely used in the game theory and computer science fields due to its pragmatic properties. Recently, many works have focused on evaluating the value of clients in FL based on Shapley value [12, 29]. The intuition is to encourage clients to participate in the training process truthfully by providing a fair and accurate assessment of clients. However, the high computation complexity of Shapley value limits its potential applications in FL. Some sampling-based approximation methods are proposed to improve the computation efficiency [14, 52] in general settings. Wang et al. [44] proposed a variant of the Shapley value amenable to FL, which captures the value of clients based on the rounds they participate in. Zheng et al. [53] proposed an efficient and secure Shapley value calculation approach under a two-server protocol.

Fairly evaluating the contribution of each client is essential to determine whether that client is helpful, which has not been well studied by existing works. In this paper, we focus on leveraging Shapley value to develop a new FL algorithm that is robust to heterogeneous data challenges and poisoning attacks.

3 PRELIMINARIES

In this section, we review the related definitions and notations used in the paper. Table 1 summarizes the frequently used notations.

3.1 Federated Learning

Standard Federated Learning [32]. Consider a set of clients $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$ such that client $k \in \mathcal{N}$ owns local dataset D_k consisting of $|D_k| = n_k$ sample points. The central server aims to make the local clients collaboratively train a machine learning model without exposing their raw data. The standard federated learning training executes the following steps until the stop criterion is met: (1) the server selects a random fraction of clients and broadcasts the global model parameters to the selected clients; (2) each selected client locally computes an update to the model by training on their local datasets and then sends the update to the server; (3) the central server aggregates and applies these updates to the global model parameters. The objective of the central server takes the following form [7].

Table 1: Some frequently used notations.

Notation	Definition
$\mathcal{N}$	the whole client set
$\mathcal{C}^t$	the selected client set in round t
$\mathcal{P}^t$	the client selected probability vector in round t
m	the expected number of clients in each round
$\boldsymbol{x}^t$	the central model parameters after training t rounds
$\boldsymbol{x}_{i,r}^t$	the local model parameters of client i after training r local steps in round t
$\mathcal{V}_i^t$	the model parameter updates of client i in round t

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x}) := \sum_{i=1}^{\mathcal{N}} w_i F_i(\boldsymbol{x}),$$

where $F_i : \mathbb{R}^d \to \mathbb{R}$ is typically taken as a continuously differentiable local loss function, $\boldsymbol{x} \in \mathbb{R}^d$ are the model parameters, w_i is the weight of client i, and $\sum_{i=1}^{\mathcal{N}} w_i = 1$. The weights of clients are usually set to be proportional to their data size.

Task-Specific Federated Learning. Standard federated learning trains a central model by minimizing the loss computed on the local data of all clients. It ignores the potential mismatch between the training objectives and the specific task of the server. Recently, several works have utilized a global validation dataset to solve this issue [23, 33, 43, 44]. The validation dataset is used to capture the desired input-output relation of the global model. Given the validation dataset D_v, the training goal of the central server can be formulated as follows.

$$\max_{\boldsymbol{x} \in \mathbb{R}^d} \Phi(D_v, \boldsymbol{x}),$$

where Φ is a metric used to measure the global model performance on D_v, e.g., accuracy or negative empirical loss.

3.2 Shapley Value

Consider a set of clients $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$. A *coalition* $\mathcal{S}$ is a subset of $\mathcal{N}$ that cooperates to complete a task. A utility function $\mathcal{U}(\mathcal{S})$ ($\mathcal{S} \subseteq \mathcal{N}$) is the utility of a coalition $\mathcal{S}$ for a task, e.g., the accuracy of the central model trained with $\mathcal{S}$. The *marginal contribution* of client i with respect to a coalition $\mathcal{S}$ is $\mathcal{U}(\mathcal{S} \cup \{\mathrm{i}\}) - \mathcal{U}(\mathcal{S})$.

Shapley [39] laid out the fundamental requirements of fair reward allocation, including balance, symmetry, additivity, and zero element. Specifically, *balance* requires that the total payoff should be fully distributed to all clients. *Symmetry* specifies that two clients should receive the same reward if they have the same marginal contributions. *Additivity* indicates that the reward value on two tasks should be the sum of the values on individual tasks. *Zero element* specifies that a client should not be rewarded if the client does not make any marginal contribution.

Shapley value measures the expectation of marginal contribution by i in all possible coalitions. That is,

$$\mathcal{SV}_i = \frac{1}{|\mathcal{N}|} \sum_{\mathcal{S}\subseteq\mathcal{N}\setminus\{i\}} \frac{\mathcal{U}(\mathcal{S}\cup\{i\}) - \mathcal{U}(\mathcal{S})}{\binom{|\mathcal{N}|-1}{|\mathcal{S}|}}. \tag{1}$$

According to Equation (1), we can find that computing the exact Shapley value requires enumerating all utilities for all client subsets by retraining the global model, which is impractical for weight adjustment in the training process.

4 FRAMEWORK OF *SHAPLEY*FL

We design a robust FL algorithm with an adaptive weighting method based on the surrogate federated SV and an independent uniform sampling strategy in Section 4.1. A theoretical analysis of the convergence and stability is given in Section 4.2.

4.1 Federated Learning with Adaptive Weights

To achieve robust federated learning, we can compute the Shapley value of each client, which measures its marginal contribution towards the global model, and then use that as a weight to aggregate the local model updates. Computing the Shapley value directly requires training the central model with each client subset from scratch and deriving the marginal contributions. This huge time cost makes it infeasible to dynamically adjust the weights in the training process. Fortunately, the collaborative training in each round is the cooperation of a subset of clients and can be used to derive the marginal contributions of each client during that round. Inspired by this, we compute and combine the marginal contributions of each client in sequential rounds as an approximation of its standard Shapley value to adjust its weights.

We define the training process of each round t ($t \geq 1$) as a cooperative game $\mathcal{G}(\boldsymbol{x}^t, C^t, \mathcal{D}_v, \mathcal{U}_F)$, where $\boldsymbol{x}^t$ is the global model parameter at the beginning of round t, and C^t is the client subset that participates in round t. $\Psi(\boldsymbol{x}^t, \mathcal{S}) = \boldsymbol{x}^t - \eta_g \sum_{i\in S} \frac{\eta_l \mathcal{V}_i^t}{|S|}$ returns the model parameter after updating $\boldsymbol{x}^t$ by client subset $\mathcal{S}$ where $\mathcal{V}_i^t$ is the model parameter update vector of client i in round t, and η_g (η_l) is global (local) learning rate, respectively. The utility of a coalition $\mathcal{S} \subseteq C^t$ is the performance of the global model updated by $\mathcal{S}$ in round t, i.e., $\mathcal{U}_F(\mathcal{S}) = \Phi(D_v, \Psi(\boldsymbol{x}^t, \mathcal{S}))$. We define the partial federated Shapley value of each client in round t as follows.

Definition 4.1. (Partial Federated SV) In a cooperative game $\mathcal{G}(\boldsymbol{x}^t, C^t, \mathcal{D}_v, \mathcal{U})$, the partial federated Shapley value of client $i \in C^t$ in round t is

$$\begin{aligned} \mathcal{SV}_i^t &= \frac{1}{|C^t|} \sum_{\mathcal{S}\subseteq C^t\setminus\{i\}} \frac{\mathcal{U}_F(\mathcal{S}\cup\{i\}) - \mathcal{U}_F(\mathcal{S})}{\binom{|C^t|-1}{|\mathcal{S}|}} \\ &= \sum_{S\subset C^t\setminus\{i\}} \frac{\Phi(\mathcal{D}_v, \Psi(\boldsymbol{x}^t, \mathcal{S}\cup\{i\})) - \Phi(\mathcal{D}_v, \Psi(\boldsymbol{x}^t, \mathcal{S}))}{|C^t|\binom{|C^t|}{|\mathcal{S}|}}. \end{aligned}$$

The partial federated SV measures the aggregate marginal contributions of the client with respect to all subsets of the selected clients in each round. The advantage is that it does not require retraining the model from scratch and can be directly computed based on the model updates from that round. By combining them from sequential rounds for each client, it provides an approximation of the overall marginal contributions of the client towards training the global model so far. However, the ranges of the partial federated SV in different rounds are unequal since the change in the central model performance tends to get smaller as the training process goes on. Thus, we adopt the min-max normalization in each round to eliminate the influence of unequal ranges of partial federated SV in Definition 4.2.

Definition 4.2. (Normalized Partial Federated SV) Given $\mathcal{SV}^t = \{\mathcal{SV}_i^t | i \in C^t\}$, the normalized partial federated SV is defined as

$$\mathcal{NSV}_i^t = \frac{\mathcal{SV}_i^t - \min(\mathcal{SV}^t)}{\max(\mathcal{SV}^t) - \min(\mathcal{SV}^t)}, \tag{2}$$

where $\max(\cdot)$ and $\min(\cdot)$ return the maximum and minimum element of a set, respectively.

Remark 1. *According to the balance property, the sum of the partial federated SV of selected clients in each round equals the performance improvement of the central model in that round. The absolute value of the partial federated SV is trivial since the incremental model performance in a single round is small. Instead, min-max normalization maintains the relative size of the partial federated SV of selected clients in each round, which is preferable in evaluating the relative contributions of clients.*

We then combine the normalized partial federated SV in different rounds in Definition 4.3.

Definition 4.3. (Surrogate Federated SV) Given T sequential cooperative games $\mathcal{G}(\boldsymbol{x}^t, C^t, \mathcal{D}_v, \mathcal{U}_F)$ ($1 \leq t \leq T$), the surrogate federated Shapley value of client i during the first t games is

$$\mathcal{SSV}_i^t = \begin{cases} \beta * \mathcal{SSV}_i^{t-1} + (1-\beta) * \mathcal{NSV}_i^t, & i \in C^t, \\ \mathcal{SSV}_i^{t-1}, & i \notin C^t, \end{cases} \tag{3}$$

where β ($0 \leq \beta \leq 1$) controls the update rate of the surrogate federated SV.

Remark 2. *The surrogate federated SV is updated by the normalized partial federated SV in each round and thus can capture the contributions of clients in the training process so far. Therefore, it is suitable for dynamically adjusting clients' weights in time. The hyperparameter β can adjust the timeliness of contributions. For example, $\beta = 0$ means the surrogate federated SV completely depends on the normalized partial federated SV in the current round.*

Adaptive Training Objective. The contribution of each client to the global loss is weighted by its surrogate federated SV so far. We formulate the adaptive training objective of the central server in round t ($1 \leq t \leq T$) as follows.

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F^t(\boldsymbol{x}) := \sum_{i=1}^{N} w_i^t F_i(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{\mathcal{SSV}_i^t}{\sum_{j=1}^{N} \mathcal{SSV}_j^t} F_i(\boldsymbol{x}). \tag{4}$$

We present a simple adaptive FL algorithm named AFedSV. Methods for enhancing convergence and efficiently calculating the surrogate federated SV will be presented in Section 5. AFedSV adopts an independent uniform client sampling strategy where the probability of each client being selected per round is $\gamma = \frac{m}{|\mathcal{N}|}$. The detailed algorithm of AFedSV is shown in Algorithm 1. Let R be the number of local steps(batches) that clients train locally in each round, and $\boldsymbol{x}^t$ be the central model parameters at the beginning of round t. The

Algorithm 1: Adaptive Federated Learning Based on Shapley Value with Uniform Sampling (AFedSV)

```
input  : initial global model parameters $x^1$
         global and local step-sizes $\eta_g, \eta_l$
         initial $\mathcal{P} = [\gamma, \cdots, \gamma]$
output : global parameter $x^{T+1}$ after training $T$ rounds
1  foreach communication round $t = 1, 2, \cdots, T$ do
2      sample clients $C^t \sim \mathcal{P}$;
3      server broadcasts $x^t$ to all clients in $C^t$;
4      foreach client $i \in C^t$ do
5          initialize local model $x^t_{i,0} \leftarrow x^t$;
6          foreach local step $r = 1, 2, \cdots, R$ do
7              compute mini-batch gradient $g_i(x^t_{i,r-1})$;
8              update $x^t_{i,r} \leftarrow x^t_{i,r-1} - \eta_l g_i(x^t_{i,r-1})$;
9          compute $\mathcal{V}^t_i = \sum_{r=1}^{R} g_i(x^t_{i,r-1})$;
10         send $\mathcal{V}^t_i$ to master;
11     server calculates weights $w^t_i (1 \leq i \leq |\mathcal{N}|)$ in round $t$;
12     server updates global model
           $x^{t+1} \leftarrow x^t - \eta_g \sum_{i \in C^t} \frac{w^t_i}{\gamma} \eta_l \mathcal{V}^t_i$;
13 return global model parameters $x^{T+1}$
```

server broadcasts the current central model parameters x^t to the selected clients (Lines 1-2). The selected clients conduct local training, which updates the received parameters using their local data (Lines 4-9). $x^t_{i,r}$ $(0 \leq r \leq R)$ is the model parameters that are updated from x^t with local data of client i after r local steps in round t. Given the learning rate of local clients η_l, $x^t_{i,r} = x^t - \eta_l g_i(x^t_{i,r-1})$, where $g_i(x^t_{i,r})$ is an unbiased estimator of $\nabla F_i(x^t_{i,r})$. The estimator of update vector of client $i \in C^t$ in round t can be represented as $\mathcal{V}^t_i = \sum_{r=1}^{R} g_i(x^t_{i,r-1})$. The local parameter updates are sent to the server after local training (Line 10). The server uses the received parameter updates to calculate the surrogate federated SV of clients and update the global model parameters using the weighted local updates according to the surrogate federated SV (Lines 11-12). We have

$$\begin{aligned} \mathbb{E}\Big[\sum_{i \in C^t} w^t_i \frac{\mathcal{V}^t_i}{\gamma}\Big] &= \mathbb{E}\Big[\sum_{i=1}^{N} w^t_i \mathcal{V}^t_i\Big] \\ = \mathbb{E}\Big[\sum_{i=1}^{N}\sum_{r=1}^{R} w^t_i g_i(x^t_{i,r-1})\Big] &= \sum_{i=1}^{N}\sum_{r=1}^{R} w^t_i \nabla F_i(x^t_{i,r-1}). \end{aligned} \tag{5}$$

Equation (5) shows the estimator of updates of partial clients is an unbiased estimator of updates of all clients. Computing the normalized partial federated SV is crucial for dynamically adjusting weights. An efficient normalized partial federated SV approximation method is proposed in Section 5.2 as a complement to weight calculation.

4.2 Theoretical Analysis

Since the loss functions of most effective models in FL are non-convex, we mainly focus on the convergence analyses in non-convex settings. Besides, we give an upper bound on the difference of the loss change in consecutive rounds to show the stability of our proposed algorithm. Due to space limitations, we provide the proof in the full paper placed in our code repository.

Convergence Analysis. Assume that the local loss function F_i $(1 \leq i \leq |\mathcal{N}|)$ is L-smooth, which is consistent with the generally adopted assumption [21]. $g_i(x^t_{i,r})$ can be decomposed to its expectation $\nabla F_i(x^t_{i,r})$ and an auxiliary estimator $\delta_i(x^t_{i,r})$, that is $g_i(x^t_{i,r}) = \nabla F_i(x^t_{i,r}) + \delta_i(x^t_{i,r})$ for i $([1 \leq I \leq |\mathcal{N}|)$, where $\mathbb{E}[\delta_i(x^t_{i,r})] = 0$. Assumptions 1 and 2 capture the variation range of local gradients and the similarity among local gradients of local loss functions, respectively, which are widely used in FL [22].

Assumption 1. *For round t $(1 \leq t \leq T)$, local step r $(1 \leq r \leq R)$, and client i $(1 \leq i \leq |\mathcal{N}|)$, $\mathbb{E}[\|\delta^t_{i,r}\|^2 | x^t_{i,r}] \leq M\|\nabla F_i(x^t_{i,r})\|^2 + c$ holds for some $M \geq 0$ and $c \geq 0$.*

Assumption 2. *For round t $(1 \leq t \leq T)$, $\sum_{i=1}^{|\mathcal{N}|} w^t_i \|\nabla F_i(x) - \nabla F^t(x)\|^2 \leq \rho$ holds for some $\rho \geq 0$.*

Denote by $\Delta x(t)$ the global update vector estimator of the server, which is obtained by training one round for objective F^t. The effective global update vector $\eta\Delta x(t)$ can be denoted as follows.

$$\eta\Delta x(t) = \eta_g \sum_{i \in C^t}\sum_{r=0}^{R-1} \frac{w^t_i}{\gamma}\eta_l g_i(x^t_{i,r}) = \frac{\eta}{R}\sum_{i \in C^t}\sum_{r=0}^{R-1}\frac{w^t_i}{\gamma} g_i(x^t_{i,r}), \tag{6}$$

where $\eta = R\eta_l\eta_g$ is the effective step size of the global update vector and the global learning rate $\eta_g \geq \sqrt{\frac{5\gamma}{4}}$.

According to Equation (6), we have

$$\mathbb{E}[\Delta x(t)] = \mathbb{E}[\frac{1}{R}\sum_{i=1}^{|\mathcal{N}|}\sum_{r=0}^{R-1} w^t_i g_i(x^t_{i,r})] = \frac{1}{R}\sum_{i=1}^{|\mathcal{N}|}\sum_{r=0}^{R-1} w^t_i \nabla F_i(x^t_{i,r}).$$

Denote by $x(t)$ the model parameters that are updated from x^t by one round for objective F^t, i.e., $x(t) = x^t - \eta\Delta x(t)$. Then, the iterations of adaptive FL satisfy Theorem 4.4 (see proof in the Appendix), which gives the standard form of convergence result of adaptive federated learning for one round in the non-convex setting [7].

Theorem 4.4. *Under Assumptions 1 and 2, we can get*

$$\begin{aligned} \mathbb{E}[F^t(x(t))] \leq \mathbb{E}[F^t(x^t)] &- \frac{3}{8}\eta(1 - \frac{10}{3}\eta L)\|\nabla F^t(x^t)\|^2 \\ &+ \frac{\eta}{8}(1 + 2\eta L)\rho + \frac{\eta^2 cL}{2R\gamma}(\frac{5\gamma}{4\eta_g^2} + \omega^t) \end{aligned}$$

by setting $\eta \in (0, \frac{\gamma}{8L(2+M/R)}]$ and $\omega^t = \sum_{i=1}^{N}(w^t_i)^2$.

Remark 3. *We analyze the upper bound of the loss change for the model in one training round since the training objectives vary in each round. It can be seen that the upper bound depends on the weights when adopting a uniform independent client sampling strategy and learning rates under Assumptions 1 and 2. When the client weights are set to uniform, the convergence guarantee recovers the state-of-the-art non-convex FL complexity guarantee provided in [22]. The advantage of adaptive weights compared to uniform weights is that the training objective can reflect the contributions of clients. To achieve a tighter convergence guarantee, we propose a new client sampling strategy in Section 5.1.*

Stability Analysis. Training objectives vary in each round for adaptive FL since clients' weights are adjusted in the training process. The convergence stability is manifested in the difference of the loss change in training the initial model parameters for objectives in consecutive rounds. However, there is no analysis of the stability of convergence in existing adaptive FL works. To fill the gap, we give its upper bound in Theorem 4.5.

Denote by $\Delta\boldsymbol{x}(t+1)$ the global update vector estimator of the server, which is obtained by training one round for objective F^{t+1}. The effective global update vector $\eta\Delta\boldsymbol{x}(t+1)$ can be denoted as follows.

$$\eta\Delta\boldsymbol{x}(t{+}1) = \eta_g \sum_{i\in C^t}\sum_{r=0}^{R-1}\frac{w_i^{t+1}}{\gamma}\eta_l g_i(\boldsymbol{x}_{i,r}^t) = \frac{\eta}{R}\sum_{i\in C^t}\sum_{r=0}^{R-1}\frac{w_i^{t+1}}{\gamma}g_i(\boldsymbol{x}_{i,r}^t).$$

Thus, the model parameters that are updated from $\boldsymbol{x}^t$ by training one round for objective F^{t+1} can be denoted by $\boldsymbol{x}(t+1) = \boldsymbol{x}^t - \eta\Delta\boldsymbol{x}(t+1)$.

Theorem 4.5. *Under Assumptions 1 and 2, we can get*

$$\begin{aligned}&\{\mathbb{E}[F^{t+1}(\boldsymbol{x}(t+1))] - \mathbb{E}[F^{t+1}(\boldsymbol{x}^t)]\} - \{\mathbb{E}[F^t(\boldsymbol{x}(t))] - \mathbb{E}[F^t(\boldsymbol{x}^t)]\}\\ &\le \frac{3}{8}\eta(1-\frac{4}{\eta}-\frac{10}{3}\eta L)\mathbb{E}[\|\nabla F^t(\boldsymbol{x}^t)\|^2]\\ &+\frac{\eta\rho}{4}+2\eta^2 L(\frac{\rho}{4}+\frac{c}{R\gamma}) - \frac{3}{8}\eta(1-\frac{10}{3}\eta L)\mathbb{E}[\|\nabla F^{t+1}(\boldsymbol{x}^t)\|^2]\end{aligned}$$

by setting $\eta \in (0, \frac{\gamma}{8L(2+M/R)}]$ *and* $\eta_g \ge \sqrt{\frac{5\gamma}{4}}$.

Remark 4. *Theorem 4.5 shows the difference of the upper bound on the loss change of training the model on objective F^{t+1} and the lower bound on the loss change of training the model on objective F^t. Thus, it gives the upper bound on the convergence change of the model on different training objectives.*

5 OPTIMIZATION: CLIENT SAMPLING AND WEIGHT CALCULATION

To reduce the communication cost, we propose an optimal client sampling method based on client-importance sampling in Section 5.1. Further, we propose an efficient approach to calculate the normalized partial federated SV based on the differences in the partial federated SV to mitigate the computation cost in Section 5.2.

5.1 Client-Importance Sampling

Only a subset of clients communicates their updates to the server in each round due to the limited communication bandwidth. Inspired by importance sampling, we design a client sampling approach that minimizes the variance of the estimator of global updates, which is crucial for ensuring faster convergence [7]. Denote by $\mathcal{P}^t = [p_1^t, \cdots, p_{|\mathcal{N}|}^t]$ the probability vector, where p_i^t is the probability that client i be selected in C^t. The expected number of clients involved in each round is $\sum_{i=1}^{|\mathcal{N}|} p_i^t \le |\mathcal{N}|$, denoted by m. Following the training objective in Equation (4) and using the effective step-size $\eta = R\eta_l\eta_g$, the global update estimate vector in round t is $\Lambda\boldsymbol{x}(t) = \frac{1}{R}\sum_{i\in C^t}\sum_{r=0}^{R-1}\frac{w_i^t}{p_i^t}g_i(\boldsymbol{x}_{i,r}^t)$. Denote by $\mathcal{X}^t = \frac{1}{R}\sum_{i=1}^{|\mathcal{N}|}\sum_{r=0}^{R-1} w_i^t g_i(\boldsymbol{x}_{i,r}^t)$, the variance of $\Lambda\boldsymbol{x}(t)$ can be represented as follows.

$$\begin{aligned}Var[\Lambda\boldsymbol{x}(t)] &= \mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\|^2]\\ &= \mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \mathcal{X}^t + \mathcal{X}^t - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\|^2]\\ &= \mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \mathcal{X}^t\|^2] + \mathbb{E}[\|\mathcal{X}^t - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\|^2]\\ &+ 2\mathbb{E}[\langle\Lambda\boldsymbol{x}(t) - \mathcal{X}^t, \mathcal{X}^t - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\rangle].\end{aligned}$$

Due to the fact that $\mathbb{E}[\Lambda\boldsymbol{x}(t)] = \mathbb{E}[\mathcal{X}^t]$, we have $\mathbb{E}[\langle\Lambda\boldsymbol{x}(t) - \mathcal{X}^t, \mathcal{X}^t - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\rangle] = 0$. Then, we can get

$$Var[\Lambda\boldsymbol{x}(t)] = \mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \mathcal{X}^t\|^2] + \mathbb{E}[\|\mathcal{X}^t - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\|^2].$$

Denote by $\mathcal{X}_i^t = \frac{1}{R}\sum_{r=0}^{R-1} g_i(\boldsymbol{x}_{i,r}^t)$, we have

$$\begin{aligned}&\mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \mathcal{X}^t\|^2] = \mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \sum_{i=1}^{|\mathcal{N}|} w_i^t\mathcal{X}_i^t\|^2]\\ &= \mathbb{E}[\|\sum_{i\in C^t} w_i^t\mathcal{X}_i^t - \sum_{i=1}^{|\mathcal{N}|} w_i^t\mathcal{X}_i^t\|^2] = \mathbb{E}[\sum_{i=1}^{|\mathcal{N}|}\frac{(1-p_i^t)(w_i^t)^2}{p_i^t}\|\mathcal{X}_i^t\|^2],\end{aligned}$$

where the last equation can be derived from the key lemma of [17]. Thus, we can get

$$\begin{aligned}\mathrm{Var}[\Lambda\boldsymbol{x}(t)] = \mathbb{E}[\sum_{i=1}^{|\mathcal{N}|}\frac{(w_i^t)^2}{p_i^t}\|\mathcal{X}_i^t\|^2] - \mathbb{E}[\sum_{i=1}^{|\mathcal{N}|}(w_i^t)^2\|\mathcal{X}_i^t\|^2]\\ + \mathbb{E}[\|\sum_{i=1}^{|\mathcal{N}|} w_i^t\mathcal{X}_i^t - \mathbb{E}[\Lambda\boldsymbol{x}(t)]\|^2].\end{aligned} \tag{7}$$

It shows that the variance of the global update estimate vector is affected by three factors: the norm of the estimator of the local update vector $\|\mathcal{X}_i^t\|^2$, the weight of clients w_i^t, and the sampling strategy (selected probability of each client p_i^t). The knowledge of all estimators of the local update vectors $\mathcal{X}_i^t (1 \le i \le |\mathcal{N}|)$ cannot be obtained due to the partial client participation in each round. In addition, in our framework, the weights are determined by the surrogate federated SV of the clients in Section 4.1 to increase the robustness of FL. Thus, we focus on how to minimize the variance by adjusting the selected probability of each client here. Since $\mathbb{E}[\sum_{i=1}^{|\mathcal{N}|}\frac{(w_i^t)^2}{p_i^t}\|\mathcal{X}_i^t\|^2]$ is the only term affected by the sampling strategy, the selected probability of each client should be positively related to its weight to minimize the variance if we omit the norm differences between estimators of local update vectors. The weights of clients are updated after the partial clients are selected in each round in our framework. So we use the weights in the previous round as an approximation. Hence, according to the relationship between the weights of clients and the sampling strategy on the variance, an optimization problem is proposed as follows and the solution is presented in Theorem 5.1 (see proof in the Appendix).

$$\min_{\mathcal{P}^t}\sum_{i=1}^{|\mathcal{N}|}\frac{(w_i^{t-1})^2}{p_i^t},\ \text{ subject to } m = \sum_{i=1}^{|\mathcal{N}|} p_i^t,\ p_i^t \in [0,1]. \tag{8}$$

Theorem 5.1. *Let $\mathcal{L}^t$ contain l $(0 \le l < m)$ clients with the largest weights in round $t-1$. The optimal solution of Equation* (8) *is the one with the smallest value obtained by Equation* (8) *among the following* *m possible solutions* ($|\mathcal{L}^t| = 0, 1, \cdots, m-1$).

$$p_i^t = \begin{cases} (m - |\mathcal{L}^t|)\frac{w_i^{t-1}}{\sum_{j\notin\mathcal{L}^t} w_j^{t-1}}, & i \notin \mathcal{L}^t \\ 1, & i \in \mathcal{L}^t. \end{cases} \tag{9}$$

We get a new adaptive federated learning algorithm improved from AFedSV by employing client-importance sampling, named AFedSV+. The detailed algorithm is shown in Appendix Algorithm 2. Algorithm 2 is similar to Algorithm 1. The main difference is to determine the client sampling probability according to Theorem 5.1 (Line 7).

Convergence Analysis. We give an analysis of the convergence of adaptive federated learning with importance sampling. Denote by $\widetilde{\boldsymbol{x}}(t) = \boldsymbol{x}^t - \eta\Lambda\boldsymbol{x}(t)$ the estimator of global parameters that $\boldsymbol{x}^t$ be updated after round t where clients are sampled according to probability vector $\mathcal{P}^t$. The improvement factor is defined as

$$\alpha^t = \frac{\mathbb{E}[\|\Lambda\boldsymbol{x}(t) - \sum_{i=1}^{|\mathcal{N}|} w_i^t \mathcal{X}_i^t\|^2]}{\mathbb{E}[\|\Delta\boldsymbol{x}(t) - \sum_{i=1}^{|\mathcal{N}|} w_i^t \mathcal{X}_i^t\|^2]}. \tag{10}$$

Theorem 5.2. *Under Assumptions 1 and 2, we can get*

$$\mathbb{E}[F^t(\widetilde{\boldsymbol{x}}(t))] \leq \mathbb{E}[F^t(x^t)] - \frac{3}{8}\eta(1 - \frac{10}{3}\eta L)\|\nabla F^t(\boldsymbol{x}^t)\|^2 + \frac{\eta}{8}(1 + 2\eta L)\rho + \frac{\eta^2 cL}{R\delta}$$

by setting $\eta \in (0, \frac{\gamma}{8L(2+M/R)}]$, $\delta = \frac{m}{\alpha^t(|\mathcal{N}|-m)+m}$ *and* $\eta_g \geq \sqrt{\frac{5\delta}{4}}$.

Remark 5. *The convergence guarantee recovers Theorem 4.4 when adopting the uniform independent client sampling ($\alpha^t = 1$). Differently, the upper bound becomes tighter when α^t is reduced by using the proposed client importance sampling. The variance of the estimator of global updates can be reduced by the client importance sampling strategy, and experimental results verify the strategy is empirically effective.*

5.2 Normalized Partial Federated SV Estimation

While our surrogate federated SV used as weights for clients avoids the prohibitive cost of computing Shapley value directly, it still requires enumerating all subsets of the participating clients and computing the marginal contributions based on the model performance on the validation dataset in each round. In this section, we present an efficient normalized partial federated SV approximation method to further reduce the computation cost of adaptive federated learning.

Observe that the differences between the partial federated SV in each round have a smaller variance since its range is smaller than that of the partial federated SV. We propose to use the differences to estimate normalized partial federated SV efficiently.

Our proposed method computes the normalized partial federated SV of participating clients in each training round. To lighten notations, we omit the round marker t (e.g., $\mathcal{SV}_i$ for $\mathcal{SV}_i^t$) and assume all clients participate in round t. Given the partial federated Shapley value $\mathcal{SV} = [\mathcal{SV}_1, \cdots, \mathcal{SV}_{|\mathcal{N}|}]$ for clients $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$, let $\Delta\mathcal{SV}_{k,i} = \mathcal{SV}_k - \mathcal{SV}_i$ be the difference of partial federated SV between client k and i. Then $\Delta\mathcal{SV}_k = [\Delta\mathcal{SV}_{k,1}, \cdots, \Delta\mathcal{SV}_{k,n}]$. According to Equation (1), we have

$$\begin{aligned} \Delta\mathcal{SV}_{k,i} &= \frac{1}{|\mathcal{N}|}\sum_{\mathcal{S}\subset\mathcal{N}\setminus\{k\}} \frac{\mathcal{U}_F(\mathcal{S}\cup\{k\}) - \mathcal{U}_F(\mathcal{S})}{\binom{|\mathcal{N}|-1}{|\mathcal{S}|}} \\ &\quad - \frac{1}{|\mathcal{N}|}\sum_{\mathcal{S}\subset\mathcal{N}\setminus\{i\}} \frac{\mathcal{U}_F(\mathcal{S}\cup\{i\}) - \mathcal{U}_F(\mathcal{S})}{\binom{|\mathcal{N}|-1}{|\mathcal{S}|}} \\ &= \frac{1}{|\mathcal{N}|-1}\sum_{\mathcal{S}\subset\mathcal{N}\setminus\{k,i\}} \frac{\mathcal{U}_F(\mathcal{S}\cup\{k\}) - \mathcal{U}_F(\mathcal{S}\cup\{i\})}{\binom{|\mathcal{N}|-2}{|\mathcal{S}|}}. \end{aligned} \tag{11}$$

The last equation can be derived by splitting subsets $\mathcal{S}$ in the first equation into three parts, including either i or k and including neither i nor k, and then rearranging the terms. We omit the mathematical operations due to space limitations. With the differences, we can reformulate the normalized federated SV of i as follows.

$$\begin{aligned} \mathcal{NSV}_i &= \frac{\mathcal{SV}_i - \min(\mathcal{SV})}{\max(\mathcal{SV}) - \min(\mathcal{SV})} \\ &= \frac{[\mathcal{SV}_k - \min(\mathcal{SV})] - [\mathcal{SV}_k - \mathcal{SV}_i]}{[\max(\mathcal{SV}) - \mathcal{SV}_k] - [\min(\mathcal{SV}) - \mathcal{SV}_k]} \\ &= \frac{\max(\Delta\mathcal{SV}_k) - \Delta\mathcal{SV}_{k,i}}{\max(\Delta\mathcal{SV}_k) - \min(\Delta\mathcal{SV}_k)}. \end{aligned} \tag{12}$$

Equation (12) allows us to get the normalized federated SV of all clients based on the differences between the partial federated SV of all clients and any client k.

We propose a sampling algorithm to compute the difference between the partial federated SV of any client k and others. The detailed algorithm is shown in Appendix Algorithm 3, named DMC. Denote by $\Delta\mathcal{SV}_{k,i,j}$ ($0 \leq j \leq |\mathcal{N}| - 2$) the expected difference in utility between coalition $\mathcal{S} \cup k$ and $\mathcal{S} \cup i$ with $|\mathcal{S}| = j$ and $\mathcal{S} \subset \mathcal{N} \setminus \{k, i\}$. That is,

$$\Delta\mathcal{SV}_{k,i,j} = \sum_{\mathcal{S}\subset\mathcal{N}\setminus\{k,i\},|\mathcal{S}|=j} \frac{\mathcal{U}_F(\mathcal{S}\cup\{k\}) - \mathcal{U}_F(\mathcal{S}\cup\{i\})}{\binom{|\mathcal{N}|-1}{|\mathcal{S}|}}. \tag{13}$$

We have $\Delta\mathcal{SV}_{k,i} = \frac{1}{|\mathcal{N}|-1}\sum_{j=0}^{|\mathcal{N}|-2}\Delta\mathcal{SV}_{k,i,j}$ by Equations (11) and (13). To approximate $\Delta\mathcal{SV}_{k,i}$, we can estimate $\Delta\mathcal{SV}_{k,i,j}$ ($0 \leq j \leq |\mathcal{N}| - 2$). Denote by $\mathcal{A}^{k,i,j} = \{\mathcal{S}|\mathcal{S} \subseteq \mathcal{N} \setminus k, i, |\mathcal{S} = j|\}$. Let $\mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S})$ be a random variable with uniform distribution on the set $\{\mathcal{U}_F(\mathcal{S}\cup k) - \mathcal{U}_F(\mathcal{S}\cup i))|\mathcal{S} \in \mathcal{A}^{k,i,j}\}$. Then we have $\mathbb{E}[\mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S})] = \Delta\mathcal{SV}_{k,i,j}$. Given a random sample of $\mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S})$ of size $m_{i,j,k}$ $\{\mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S}_1), \cdots, \mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S}_{m_{k,i,j}})\}$, where $\mathcal{S}_1, \cdots, \mathcal{S}_{m_{k,i,j}} \in \mathcal{A}^{k,i,j}$, the sample mean $\overline{\Delta\mathcal{SV}_{k,i,j}} = \frac{1}{m_{k,i,j}}\sum_{o=1}^{m_{k,i,j}} \mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S}_o)$ is an unbiased estimation of $\Delta\mathcal{SV}_{k,i,j}$. Because

$$\mathbb{E}[\frac{1}{m_{k,i,j}}\sum_{o=1}^{m_{k,i,j}} \mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S}_o)] = \frac{1}{m_{k,i,j}}\sum_{o=1}^{m_{k,i,j}} \mathbb{E}[\mathcal{X}_{\mathcal{N}}^{k,i,j}(\mathcal{S}_o)] = \Delta\mathcal{SV}_{k,i,j}.$$

By equally stratified sampling $\mathcal{X}_{\mathcal{N}}^{k,i,0}(\mathcal{S}), \cdots, \mathcal{X}_{\mathcal{N}}^{k,i,|\mathcal{N}|-2}(\mathcal{S})$, we can get $\overline{\Delta\mathcal{SV}_{k,i,j}}$ ($0 \leq j \leq |\mathcal{N}| - 2$). Now, consider the sample mean $\overline{\Delta\mathcal{SV}_{k,i}} = \frac{1}{|\mathcal{N}|-1}\sum_{j=0}^{|\mathcal{N}|-2}\overline{\Delta\mathcal{SV}_{k,i,j}}$. We have

$$\mathbb{E}[\frac{1}{|\mathcal{N}|-1}\sum_{j=0}^{|\mathcal{N}|-2}\overline{\Delta\mathcal{SV}_{k,i,j}}] = \frac{1}{|\mathcal{N}|-1}\sum_{j=0}^{|\mathcal{N}|-1}\mathbb{E}[\overline{\Delta\mathcal{SV}_{k,i,j}}] = \Delta\mathcal{SV}_{k,i}.$$

That is, $\overline{\Delta\mathcal{SV}_{k,i}}$ is an unbiased estimation of $\Delta\mathcal{SV}_{k,i}$.

6 EXPERIMENTS

In this section, we experimentally study AFedSV/AFedSV+. In Section 6.1, we provide details of datasets used and experimental setup. In Section 6.2, we evaluate a variety of data and model poisoning scenarios on standard image dataset CIFAR-10[1] and Fashion-MNIST[2] to verify the robustness of AFedSV/AFedSV+. In Section 6.3, we further apply our algorithms on a realistic cross-silo healthcare dataset Fed-ISIC2019 [9, 11, 42]. Due to space limitations, the experiment evaluating the effectiveness of DMC is given in Appendix B.5. We also add more experimental details in the supplementary material to enhance reproducibility. The code for experiments is available at https://github.com/ZJU-DIVER/ShapleyFL-Robust-Federated-Learning-Based-on-Shapley-Value, which is implemented using PyTorch.

6.1 Datasets and Experimental Setup

We implement AFedSV/AFedSV+ on standard image datasets CIFAR-10 and Fashion-MNIST. The CNN network is adopted as the central model since it is widely used in the field of image classification. As in previous works, we focus on the more challenging Non-IID setting and simulate the synthetic Non-IID partitions of Fashion-MNIST and CIFAR-10 datasets. The details of the datasets are given in Appendix B.1. Moreover, we study 5 popular data and model poisoning scenarios based on the Non-IID data setting [2, 5, 35, 36, 48]: 1) imbalanced data with long-tailed distribution; 2) irrelevant data with open-set label noise; 3) malicious clients with closed-set label noise; 4) malicious clients with data noise; 5) attacks with gradient poisoning. The details of Non-IID setting and poisoning strategies are given in Appendix B.2 and B.3, including the partition strategy of innocent client/malicious client and the test dataset/validation dataset.

We further experiment on a realistic cross-silo healthcare dataset Fed-ISIC2019. The detailed dataset description is given in Appendix B.1. We follow the setting in [11] and end up with a 6-client federated version of ISIC2019. The best-performing EfficientNets architecture is used as the central model. Since the data distribution among hospitals is fixed, it is no longer necessary to further simulate Non-IID partitions. Considering that the hospitals are honest and the data cannot be distorted locally, we care about Byzantine failures where some clients just fail and send random gradients, which is studied in the experiment.

Proposed algorithms.

- **AFedSV**: Adaptive FL with uniform sampling in Algorithm 1.
- **AFedSV+**: Communication-efficient adaptive FL with the client importance sampling strategy in Algorithm 2.

Baseline Algorithms.

- **FedAvg** [32]: The most popular FL algorithm.
- **FedProx** [26]: The algorithm that copes with the Non-IID problem by adding a proximal term to the loss function.
- **FedSV** [44]: The algorithm that is extended for robust FL using the SV variant in [44].
- **S-FedAvg** [33]: The algorithm that considers the irrelevant data/clients and modifies FedAvg by selecting relevant clients with an SV-based score.
- **RFA** [35]: The Robust Federated Aggregation (RFA) relies on a robust aggregation oracle in FL based on the geometric median.

6.2 Performance on Image Classification

Results on CIFAR-10. Figure 1 shows the accuracy of AFedSV/AFedSV+ along with other baselines in 5 different data scenarios (mentioned in Section 6.1) on CIFAR-10. The averaged results after 5 independent experiments reveal that AFedSV+/AFedSV significantly outperforms baselines in various data settings. With up to 8.1%, 7.3%, 20.7%, 7.9%, and 11.0% performance improvement of AFedSV+ compared to FedAvg, FedSV, FedProx, S-FedAvg, and RFA, respectively.

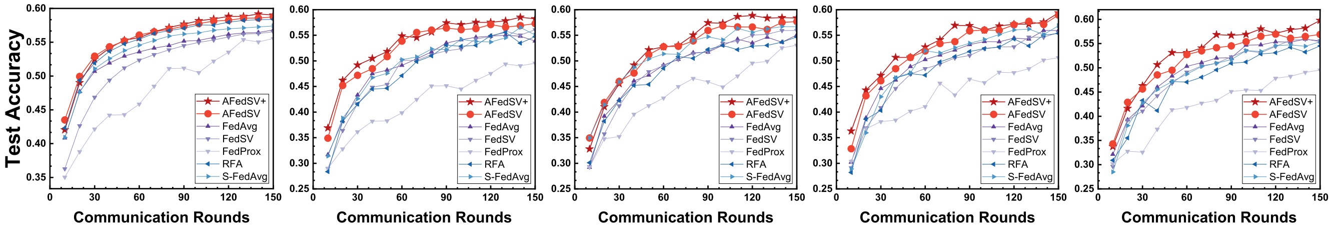
Take the performance improvement in the open-set label noise setting as an illustration. As shown in Figure 1(b), the average accuracy of AFedSV+ on the central server test set D_T after 150 rounds of global communication is 58.22%. It is 8.1%, 17.5%, and 6.3% improvement over the average accuracy of FedAvg, FedProx, and RFA, respectively. We believe that the failure of FedProx in the irrelevant/malicious data setting (even compared to FedAvg) is because it adjusts the loss function and aggregates the gradients produced by the irrelevant clients with the same weight as an innocent client. By contrast, AFedSV+ limits the impact of irrelevant clients by estimating its contribution and assigning lower probability via computing SV-based global weight. Although S-FedAvg and FedSV also introduce the concept of Shapley value, AFedSV+ still has 5.7% and 4.4% improvement over the average accuracy of S-FedAvg and FedSV, respectively. AFedSV+ converges much faster than S-FedAvg and FedSV since it amplifies the impact of gradients collected by clients with higher Shapley value and samples clients based on previous knowledge.

The results also reveal that introducing client-importance sampling could accelerate convergence and improves global model accuracy. Comparing the performance of AFedSV+ and AFedSV, AFedSV+ has a better convergence rate and higher model accuracy. This is because the optimal client sampling method in AFedSV+ allows the selection of optimal clients with higher probability than the uniform sampling strategy in AFedSV. The greater involvement of these clients ensures higher model accuracy. Such a pattern is even more evident at Fashion-MNIST.

Results on Fashion-MNIST. Figure 2 shows the accuracy of AFedSV/AFedSV+ along with other baselines in 5 different data scenarios (mentioned in Section 6.1) on Fashion-MNIST.

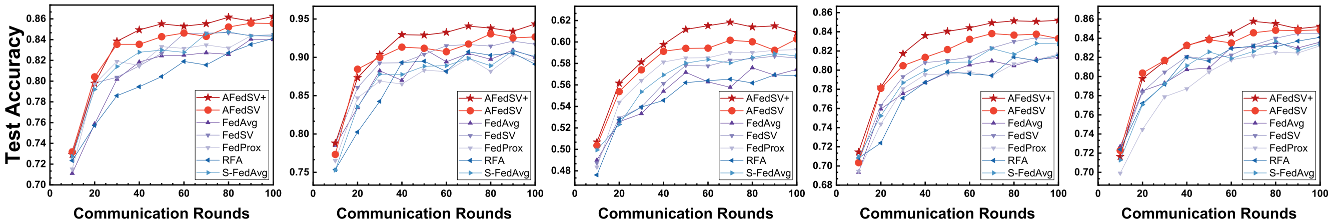
Agreeing with the results on CIFAR-10, the results reveal that AFedSV+/AFedSV significantly outperforms baselines. However, it is worth mentioning that in the closed-set label injection setting, the global accuracy of all algorithms dropped significantly on Fashion-MNIST. We believe this is due to the relative simplicity of the Fashion-MNIST dataset where a simple CNN network could achieve high performance. Thus, the flipped labels exert a severely negative impact on the gradients. Even if the aggregated weight of the gradient from the malicious client is set low, it still exerts a vastly negative impact on global accuracy. As for the CIFAR-10 dataset, since the accuracy of the simple CNN network is relatively

[1] CIFAR-10: http://www.cs.toronto.edu/ kriz/cifar.html
[2] Fashion-MNIST: https://github.com/zalandoresearch/fashion-mnist



(a) Long-tailed (b) Open-set Label Noise (c) Close-set Label Noise (d) Data Noise Attack (e) Gradient Attack

Figure 1: Evaluation results on CIFAR-10 (Section 6.2).



(a) Long-tailed (b) Open-set Label Noise (c) Close-set Label Noise (d) Data Noise Attack (e) Gradient Attack

Figure 2: Evaluation results on Fashion-MNIST (Section 6.2).

low, the effect of closed-set label noise is relatively insignificant. Also, comparing AFedSV+ and AFedSV confirms the benefit of client-importance sampling.

6.3 Performance on Medical Diagnosis

Following the setting in [11], we set the fraction of clients participating in training to 1.0 due to the limited number of clients. Thus, the impact of the client importance sampling strategy is not studied, i.e., we only evaluate AFedSV instead of both. Also, we omit S-FedAvg since it degenerates to FedAvg. We simulate the gradient attack as mentioned in Section 6.1 with 2 malicious clients that upload perturbed gradient in each round. For fairness consideration, we implement 10 independent experiments with 25 rounds using random seeds to generate candidates for malicious clients. Figure 3 shows that AFedSV outperforms baselines. The average accuracy of AFedSV on 25 rounds reaches 64.24%, which has 25.1%, 13.7%, 13.4%, 4.4% improvement compared to FedAvg, FedProx, RFA, and FedSV, respectively. The error bar indicates the fluctuation of global accuracy caused by choosing different malicious client candidates. Naturally, setting the client with the largest data size (9930) as a malicious client has a more significant negative impact on global model accuracy than the client with the smallest data size (351). Thus, the consistently smaller error bar of AFedSV verifies the robustness of our adaptive weighting mechanism compared to baselines.

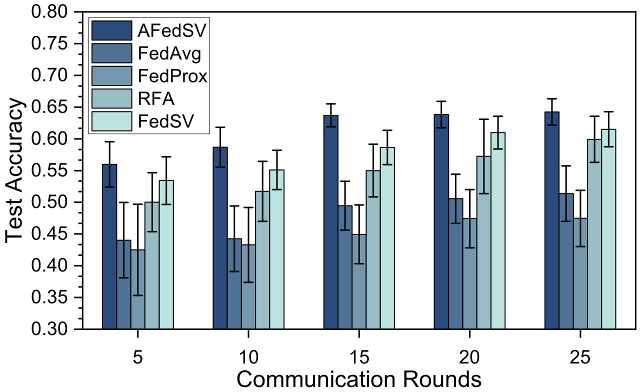


Figure 3: Evaluation results on Fed-ISIC2019 (Section 6.3).

7 CONCLUSION

In this paper, in order to enhance the robustness of federated learning, we proposed an adaptive Shapley value-based weighting method. We produced a client-importance sampling strategy to save communication costs and a normalized partial federated SV estimation method to mitigate the computation cost. We provided a thorough theoretical analysis of the convergence and stability of AFedSV and AFedSV+. Extensive experiments on several real-world applications (e.g., vision and healthcare) were conducted to validate the robustness of our proposed methods.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the National Key RD Program of China (2021YFB3101100, 2022YFB310340), NSFC grants (62102352), NSF grants (CNS-2124104, CNS-2125530), NIH grants (R01LM013712, UL1TR002378), and the Major Programs of the National Social Science Foundation of China (22&ZD147).

REFERENCES

[1] Ergute Bao, Yizheng Zhu, Xiaokui Xiao, Yin Yang, Beng Chin Ooi, Benjamin Hong Meng Tan, and Khin Mi Mi Aung. 2022. Skellam Mixture Mechanism: a Novel Approach to Federated Learning with Differential Privacy. *Proc. VLDB Endow.* 15, 11 (2022), 2348–2360. https://www.vldb.org/pvldb/vol15/p2348-bao.pdf

[2] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems* 30 (2017).

[3] Christopher Briggs, Zhong Fan, and Peter Andras. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2020. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995* (2020).

[6] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & OR* 36, 5 (2009), 1726–1730.

[7] Wenlin Chen, Samuel Horvath, and Peter Richtarik. 2020. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723* (2020).

[8] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. 2022. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 10351–10375.

[9] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. 2018. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 168–172.

[10] Xiaotie Deng and Christos H Papadimitriou. 1994. On the complexity of cooperative solution concepts. *Mathematics of operations research* 19, 2 (1994), 257–266.

[11] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. [n.d.]. FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Healthcare Settings. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[12] Zhenan Fan, Huang Fang, Zirui Zhou, Jian Pei, Michael P Friedlander, Changxin Liu, and Yong Zhang. 2022. Improving Fairness for Data Valuation in Horizontal Federated Learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2440–2453.

[13] Fangcheng Fu, Xupeng Miao, Jiawei Jiang, Huanran Xue, and Bin Cui. 2022. Towards Communication-efficient Vertical Federated Learning Training via Cache-enabled Local Update. *Proc. VLDB Endow.* 15, 10 (2022), 2111–2120. https://www.vldb.org/pvldb/vol15/p2111-fu.pdf

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

[15] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 19586–19597.

[16] Yufei Han and Xiangliang Zhang. 2020. Robust federated learning via collaborative machine teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4075–4082.

[17] Samuel Horváth and Peter Richtárik. 2019. Nonconvex variance reduced optimization with arbitrary sampling. In *International Conference on Machine Learning*. PMLR, 2781–2789.

[18] Zhida Jiang, Yang Xu, Hongli Xu, Zhiyuan Wang, Chunming Qiao, and Yangming Zhao. 2022. FedMP: Federated Learning through Adaptive Model Pruning in Heterogeneous Edge Computing. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 767–779. https://doi.org/10.1109/ICDE53745.2022.00062

[19] Jiayin Jin, Jiaxiang Ren, Yang Zhou, Lingjuan Lyu, Ji Liu, and Dejing Dou. 2022. Accelerated Federated Learning with Decoupled Adaptive Optimization. In *International Conference on Machine Learning*. PMLR, 10298–10322.

[20] 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.

[21] Hamed Karimi, Julie Nutini, and Mark Schmidt. 2016. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 795–811.

[22] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*. PMLR, 5132–5143.

[23] Junyi Li, Jian Pei, and Heng Huang. 2022. Communication-Efficient Robust Federated Learning with Noisy Labels. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 914–924.

[24] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 965–978. https://doi.org/10.1109/ICDE53745.2022.00077

[25] 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.

[26] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[27] Zengpeng Li, Vishal Sharma, and Saraju P Mohanty. 2020. Preserving data privacy via federated learning: Challenges and solutions. *IEEE Consumer Electronics Magazine* 9, 3 (2020), 8–16.

[28] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. 2021. Projected Federated Averaging with Heterogeneous Differential Privacy. *Proc. VLDB Endow.* 15, 4 (2021), 828–840. https://doi.org/10.14778/3503585.3503592

[29] Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. 2022. GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–21.

[30] Bing Luo, Xiang Li, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. 2021. Cost-effective federated learning design. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.

[31] Bing Luo, Wenli Xiao, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. 2022. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1739–1748.

[32] 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.

[33] Lokesh Nagalapatti and Ramasuri Narayanam. 2021. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9046–9054.

[34] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 1–7.

[35] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. 2022. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing* 70 (2022), 1142–1154.

[36] Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. 2019. DETOX: A redundancy-based framework for faster and more robust gradient aggregation. *Advances in Neural Information Processing Systems* 32 (2019).

[37] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. 2020. The future of digital health with federated learning. *NPJ digital medicine* 3, 1 (2020), 1–7.

[38] Xinyi Shang, Yang Lu, Yiu-ming Cheung, and Hanzi Wang. 2022. FEDIC: Federated Learning on Non-IID and Long-Tailed Data via Calibrated Distillation. *arXiv preprint arXiv:2205.00172* (2022).

[39] Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[40] Xian Shuai, Yulin Shen, Siyang Jiang, Zhihe Zhao, Zhenyu Yan, and Guoliang Xing. 2022. BalanceFL: Addressing Class Imbalance in Long-Tail Federated Learning. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 271–284.

[41] Farnaz Tahmasebian, Jian Lou, and Li Xiong. 2022. Robustfed: a truth inference approach for robust federated learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1868–1877.

[42] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data* 5, 1 (2018), 1–9.

[43] 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.

[44] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. 2020. A principled approach to data valuation for federated learning. In *Federated Learning*. Springer, 153–167.

[45] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8688–8696.

[46] Hongda Wu and Ping Wang. 2021. Fast-convergent federated learning with adaptive weighting. *IEEE Transactions on Cognitive Communications and Networking* 7, 4 (2021), 1078–1088.

[47] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. 2020. Privacy Preserving Vertical Federated Learning for Tree-based Models. *Proc. VLDB Endow.* 13, 11 (2020), 2090–2103. http://www.vldb.org/pvldb/vol13/p2090-wu.pdf

[48] Jian Xu, Shao-Lun Huang, Linqi Song, and Tian Lan. 2022. Byzantine-robust federated learning through collaborative malicious gradient filtering. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1223–1235.

[49] Miao Yang, Hua Qian, Ximin Wang, Yong Zhou, and Hongbin Zhu. 2021. Client Selection for Federated Learning With Label Noise. *IEEE Transactions on Vehicular Technology* 71, 2 (2021), 2193–2197.

[50] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*. PMLR, 5650–5659.

[51] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems* 216 (2021), 106775.

[52] Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Efficient Sampling Approaches to Shapley Value Approximation. In *SIGMOD*. ACM.

[53] Shuyuan Zheng, Yang Cao, and Masatoshi Yoshikawa. 2022. Secure Shapley Value for Cross-Silo Federated Learning. *arXiv preprint arXiv:2209.04856* (2022).

APPENDIX

The organization of the appendix is as follows. Section A provides the details of Algorithms 2 and 3. Section B provides the details of the experiment to guarantee reproducibility. Section C provides proof of theoretical results.

A ALGORITHM

Algorithm 2: Adaptive Federated Learning Based on Shapley Value with Importance Sampling (AFedSV+)

```
input  : initial global model parameters $\boldsymbol{x}^1$
         initial global and local step-sizes $\eta_g, \eta_l$
         initial $\mathcal{P}^1 = [\frac{m}{|\mathcal{N}|}, \cdots, \frac{m}{|\mathcal{N}|}]$
output : global parameters $\boldsymbol{x}^{T+1}$ after training $T$ rounds
foreach communication round $t = 1, 2, \cdots, T$ do
    sample clients $C^t \sim \mathcal{P}^t$;
    server broadcasts $\boldsymbol{x}^t$ to all clients in $C_t$;
    local training in round $t$ the same as Algorithm 1;
    server calculates weights $w_i^t (1 \leq i \leq |\mathcal{N}|)$ in round $t$;
    server updates global model
      $\boldsymbol{x}^{t+1} \leftarrow \boldsymbol{x}^t - \eta_g \sum_{i \in C^t} \frac{w_i^t}{p_i^t} \eta_l \mathcal{V}_i^t$;
    server calculates $\mathcal{P}^{t+1}$ by Theorem 5.1;
return global model parameters $x^{T+1}$;
```

Algorithm 3: Shapley Value Difference Computation (DMC)

```
input  : clients $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$
         number of total samples $M > 0$
output : approximate difference between Shapley value for
         each client $i$ $(1 \leq i \leq |\mathcal{N}|)$ and $k$
$\overline{\Delta\mathcal{SV}_{k,i}} \leftarrow 0 \ (1 \leq i \leq |\mathcal{N}|)$;
$\overline{\Delta\mathcal{SV}_{k,i,j}} \leftarrow 0 \ (1 \leq i \leq |\mathcal{N}|, 0 \leq j \leq |\mathcal{N}| - 2)$;
for _ =1 to $\lfloor M/(|\mathcal{N}| - 1) \rfloor$ do
    for $i$ = 1 to $n$ do
        for $j$ = 0 to $|\mathcal{N}|$-2 do
            let $\mathcal{S}$ be a random sample drawn from $\mathcal{A}^{k,i,j}$;
             $u \leftarrow \mathcal{U}(\mathcal{S} \cup \{z_k\}) - \mathcal{U}(\mathcal{S} \cup \{z_i\})$;
             $\overline{\Delta\mathcal{SV}_{k,i,j}} += \frac{u}{\lfloor m/(|\mathcal{N}|-1) \rfloor}$;
for i=1 to $|\mathcal{N}|$ do
    for j=0 to $|\mathcal{N}|$-2 do
        $\overline{\Delta\mathcal{SV}_{k,i}} += \frac{1}{|\mathcal{N}|-1} \overline{\Delta\mathcal{SV}_{k,i,j}}$;
return $\overline{\Delta\mathcal{SV}_{k,1}}, \ldots, \overline{\Delta\mathcal{SV}_{k,|\mathcal{N}|}}$;
```

B REPRODUCIBILITY

B.1 Dataset description

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 50000 training images and 10000 test images. The Fashion-MNIST dataset is a 28x28 grayscale image dataset used to replace the traditional handwriting dataset MNIST, which also has ten labels and consists of a train set of 60000 images and a test set of 10000 examples. We randomly split 2000 images(20%) of the original test dataset as the global validation dataset and the rest 8000 images as the real test dataset.

The Fed-ISIC2019 dataset contains 23,247 dermoscopy images with 200*200*3 input dimensions collected from different hospitals. The training entails identifying images from eight distinct melanoma classes. We follow [11] and re-split into train dataset with 18597 images(80%) with 9930,3163,2691,1807,655,351 images for the corresponding client and 4650 images(20%) for the validation dataset and test dataset. We also randomly split 20% of the 4650 images as the global validation dataset, and the rest be the test dataset. We measure classification performance through balanced accuracy, defined as the average recall in each class.

B.2 Non-IID setting

The performance of the FL central model in Non-IID FL settings has been a well-known challenge due to the diversity of gradients. Consequently, as in previous works, we focus on the inconsistency and simulate the synthetic Non-IID partitions of Fashion-MNIST and CIFAR-10 datasets. For illustration, the experiment in Section 6.2 set the total number of clients to 100 and the proportion of clients participating in training in each communication round to 0.1. The training data is sorted by label and then divided into 200 shards. Consequently, each shard has 250 images for the CIFAR-10 dataset and 300 images for the Fashion-MNIST dataset. Then, each client is assigned two shards of data, which guarantees that each client can only have 1 or 2 consecutive labels and further ensures the non-identical distribution with each other.

B.3 Vulnerable scenario simulation

We consider the following popular vulnerable scenario based on the Non-IID data setting.

Imbalanced data with the long-tailed distribution. To create the imbalanced version of CIFAR-10, we reduce the number of training samples per class in the original datasets. Then we follow [4] to obtain long-tail distribution with different imbalance ratios (IR), which denote the ratio between the number of samples in the largest and that in the smallest class. Long-tailed imbalance follows an exponential decay in sample sizes across different classes.

Irrelevant data with open-set label noise. In this experiment, we simulate irrelevant clients by injecting label noise. We follow [33] and adopt the method of open-set noise[45], which assigns labels of known categories to data of unknown categories. By way of illustration, we assign the label aeroplane to an image of a truck in the classification task that involves a truck, ship, and automobile. We inject **half** of the training images with label noise in both CIFAR-10 and Fashion-MNIST.

Malicious clients with closed-set label noise. In this experiment, we simulate malicious clients by injecting label noise. The malicious clients flip the local sample labels during training to generate faulty gradients. In particular, the label of each training sample in Byzantine clients is flipped from L to $(L + 1)\%C$, where C is the total categories of labels and $L \in \{0, 1, \cdots, C - 1\}$.

Malicious clients with data noise. In this experiment, we simulate malicious clients by injecting random noise into real, raw data. $D_m = D_h + N(\mu, \sigma^2 I)$. In particular, we set $\mu = (0, \cdots 0) \in \mathbb{R}^d$ and $\sigma = 1$.

Malicious client with gradient poisoning. In this experiment, we simulate malicious clients by sending perturbed gradients. We consider a typical scenario of gradient poisoning called a random Byzantine Attack. Specifically, assuming the original value of a gradient element is a, the value after adding noise is given by a * (1 + b), where b is randomly sampled from a uniform distribution [-0.5, 0.5].

B.4 Hyperparameter choosing

We first validate the impact of weights update rate β as shown in Equation 3. The hyperparameter β limits the updates rate of the adaptive weight of each client, thereby influencing FL global accuracy. We implement AFedSV+ with varying β and explore the averaged convergence global accuracy on CIFAR-10 with open-set label noise. The experimental results in Table 2 reveal that too large or too small weight update rate is not conducive to the model training. Consequently, we set $\beta = 0.3$ for AFedSV/AFedSV+ in the experiments afterwards.

Table 2: Impact of weights update rate.

β	0.1	0.3	0.5	0.7	0.9
Test Accuracy	0.5482	0.5822	0.5784	0.5660	0.5588

B.5 Effectiveness of SV Calculation

We adopt the universal benchmark algorithms, including Monte Carlo algorithm (MC) [6] and Truncated Monte Carlo (TMC) algorithm [14] for approximating Shapley value as baselines. We compare DMC with MC and TMC in the scenario of gradient poisoning. We set the proportion of clients participating in each round to 0.2, i.e., 20 clients are selected in each round ($|\mathcal{N}| = 20$). We compute the average of the Mean Squared Errors (MSEs) to verify the effectiveness of the proposed algorithms. Given benchmark normalized Shapley value $\mathcal{NSV}_i$ and estimated normalized Shapley value $\overline{\mathcal{NSV}_i}$ ($1 \le i \le |\mathcal{N}|$) computed by the proposed algorithms, we compute $MSE(\mathcal{NSV}, \overline{\mathcal{NSV}}) = \frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} (\mathcal{NSV}_i - \overline{\mathcal{NSV}_i})^2$.

Computing the exact normalized Shapley value $\mathcal{NSV}_i$ for evaluation purposes is prohibitively expensive because it grows exponentially with the number of players. Therefore, we use the estimated Shapley value computed by the Monte Carlo algorithm with 2000 sampled permutations (sufficiently large) as the benchmark Shapley value. We conduct the experiment for the first communication round due to the enormous computation cost for the benchmark Shapley value. In addition, we omit the experiment on Fed-ISIC2019 since it only has six clients, which is too few for evaluating Shapley value computation.

Table 3 shows that DMC consistently outperforms baselines in MSE where the first row in the table shows the number of sampled permutations in each algorithm. We observe that the advantage of DMC over baselines is more obvious when the number of samples is small because the estimated normalized SV becomes closer to the accurate normalized SV with increasing samples.

Table 3: MSEs for CIFAR-CNN and FMNIST-CNN (Section B.5).

Dataset	Method	80	160	240	320	400
CIFAR	MC	5.66e-2	3.23e-2	1.41e-2	9.04e-3	8.87e-3
	TMC	3.48e-2	3.38e-2	3.13e-2	3.08e-2	2.71e-2
	DMC (ours)	**1.52e-2**	**1.40e-2**	**1.36e-2**	**8.72e-3**	**7.81e-3**
FMNIST	MC	1.05e-2	9.43-3	7.08e-3	5.25e-3	4.22e-3
	TMC	2.81e-2	2.94e-2	2.33e-2	2.19e-2	2.03e-2
	DMC (ours)	**7.14e-3**	**5.54e-3**	**5.16e-3**	**4.93e-3**	**3.76e-3**

C PROOF

C.1 Proof of Theorem 5.1

Proof. It is easy to understand that if client i has a larger weight than client j, then the selection probability of i should be larger or equal to j, or we can switch the probabilities to get a better solution. Thus, we can know there are no clients with a probability of less than 1 that have larger weights than those with a selected probability of 1. Then, we can enumerate all possible values of $\mathcal{L}^t$ and solve the simpler optimization problem

$$\min \sum_{i \notin \mathcal{L}^t} \frac{(w_i^{t-1})^2}{p_i^t} st. \sum_{i \notin \mathcal{L}^t} p_i^t = m - |\mathcal{L}^t|. \tag{14}$$

Equation (14) can be solved using the Lagrange multipliers method. At last, we can check which solution can minimize equation (8) while the probability of each client is in the proper domain. □