2025 IEEE International Conference on Edge Computing and Communications (EDGE) | 979-8-3315-5559-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/EDGE67623.2025.00023

Empowering Clients: Self-Adaptive Federated Learning for Data Quality Challenges

Zahidur Talukder*, Muhammad Rana†, Keaton Hamm†‡, Mohammad A. Islam*
*Department of Computer Science and Engineering, University of Texas at Arlington
†Department of Mathematics, University of Texas at Arlington
‡Division of Data Science, College of Science, University of Texas at Arlington
{zahidurrahim.talukder, muhammadsohel.rana, keaton.hamm, mislam}@uta.edu

***Abstract*—Federated Learning (FL) enables collaborative model training across distributed clients, but the global model's performance often degrades due to variable data quality and reliability at the local level. Previous approaches mitigate this by restricting or excluding contributions from certain clients, leading to wasted computation and communication resources for those disregarded. In this paper, we introduce FedSRC: Federated Learning with Self-Regulating Clients, a novel framework that optimizes resource use while safeguarding client anonymity. With FedSRC, clients autonomously assess their local training's benefit to the global model using a lightweight checkpoint based on local test loss and a Refined Heterogeneity Index (RHI), deciding their participation in each FL round accordingly. Extensive evaluations across four datasets demonstrate that FedSRC achieves up to 30% savings in communication costs and 55% in computation costs, all while maintaining privacy and enhancing efficiency.**

***Index Terms*—Federated Learning, Self-Regulation, Data Quality, Client Selection, Privacy, Resource Efficiency, Heterogeneity**

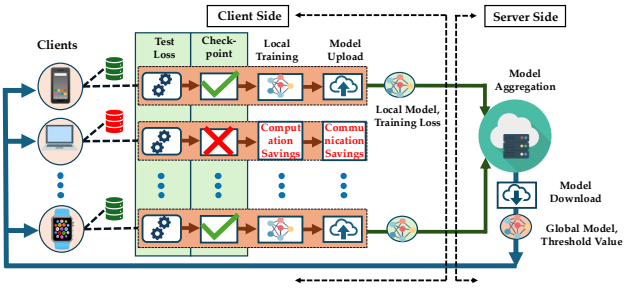


Fig. 1: FedSRC's architecture employs a client-side checkpoint integrating test loss on the global model (GM) with a Refined Heterogeneity Index (RHI). Clients with low-quality data (high test loss) abstain, reducing local computation and communication costs, while RHI ensures fair participation for heterogeneous but valid data.

I. Introduction

Federated Learning (FL) has emerged as a transformative paradigm in distributed machine learning, enabling a multitude of clients—such as smartphones, wearables, and IoT devices—to collaboratively train a shared model without transferring their sensitive data to a central server [1]. Introduced with the Federated Averaging (FedAVG) algorithm, FL facilitates privacy-preserving training by allowing clients to compute local updates on their private datasets and share only model parameters with a central server, which aggregates these updates into a global model. This approach has gained traction in real-world applications, powering features like predictive text in Google's Gboard [2] and voice recognition in Apple's Siri [3]. By harnessing diverse, large-scale data from millions of devices, FL offers scalability and privacy advantages over traditional centralized training, making it a cornerstone of modern privacy-conscious AI.

Motivation. Despite its promise, FL's decentralized architecture introduces significant challenges, particularly related to the quality and reliability of client-generated data. In FL, clients operate in silos, collecting data from varied sources—such as sensors, cameras, or user inputs—using hardware of differing capabilities. For instance, mobile and wearable devices, ideal for FL due to their privacy-sensitive nature, often feature sensors ranging from low-cost components in budget models to advanced systems in premium devices [4], [5]. A recent study by Facebook revealed thousands of unique hardware configurations among its app users [6], underscoring this diversity. Over time, hardware evolution has improved data quality, with newer sensors offering higher resolution and accuracy [7], [8]. However, this progress is uneven, and older or malfunctioning devices can produce noisy, incomplete, or erroneous data [9]. Worse, malicious actors can exploit FL's distributed nature, corrupting data to poison the global model [10], as seen in adversarial attacks targeting crowdsourced systems.

These data quality issues are exacerbated by statistical heterogeneity, often termed non-IID (non-independent and identically distributed) data, where client datasets differ in distribution, class balance, or feature representation [11]. For example, one client's wearable might track fitness data skewed toward running, while another's captures sedentary behavior, leading to divergent local models. FedAVG, the foundational FL algorithm, mitigates this by weighting client updates by dataset size [1], but it assumes uniform data quality—an assumption frequently violated in practice. Poor-quality data slows convergence, introduces model drift during aggregation, and degrades global performance, posing a critical barrier to FL's widespread adoption in real-world, heterogeneous environments.

Limitations of existing approaches. Researchers have

2767-9918/25/$31.00 ©2025 IEEE
DOI 10.1109/EDGE67623.2025.00023

proposed several strategies to address FL's data quality and heterogeneity challenges, yet each falls short in critical ways. One prominent approach adjusts aggregation weights at the server, reducing the influence of clients with unreliable data [12]–[15]. Techniques like SCAFFOLD [13] correct for client drift, while Byzantine-robust methods [14] filter outliers from malicious or faulty updates. Although effective in improving model accuracy, these methods require all clients to perform local training and upload updates, regardless of their eventual contribution. For resource-constrained devices—common in FL—this incurs significant computational overhead (e.g., training neural networks on limited CPU/GPU) and communication costs (e.g., uploading over bandwidth-constrained networks). Clients with poor data, whose updates are down-weighted or discarded, essentially waste these resources, undermining FL's efficiency.

An alternative strategy, active client selection, seeks to preempt this waste by profiling clients and selecting only those with favorable updates [16], [17]. By analyzing update quality or historical performance, the server cherry-picks participants, reducing unnecessary training. However, this requires tagging updates with client identifiers, enabling longitudinal tracking that compromises anonymity—a cornerstone of FL's privacy guarantee. Such profiling heightens vulnerability to model-inversion attacks, where adversaries reconstruct private data from gradients [18], as demonstrated in recent security analyses. Moreover, traditional data preprocessing—standard in centralized ML to handle quality issues—is impractical in FL. Clients lack visibility into the global data distribution, risking over-correction (e.g., discarding valid outliers when bad data dominates locally), and their preprocessing capabilities vary widely, from sophisticated smartphones to basic IoT nodes.

Our contribution. To overcome these limitations, we introduce FedSRC (**Fed**erated Learning with **S**elf-**R**egulating **C**lients), a groundbreaking framework that shifts the burden of quality control to clients themselves, preserving both efficiency and privacy. In FedSRC, clients autonomously decide their participation using a lightweight checkpoint mechanism, illustrated in Figure 1. This checkpoint evaluates local test loss on the global model—reflecting data quality—and adjusts it with a Refined Heterogeneity Index (RHI) to account for non-IID effects. Clients with high test loss, indicating poor-quality data, abstain from training and uploading, achieving up to 30% communication and 55% computation savings across four datasets. Unlike server-side weighting, FedSRC avoids wasteful resource use; unlike active selection, it requires no client profiling, maintaining anonymity by adhering to standard FL protocols.

Implementing client-side regulation poses two key challenges. First, clients have only local data, lacking the global context to assess quality statistically. Second, the checkpoint must be computationally lightweight for edge devices. We address these with a novel test loss-based policy: clients test the global model locally, exiting if loss exceeds a personalized RHI-adjusted threshold. Our empirical insight—that poor-quality data consistently yields high test loss—drives this design, while RHI ensures heterogeneous but valid data isn't penalized. FedSRC integrates seamlessly with server-side defenses against poisoning [10], enhancing robustness without sacrificing privacy.

To our knowledge, FedSRC is the first FL approach empowering clients with strategic decision-making. We provide the first theoretical analysis of FL convergence under data quality constraints, proving that selective participation accelerates convergence by filtering detrimental updates. Extensive evaluations on `MNIST`, `CIFAR10`, `FEMNIST`, and `SHAKESPEARE` datasets confirm FedSRC's efficacy, demonstrating significant resource savings and performance gains over baselines like FedAVG and robust aggregation methods. By rethinking FL's client-server dynamic, FedSRC paves the way for scalable, privacy-preserving learning in real-world, heterogeneous settings. The source code and supplementary theoretical proofs can be accessed at **FedSRC** [19].

II. Preliminaries

A. Scope and Threat Model

This work targets data quality challenges in Federated Learning (FL), focusing on two primary sources of poor-quality data at the client level: hardware-related issues (e.g., malfunctions or low-quality sensors) and adversarial tampering with client sensors or devices. Our threat model assumes an attacker can degrade data quality by manipulating a client's sensor or hardware, but the client's device executing local FL training remains uncompromised. This assumption aligns with FedSRC's client-centric design, which relies on clients adhering to the protocol. However, we note that this can be relaxed without loss of generality by pairing FedSRC with server-side defenses against malicious clients [20], [21], though such clients may not self-regulate participation as intended. Our focus excludes Byzantine adversaries controlling client devices, prioritizing a client-level solution for data quality over broader trust assumptions.

B. Federated Learning

Problem formulation. In an FL system, K clients, each holding a local dataset $\mathcal{D}_k$, collaborate to minimize a global loss function:

$$F(w) = \sum_{k=1}^{K} p_k F_k(w), \quad p_k = \frac{|\mathcal{D}_k|}{\sum_{k=1}^{K} |\mathcal{D}_k|}, \tag{1}$$

where $F_k(w) = \frac{1}{|\mathcal{D}_k|} \sum_{\xi \in \mathcal{D}_k} f(w, \xi)$ is the local loss for client k, $f(w, \xi)$ is the loss for a data sample ξ under model parameters w, and p_k weights each client by dataset size. The goal is to optimize $F(w)$ across all clients' data without centralizing it, preserving privacy. Traditional FL assumes uniform client contributions, but poor-quality or unreliable data—due to hardware faults, sensor noise, or attacks—introduces noise and bias, skewing updates and hindering convergence.

Solution. The Federated Averaging (FedAVG) algorithm [1] addresses this optimization efficiently through iterative rounds. In round t, a fraction M of clients ($m = M \cdot K$) is randomly selected as $S^{(t)}$. Each selected client performs τ

local Stochastic Gradient Descent (SGD) iterations on $\mathcal{D}_k$ and sends its updated model to the server, which aggregates them into a new global model. The update for client k is:

$$w_k^{(t+1)} = \begin{cases} w_k^{(t)} - \eta_t g_k(w_k^{(t)}, \xi_k^{(t)}), & \text{if } (t+1) \bmod \tau \neq 0 \\ \frac{1}{m} \sum\limits_{l \in S^{(t)}} \left(w_l^{(t)} - \eta_t g_l(w_l^{(t)}, \xi_l^{(t)}) \right) = \bar{w}^{(t+1)}, & \text{o/w} \end{cases} \tag{2}$$

where $w_k^{(t)}$ is the local model, η_t is the learning rate, and $g_k(w_k^{(t)}, \xi_k^{(t)}) = \frac{1}{b} \sum_{\xi \in \xi_k^{(t)}} \nabla f(w_k^{(t)}, \xi)$ is the stochastic gradient over a mini-batch $\xi_k^{(t)}$ of size b, sampled from $\mathcal{D}_k$. The server updates the global model only after τ iterations, averaging the local updates.

While FedAVG scales effectively, it struggles with poor-quality data. Noisy or corrupted $\mathcal{D}_k$ yields skewed gradients, degrading the global model and slowing convergence, especially under non-IID conditions where data distributions vary widely across clients. This motivates a need for selective participation to mitigate the impact of unreliable updates.

C. Improving FL with Data Quality Issues

Biased aggregation. Equal treatment of clients in FedAVG exacerbates performance degradation when data quality varies [15], [22]. Unlike centralized ML, where preprocessing can mitigate quality issues, FL's local training amplifies the effect of bad data on model updates, harming aggregation. Biased aggregation methods [12], [13], [15] counter this by down-weighting or discarding updates from unreliable clients, improving global performance. However, this raises fairness concerns: clients with poor data, assigned low weights, contribute minimally yet still incur training costs, while FL aims for collaborative benefit. We argue that excluding such clients is justified—poor data already limits their local performance, and including them risks degrading the global model for all.

This differs from inclusive biased aggregation [16], [23], which favors poorly performing clients to balance non-IID disparities, assuming distribution differences rather than quality defects. Our focus is on quality-driven exclusion, prioritizing overall model efficacy over universal inclusion.

Self-regulating clients. Centralized biased aggregation wastes resources by requiring all clients to train and upload, even those later discarded. We propose biased aggregation via client selection, where only clients with high-quality data participate. Server-side selection, however, requires profiling, breaching anonymity. Instead, FedSRC empowers clients to self-regulate, profiling their own data quality and opting out if detrimental, preserving privacy and efficiency.

Challenges. Self-regulation faces three hurdles: (1) clients access only their own data, limiting quality assessment; (2) they lack visibility into other clients' updates, unlike the server; and (3) the strategy must be lightweight for resource-constrained devices. In the next section, we develop a policy to overcome these, leveraging local test loss and a Refined Heterogeneity Index (RHI) for effective, efficient client selection.

III. Our Solution

A. FedSRC: FL with Self-Regulating Clients

Client classification. To implement a client selection strategy, we first need to define who should be considered as a "bad client" and discarded from model aggregation. Since FL clients' data is private, we cannot directly assess clients' data quality. Hence, we classify bad clients based on the impact of their inclusion in the global model as follows:

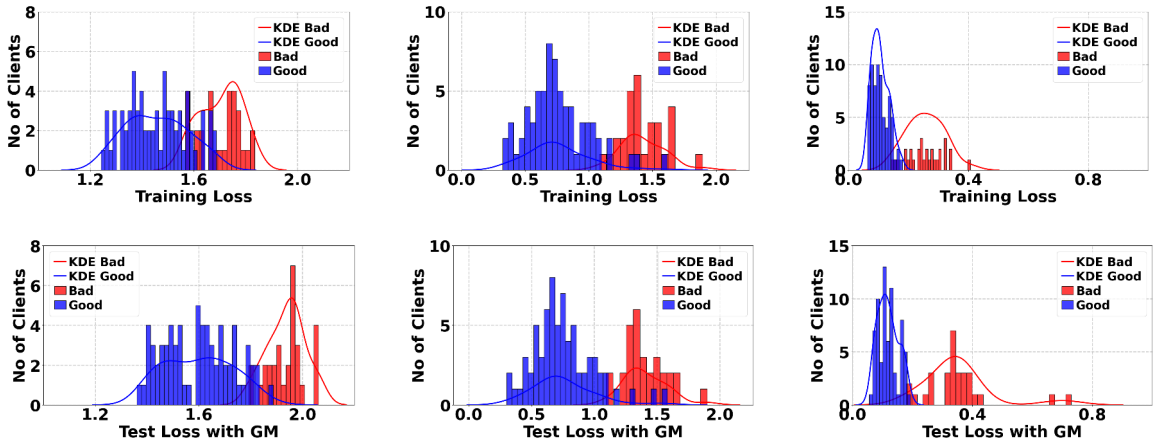
Definition III.1 (ϵ-Bad Client)**.** An FL client is ϵ-bad if its inclusion in the unbiased global aggregation increases the converged global objective loss by more than ϵ.

The parameter ϵ in our definition serves two purposes. *First*, it allows us to set the degree of negative impact that constitutes a bad client. *Second*, it can absorb the variation of global objective loss (for good client participation) due to non-IID data distribution and the sequence of client participation in the training rounds. Our definition, however, can only be an *approximate* definition of bad clients since the impact of bad data (from bad clients) and non-IID data (from good clients) on the global loss is not always distinguishable. Nevertheless, our definition serves to develop a client selection strategy for improving the global model performance, albeit there is a possibility (tunable through ϵ) of treating some good clients with non-IID data as bad clients.

Client selection strategy. The client classification in Definition III.1 requires $N+1$ complete[1] FL training, rendering it impractical due to its huge computation and communication overheads. Hence, we need to develop a lightweight approach for identifying good and bad clients at the client level that can serve as a proxy for Defintion III.1.

We devise FedSRC's client selection strategy based on our empirical observation that *clients with bad quality data suffer from worse local test performance when vetted against the global model (GM)*. More specifically, we find that the bad clients tend to have a higher test loss on the shared global model across training rounds, even when they are included in the model aggregation. To demonstrate this, we run experiments on several data sets with 30% of clients suffering from noisy data in Fig. 2. The results reveal two distinct distributions of good and bad clients, with some degree of overlap between them. Notably, we observe that this distribution pattern is consistent for both training and test losses when using the GM, suggesting that the threshold calculated based on training losses can be effectively transferred to test losses. Further analysis into the reasons for the overlap between good and bad clients indicates that clients with higher data heterogeneity, despite having noisy data, may outperform good clients with lower heterogeneity. To further investigate this phenomenon, we utilize the Refined Heterogeneity Index (RHI) to quantify data heterogeneity and generate bad clients with varying RHI levels, subsequently evaluating their test losses. As shown in Fig. 3, clients with higher heterogeneity can have lower test losses with the GM

[1]Global model convergence after many rounds of training.



(a) CIFAR10 (b) FEMNIST (c) MNIST

Fig. 2: Clients' training losses and test losses on the global model (GM) of one training round in the presence of bad clients show similar distribution and a test-loss-based cutoff can separate the good and bad clients reasonably well.

compared to good clients with lower heterogeneity. Thus adding RHI to fine-tune the threshold for individual clients make our design more robust against data heterogeneity.

A test loss-based approach satisfies our requirements for client-side regulation since it can be a reasonably accurate proxy for Definition III.1. Consequently, we set our client selection strategy as follows: *during FL training, we select the clients with test loss lower than a given threshold.*

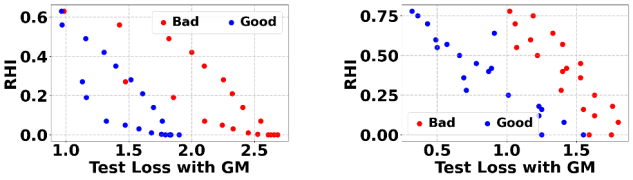
Threshold-based participation. While we would like the clients to implement our selection strategy and set the participation threshold themselves, they do not have access to the training losses of other clients.

Hence, in FedSRC, we engage the central server to *anonymously* collect the training losses of participating clients' of a particular FL round and determine the participation threshold for the next round. Thus in our setup, participating clients send their training losses along with the trained model to the central server but do not share their RHI value. Here, we consider that the clients send their model updates and training losses anonymously using an anonymous communication protocol such as Secure Multi-Party Computation (SMPC) [24], i.e., the server can neither track which updates/losses are supplied by which client nor pair training losses with their corresponding model updates. Note that such anonymous participation cannot be implemented in central serve-based active client selection.

Server-side threshold computation: In FedSRC, the server computes a participation threshold for round t using training loss statistics from clients active in round $t-1$. This threshold, denoted $\phi^{(t)}$, guides clients in self-regulating their involvement, optimizing resource use while filtering poor-quality contributions. Let $S^{(t-1)}$ be the set of participating clients in round $t-1$, each providing a local training loss $\text{Tr\_loss}_k^{(t-1)}$. The server calculates $\phi^{(t)}$ in three steps:

First, the *median training loss* establishes a baseline:

$$\phi_{\text{base}}^{(t)} = \text{median}\big(\{\text{Tr\_loss}_k^{(t-1)}\}_{k\in S^{(t-1)}}\big),$$



(a) CIFAR10 dataset (b) FEMNIST dataset.

Fig. 3: Bad clients with higher RHI outperform good clients with lower RHI using the global model (GM).

where median$(\cdot)$ selects the middle value of the ordered loss set, offering a robust central tendency resistant to outliers.

Next, the *loss variability* quantifies the spread of these losses:

$$\sigma^{(t)} = \sqrt{\frac{1}{|S^{(t-1)}|}\sum_{k\in S^{(t-1)}}\big(\text{Tr\_loss}_k^{(t-1)} - \phi_{\text{base}}^{(t)}\big)^2},$$

where $|S^{(t-1)}|$ is the number of clients in $S^{(t-1)}$, and $\sigma^{(t)}$ is the standard deviation, reflecting the diversity in client performance from round $t-1$.

Finally, an *adaptive adjustment* combines these to set the threshold:

$$\phi^{(t)} = \phi_{\text{base}}^{(t)} + \alpha\sigma^{(t)},$$

where $\alpha \in [0, \infty)$ (e.g., 1.5) is a scaling factor that dynamically adjusts the threshold $\phi^{(t)}$ based on loss variability. A larger α increases $\phi^{(t)}$, loosening participation criteria and allowing more clients—including those with potentially poor-quality data—to join round t, whereas a smaller α reduces $\phi^{(t)}$, tightening criteria and restricting participation, which may exclude even high-quality clients. To balance this, clients indirectly influence participation by adapting α: if the participation rate falls below a predefined threshold (e.g., 50% of K), α increases to include more clients; conversely, if excessive bad clients participate, α decreases to enhance selectivity. The server broadcasts $\phi^{(t)}$ to

all clients, who evaluate their local test loss against it to decide participation, optimizing the inclusion of reliable contributors.

Calculating RHI. Each client computes a *Refined Heterogeneity Index (RHI)* to measure both the number of classes and the distribution of samples across them. The original Heterogeneity Index (HI) [25] is defined as

$$\text{HI}_k = 1 - \frac{c_k - 1}{C_{\text{max}} - 1},$$

where c_k is the number of classes held by client k, and C_{max} is the total number of classes in the dataset. To refine this, we incorporate the entropy of the local class distribution, computed as

$$\text{Entropy}_k = -\sum_{i=1}^{c_k} d_i \log d_i$$

where d_i is the fraction of data points in class i. The normalized entropy is then

$$\text{NE}_k = \frac{\text{Entropy}_k}{\log c_k},$$

which reaches its maximum value of 1 when the data is uniformly distributed. Finally, the Refined Heterogeneity Index is computed as

$$\text{RHI}_k = \kappa \cdot \text{HI}_k + (1 - \kappa)(1 - \text{NE}_k),$$

where $\kappa \in [0, 1]$ balances the contribution of class count and class distribution. Clients use their RHI to adjust participation thresholds during training.

Client-side personalization: In FedSRC, each client k customizes its participation threshold for round t, denoted $\Phi_k^{(t)}$, based on the server-provided threshold $\phi^{(t)}$ and its local data characteristics. This personalization enables clients to self-regulate efficiently, balancing data quality and diversity. The threshold is defined as:

$$\Phi_k^{(t)} = \phi^{(t)} \times (1 - \beta \cdot \text{RHI}_k),$$

where RHI_k is the Refined Heterogeneity Index of client k, and $\beta \in [0, 0.9]$ (e.g., 0.3) is a tuning parameter controlling RHI's impact. The RHI, a value between 0 and 1, reflects the client's data diversity (e.g., based on label distribution). A high RHI_k reduces $\Phi_k^{(t)}$, lowering the bar for participation and accommodating diverse clients despite potentially higher test losses, while a low RHI_k keeps the threshold closer to $\phi^{(t)}$.

To decide participation, client k evaluates the global model $w^{(t)}$ on its local dataset, computing the test loss:

$$\text{Test\_loss}_k^{(t)} = \frac{1}{|\mathcal{D}_k|} \sum_{\xi \in \mathcal{D}_k} f(w^{(t)}, \xi).$$

If $\text{Test\_loss}_k^{(t)} \leq \Phi_k^{(t)}$, the client proceeds with local training and submits its update to the server, contributing to the global model. Otherwise, it abstains, conserving computation and communication resources when its data is unlikely to benefit the model. The server imposes a deadline for responses, ensuring the round progresses with participating clients, and collects their training losses $\{L_1^{(t)}, L_2^{(t)}, \ldots, L_m^{(t)}\}$ (where $m \leq K$ is the number of participants) to inform the threshold for round $t + 1$.

Algorithm 1 FedSRC

Server Side:
Initialize $(w^{(0)}, \alpha, \beta)$
for each round $t = 0$ to $T - 1$ **do**
  if $t = 0$ **then**
    $S^{(t)} \leftarrow K$
  else
    Select subset $S^{(t)}$ from K clients at random
  end if
  Send $w^{(t)}$, $\phi^{(t)}$ to $S^{(t)}$
  for each client $k \in S^{(t)}$ **do**
    $w_k^{(t+1)}$, $L_k^{(t)} \leftarrow$ **ClientUpdate**$(w^{(t)})$
  end for
  Update $w^{(t+1)}$, compute med(Tr_loss)$^{(t)}$, $\sigma^{(t)}$
end for
Client Side:
ClientUpdate $(w^{(t)})$
Client computes $\text{Test\_loss}_k^{(t)}$ with GM
Client computes $\Phi_k^{(t)}$ using RHI
if $\text{Test\_loss}_k^{(t)} \leq \Phi_k^{(t)}$ **then**
  for each local epoch e from 1 to E **do**
    Update $w_k^{(t+1)}$, compute $L_k^{(t)}$
  end for
end if
return $w_k^{(t+1)}$, $L_k^{(t)}$

Overhead of FedSRC's implementation: The implementation of FedSRC introduces minimal computational overhead for clients. This is primarily due to the addition of a single test on a subset of the client's training data. However, the computational cost of this test is negligible when compared to the substantial training cost savings achieved. Additionally, the initial minibatch error, recorded before modifying the global weights, can be leveraged to estimate the client's test loss using a randomly sampled minibatch from the training data. For clients with fixed data distributions, the Refined Heterogeneity Index (RHI) only needs to be computed once. As a result, clients can make efficient participation decisions in any FL training round with minimal computational impact.

Client re-inclusion in FedSRC: To ensure that good clients with highly non-IID data (e.g., those exhibiting skewness in feature space) are not permanently excluded, FedSRC incorporates a re-inclusion mechanism. This strategy allows any excluded client to participate periodically, such as 10% of the time, or automatically re-includes a client after a predefined number of exclusions. This ensures that rare or unique data distributions are not overlooked during training, contributing to a more robust and inclusive federated learning process.

B. Theoretical Analysis

Here, we prove the convergence of FedSRC and discuss how our client selection policy affects the convergence. To facilitate

our analysis, We make the following assumptions:

Assumption III.2. $F_1, \ldots, F_k$ are all L-smooth, i.e., for all v and w,

$$F_k(v) \leq F_k(w) + (v-w)^T \nabla F_k(w) + \frac{L}{2}\|v-w\|_2^2.$$

Assumption III.3. $F_1, \ldots, F_k$ are all μ-strongly convex, i.e., for all v and w,

$$F_k(v) \geq F_k(w) + (v-w)^T \nabla F_k(w) + \frac{\mu}{2}\|v-w\|_2^2.$$

Assumption III.4. For the mini-batch ξ_k uniformly sampled at random from $\mathcal{D}_k$ of user k, the resulting stochastic gradient is unbiased; that is, $\mathbb{E}[g_k(w_k, \xi_k)] = \nabla F_k(w_k)$. Also, the variance of stochastic gradients is bounded: $\mathbb{E}[\|g_k(w_k, \xi_k) - \nabla F_k(w_k)\|^2] \leq \sigma^2$ for all $k = 1, \ldots, K$.

Assumption III.5. The stochastic gradients' expected squared norms are uniformly bounded, i.e., $\mathbb{E}[\|g_k(w_k, \xi_k)\|^2] \leq G^2$ for $k = 1, \ldots, K$.

Denote by $\mathcal{B}$ the set of ϵ-bad clients for a fixed $\epsilon > 0$, and let $\mathcal{G}$ be the set of good clients (i.e., those that are not ϵ-bad. By Definition III.1, these sets are fixed.

Since our assumption is that there are bad clients whose updates adversely affect the global model, our convergence analysis takes this into account by separating the good and bad clients in all terms defined below. We utilize similar ideas to [16] by defining a local-global objective gap and a skewness of biased selection of clients who send their model update to the central server. However, in contrast to prior work, our definitions are in terms of the good (or potentially bad) clients, which allows us to understand the effect of our client selection strategy in the context of our problem setup.

We define global loss for two client sets: $F_g(w) = \sum_{k \in \mathcal{G}} p_k F_k(w)$ for the good clients in $\mathcal{G}$, and similarly define F_b for the bad clients in $\mathcal{B}$. The optimal global losses for good and bad clients are $F_g^* = \min_w F_g(w)$ and $F_b^* = \min_w F_b(w)$. Additionally, we define the global model optimum $w^* = \arg\min_w F(w)$, and the client-level optima $w_k^* = \arg\min_w F_k(w)$ for each client k.

Definition III.6 (Local-Global Objective Gap)**.** We define the *local-global objective gap* for good clients as follows:

$$\Gamma_g = F_g^* - \sum_{k \in \mathcal{G}} p_k F_k^* = \sum_{k \in \mathcal{G}} p_k (F_k(w^*) - F_k(w_k^*)) \geq 0.$$

For highly non-IID data, Γ_g is non-zero, and larger Γ_g implies higher data heterogeneity. $\Gamma_g = 0$ implies consistent optimum models among the clients and the central server.

Definition III.7 (Selection Skewness)**.** Let w be the current weights of the global model, and π be any client selection strategy. We let $S(\pi, w)$ denote the selected clients using selection strategy π and define the skewness of the client selection strategy π for good and bad clients via

$$\rho_g(S(\pi,w),w') = \frac{\mathbb{E}_{S(\pi,w)}\left[\frac{1}{p}\sum_{k \in S(\pi,w) \cap \mathcal{G}} \left(F_k(w') - F_k^*\right)\right]}{F_g(w') - \sum_{k \in \mathcal{G}} p_k F_k^*},$$

$$\rho_b(S(\pi,w),w') = \frac{\mathbb{E}_{S(\pi,w)}\left[\frac{1}{q}\sum_{k \in S(\pi,w) \cap \mathcal{B}} \left(F_k(w') - F_k^*\right)\right]}{F_g(w') - \sum_{k \in \mathcal{G}} p_k F_k^*},$$

where p is the number of selected good clients, q is the number of selected bad clients, and $m = p + q$. Above, the current global model weights w influences the selection strategy π, while w' is the global model weight at which the selection skewness is evaluated. $\mathbb{E}_{S(\pi,w)}[\cdot]$ represents the expectation over the randomness from the selection strategy π in determining $S(\pi, w)$.

Note that the denominator of both ρ_g and ρ_b are the same and represent the current gap between the local and global models for good clients only. This is because we do not wish to select bad clients, and their local-global objective gap should not influence our convergence analysis. The following terms are useful for providing a concrete error bound in the main theorem below. $\bar{\rho}_g = \min_{w,w'} \rho_g(S(\pi,w),w')$ and $\tilde{\rho}_g = \max_w \rho_g(S(\pi,w),w^*)$.

We define $\bar{\rho}_b$ and $\tilde{\rho}_b$ similarly.

Theorem III.8. *Under the Assumptions stated above, for a learning rate $\eta_t = \frac{1}{\mu(t+\gamma)}$ with $\gamma = \frac{4L}{\mu}$, and for client selection strategy π that selects the same number of good and bad clients (p and q, respectively) after time T, the error of federated learning with self-regulating clients satisfies, for every $t \geq T$,*

$$\mathbb{E}[F(\bar{w}^{(t)})] - F^* \leq \underbrace{\frac{1}{(t+\gamma)}\left[\frac{4L(32\tau^2G^2 + \sigma^2/m)}{3\mu^2\bar{\rho}_g} + \right.}_{\text{Vanishing Term}}$$
$$\underbrace{\left.\frac{8L^2\Gamma_g}{\mu^2}\frac{\bar{\rho}_b}{\bar{\rho}_g} + \frac{L(\gamma+1)(\|\bar{w}^{(1)} - w^*\|^2)}{2}\right]}_{\text{Vanishing Term}}$$
$$+ \underbrace{\frac{8L\Gamma_g}{3\mu}\left(\frac{p\tilde{\rho}_g + q\tilde{\rho}_b}{m\bar{\rho}_g} - 1\right)}_{\text{Bias Term}} \tag{3}$$

To the best of our knowledge, Theorem III.8 provides the first theoretical convergence bound for Federated Averaging in the presence of adversarial or low-quality clients. The complete proof is available in the supplementary material at **FedSRC** [19].

Takeaway 1: Effect of the client selection strategy. First, for an unbiased client selection strategy (clients participate in the model update uniformly at random), both good and bad clients will provide a model update. As the model's training progresses, the loss of the good clients decreases, whereas the loss of the bad clients does not improve. This results in a decreasing ρ_g, but increasing ρ_b, both of which negatively affect both the rate of convergence of the vanishing term and the magnitude of the bias term in (3). A biased client selection strategy that is able to discard clients with higher loss will

ensure an increase in the number of good clients selected and decrease in number of bad clients selected, which reduces the value of ρ_b and increases the value of ρ_g, resulting in both faster convergence and smaller bias.

Takeaway 2: Reducing ρ_b and increasing ρ_g for faster convergence. Under our model for good and bad clients, if our selection strategy prioritizes client updates for those with small test loss value F_k, the number of bad clients selected in $S(\pi, w)$ will be smaller, which results in larger ρ_g but smaller ρ_b. Consequently, the first two terms in the vanishing term of Theorem III.8 will be smaller, leading to faster convergence compared to an unbiased selection strategy.

Takeaway 3: Bias term. Similarly, a client selection strategy that prioritizes lower-loss clients will reduce the bias term as p increases. Indeed, $\widetilde{\rho}_b$ should be larger than $\widetilde{\rho}_g$ for a given selection strategy, so decreasing q, the number of bad clients selected, decreases the numerator significantly, even as p increases. Likewise, as p increases based on the selection strategy, the denominator increases as well, thereby decreasing the bias term.

IV. Evaluation

A. Settings

Dataset and model description. We utilize four prominent datasets: `MNIST` [26], `CIFAR10` [27], `FEMNIST` [28], and `SHAKESPEARE` [28], which are widely referenced in the literature [1], [29], [30]. For the `MNIST` and `CIFAR10` datasets, we create non-IID settings by assigning each client a dominant class comprising 80% of their data, with the remaining 20% distributed among the other classes. In an extreme scenario, each client receives data from only two classes. The `FEMNIST` and `SHAKESPEARE` datasets are naturally non-IID. For handwriting classification in `MNIST` and `FEMNIST`, we use a multilayer perceptron (MLP). For `CIFAR10` image classification, we employ a Convolutional Neural Network (CNN), and for the next character prediction in `SHAKESPEARE`, we utilize a Recurrent Neural Network (RNN). The `MNIST` and `CIFAR10` models are evaluated on their respective test datasets, while for `FEMNIST` and `SHAKESPEARE`, each client's test data is used to evaluate the global performance. Dataset is tabulated in I.

Evaluation scenarios. We consider three scenarios reflecting potential data corruption due to sensor quality, malfunction, and aging. *Label shuffling:* It can be referred to as random sensor malfunction, leading to assigning random labels to data. *Label flipping:* It refers to mislabeling data, leading to the same mislabel across all the client data. *Noisy data:* It results from hardware quality in the feature space. To simulate this, we added Gaussian noise to the feature and then clipped the value within the desired feature space level. As the default configuration for our evaluation, we use a mix of 70% good clients with 30% bad clients. The bad clients are equally divided among the three cases.

Benchmark algorithms. To assess the performance of FedSRC, we compare it with several benchmark algorithms. FedAVG [1] is the standard federated averaging technique, assigning client weights based on dataset size. Median [14] employs a Byzantine-robust aggregation rule, computing the median of each parameter independently. Trimmed Mean [14] is another Byzantine-robust aggregation approach, sorting parameters and averaging the middle values after removing extremes. FedASL [15] automatically assigns weights to clients based on the median of their training losses, prioritizing clients within a predefined "good zone" around the median. Krum [31] operates by calculating the Euclidean distance norms between each client's model weights and those of others, removing the highest value for each client, and averaging the rest to select the next global model.

TABLE I: Dataset details.

Dataset	Training	Test	# Clients	Distribution
`MNIST`	60,000	10,000	300	IID/non-IID
`CIFAR10`	50,000	10,000	300	IID/non-IID
`FEMNIST`	341,873	40,832	3,383	non-IID
`SHAKESPEARE`	16,068	2,356	715	non-IID

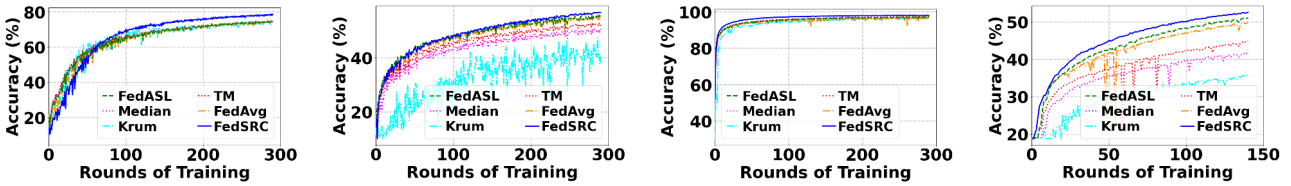
Computing infrastructure. The experiments are conducted using two tower servers, each equipped with two NVIDIA 3080 GPUs, 256 GB of memory, and 1024 cores. The models were developed and trained using TensorFlow Keras, leveraging the high computational power and parallel processing capabilities of this setup.

B. Results

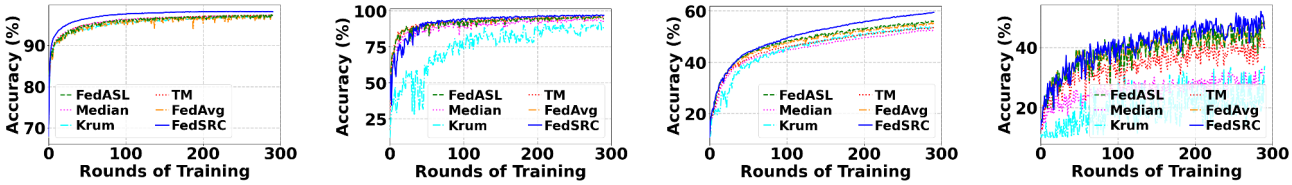
Comparison with benchmark algorithms. We assess the efficacy of FedSRC by comparing it with established federated learning algorithms—Trimmed Mean, Krum, FedASL, and FedAVG—across diverse datasets including `FEMNIST`, `CIFAR10`, `MNIST`, and `SHAKESPEARE`. Experiments are conducted under a standard scenario with 30% of clients exhibiting corrupted data. In FedSRC, the threshold $\phi^{(t)}$ is dynamically tuned using α, adjusted to maintain a 70% participation rate (reflecting the proportion of good clients), alongside a fixed $\beta = 0.5$ to optimize client selection based on the Refined Heterogeneity Index (RHI). For comparison, Trimmed Mean and Krum mitigate corruption by blocking 30% of clients, FedASL excludes clients beyond one standard deviation (approximately 32% exclusion), and FedAVG, the baseline, retains all clients, rendering it highly susceptible to corrupted data influence.

Table II presents the test accuracy, while Table III details the test loss of the global model on uncorrupted test data across all algorithms and datasets. Figure 4 underscores that FedSRC consistently surpasses the benchmarks, delivering superior global model accuracy. These findings affirm FedSRC 's effectiveness in managing corrupted data, enhancing performance while preserving client anonymity and optimizing resource utilization through adaptive participation control.

Computation and communication savings. To quantify the computational and communication savings achieved by FedSRC, we conduct experiments across various proportions of corrupted clients using different datasets. Clients need to do a forward pass (inference loss) of the first batch only to



(a) FEMNIST non-IID (b) CIFAR10 non-IID (c) MNIST non-IID (d) SHAKESPEARE non-IID



(e) MNIST IID (f) MNIST extreme (g) CIFAR10 IID (h) CIFAR10 extreme

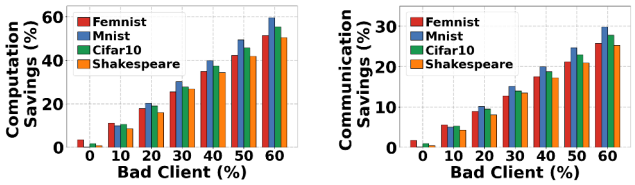
Fig. 4: Comparison of global accuracy of FedSRC with benchmark algorithms.

TABLE II: Comparison of the accuracy of FedSRC with benchmark algorithms.

Algorithms	MNIST IID	MNIST non-IID	MNIST Extreme	CIFAR10 IID	CIFAR10 non-IID	CIFAR10 Extreme	FEMNIST	SHAKESPEARE
FedASL	0.973	0.971	0.942	0.556	0.556	0.488	0.746	0.511
FedAVG	0.971	0.969	0.936	0.548	0.552	0.491	0.749	0.499
Krum	0.969	0.963	0.891	0.534	0.411	0.339	0.742	0.357
Median	0.972	0.970	0.929	0.524	0.504	0.275	0.742	0.417
Trimmed Mean	0.974	0.974	0.957	0.533	0.529	0.411	0.743	0.449
FedSRC	**0.981**	**0.979**	**0.969**	**0.593**	**0.568**	**0.502**	**0.780**	**0.524**

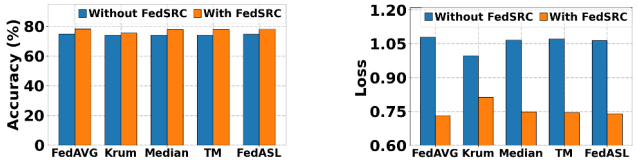
TABLE III: Comparison of the loss of FedSRC with benchmark algorithms.

Algorithms	MNIST IID	MNIST non-IID	MNIST Extreme	CIFAR10 IID	CIFAR10 non-IID	CIFAR10 Extreme	FEMNIST	SHAKESPEARE
FedASL	0.117	0.121	0.287	1.313	1.328	1.494	1.064	1.655
FedAVG	0.174	0.138	0.396	1.362	1.384	1.543	1.080	1.699
Krum	0.098	0.116	0.342	1.330	1.815	2.183	0.996	2.315
Median	0.096	0.112	0.329	1.429	1.530	2.010	1.066	2.027
Trimmed Mean	0.116	0.115	0.229	1.395	1.530	1.775	1.071	1.895
FedSRC	**0.064**	**0.072**	**0.119**	**1.172**	**1.233**	**1.495**	**0.731**	**1.607**



(a) Computation savings (b) Communication savings

Fig. 5: Client-side savings of FedSRC.



(a) Accuracy (b) Loss

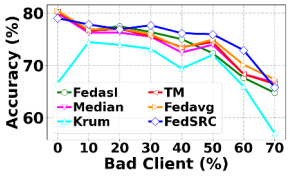
Fig. 6: Performance comparison of integrating FedSRC with existing algorithms for FEMNIST.

decide on participation. This initial computational step, while adding minor overhead, becomes increasingly negligible as the dataset size grows, relative to the overall savings achieved. For datasets with smaller average batch sizes, such as FEMNIST (average of 5 batches), the proportional computational savings are slightly lower compared to larger datasets like MNIST (15 batches) and CIFAR10 (13 batches).

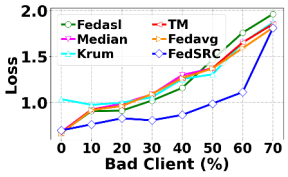
Despite these variations, Fig. 5 demonstrates that FedSRC achieves substantial computational savings, reducing local computational costs by up to 55% when higher proportions of clients are malicious.

In terms of communication savings, FedSRC enables clients to abstain from participation. As Fig. 5 highlights, FedSRC achieves communication savings of up to 30% when 60% of clients are bad. Unlike computation savings, these communication savings are independent of dataset size, as they are solely determined by the proportion of abstaining clients. Together, these findings underscore FedSRC's efficiency in reducing resource consumption while maintaining robust federated learning performance.

Integration with existing algorithms. We demonstrate the effectiveness of integrating FedSRC with other algorithms by implementing it at the client level while maintaining aggregation protocols such as FedAVG, FedASL, Trimmed

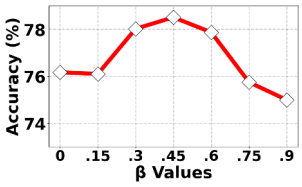


(a) Accuracy with different percentages of bad clients.

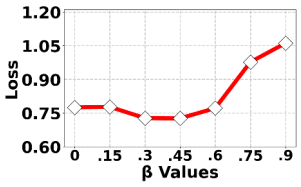


(b) Loss with different percentages of bad clients.

Fig. 7: FedSRC with different percentages of bad clients.



(a) Accuracy



(b) Loss

Fig. 8: Effect of β on the performance of FedSRC for the FEMNIST dataset.

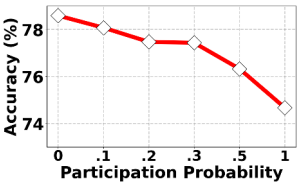
Mean, Krum, and Median on the server side.

As shown in Fig. 6, our integration approach enhances the performance of these pre-existing algorithms (about 6% increase in accuracy) and reduces the error loss (about 33% decrease in loss) in the presence of unreliable clients, all the while also reducing computation and communication costs. These observations support our assertion that FedSRC seamlessly complements and improves the performance of benchmark algorithms controlled from the server side.

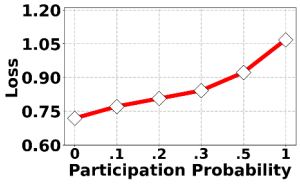
Impact of different percentages of bad clients. To assess our algorithm against varying levels of corrupted data, we use the FEMNIST dataset with different percentages of bad clients. Fig. 7 shows the effect of increasing the percentage of bad clients on global model performance. As the proportion of bad clients rises, conventional algorithms such as FedAVG, Trimmed Mean, and Krum experience significant declines in accuracy due to the adverse impact of corrupted data. In contrast, FedSRC demonstrates robust performance, maintaining higher accuracy levels compared to the benchmarks. This robustness can be attributed to FedSRC's adaptive threshold mechanism, which effectively identifies and excludes the influence of bad clients without compromising the contributions of good clients.

C. Ablation Study

Impact of the value of β. To evaluate β's impact, we experiment on the FEMNIST dataset with 30% bad clients (e.g., with corrupted labels), testing β values from 0 to 0.9. Figure 8 shows accuracy, and test loss. At $\beta = 0$, RHI is ignored, overly restricting participation and excluding good but heterogeneous clients, leading to lower accuracy (76%) and higher loss (0.75). Conversely, at $\beta = 0.9$, excessive weight on RHI reduces $\Phi_k^{(t)}$ too much, allowing more bad clients to participate, which degrades accuracy to 74% and increases loss

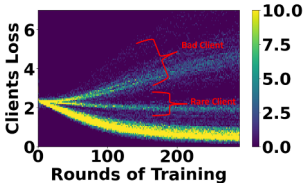


(a) Client accuracy vs probability of inclusion

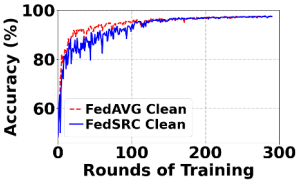


(b) Client loss vs probability of inclusion

Fig. 9: Effect of client re-inclusion on the performance of FedSRC for the FEMNIST dataset.



(a) Evolution of training loss during training for clients with bad and very rare data for the FEMNIST dataset.



(b) Performance of FedSRC compared to FedAVG in the presence of only extreme non-IID data for the MNIST dataset.

Fig. 10: Training loss and performance under extreme data conditions.

to 1.05. The optimal balance occurs at $\beta = 0.45$, achieving the highest accuracy (78%) and lowest loss (0.65), excluding approximately 37% of clients—close to the actual 30% bad clients. This highlights the need for careful β tuning to balance heterogeneity and quality.

Impact of client re-inclusion. To see how client re-inclusion can affect the performance of FedSRC, we conduct our experiment with FEMNIST dataset with varying probability of not following threshold protocol as in Fig. 9. The definition of re-inclusion here is not profiling and includes a client, it means sometimes even if the client is outside the threshold value, it will still participate with a certain probability. As we can see here, with added probability more and more clients start to participate, so if we can choose a good β, and if we choose more clients, effectively it starts to include bad clients, and performance becomes bad. However, because of our policy if an extreme non-IID client is discarded, because of this re-inclusion policy, they still can participate and make the model more robust. So choosing a reasonable probability can make our design more generalized and robust.

Evolution of client loss during training in the presence of rare and bad clients. To evaluate FedSRC's robustness under extreme non-IID conditions, we analyzed how clients' training losses evolve during training. Fig. 10 illustrates the FEMNIST dataset's training loss distribution in the presence of non-IID, extreme non-IID, and bad clients. Initially, clients with bad and extreme non-IID data exhibit similar loss patterns. However, as training progresses, the losses of bad clients diverge significantly, creating a clear distinction from those with extreme non-IID data. This distinction supports the use of a reasonable threshold in FedSRC to maintain good performance.

Additionally, Fig. 10 shows FedSRC's performance on the MNIST dataset with extreme non-IID clients, where each client has data from only two classes. Even under such conditions, FedSRC achieves results comparable to FedAVG, demonstrating its stability and effectiveness without degrading overall performance. These findings confirm FedSRC's ability to handle challenging non-IID scenarios while ensuring system stability.

V. Related work

The performance of FL degrades in the presence of corrupted clients [10]. To defend against corrupted clients, various algorithms have been proposed in the literature for attaining robust federated learning [32]–[34].

Statistics-based solutions. Among these techniques, notable statistics-based approaches include Krum [31], which selects a local model as the global model based on its similarity to others. Bulyan [35] addresses its limitations by combining Krum with a Trimmed Mean [14] variant. However, Bulyan's scalability and computation overhead are issues due to the need to compute both Krum and Trimmed Mean in each training round

Byzantine robust algorithms. Other Byzantine-robust algorithms include Trimmed Mean [14], which individually sorts and trims outliers from each model before averaging the remaining parameters for the global model. Median [14] adopts a similar approach, using the median of independent model parameters as the global model. Another technique employs Geometric Median (GM) [32], [36] to determine the federated learning model parameter. However, these methods suffer from computational intensity and lack suitability for edge-based federated learning due to resource constraints.

Client selection. Client selection techniques utilize loss function evaluations to address non-IID (Independent and Identically Distributed) challenges. The Active Federated Learning (AFL) algorithm [17] employs a value function assessed at each client, converting these valuations into selection probabilities. This approach favors clients with higher loss values, simulating a greater representation of minority data points. Similarly, Power-of-Choice Selection Strategies [16] build on this idea, prioritizing clients with higher losses for subsequent training rounds. However, both methods require tagging client updates with identifiers, which compromises client anonymity and threatens the privacy principles inherent in federated learning. Moreover, in scenarios where bad clients exhibit higher losses, these algorithms can completely fail, leading to divergence. Their design primarily targets non-IID issues but does not adequately address data quality concerns.

Re-weighting. In [37], poisoned updates in collaborative learning are detected using client-side cross-validation results to adjust update weights during aggregation. [38] addresses unreliable clients in Federated Learning by computing a utility score using auxiliary validation data, reducing their negative impact. In [15], reweighting models during aggregation based on the distance of training loss from the median of all client losses mitigates the impact of unreliable clients. These defenses require online detection, potentially compromising privacy due to access to auxiliary datasets.

Other data poisoning. In works such as [34], [39]–[42], trusted client subsets are leveraged to counteract the impact of malicious clients. In clustered Federated Learning, [34] proposes dividing clients into benign and corrupted groups based on cosine similarity between model parameters. Li et al. [41] employs an encoder-decoder approach to identify malicious updates. FLTrust [39] introduces maintaining a root dataset and server model to collect a clean small training dataset from clients. While these methods aim to enhance federated learning's robustness, the trustworthiness of trusted clients and validation datasets is uncertain in the federated setup. Additionally, communication and privacy constraints challenge compliance with federated learning protocols.

In contrast, FedSRC can manage client-side data corruption without reliance on validation datasets or identity disclosure. To the best of our knowledge, FedSRC is the pioneering algorithm to address data quality issues directly from the client side to save local computation and communication costs.

VI. Concluding Remarks

This paper introduces a pioneering approach to address data quality challenges in federated learning (FL) by empowering clients to self-regulate their participation. Our method significantly reduces communication and computation costs, enhances the accuracy of the global model, and preserves client anonymity—key advantages in resource-constrained, privacy-sensitive environments. To the best of our knowledge, FedSRC represents the first effort to implement client-side participation control, shifting the burden of quality management from the server to the clients. Moreover, we demonstrate that FedSRC imposes negligible additional computational overhead on the server, ensuring scalability without compromising efficiency.

Limitations. While FedSRC is effective under normal conditions, it relies on client-side statistics to evaluate local utility and thus may become ineffective if a large fraction of clients (e.g., over 60%) are corrupted or malicious. Additionally, although FedSRC reduces the communication overhead by avoiding unnecessary model uploads, it does not eliminate the cost of downloading the global model—clients still need to receive the model to compute their local test loss. In contrast, active client selection approaches can save both download and upload costs, but often at the expense of client anonymity. Furthermore, since FedSRC operates entirely at the client side, it does not incorporate server-level mechanisms to detect or mitigate model poisoning attacks from malicious clients. However, it is worth noting that FedSRC does not introduce any new vulnerabilities or attack surfaces beyond those already present in standard FL systems.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under grant numbers ECCS-2152357 and CCF-2324915, and by the Army Research Office under grant number W911NF-23-1-0213.

REFERENCES

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

[2] A. Hard, C. M. Kiddon, D. R. Ramage, F. S. Beaufays, H. Eichner, K. Rao, R. Mathews, S. Augenstein, and S. Ramaswamy, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[3] F. Granqvist, M. Seigel, R. van Dalen, A. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *CoRR*, 2020.

[4] S. Cho, I. Ensari, C. Weng, M. G. Kahn, and K. Natarajan, "Factors affecting the quality of person-generated wearable device data and associated challenges: Rapid systematic review," *JMIR mHealth and uHealth*, vol. 9, no. 3, p. e20738, 2021.

[5] S. Cho, C. Weng, M. G. Kahn, K. Natarajan, *et al.*, "Identifying data quality dimensions for person-generated wearable device data: Multi-method study," *JMIR mHealth and uHealth*, vol. 9, no. 12, p. e31618, 2021.

[6] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia, T. Leyvand, H. Lu, Y. Lu, L. Qiao, B. Reagen, J. Spisak, F. Sun, A. Tulloch, P. Vajda, X. Wang, Y. Wang, B. Wasti, Y. Wu, R. Xian, S. Yoo, and P. Zhang, "Machine learning at facebook: Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 331–344, 2019.

[7] M. Haghi, K. Thurow, and R. Stoll, "Wearable devices in medical internet of things: scientific research and commercially available devices," *Healthcare informatics research*, vol. 23, no. 1, pp. 4–15, 2017.

[8] Y. Cheng, K. Wang, H. Xu, T. Li, Q. Jin, and D. Cui, "Recent developments in sensors for wearable device applications," *Analytical and bioanalytical chemistry*, vol. 413, no. 24, pp. 6037–6057, 2021.

[9] C. Liu, P. Nitschke, S. P. Williams, and D. Zowghi, "Data quality and the internet of things," *Computing*, vol. 102, no. 2, pp. 573–599, 2020.

[10] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pp. 480–501, Springer, 2020.

[11] C. Yang, Q. Wang, M. Xu, Z. Chen, K. Bian, Y. Liu, and X. Liu, "Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data," in *Proceedings of the Web Conference 2021*, pp. 935–946, 2021.

[12] S. Li, E. Ngai, F. Ye, and T. Voigt, "Auto-weighted robust federated learning with corrupted data sources," *ACM Trans. Intell. Syst. Technol.*, feb 2022.

[13] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Thirty-Seventh International Conference on Machine Learning (ICML)*, 2020.

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

[15] Z. Talukder and M. A. Islam, "Computationally efficient auto-weighted aggregation for heterogeneous federated learning," in *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*, pp. 12–22, IEEE, 2022.

[16] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *25th Artificial Intelligence and Statistics (AISTATS)*, 2022.

[17] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu, and A. Kumar, "Active federated learning," *arXiv preprint arXiv:1909.12641*, 2019.

[18] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16937–16947, 2020.

[19] Z. R. Talukder, "Empowering clients: Self-adaptive federated learning for data quality challenges." https://github.com/zahidurtalukder/FedSRC, 2025. Accessed: 2025-06-05.

[20] X. Cao, J. Jia, and N. Z. Gong, "Provably secure federated learning against malicious clients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 6885–6893, 2021.

[21] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2545–2555, 2022.

[22] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, and Z. Shen, "Dealing with label quality disparity in federated learning," *Federated Learning: Privacy and Incentive*, pp. 108–121, 2020.

[23] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *15th USENIX Symposium on Operating Systems Design and Implementation OSDI 21)*, pp. 19–35, 2021.

[24] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

[25] S. Zawad, A. Ali, P.-Y. Chen, A. Anwar, Y. Zhou, N. Baracaldo, Y. Tian, and F. Yan, "Curse or redemption? how data heterogeneity affects the robustness of federated learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 10807–10814, 2021.

[26] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[27] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.

[28] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[29] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020.

[30] Z. Talukder, B. Lu, S. Ren, and M. A. Islam, "Hardware-sensitive fairness in heterogeneous federated learning," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 10, no. 1, pp. 1–31, 2025.

[31] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[32] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, 2022.

[33] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-resilient secure federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2168–2181, 2020.

[34] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8861–8865, IEEE, 2020.

[35] R. Guerraoui, S. Rouault, *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*, pp. 3521–3530, PMLR, 2018.

[36] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[37] L. Zhao, S. Hu, Q. Wang, J. Jiang, C. Shen, X. Luo, and P. Hu, "Shielding collaborative learning: Mitigating poisoning attacks through client-side detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2029–2041, 2020.

[38] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1486–1500, 2019.

[39] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," in *ISOC Network and Distributed System Security Symposium (NDSS)*, 2021.

[40] Y. Han and X. Zhang, "Robust federated learning via collaborative machine teaching," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 4075–4082, 2020.

[41] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[42] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19586–19597, 2020.