FairRFL: Fair and Robust Federated Learning in the Presence of Selfish Clients

Andrea Augello, Ashish Gupta, Giuseppe Lo Re, *Senior Member, IEEE*, and Sajal K. Das, *Fellow, IEEE*

***Abstract*—Federated Learning (FL) is a paradigm that enables collaborative machine learning without disclosing the local data of the participants. However, in real-world FL deployment scenarios, some unscrupolous clients may alter the training process to skew the global model towards their local optimum, unfairly prioritizing their data distribution. Their influence can degrade overall model performance for normal clients and reduce fairness in the system. We call this novel category of misbehaving clients "selfish". This work proposes a Fair and Robust strategy for aggregation in the Federated Learning (FL) server to mitigate the effect of Selfish clients (FairRFL). FairRFL incorporates a novel technique to recover (or estimate) the true updates from selfish clients by using robust statistics, specifically the median of norms. The presented strategy, through the inclusion of the recovered updates in the aggregation process, is robust against selfish behavior. Through extensive empirical evaluations with WISDM-W and CIFAR-10 datasets, we observe that a selfish client can increase the model accuracy on its data by up to 39% and more than quadruple the accuracy variance among clients, which FairRFL can address perfectly and recover performance fairness across normal clients.**

***Index Terms*—Distributed systems, federated learning (FL), reliability and robustness, fairness.**

Received 26 May 2025; revised 26 December 2025; accepted 29 January 2026. Date of publication 11 February 2026; date of current version 11 March 2026. The work of Sajal K. Das was supported by U.S. National Science Foundation (NSF) under Grant CNS-2008878 (FLINT: Robust Federated Learning for Internet of Things), Grant AI-ENGAGE-2520346 (HARVEST: Holistic AI-powered Agricultural Response Validation and Early Prediction System across Territories), and Grant PFI-2431990 (Smart Connected Farms: Pest Management in Agriculture through AI and IoT). The work of Ashish Gupta supported by BITS Pilani Dubai Campus (BPDC) for the project FaR-FedIoT: Fair and Resource- adaptive Federated System for IoT under NFSG scheme. This work was supported by the European Union Next-Generation EU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3) Progetto "Future Artificial Intelligence - FAIR" under Grant PE00000013 and Grant CUP J73C24000060007. *(Corresponding author: Andrea Augello.)*

Andrea Augello and Giuseppe Lo Re are with the Department of Engineering, University of Palermo, 90133 Palermo, Italy (e-mail: andrea.augello01@unipa.it; giuseppe.lore@unipa.it).

Ashish Gupta is with the Department of Computer Science, Birla Institute of Technology and Science Pilani, Dubai, UAE (e-mail: ashish@dubai.bits-pilani.ac.in).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science, Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

Digital Object Identifier 10.1109/TETC.2026.3661199

I. Introduction

FEDERATED Learning (FL) has been proposed [1] as a framework that enables collaboration among multiple clients to train a model without sharing sensitive data. In the FL paradigm, each participating client locally trains the model and transmits only the updated weights to a central server tasked with aggregation. Since its introduction, FL has been effectively applied across various domains, demonstrating its versatility and potential. While offering numerous benefits, the absence of supervision during the training process can lead to unforeseen shortcomings. Clients could deviate from the expected training behavior [2], adversely affecting the model performances. Such deviations could be malicious, with a client intentionally disrupting the training [3], or unintentional, due to resource limitations [4]. A malicious client might introduce noise to prevent model convergence, as seen in Byzantine attacks [3], or pursue secondary objectives, such as embedding a backdoor [5]. Given that the server does not directly control the clients, mitigating these harmful actions is challenging.Literature on FL indicates a plethora of works addressing the issue of adversarial or malicious clients. Two broad categories of existing approaches exist: either detecting and removing clients that deviate from the correct behavior [6], or mitigating the impact of misbehaving clients on the global model through robust aggregation techniques [7]. For instance, the work in [8] excludes the contributions of colluding malicious clients that submit similar updates. Statistical methods such as median and trimmed estimators have proven effective for robust aggregation in the presence of Byzantine clients, without requiring explicit identification [9]. Additionally, the approach in [10] guarantees satisfactory performance as long as the proportion of malicious clients is not preponderant.

This work mainly focuses on a novel type of misbehaving client, referred to as a *selfish client*, who aims to obtain a global model that prioritizes its local data distribution. When the participants have non Independent and Identically Distributed (IID) data distributions, the local optima of the clients can significantly differ from the global optimum, and even low-level feature spaces might be different [11], [12]. Thus, in a competitive environment, acting selfishly to obtain a better-performing model can be a tempting strategy. For example, in a federated healthcare system, providers may seek performance benefits if they possess rare or high-quality data. As illustrated in part ② of Figure 1, even when the selfish client does not intend to poison the model, the resulting model may perform worse for other

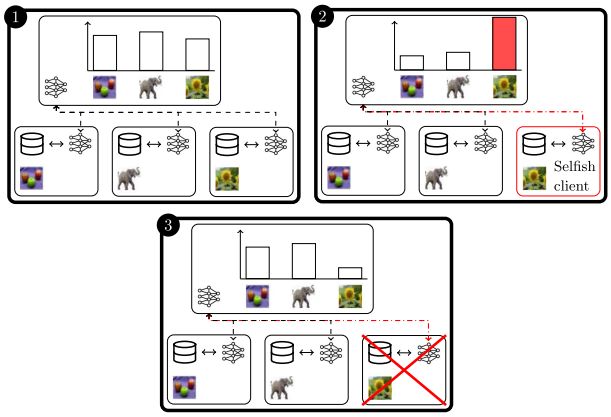

Figure 1. Part ① depicts a standard FL scenario with heterogeneous data distributions for training a global model. In part ②, a selfish client crafts updates to improve the performance on its data distribution, worsening generalization. And part ③ shows that excluding the selfish client would yield poor performance on data similar to the selfish client's.

clients, especially in non-IID settings. Note that selfish clients do not have malicious aims, suggesting that the selfish model updates cannot be disregarded entirely from the aggregation, as doing so would cause the loss of their valuable contributions and, in turn, poor performance on the respective task as depicted in part ③ of Figure 1. On the other hand, those updates cannot be included as-is because that would result in an altered global model with worse generalization capabilities. Hence, existing defense methods designed for malicious clients [4], [5] can NOT effectively tackle the selfishness issue.

A. Major Contributions

To the best of our knowledge, this work is the first to address the selfishness issues in FL. Our major contributions include:

- We present a new issue of selfishness in FL, driven by selfish clients. In non-IID settings, their presence is inevitable, since certain clients can readily discern the disparity between the global optimum and their local one and may attempt to leverage this for their benefit. The selfish client can adjust the level of selfishness through an upscaling parameter.
- We propose FairRLF: a **Fair** and **R**obust aggregation strategy for **FL** to limit Selfish clients' effect on the aggregation process without affecting the global model by intertwining two novel strategies: (i) RFL-Self for detecting and mitigating the impact of selfish clients, and (ii) Dq-FFL for dynamically adjusting the fairness parameter on the client side to implicitly account for the presence of selfish clients, thereby improving performance fairness across all participants.
- Through our experiments on two benchmark image classification datasets, WISDM-W and CIFAR-10, we analyze the influence of selfish clients on the global model and evaluate the performance of FairRFL for varying levels of selfishness to observe closely fairness improvement across normal clients.

This research is an extension of our conference paper [13] with the following additional significant contributions.

1) On the Selfishness Side:

- We expand the concept of selfishness with more sophisticated aggregation strategies beyond FedAvg.
- We introduce an additional new strategy for selfish clients to automatically tune an upscaling parameter to achieve the desired level of selfishness.

2) On the Robustness Side:

- We estimate the impact of selfish clients on the fairness aspect of FL, showing that even when they do not affect the global model, they can still significantly harm fairness.
- We expand the theoretical analysis of the FairRFL strategy and derive a convergence bound for our server-side mitigation strategy.
- We introduce Dq-FFL, a variant of q-FFL [14], to address the performance fairness across participants.
- We also consider non-persistent selfish clients, who behave selfishly only for a limited number of rounds, to rigorously analyze the selfishness aspect.

The remainder of this paper is organized as follows. Section II discusses the related works. Section III describes in detail the strategy employed by selfish clients, while Section IV presents the proposed mitigation strategies. Section V provides a theoretical analysis of the proposed framework. In Section VI, we analyze the experimental results. Finally, Section VII concludes the paper.

II. Preliminaries and Related Works

In FL, the server aims to find an optimal model $\mathbf{w}^*$ which solves the following optimization problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\{ F(\mathbf{w}) \triangleq \frac{1}{k} \sum_{i \in [k]} F_i(\mathbf{w}) \right\}, \tag{1}$$

This optimization problem is solved iteratively through multiple communication rounds. At a given round t, each client i computes an update δ_i^t using the Stochastic Gradient Descent (SGD) algorithm. The server combines these updates to compute the updated global model for the next round $t+1$ as $\mathbf{w}^{t+1} = \mathbf{w}^t + \delta_{[k]}^t$, where $\delta_{[k]}^t = \frac{1}{\Gamma} \sum_{i \in [k]} \omega_i \delta_i^t$, ω_i is the aggregation weight associated the i-th client, and Γ is a normalization factor which might differ in each round. For instance, in FedAvg [1], Γ is the overall number of training samples, and the aggregation weight ω_i is the number of training samples of the i-th client. When all the clients have datasets with the same size, Γ is the number of clients that participate in the training process k, and all the aggregation weights ω are set to 1. In an existing fairness algorithm, q-FFL [14], Γ is computed by the server at each round based on the loss of each client, and each client has a different (known to them) aggregation weight ω.

To simplify the notation, we omit the round superscript when it is clear from the context. Also, with some abuse of notation, we consider all updates to have already been scaled by the aggregation weight, i.e., $\delta_i = \omega_i \delta_i$. The main notation used in this paper is summarized in Table I.

TABLE I
USED NOTATIONS

Notation	Description
$\mathbf{w}^t$	Global model at round t
$\mathbf{w}^*$	Optimal global model
$F(\cdot)$	Global loss function
$F_i(\cdot)$	Local loss function of the i-th client
Γ	Normalization factor for aggregation
ω_i	Aggregation weight of the i-th client
δ_i	Local update of the i-th client
$\delta^t_{[k]}$	Aggregated local updates at round t
$\delta_{[k]\setminus\{s\}}$	Aggregated local updates of all clients except s
$\bar{\delta}_{[k]\setminus\{s\}}$	Estimate of the average local update of all clients except s
$\tilde{\delta}_s$	Crafted update of selfish client s
ϕ	Selfishness parameter
$\mathcal{N}_{med}$	Medan norm of the local updates
$\mathcal{S}$	Set of suspected selfish clients
δ_{med}	Median local update
δ'_i	Recovered update of the i-th client

A. Robust Federated Learning

Clients intentionally misbehaving are known to be a problem for FL. Indeed, several works focus on clients with malicious intent who aim to disrupt the training process. The kind of attacks that can be performed by malicious clients can be split into two main classes [15]: targeted attacks and untargeted attacks. Untargeted attacks aim to disrupt the training process by sending random or crafted updates to the server, trying to reduce the accuracy of the global model on all inputs [16]. By contrast, targeted attacks aim to disrupt the training process by sending updates that decrease the performance of the global model on a specific subset of inputs. Backdoor attacks [17] are a subset of targeted attacks where the targeted subset is characterized by a backdoor trigger. To reduce the influence of malicious clients on the global model, robust aggregation techniques have been developed. A popular approach is to use a robust aggregation method, such as the Median [18], or the geometric median [19]. Other approaches are also possible, such as removing malicious clients from the training process [20]. For instance, the updates from clients suspected of being malicious can be weighted less in the aggregation process [8].

B. Fair Federated Learning

Selfish clients can also be seen as a fairness issue, as they wish to make the global model more accurate on their data at the expense of other clients, actively reducing the fairness of the global model. Fairness is a critical issue in FL [21]: the global model can be biased towards some clients, leading to poor performance for other clients, even when selfish clients are not involved. This is especially true in non-IID settings, where the data distribution of the clients can be very different [12].

Multiple definitions of fairness have been explored in the context of FL, such as accuracy parity, which strives to achieve similar accuracy for all clients [22]; group fairness, that strives to achieve similar accuracy for different groups of clients, e.g., demographic groups [23]; and contribution fairness, where each client's payoff is proportional to their contributions [24].

We focus on accuracy parity as defined in [20], i.e., the variance of the testing performance across clients. It is the fairness definition against which the impact of selfish clients is the most visible.

Definition 1: A model $\mathbf{w}$ is fairer than another model $\mathbf{w}'$ if the variance of its performance across all the clients is lower.

$$fair(\mathbf{w}) > fair(\mathbf{w}') \Leftrightarrow \text{var}\left(\{F_i(\mathbf{w})\}_{i\in[k]}\right) < \text{var}\left(\{F_i(\mathbf{w}')\}_{i\in[k]}\right),$$

where $[k]$ is the set of all clients.

To achieve fairness, multiple approaches have been proposed. One such approach is that of Agnostic FL [25], where the goal is to optimize the global model for the worst-performing client through a minmax optimization scheme. The minmax formulation can also be used to achieve other fairness definitions, such as proportional fairness [26], where the goal is to determine the Nash bargaining solution that tries to maximize the total utility and the worst-case utility of the clients.

A more flexible trade-off between fairness and accuracy than that of Agnostic FL can be achieved with q-Fair FL (q-FFL) [14], where a q parameter controls by how much each client's loss influences its aggregation weight. If $q = 0$, the aggregation weight is independent of the client's loss, and the training process is equivalent to standard FL. With $q \to \infty$, the training process is equivalent to Agnostic FL.

It is worth noting that an effective way to ensure that each client can achieve the best possible accuracy is to use a personalized model for each client [27]. Our setting, however, assumes the FL server aims to obtain a single unified global model, intended to be deployed across the entire domain and serve all clients equally, which is the most common FL scenario. In such a scenario, personalization techniques such as training separate models for individual clients or groups [28], [29] are not applicable. If selfish clients could receive personalized models tailored to their own data distributions, their incentive to interfere with the global training process would be greatly diminished. Our focus, therefore, is on the standard FL configuration where a single global model is collaboratively trained and deployed for all participants, making the selfishness problem both relevant and challenging.

Recent works have increasingly focused on the performance of FL in the face of non-IID and heterogeneous data. For instance, cross-silo prototypical calibration [11], [29] reduces representation gaps between clients, ensuring that models do not disproportionately favor certain data distributions. Similarly, the data-free knowledge distillation approach [12] allows models to align by reducing the difference in local distributions without requiring direct data sharing. While laudable and noteworthy, these approaches primarily address issues stemming from data heterogeneity, making for fairer models across clients. However, a selfish client does not care about model fairness and instead aims for the best possible performance on its data. Thus, mitigating the non-IID data distribution issue does not solve the selfishness problem, which requires dedicated solutions.

Complementing these strategies, FedCDA [30] proposes divergence-aware aggregation to explicitly account for cross-round variability, mitigating the risk of skewed updates by not including updates that greatly diverge from other updates through the use of historical data. Along a different line, FedMut [31] introduces stochastic mutation to enhance generalization, which helps prevent bias toward dominant data patterns.

III. The Selfishness Problem

The FL server strives to achieve a globally optimal model. Such a model may not be optimal for each client, particularly under non-IID scenarios. This disparity might lead some clients to strive for a global model that is closer to their local models. Consequently, the global model stops being optimal for the entire system, and is expected to generalize worse [29]. We refer to these clients as *selfish* due to their self-centered nature.

Definition 2: A selfish client $s \in [k]$ is a participant in FL who, instead of *locally* minimizing $F_s(\mathbf{w})$ so that the aggregated global model will minimize the average loss, tries to make the *aggregated global model* minimize $F_s(\mathbf{w})$ as in (2):

$$\mathbf{w}_s^* = \arg\min_{\mathbf{w}} \left\{ F_s \left(\frac{\mathbf{w} + \sum_{i \in [k] \setminus s} \mathbf{w}_i}{\Gamma} \right) \right\}, \tag{2}$$

• *Distinguishing selfish clients from malicious:* Despite some authors use the term "selfish" to denote common malicious clients [32], the notion of selfish clients in our work **completely differs from the malicious ones**. The selfish clients do not harbor any malicious intent, such as orchestrating model poisoning attacks [33] or impeding model convergence [3]. A selfish client aims to improve the global model's performance on its own local task, which is not explicitly contradictory to the global objective. As a side effect, the global model's performance on other tasks might be worsened, but this is not its primary goal. Further, the selfish clients do not try to optimize the model for an additional objective different from the main task, unlike backdoor attackers [5]. For instance, in an FL application for activity recognition, a selfish client might manipulate its update to improve the model's performance on recognizing activities that are more relevant to them (e.g., walking and sitting), even if this comes at the cost of reduced performance on other activities (e.g., running and climbing). A malicious client, on the other hand, might try to degrade the model's overall performance or insert a backdoor that causes the model to misclassify certain activities (e.g., classifying eating as running if the user is eating with their left hand).

In communication networks, "selfish clients" often refers to free-riders—clients who benefit from the system while contributing little or nothing to training [34], [35]. Some researchers have utilized the term "selfish" to denote free-riders in their work [36], [37]. Nonetheless, the challenges explored in these works differ from the focus of this paper. To the best of our knowledge, this is the first study to address this specific problem. A complementary use of the term "selfish" in the context of FL is found in [38], where it refers to a server that favors a subset of clients, which is the opposite problem to the one we address. The differences between the various types of clients are summarized in Table II.

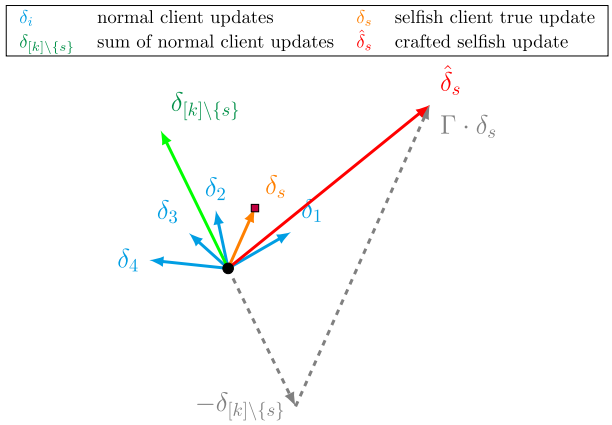

Figure 2. An intuition for how a selfish client can estimate the updates $\hat{\delta}_s$.

• *Capabilities:* Selfish clients are normal participants with selfish intent; thus, they can access and modify the local data and local model, but cannot access other clients' data and model. As the clients communicate with the server securely, interception of the communication channel is not possible.

Remark 1: Selfish clients are not malicious; their aim is not to degrade the global model on some/all the inputs, but rather to improve the global model on a specific data distribution. The worsened generalization is an unintended side-effect.

A. Selfish Behavior

To introduce selfishness into the local model, a selfish client $s \in [k]$ needs to estimate local updates that will lower the loss of the global model on its local data. As a proxy for the objective in (2), drawing inspiration from model replacement attacks [39], a selfish client strives to craft an update $\hat{\delta}_s$ that minimizes the distance of the global model from its locally trained model $\|\mathbf{w}^{t+1} - \mathbf{w}_s^t\|^2$ by satisfying (3):

$$\hat{\delta}_s = \arg\min_{\hat{\delta}_s} \left\| \mathbf{w}^t + \frac{\hat{\delta}_s + \delta_{[k]\setminus\{s\}}}{\Gamma} - (\mathbf{w}^t + \delta_s) \right\|^2. \tag{3}$$

By uploading $\hat{\delta}_s$ instead of its true updates, these clients effectively steer the training process towards a global model that aligns more closely with their local optimum. Figure 2 visually illustrates the potential deviation of the selfish update $\hat{\delta}_s$ from the updates sent by normal clients.

Challenge: The term $\mathbf{w}^t$ is known to all the clients as it is the previously received global model, and can be easily simplified. However, the additive term $\frac{\delta + \delta_{[k]\setminus\{s\}}}{\Gamma}$ requires knowledge of Γ, which is generally unknown to the clients, and the sum of all the updates from the non-selfish clients in the round $t+1$. However, obtaining such information is generally unfeasible in FL due to privacy concerns. Since the server is assumed to be trustworthy, there are no viable means to acquire $\delta_{[k]\setminus\{s\}}$ and Γ, thereby posing a significant challenge in solving (3).

TABLE II
COMPARISON OF MISBEHAVING AND FAULTY CLIENT TYPES IN FEDERATED LEARNING

Client Type	Intent	Behavior	Impact	Mitigation
Selfish	Opportunistic: maximize local performance	Biased updates	Harms generalization; reduces performance fairness	True update recovery
Free-Rider	Opportunistic: benefit without contributing	Minimal/no updates	Minimal impact on model quality; primarily a contribution fairness issue.	Contribution incentives
Byzantine	Malicious: disrupt training	Adversarial updates	Prevents convergence, degrades accuracy	Robust aggregation
Backdoor Attacker	Malicious: insert hidden behavior	Trigger-targeted updates	Compromises model on specific inputs; general performance may be maintained.	Anomaly detection
Faulty (Hardware)	Benign: unintentional disruption	Noisy/corrupted updates	Random accuracy degradation	Error correction, update filtering

Considering the aforementioned challenge, the selfish client is constrained to obtain an approximation of the mean update from the other clients, denoted as $\bar{\delta}_{[k]\setminus\{s\}}$, by assuming minimal changes in the average updates between two consecutive communication rounds. This assumption has been validated experimentally in Section VI, showing that the estimated update closely tracks the true one. However, it may not hold in some extreme scenarios, e.g., when the data distribution of the clients changes significantly between two rounds; consequently, the effectiveness of the proposed strategy might be reduced. At any round t, the selfish client can leverage knowledge of past updates to calculate $\bar{\delta}_{[k]\setminus\{s\}}$ using the following expression:

$$\arg\min_{\Gamma,\bar{\delta}_{[k]\setminus\{s\}}} \|\Gamma\cdot(\delta_{[k]}^{t-2}-\bar{\delta}_{[k]\setminus\{s\}})-\delta_s^{t-2}\| + \left\|\Gamma\cdot(\delta_{[k]}^{t-1}-\bar{\delta}_{[k]\setminus\{s\}})-\delta_s^{t-1}\right\|,$$

where for $t=2$, $\hat{\delta}_s^0=\delta_s^0$ and $\hat{\delta}_s^1=\delta_s^1$. Once this estimate is determined, the selfish client attempts to align the next global update more closely with its local model. To accomplish this, the client could transmit the update vector derived from (4) to the server. This action would entirely substitute the global model, inadvertently leading to behavior resembling that of a malicious client.

$$\hat{\delta}_s=\frac{\Gamma}{\omega_s}\delta_s-(\Gamma-\omega_s)\bar{\delta}_{[k]\setminus\{s\}}. \tag{4}$$

To illustrate the actions of the selfish client we make use of a toy 2D numerical example:

Example 1: (Single selfish client) Let $\delta_1,\ldots,\delta_4$, and δ_s be the the update vectors in Figure 2, with the following values:

- $\delta_1=[0.95,0.55]$
- $\delta_2=[-0.20,0.90]$
- $\delta_3=[-0.60,0.55]$
- $\delta_4=[-1.20,0.10]$
- $\delta_s=[0.40,0.90]$

Considering an unweighted FedAvg for aggregation, if all clients behave honestly, the global update is $\delta_{[k]}^0=\delta_{[k]}^1=[-0.13,0.60]$. With $\Gamma=5$ and $\omega=1$, and the selfish client s computes $\bar{\delta}_{[k]\setminus\{s\}}^2=[-0.26,0.52]$. With these parameters, the crafted model update $\hat{\delta}_s$ is computed through (4) as $\hat{\delta}_s=5\delta_s-4\bar{\delta}_{[k]\setminus\{s\}}=5\cdot[0.40,0.90]-4\cdot[-0.26,0.52]=[2.0,4.5]-[-1.04,2.08]=[3.04,2.42]$. It is easily verifiable that the aggregate update $\hat{\delta}_{[k]}$ is equal to δ_s. ◊

If a selfish client adopts this approach, it would gain no advantage from participating in the FL system, as the global model would match its local model. To address this, we define a selfishness parameter $\phi\in[0,1]$, which determines the proportion of the global update to be substituted with $\hat{\delta}_s$ during each round. By maintaining a fixed ϕ throughout the rounds, the selfish client can compute its updates using (5):

$$\begin{aligned}\hat{\delta}_s&=\phi\left(\frac{\Gamma}{\omega_s}\delta_s-(\Gamma-\omega_s)\bar{\delta}_{[k]\setminus\{s\}}\right)+(1-\phi)\omega_s\bar{\delta}_{[k]\setminus\{s\}}\\&=\phi\Gamma\left(\frac{\delta_s}{\omega_s}-\bar{\delta}_{[k]\setminus\{s\}}\right)+\omega_s\bar{\delta}_{[k]\setminus\{s\}}.\end{aligned} \tag{5}$$

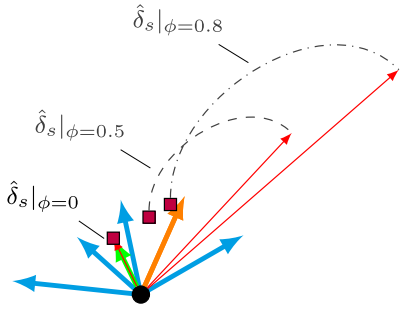

Figure 3. Demonstrating the influence of ϕ on the selfish update vector and the aggregated model when a single selfish client is present.

B. Analyzing the Parameter ϕ

When $\phi=0$, the component of the update including the selfish client's local model would be multiplied by zero, making the estimated updates equivalent to the other clients' updates. On the flip side, setting $\phi=1$ causes the selfish client to completely replace the global update with its own local update, thereby making it a model replacement attack. The effect of the parameter ϕ can be easily interpreted in the case of the FedAvg algorithm with $\Gamma=k$ and $\omega=1$. Indeed, if $\phi=\frac{1}{k}$, then the update would simplify to $\hat{\delta}_s|_{\phi=\frac{1}{k}}=\delta_s$. while the other client's update corresponds to the mean update of the normal clients. In general, each increment of ϕ by $\frac{\omega_s}{\Gamma}$ amplifies the influence of the selfish client by adding the equivalent of one normal update δ_s from the selfish client while reducing the aggregated contribution of one normal client. Figure 3 provides a visual illustration of the impact of the parameter ϕ on the selfish update vector and the resulting aggregated model $\hat{\mathbf{w}}$.

Remark 2: Selfish clients influence the model with partial (according to a tuning parameter ϕ) replacement attempts.

Example 2: Using (5), the selfish update can be computed as: $\hat{\delta}_s=5\phi[0.66,0.38]+[-0.26,0.52]=[3.30\phi-0.26,1.90\phi+$

0.52]. Accordingly, the resulting aggregate update $\hat{\delta}_{[k]}$ is $[0.66\phi - 0.26, 0.38\phi + 0.52]$.

Thus, for $\phi = 0$, the selfish update is $[-0.26, 0.52]$, which is equivalent to the average update of the normal clients, and $\phi = 1$ results in the aggregate update being [0.40,0.90], which coincides with δ_s. Intermediate values of ϕ would result in updates that are a linear combination of the selfish client's local update and the average update of the normal clients. For instance with $\phi = \frac{1}{\Gamma} = 0.2$, the selfish update is $\hat{\delta}_s = [0.40, 0.90] = \delta_s$, resulting in the aggregate update $\delta_{[k]} = [-0.13, 0.59]$. With $\phi = 0.5$, the aggregated model would be exactly the average of the selfish client's local model and the average of the normal clients' updates: $\hat{\delta}_{[k]} = [0.07, 0.71]$.

It is interesting to notice that the norm of the aggregated update $\hat{\delta}_{[k]}$ is not constant and instead has a monotone non-linear relationship with ϕ. Specifically, in this example, the norm of the global update without selfish behavior is $\|\delta_{[k]}\| = \sqrt{-0.13^3 + 0.6^2} = 0.61$, while the norm of the global update when perturbed by the selfish client is $\sqrt{0.58\phi^2 + 0.052\phi + 0.34}$, spanning from 0.58 to 0.99.

C. Multiple Selfish Clients Scenario

In scenarios involving multiple selfish clients, we consider them to act independently without any form of collaboration or collusion among themselves. Thus, we expect them to *compete* with each other to increase their effect on the global model and try to cancel each other's effect. Modeling the behavior of collaborating/colluding selfish clients on a common FL task is an interesting direction for future work, but it is beyond the scope of this paper.

Example 3. (Multiple selfish clients): Let us assume, for this example only, that client 3 is also behaving selfishly. The aggregated global update would be $\frac{\delta_1+\delta_2+\delta_4+\hat{\delta}_s+\hat{\delta}_3}{5}$

The original updates of clients 3 and s have a respective distance of 0.47 and 0.61, from the global update without selfish behavior. We use this distance as a reference to evaluate how well the obtained global update reflects the goals of the two selfish clients. For ease of presentation, we assume that both clients s and 3 have independently chosen to set $\phi = 0.5$, but analogous results can be obtained for any other value of ϕ.

Client 3 computes $\bar{\delta}^2_{[k]\setminus\{3\}} = [-0.01, 0.61]$. Accordingly, $\hat{\delta}^2_3 = [-2.99\phi - 0.01, -0.45\phi + 0.64]$. In round 2, the first with selfish behavior, the aggregated update would be $\hat{\delta}^2_{[k]} = [0.31\phi - 0.14, 0.29\phi + 0.54] = [-0.11, 0.69]$, which is not much closer to the selfish clients' local models (0.46 and 0.65) than the global model without selfish behavior. From the next round onwards, the selfish clients would start competing with each other, worsening the outcome for all the participants: the selfish clients, s and 3, would update their estimates of the other clients' updates to craft their own and cancel the other client's contributions.

For the sake of manual computation, we assume Γ to be known; the updated estimates are just the average of the previous estimates and the new updates: $\bar{\delta}^3_{[k]\setminus\{s\}} = \frac{[-0.26,0.52]}{2} + \frac{\Gamma[-0.11,0.69]-[1.39,1.47]}{2(\Gamma-1)} = \frac{[-0.26,0.52]+[-0.53,0.30]}{2} = [-0.39, 0.41]$. Likewise, $\bar{\delta}^3_{[k]\setminus\{3\}} = \frac{[-0.01,0.61]}{2} + \frac{\Gamma[-0.11,0.69]-[-0.50,0.42]}{2(\Gamma-1)} = \frac{[-0.01,0.64]+[0.18,0.55]}{2} = [0.09, 0.58]$.

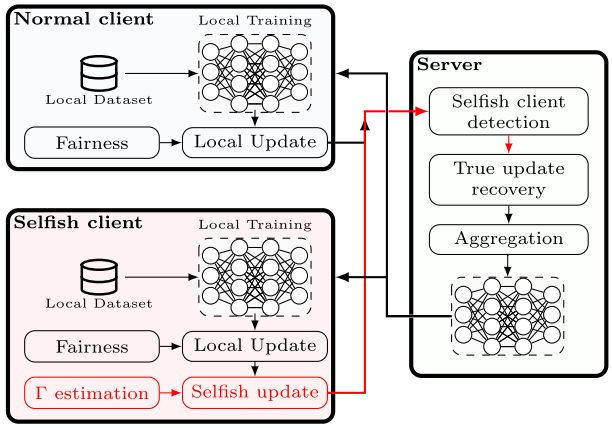

Figure 4. Overview of the solution at client and server sides.

Thus, the selfish updates for round 3 would be $\hat{\delta}^3_s = [1.59, 1.62]$ and $\hat{\delta}^3_3 = [-1.63, 0.50]$. With these selfish updates, and the normal updates unchanged, the global update would be $\hat{\delta}^3_{[k]} = [-0.16, 0.51]$, whose distance from the goals of the selfish clients is 0.45 and 0.65.

From round 8 onwards, client 3 also starts experiencing worsening outcomes. Indeed, repeating this process for a few iterations, assuming the honest updates are unchanged, the global update would converge to $[-0.25, 0.11]$, whose distance from the goals of the selfish clients is 0.57 and 1.03; and its distance from $\delta^0_{[k]}$ is 0.50. This example aligns with our empirical evidence that the presence of multiple selfish clients harms the model performance for all the participants, including the selfish clients themselves.

Remark 3: Multiple selfish clients compete to influence the model, introducing instability in the training process.

IV. Proposed Solution: FairRFL

To mitigate the impact of selfish clients on the global model, the server should be able to identify them and recover their true updates. Nevertheless, even when these clients are accurately identified, they should not be entirely removed from the training process, as this could result in a model that performs inadequately for users (not involved in training) whose data distribution is close to that of the selfish clients. Thus, the only viable solution to deal with selfish clients is to recover an estimate of their true updates and use them in aggregation (Figure 4). This section presents RLF-Self, an aggregation strategy for the FL server to robustly remove the impact of selfish clients from the model training. At each round, the RFL-Self first identifies suspected selfish clients and then attempts to recover their true update. Additionally, in Section IV-D, we propose a more advanced weighting scheme, Dq-FFL, to improve performance fairness across all clients. We denote the joint use of these two strategies as FairRFL.

A. Selfish Client Detection At the Server

After collecting updates from all clients, RFL-Self calculates the L2-norm for each update and determines the median norm, denoted as $\mathcal{N}_{med}$. Selfish clients attempt to amplify their impact on the global model by steering the training process closer to their local model, which naturally results in their updates having a greater magnitude compared to those of other clients. This intuition is confirmed by the formal result in Theorem 1. This characteristic indicates that if $\|\delta_i\|^2 > \mathcal{N}_{med}$ for a given client i, there is a possibility that the client is behaving selfishly. Consequently, the received update δ_i could be the crafted $\hat{\delta}_i$.

Theorem 1: An effective selfish update is always larger in magnitude than normal updates.

Proof: Let $\hat{\delta}_s$ be a crafted selfish update indistinguishable (in terms of norm) from the true update δ_s. Then $\hat{\delta}_s$ was obtained by tuning ϕ such that $\|\hat{\delta}_s\|^2 = lVert\delta_s\|^2$. Using this constraint and (5), we obtain the following equation:

$$\left\|\phi\frac{\Gamma}{\omega_s}\delta_s + (1-\phi\Gamma)\omega\bar{\delta}_{[k]\setminus\{s\}}\right\|^2 = \|\delta_s\|^2,$$

$$\left(\phi\frac{\Gamma}{\omega_s}\|\delta_s\|\right)^2 + 2\phi\Gamma(1-\phi\Gamma)\left\langle\delta_s, \bar{\delta}_{[k]\setminus\{s\}}\right\rangle$$

$$+(1-\phi\Gamma)^2\omega_s^2\|\bar{\delta}_{[k]\setminus\{s\}}\|^2 = \|\delta_s\|^2,$$

After solving above, we get two possible values of ϕ:

$$\phi = \frac{\|\omega_s\bar{\delta}_{[k]\setminus\{s\}}\|^2 - \|\delta_s\|^2}{\frac{\Gamma^2}{\omega_s^2}(\|\delta_s\|^2 + 2\langle\delta_s, \bar{\delta}_{[k]\setminus\{s\}}\rangle + \|\omega_s\bar{\delta}_{[k]\setminus\{s\}}\|^2)} \tag{6}$$

$$\phi = \frac{\omega_s}{\Gamma} \tag{7}$$

If $\|\delta_s/\omega_s\|^2 \simeq \|\bar{\delta}_{[k]\setminus\{s\}}\|^2$, then (6) would give $\phi \simeq 0$, which corresponds to sending $\omega_s\bar{\delta}_{[k]\setminus\{s\}}$. On the other hand, for $\phi = \frac{\omega_s}{\Gamma}$ the selfish update coincides with the true one. Thus, the only way to achieve selfish behavior is by sending updates with a larger magnitude (i.e., larger norm) compared to the true ones. □

Remark 4: All the selfish updates need to have a larger norm than the true updates. Theorem 1 only proves that this condition is necessary (not sufficient) for an update to be selfish.

Flagging all the updates with a norm larger than $\mathcal{N}_{med}$ as potentially selfish would necessarily lead to many false positives. Indeed, at each round, half of all the clients would be treated as selfish. For this reason, instead, we flag as potentially selfish all the updates that seem out of distribution by evaluating their distance from the median in terms of the median absolute deviation (MAD) [40]. The selfish detection algorithm is detailed in Algorithm 1, where the 1.4826 constant is used to ensure that the MAD is a consistent estimator of the standard deviation for normal distributions [40]. And τ is a user-specified threshold that controls the sensitivity of the detection algorithm. In this work, we set $\tau = 2.5$ after testing different values in the range [0,6]. It is worth noting that the MAD has the same breakdown point as the median, meaning that using Algorithm 1 instead of flagging all the updates with norms larger than the median offers the same robustness guarantees while reducing the false positives. This detection strategy is designed to be simple and lightweight,

Algorithm 1: Selfish Client Detection.

Input: A set of k update vectors $\{\delta_1, \ldots, \delta_k\}$, a threshold τ
Output: A set $\mathcal{S}$ of potentially selfish clients

```
Compute 𝒩 ← {||δ_i||, ∀i ∈ [k]}
Compute a median norm N_med ← median(𝒩)
MAD ← 1.4826 × median({|N_med − N_i|, ∀i ∈ [k]})
S ← {}
for each i ∈ [k] do
    if (N_i − N_med)/MAD > τ then
        S ← S ∪ {i}
return S
```

and the experimental evaluation, with the theoretical backing, confirms that it is effective in identifying selfish clients.

Algorithm 1 runs at the server after each FL round. The complexity of the algorithm is $\mathcal{O}(dk + k\log k)$, where k is the number of clients and d is the dimensionality of the model.

Example 4: Let us consider the same setup as in Example 1, the norms of the honest updates are $\mathcal{N}_1 = 1.1, \mathcal{N}_2 = 0.9, \mathcal{N}_3 = 0.8$, and $\mathcal{N}_4 = 1.2$. The selfish update δ_s, with $\phi = 0.5$, has a norm of $\mathcal{N}_s = 2.02$.

Since there is an odd number of updates, the median norm is $\mathcal{N}_{med} = 1.1$ and MAD $= 1.4826 \times$ median$(\{0.28, 0.18, 0, 0.10, 0.92\}) = 0.26$. The threshold for the detection of selfish clients is $1.1 + \tau \cdot 0.26 = 1.1 + 2.5 \cdot 0.26 = 1.75$, which is exceeded only by the selfish client's update.

B. Selfish Clients' True Updates Recovery

Differently from works on malicious clients that exclude suspected poisoned updates, in this work, the server tries to estimate the true updates δ'_s from the selfish clients' updates $\hat{\delta}_s$. The crucial difference is that selfish updates, differently from malicious ones, still contain useful information that can improve the generalization of the model.

In RFL-Self, the server computes δ'_s for each detected selfish client s by combining $\hat{\delta}_s$ and the marginal median update δ_{med} using a convex combination, expressed as

$$\delta'_s = \beta\hat{\delta}_s + (1-\beta)\delta_{med}, \tag{8}$$

where the parameter β defines the convex combination and is chosen such that the norm of δ'_s matches the median norm of all the received updates, denoted as $\mathcal{N}_{med}$. To determine β, the following equation is solved:

$$||\beta\hat{\delta}_s + (1-\beta)\delta_{med}|| = \mathcal{N}_{med}. \tag{9}$$

Through the mechanism in (8), the suspected selfish update is adjusted by reducing its magnitude and aligning it closer with the median update, effectively recovering an approximation of the true update δ_s. Later, we also theoretically analyze the soundness of the above strategy.

Equation (9) is a quadratic equation in β, with one of the two solutions being the trivial $\beta = 0$, and can be solved analytically through the cosine rule using only the magnitude of the selfish update $\hat{\delta}_s$, and its dot product with the median update δ_{med}.

The complexity of solving (9) is thus $\mathcal{O}(d)$, where d is the dimensionality of the model, for each suspected selfish client.

Example 5: Continuing with Example 4, the median update is $\delta_{med} = [-0.20, 0.55]$. By solving (9), we find that the best value for β is $\beta = 0.45$. $\beta = 0.45$ results in the estimated update $\delta'_s = [0.52, 0.96]$.

It is worth noting that the conditions for the perfect recovery, as delineated in Section V, are not met: $\mathcal{N}_{med} = 1.1$ while $||\delta'_s|| = 1.0$, additionally, the median update has a distance from $\delta_{[k]\setminus\{s\}}$ of 0.07, Nevertheless, the estimated update is close to the original update [0.40,0.90] of the selfish client.

According to the theoretical analysis in (12), the optimal value of β, assuming the server had additional knowledge of ϕ, would have been $\beta = 0.4$, within 10% of the computed value. Using the optimal β would have resulted in $\delta'_s = [0.44, 0.91]$, which is even closer to the original update.

Remark 5: Selfish updates contain useful information that, if recovered, can improve the global model's generalization. The server can use robust statistics to identify selfish clients and estimate their true updates.

C. Robust Aggregation

The server aggregates the updates from all participants by substituting the suspected selfish clients' updates with their recovered counterparts, as shown below:

$$\delta_{[k]} = \frac{1}{\Gamma}\left(\sum_{i\in[k]\setminus\mathcal{S}} \delta_i + \sum_{j\in\mathcal{S}} \delta'_j\right). \tag{10}$$

Subsequently, the global model is updated as $\mathbf{w} = \mathbf{w} + \delta_{[k]}$ for the next iteration. This aggregation approach effectively reduces the impact of selfish clients while leveraging their updates to create a more generalized and robust global model.

Example 6: As seen in Example 4, the only flagged suspected selfish client is the actual selfish client, and its recovered update is $\delta'_s = [0.52, 0.96]$. The global update is then $\delta_{[k]} = \frac{d_1+d_2+d_3+d_4+\delta'_s}{5} = [-0.11, 0.61]$, whose distance from the untampered global update $[-0.13, 0.60]$ is 0.03.

Algorithm 2 outlines the complete procedure for RFL-Self. It is worth highlighting that RFL-Self modifies only the server-side aggregation process compared to the FedAvg [1], making it compatible as a direct substitute without requiring any change on the client side.

The main computational overhead of RFL-Self is the computation of the median update δ_{med}, which is done at the server. Computing the median update has a complexity of $O(dk \log k)$, where k is the number of clients, and d is the dimension of the model. Compared to this, the complexity of Algorithm 1 is negligible. Thus, the overall complexity of RFL-Self is the same as simply using the median as a robust aggregation method, which is $O(dk \log k)$.

D. Dynamic q for Fair FL (Dq-FFL)

Our algorithm Dq-FFL is heavily inspired by the q-FFL algorithm [14], which is a variant of FedAvg with a dynamic boosting parameter q to increase fairness among the clients. The optimization goal pursued by q-FFL is not the minimization of the average loss as in (1), but the minimization of the average of $q+1$-th power of the losses:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i\in[k]} F_i^{q+1}(\mathbf{w}). \tag{11}$$

When q=0, the objective function is equivalent to that of FedAvg, while when $q \to \infty$, the objective function is equivalent to the minimization of the maximum loss among the clients (Angostic Federated Learning [25]). In general, the higher the q, the more fairness is enforced at the expense of average accuracy. To optimize for this goal, the update vector of each client is scaled by a factor proportional to the client loss raised to the power of q. Moreover, the server dynamically computes Γ using information from the clients relative to their loss and the magnitude of their updates.

The key difference of Dq-FFL from q-FFL is that the boosting parameter q is dynamically computed by each client, scaling the global q parameter by the ratio between the loss experienced by the client in the previous round l_i and the median loss l_{med}. This objective function emphasizes the updates of the clients with the largest loss. The full algorithm is reported in Algorithm 3.

Since selfish clients cannot tamper with the local training step, the dynamic q ensures that, when the selfish clients upscale their updates, if they experience better performances than the normal clients, they would be forced to send smaller updates, mitigating the instability brought on by the selfish clients and improving the recovery of the true update. While testing this algorithm, we assume that the selfish clients also upscale the transmitted loss l_i so that it becomes compatible with their larger update and tries to skew l_{med}.

Algorithm 2: RFL-Self Algorithm.

Input: A set of k clients
Output: A trained global model with weights $\mathbf{w}_{[k]}$
$\mathbf{w}_{[k]} \leftarrow$ random initialization
Send $\mathbf{w}_{[k]}$ to all the clients
for *each communication round* **do**
  Receive updates $\{\delta_i,\ \forall i \in [k]\}$
  Compute $\boldsymbol{\mathcal{N}} \leftarrow \{||\delta_i||,\ \forall i \in [k]\}$
  Compute $\delta_{med} \leftarrow$ MARGINALMEDIAN$(\{\delta_i,\ \forall i \in [k]\})$
  $\mathcal{S} \leftarrow$ SELFISHUPDATESDETECTION$(\{\delta_i,\ \forall i \in [k]\})$
  for *each client* $i \in \mathcal{S}$ **do**
    Obtain β by solving (9)
    Estimate δ'_i using (8)
  Update $\mathbf{w}_{[k]} \leftarrow \mathbf{w}_{[k]} + \frac{1}{\Gamma}(\sum_{i\in[k]\setminus\mathcal{S}} \delta_i + \sum_{j\in\mathcal{S}} \delta'_j)$
  Send the updated global model $\mathbf{w}_{[k]}$ to all the clients
return $\mathbf{w}_{[k]}$

V. Theoretical Analysis

This section discusses the theoretical aspects of the proposed strategy, providing insights into its properties and behavior.

Algorithm 3: Dq-FFL Algorithm.

Input: A set of k clients, a learning rate L, a parameter q to control the level of fairness
Output: A trained global model with weights $\mathbf{w}_{[k]}$

$\mathbf{w}_{[k]} \leftarrow$ random initialization
Send $\mathbf{w}_{[k]}$ to all the clients
for *each communication round* **do**
  for $i \in [k]$ **do**
    At Client
    $\delta_i \leftarrow \text{LocalTraining}(\mathbf{w}_{[k]})$
    if $round == 0$ **then**
      $q_i \leftarrow q$
    else
      $q_i \leftarrow q\frac{l_{med}}{l_i}$
    $\omega_i \leftarrow qF_k^{q-1}(\mathbf{w}_{[k]})\|\delta_i\|^2 + LF_i^q(\mathbf{w}[k])$
    $\delta_i \leftarrow F_i^q(\mathbf{w}[k])L\delta_i$
    $l_i \leftarrow F(\mathbf{w}_{[k]})$
    Send δ_i, ω_i, l_i to the server
  At FL Server
  Receive $\{\delta_i,\ \forall i \in [k]\}$, $\{\omega_i,\ \forall i \in [k]\}$, $\{l_i,\ \forall i \in [k]\}$
  Update $\mathbf{w}_{[k]} \leftarrow \mathbf{w}_{[k]} + \frac{\sum_{i\in[k]}\delta_i}{\sum_{i\in[k]}\omega_i}$
  $l_{med} \leftarrow \text{median}(\{l_i,\ \forall i \in [k]\})$
  Send the global model $\mathbf{w}_{[k]}$ and l_{med} to all the clients
return $\mathbf{w}_{[k]}$

A. Recovery Mechanism Guarantees

Intuitively, the convex combination (in (8)) alleviates the effect of the selfish update by deviating its direction opposite to the average update of normal clients. To analyze this intuition quantitatively, we assume three conditions in place.

- *Condition 1:* The average update $\delta_{[k]\setminus\mathcal{S}}$ (excluding selfish clients) does not vary significantly in consecutive rounds.
- *Condition 2:* The median update δ_{med} reliably approximates the mean update of all non-selfish clients.
- *Condition 3:* The true update δ_s from a selfish client has a magnitude comparable to $\mathcal{N}_{med}$.

When conditions 1 and 2 are satisfied, it follows that $\bar{\delta}_{[k]\setminus\{s\}} \approx \delta_{med}$. Substituting this into Eqs. (5) and (8), we derive

$$\begin{aligned}\delta'_s &\simeq \beta\left[\phi\Gamma\left(\delta_s/\omega_s - \bar{\delta}_{[k]\setminus\{s\}}\right) + \omega_s\bar{\delta}_{[k]\setminus\{s\}}\right] + (1-\beta)\delta_{med}\\ &\simeq \beta\phi\Gamma\frac{\delta_s}{\omega_s} + (1-\beta\phi\Gamma)\omega_s\bar{\delta}_{[k]\setminus\{s\}}. \end{aligned} \tag{12}$$

This estimated update δ'_s corresponds to the unweighted δ_s when $\beta = 1/(\phi\Gamma)$. However, since the server lacks knowledge of ϕ, it determines β (ignoring ϕ) by solving $||\delta'_s|| = \mathcal{N}_{med}$, as described in (9). When all three conditions are satisfied, solving (9) for β yields $\beta \simeq 1/(\phi\Gamma)$, which ensures that δ'_s closely approximates δ_s. Part (a) of Figure 5 illustrates how effectively RFL-Self can recover the selfish update δ'_s when the three conditions are satisfied.

Even if the conditions are not fully met, RFL-Self can still yield a reasonable approximation. For instance, if Condition 2 is not satisfied, we obtain $\delta'_s = \beta\phi\Gamma\delta_s/\omega_s + (\delta_{med} - \beta\phi\Gamma\omega_s\bar{\delta}_{[k]\setminus\{s\}}) + \beta(\omega_s\bar{\delta}_{[k]\setminus\{s\}} - \delta_{med})$. This result deviates from (12) by $(1-\beta)(\delta_{med} - \omega_s\bar{\delta}_{[k]\setminus\{s\}})$. The magnitude of this deviation (error) is bounded above by $\sqrt{\text{Tr}(\text{var}(\delta))}$ [41]. If Condition 3 is not satisfied, the estimated update will be closer to the median update than the true update, with a maximum deviation of $||\delta_s - \delta_{med}||$. Part (b) of Figure 5 illustrates the discrepancy between the recovered update δ'_s and the true update δ_s when none of the conditions are met.

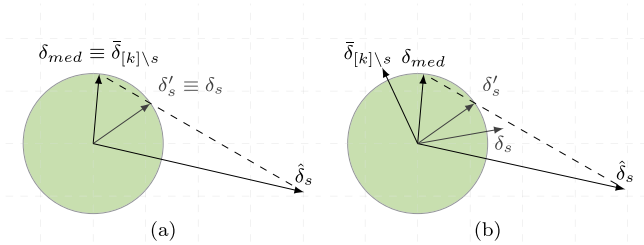

Figure 5. Illustrating how close the recovered update δ'_s is to the true δ_s: (a) when all three conditions meet, and (b) when all three conditions do not hold.

RFL-Self versus Downscaling strategy: a trivial way one might be tempted to adopt to mitigate the impact of selfish updates entails simply downscaling them by a factor β before aggregation, which essentially means $\delta'_s = \frac{\hat{\delta}_s}{\beta} = \frac{\phi\Gamma}{\beta\omega_s}\delta_s + \frac{1-\phi\Gamma}{\beta}\bar{\delta}_{[k]\setminus\{s\}}$. No matter how the β is chosen, the estimated update δ'_s differs from the true update due to a component opposite to $\bar{\delta}_{[k]\setminus\{s\}}$. This component disappears only when $\phi = \frac{1}{\Gamma}$, where selfish clients behave like normal ones. RFL-Self enhances robustness by approximating δ'_s to align more closely with the actual update δ_s.

Theorem 2: The maximum error in the recovered aggregated update is bounded by $\frac{4k+k^2}{4\Gamma^2}\,\text{Tr}(\text{var}(\delta))\mathbb{E}[\omega]$

Proof: Based on Theorem 1, when there are s suspected selfish clients, the expected error can be expressed as:

$$\begin{aligned}\frac{\mathbb{E}\|\delta_{[k]} - \delta'_{[k]}\|^2}{\mathbb{E}[\omega]} &= \frac{1}{\Gamma^2}\mathbb{E}\left\|\sum_{i\in[k]}\delta_i - \left(\sum_{i\in[k]\setminus\mathcal{S}}\delta_i + \sum_{j\in\mathcal{S}}\delta'_j\right)\right\|^2\\ &= \frac{1}{\Gamma^2}\mathbb{E}\left\|\sum_{i\in\mathcal{S}}\delta_i - \sum_{j\in\mathcal{S}}\delta'_j\right\|^2 \end{aligned} \tag{13}$$

Since the RFL-Self leaves all $[k]\setminus\mathcal{S}$ updates unchanged, the error essentially comes from the updates whose norm is larger than $\tau\cdot\text{MAD} + \mathcal{N}_{med}$. To formalize the maximum bound, the error induced by the selfish updates can be approximated by the median update as given below

$$\begin{aligned}\frac{\mathbb{E}[\omega]}{\Gamma^2}\mathbb{E}\left\|\sum_{i\in\mathcal{S}}\delta_i - \sum_{j\in\mathcal{S}}\delta'_j\right\|^2 &\le \frac{\mathbb{E}[\omega]}{\Gamma^2}\mathbb{E}\left\|\sum_{i\in\mathcal{S}}(\delta_i - \delta_{med})\right\|^2\\ &= \frac{\mathbb{E}[\omega]}{\Gamma^2}\mathbb{E}\left\|\sum_{i\in\mathcal{S}}\left(\delta_i - \bar{\delta}_{[k]} - \delta_{med} + \bar{\delta}_{[k]}\right)\right\|^2\end{aligned}$$

Using Jensen's inequality [42]

$$\leq \frac{\mathbb{E}[\omega]}{\Gamma^2}\mathbb{E}\left\|\sum_{i\in\mathcal{S}}\left(\delta_i-\bar{\delta}_{[k]}\right)\right\|^2+\frac{\mathbb{E}[\omega]\,|\mathcal{S}|^2}{\Gamma^2}\mathrm{Tr}(\mathrm{var}(\delta))$$

$$\leq \frac{k\mathbb{E}[\omega]}{\Gamma^2}\mathrm{Tr}(\mathrm{var}(\delta))+\frac{\mathbb{E}[\omega]\,|\mathcal{S}|^2}{\Gamma^2}\mathrm{Tr}(\mathrm{var}(\delta))$$

$$\leq \frac{4k+k^2}{4\Gamma^2}\,\mathrm{Tr}(\mathrm{var}(\delta))\mathbb{E}[\omega] \quad \text{since } |\mathcal{S}|\leq\frac{k}{2}. \tag{14}$$

□

In the case of FedAvg, since $\Gamma = k$ and $\omega = 1$, the error bound is $(\frac{1}{k}+\frac{1}{4})\mathrm{Tr}(\mathrm{var}(\delta))$

Remark 6: From Theorem 2 it follows that the error remains unaffected by the count of selfish clients, provided they do not significantly skew the median, even if the median is not an accurate estimate of the mean update of normal clients.

B. Convergence Analysis

The convergence of RFL-Self can be analyzed by considering the convergence of using the median as a robust aggregation method [43]. The proof lies in three fundamental assumptions:

1) The median is an unbiased estimator of the mean update: $\mathbb{E}[\delta_{med}] = \mathbb{E}[\bar{\delta}_{[k]}]$.
2) F has a Lipschitz gradient: $\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\| \leq L\|\mathbf{w}-\mathbf{w}'\|$.
3) The variance of the updates has an upper bound: $\mathbb{E}[\|\delta_{med} - \mathbb{E}[\delta_{med}]\|^2] \leq \sigma^2$.

The first two assumptions are standard for stochastic optimization, while the third is the classical bounded variance assumption for mean-based aggregation methods modified for the median. Additionally, we assume an odd number of clients to ensure that the median is well-defined.

Theorem 3: Suppose that the above three assumptions hold, given a learning rate $\lambda = \min(\frac{1}{\sqrt{Tk}}, \frac{1}{2L})$, and setting $D_F \triangleq F(\mathbf{w}^0_{[k]}) - F(\mathbf{w}^*)$, then RFL-Self yields the following convergence rate:

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left[F\left(\mathbf{w}^t_{[k]}\right)\right] \leq 2\sqrt{\frac{k}{T}}(D_F+L\sigma^2)+\frac{1}{T}\sum_{t=0}^{T} \times \mathbb{E}\left[\left\|\nabla F(\mathbf{w}^t_{[k]}-\mathbb{E}[\delta_{med}]\right\|^2\right] \tag{15}$$

Proof: We analyze a (unrealistic) worst-case scenario where the server cannot distinguish between the selfish and normal clients and all the clients are detected as selfish. One such scenario is if the MAD is zero, which implies that more than half of the updates have the same norm. In this case, for all the flagged clients (9) would yield $\beta = 0$, since $\beta = 1$ corresponds to no mitigation at all and is thus not acceptable. With $\beta = 0$, all the estimated updates would be replaced by the median update.

By the Lipschitz gradient assumption and the update rule of (10), the convergence of RFL-Self is guaranteed.

$$F(\mathbf{w}^{t+1}_{[k]}) \leq F\left(\mathbf{w}^t_{[k]}\right)+\frac{\lambda^2 L}{2}\|\delta^t_{med}\|^2-\lambda\langle\nabla F\left(\mathbf{w}^t_{[k]}\right),\delta^t_{med}\rangle$$

$$\leq F(\mathbf{w}^t_{[k]})+\lambda^2 L\left(\|\delta^t_{med} -\mathbb{E}[\delta_{med}]\|^2+\|\mathbb{E}[\delta_{med}]\|^2\right) -\lambda\left\langle\nabla F\left(\mathbf{w}^t_{[k]}\right),\delta^t_{med}\right\rangle$$

Rearranging $F(\mathbf{w}^t_{[k]})$ and taking the expectation for both sides of the inequality, we obtain

$$\mathbb{E}[F(\mathbf{w}^{t+1}_{[k]})]-\mathbb{E}\left[F\left(\mathbf{w}^t_{[k]}\right)\right]$$

$$\leq -\lambda\mathbb{E}\left[\left\langle\nabla F(\mathbf{w}^t_{[k]}),\delta^t_{med}\right\rangle\right] +\lambda^2 L\mathbb{E}\left[\|\delta^t_{med}-\mathbb{E}[\delta_{med}]\|^2+\|\mathbb{E}[\delta_{med}]\|^2\right]$$

$$= -\frac{\lambda}{2}\mathbb{E}\left[\nabla\|F(\mathbf{w}^t_{[k]})\|^2+\|\mathbb{E}[\delta_{med}]\|^2 -\|\nabla F(\mathbf{w}^t_{[k]})-\mathbb{E}[\delta_{med}]\|^2\right] +\lambda^2 L\mathbb{E}\left[\|\delta^t_{med}-\mathbb{E}\left[\delta_{med}\right]\|^2+\|\mathbb{E}[\delta_{med}]\|^2\right]$$

$$= -\frac{\lambda}{2}\mathbb{E}\left[\|\nabla F(\mathbf{w}^t_{[k]})\|^2\right]-\frac{\lambda-2\lambda^2 L}{2}\mathbb{E}\left[\|\mathbb{E}\left[\delta_{med}\right]\|^2\right] +\frac{\lambda}{2}\mathbb{E}\left[\|\nabla F(\mathbf{w}^t_{[k]})-\mathbb{E}[\delta_{med}]\|^2\right] +\lambda^2 L\mathbb{E}\left[\|\delta^t_{med}-\mathbb{E}[\delta_{med}]\|^2\right]$$

Setting $\lambda = \min(\frac{1}{\sqrt{Tk}}, \frac{1}{2L})$, dividing both sides by $\frac{\lambda T}{2}$, and explicitly expressing the telescoping sum, we can derive

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}[F(\mathbf{w}^t_{[k]})]$$

$$\leq 2\sqrt{\frac{k}{T}}\left(\mathbb{E}[F(\mathbf{w}^0_{[k]})]-\mathbb{E}[F(\mathbf{w}^{t+1}_{[k]})]\right) +\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}[\|\nabla F(\mathbf{w}^t_{[k]}-\mathbb{E}[\delta_{med}]\|^2]+2L\sqrt{\frac{k}{T}}\sigma^2.$$

Since the difference between the starting loss and the optimal loss is bounded by D_F, (15) is obtained as the final upper bound for the convergence of RFL-Self. □

Remark 7: If the difference between the expected median and the true gradients is small, the sum on the right-hand side of (15) is negligible, and the algorithm reaches a stationary point with a rate of $\sqrt{\frac{k}{T}}$, comparable to the $\sqrt{\frac{1}{kT}}$ rate of FedAvg, with some additional variance due to the use of the median with non-iid client data. If the median does not approximate well the mean update of normal clients, the convergence rate would be affected by a constant factor.

Remark 8: It is possible to detect all the selfish clients and recover a good approximation of their true updates within some theoretical bounds.

VI. Experimental Evaluation

This section provides an empirical evaluation of the influence of selfish clients on the performance of FL systems and assesses the effectiveness of the proposed FairRFL approach. We use

two benchmark datasets: CIFAR-10 [44], an image classification task, and WISDM-W, a time-series human activity recognition task from the FedAIoT benchmark [45]. These two datasets are chosen due to their popularity in the FL community and their different levels of complexity, which allows us to evaluate the performance of our proposed method in diverse scenarios. Experiments were carried out with 50 clients at varying levels of selfishness ϕ and up to 30% selfish clients. To simulate the non-IID scenario, we partition the dataset such that each client gets only the data for two randomly selected classes. This severe non-IID setting is chosen to showcase the effect of selfish clients better: if the data is IID, then all the clients have the same local objective, and the selfish clients would not have any incentive to act selfishly. Additionally, different levels of non-IID data distribution are analyzed in Section VI-B to assess the generalizability of our findings. Larger datasets like CIFAR-100 [44] are not considered, as the lack of overlap between the classes of the clients would make the selfishness detection trivial. For the WISDM-W dataset, we utilize a ResLSTM architecture comprising 4 Long Short-Term Memory cells with residual connections. For the CIFAR-10 dataset, we employ a CNN with 3 convolutional layers and 2 fully connected layers. The hyper-parameters used are as follows: *optimizer*: SGD, *batch size*: 32/256, and *learning rate*: 0.01/0.1 for WISDM-W/CIFAR-10 datasets, respectively. The models are trained over 30 FL rounds, with each client performing five local epochs per round to strike a balance between communication efficiency and convergence speed. Unless otherwise specified, the results are obtained by selecting all clients for aggregation in each round to reduce spurious effects due to client selection. However, we also analyze the impact of the selfish behavior appearing in only a fraction of the rounds in Section VI-E. All the experiments are implemented in Python using a well-known library PyTorch 1.12.1, on a Windows 11 powered with an NVIDIA RTX A5000 GPU. Our source code is publicly available at https://github.com/ndslab-group/FairRFL.

A. Global Update Estimation

At first, we carry out experiments to analyze the ability of a selfish client to correctly estimate the value of Γ. These experiments are conducted with a single selfish client to avoid the interference of other clients. We report the results of the estimation technique on both datasets, WISDM-W and CIFAR-10, in Figure 6, which reports the results for FedAvg in part (a). Here $\Gamma = k$ is fixed for all the communication rounds. For both datasets, the estimation technique can correctly converge towards the correct value of Γ. Figure 6(b) reports the results for Dq-FFL, where Γ is computed dynamically as $\sum \omega_i$. The value of Γ varies between rounds and, in the case of CIFAR-10, it spans multiple orders of magnitude. Nevertheless, the estimation technique can closely track the value of Γ, demonstrating its effectiveness.

To further assess the capability of a selfish client to approximate the global update, we compute the cosine similarity S_C between the true global update and the estimated one by the selfish client across different communication rounds. Figure 7 shows the cosine similarity for both datasets using the FedAvg and Dq-FFL algorithms. For both datasets and algorithms, the selfish client can accurately estimate the global update, with a cosine hovering around 0.8, indicating a strong alignment between the true and estimated updates.

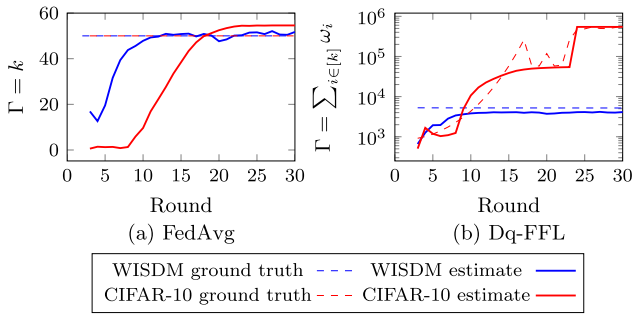

Figure 6. Estimation of Γ in the presence of a single selfish client.

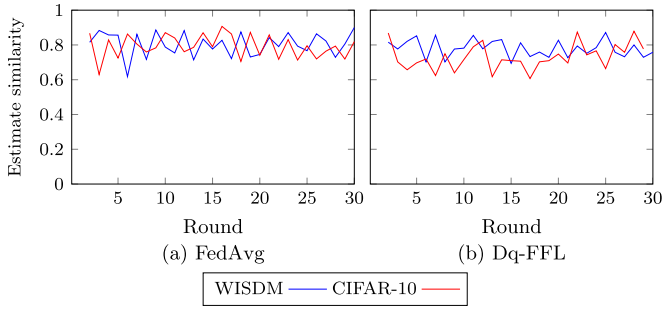

Figure 7. Cosine similarity between the true and estimated global updates by a selfish client.

B. Impact of Undetected Selfish Clients on Performance

We conduct experiments to evaluate the impact of selfish client(s) on the accuracy of the global model. The results for the FedAvg algorithm are presented in Figure 8. For the CIFAR-10 dataset in part (a), the mean test accuracy (represented by the 'green' box plots) for normal clients steadily declines as the selfishness level ϕ rises. However, beyond a certain threshold (i.e., $\phi > 0.6$), the accuracy of selfish clients appears to stabilize and no longer improves. A similar trend can be observed for the WISDM-W dataset in part (c), although the trend is less smooth. Nevertheless, the selfish clients can achieve a higher accuracy than the normal clients, and the overall variance of the test accuracy of the global model across normal clients tends to increase with the selfishness level ϕ for both datasets. Interestingly, when multiple selfish clients are present (parts (b) and (d)) they fail to gain significant advantages. Instead, their selfish actions counteract each other, leading to a situation where no one benefits and the average outcome steadily worsens as ϕ increases. Selfish behavior amplifies accuracy variance, lowering performance and rendering the model unfair and unreliable for normal participants.

Next, we carry out experiments to analyze the impact of selfish client(s) on the global model accuracy for the Dq-FFL algorithm

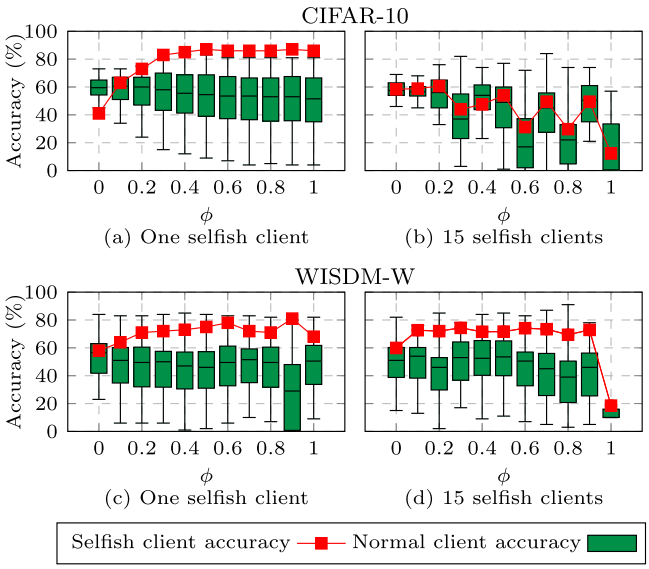

Figure 8. Test accuracy of the global model with one and 15 selfish clients (out of 50 clients) on the and CIFAR-10 datasets using **FedAvg** algorithm.

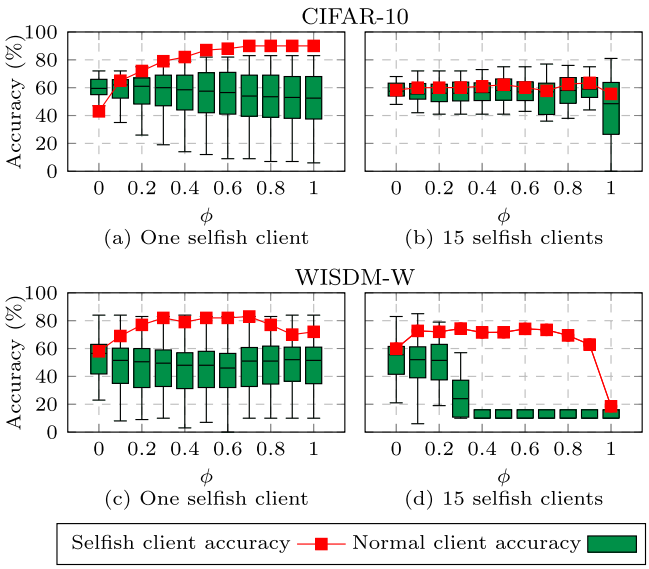

Figure 9. Test accuracy of the global model with one and 15 selfish clients (out of 50 clients) on the WISDM-W and CIFAR-10 datasets using **Dq-FFL** algorithm.

and report results in Figure 9. With one selfish client, in parts (a) and (c), there is little difference in outcomes compared to FedAvg, showing that the selfish client is effective even if the underlying algorithm strives to ensure fairness. However, with multiple selfish clients, in parts (b) and (d), the instability brought on by the multiple selfish clients is mitigated, for low values of ϕ, hinting that a fairness mechanism can be an effective component of a defense against selfish clients. For the WISDM-W dataset, at higher levels of selfishness the negative impact of selfish clients is exacerbated, significantly degrading the global model accuracy for normal clients, confirming the

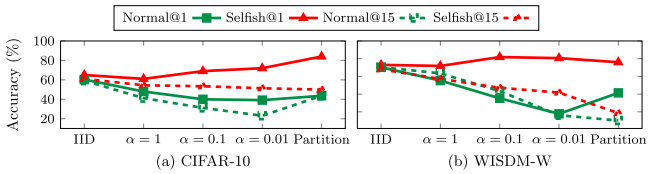

Figure 10. Influence of label distribution skew on the impact of selfish behavior.

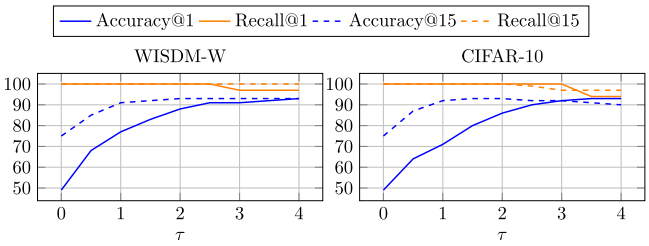

Figure 11. Effectiveness of selfish client detection with $\phi = 0.7$ and sensitivity to the τ parameter with 1 and 15 selfish clients.

necessity of a dedicated mitigation mechanism. This instability is in accordance with the observations on the toy numerical example in Section VI-C. The effect is more pronounced for the WISDM-W dataset, likely due to the larger model size, which allows selfish clients to create more disruptive updates.

Finally, Figure 10 shows the impact of varying levels of non-IID data distribution on the performance degradation caused by selfish clients with fixed selfishness level 0.7. The degree of label distribution skew varies from an IID distribution (i.e., each client has data from all the classes with equal probability) to a partitioned non-IID distribution (i.e., each client has data from only two classes), with two intermediate settings where the skew is determined by a Dirichlet distribution with parameters $\alpha \in \{1, 0.1, 0.01\}$. As predictable, in IID settings where clients have no incentive to behave selfishly (all the clients have the same local objective), the impact of selfishness is negligible. However, even at mild levels of non-IID, selfish clients can cause significant performance degradation, which worsens as the data distribution becomes more skewed. Moreover, in all non-IID settings, the presence of multiple selfish clients leads to a more pronounced degradation of the global model accuracy compared to the case with a single selfish client.

C. Performance of Proposed Algorithms

We conduct experiments to evaluate the performance of the proposed methods, RFL-Self and FairRFL. First, we analyze the performance of the detection mechanism, and then we turn to the full mitigation mechanism.

1) Accuracy of the Selfish Client Detection: The accuracy of the detection strategy (shared by both approaches) is crucial to the effectiveness of the mitigation mechanism. Undetected selfish clients can have a significant negative impact on the global model, as shown in the previous section, and false positives can unfairly penalize normal clients. We carry out experiments to analyze the impact of the threshold τ on the accuracy and recall

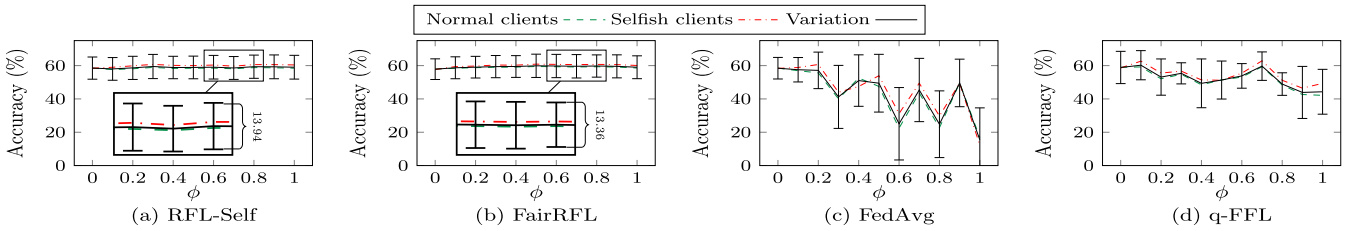

Figure 12. Accuracy of the global model for different values of ϕ with 15 selfish clients on the local dataset for the CIFAR-10 dataset.

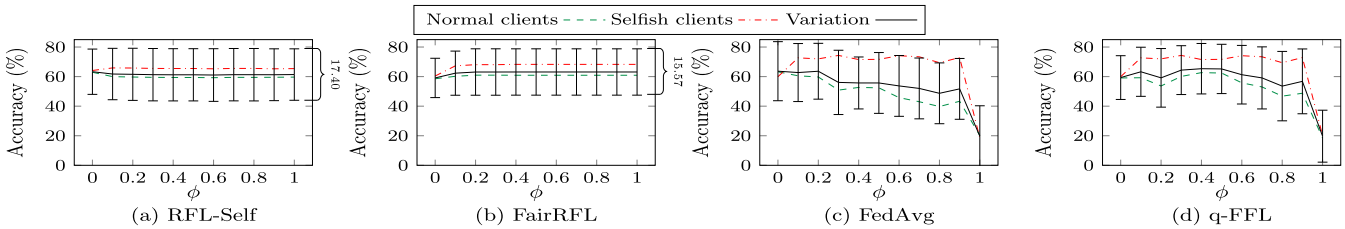

Figure 13. Accuracy of the global model for different values of ϕ with 15 selfish clients on the local dataset for the WISDM-W dataset.

of the detection mechanism on both datasets, with one and fifteen selfish clients, and report results in Figure 11.

The detection accuracy is high for both datasets, although, on average, it is slightly lower for the CIFAR-10 dataset. This could hint that larger models, such as the one used for WISDM-W, might make selfish updates more distinct from normal ones, thus making the detection easier. Nevertheless, even if the accuracy for the CIFAR-10 dataset is lower, the detection mechanism is still able to correctly identify all the selfish clients while maintaining an acceptable false positive rate. We hypothesize that, given that the first layers of shallow CNN models tend to learn more generic features, the selfish updates might have a smaller impact on those layers, making the overall update norm less distinguishable from normal ones. In larger models, more layers can be affected by selfish behavior, making the update norm more distinct. The heuristic choice of $\tau = 2.5$ appears to be a good compromise between achieving a high accuracy in the detection process, avoiding the inclusion of too many normal clients in the set of suspected selfish clients, and ensuring 100% recall with all selfish clients detected. At higher values of τ, some selfish updates are not detected, which is not acceptable given the disruptive impact of selfish clients. This level of accuracy is sufficient to ensure that the mitigation mechanism can correctly identify the selfish clients and mitigate their negative impact on the global model, as demonstrated in the next section.

2) Mitigation Accuracy: Next, we evaluate the performance of FairRFL in the presence of selfish clients with and without the fairness component, and compare the results with FedAvg when no mitigation mechanism is employed. Figs. 12 and 13 present the test accuracy for both selfish and normal clients across different ϕ values, considering fifteen selfish clients. An immediate insight from the results is that when $\phi > 0.4$, selfishness can lead to significant accuracy loss for both normal and selfish clients in the absence of robust aggregation at the server. While trying to enforce fairness with methods such as q-FFL (part d of the figures) reduces the instability induced by selfish clients, it still leads to a significant drop in accuracy and a high variance in the performance for all the clients, negating the attempts at fairness. Indeed, as little as $\phi = 0.1$ is enough to increase the variation of the accuracy experienced by the clients. An effective mitigation strategy should maintain the accuracy of normal clients while ensuring that selfish clients do not experience significant changes. In this regard, all the evaluated mitigation strategies appear to perform effectively by stopping selfish clients from causing substantial accuracy drops. Additionally, the Dq-FFL algorithm can noticeably reduce the inter-client accuracy variance. In Figure 13, we can see that, in the presence of selfish clients, with FairRFL, the standard deviation of the accuracy is noticeably reduced for both datasets compared to the results with RFL-Self. An interesting point is that the level of selfishness has a negligible impact on the model when FairRFL is employed, confirming its effectiveness.

D. Performance Comparison of Different Approaches

We also compare the performance of different approaches in the presence of a single selfish client with $\phi = 0.7$. We choose these values of ϕ with the help of parts (b) and (d) of Figure 8, where at $\phi = 0.7$, with multiple selfish clients, the model is greatly perturbed, and a single selfish client can achieve a significant accuracy gain. The results are reported in Figure 14. We report the average accuracy of the global model on the local test dataset of normal and selfish clients, as well as the standard deviation of the accuracy of all the clients, shown as error bars. On both datasets, neglecting mitigation efforts leads to up to double the standard deviation in accuracy and a notable disparity between the accuracy of normal and selfish clients. We also compare FairRFL with two additional state-of-the-art algorithms: FedMut [31], which sends carefully mutated versions of the global model to clients to limit the

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES FOR AN FL SETUP WITH THE VARYING NUMBER OF SELFISH CLIENTS AND $\phi = 0.7$

	Approaches	0% Selfish clients			10% Selfish clients			20% Selfish clients			30% Selfish clients		
		ACC_n	ACC_s	STD	ACC_n	ACC_s	STD	ACC_n	ACC_s	STD	ACC_n	ACC_s	STD
WISDM-W	FedAvg	62.10	-	16.95	46.22	70.20	18.73	44.93	71.17	18.36	43.00	73.40	20.57
	q-FFL	62.30	-	16.74	45.02	57.20	17.81	18.93	19.00	18.36	18.92	19.13	16.36
	Dq-FFL	62.30	-	16.68	46.62	69.00	17.56	45.93	68.30	18.36	48.20	75.27	17.42
	Median	53.52	-	17.08	43.07	55.60	17.10	37.33	51.80	16.98	37.74	43.67	16.92
	FedMut	60.92	-	16.55	37.93	48.00	16.36	32.76	31.00	18.36	22.36	27.13	16.36
	FedCDA	63.56	-	17.40	47.79	71.60	18.38	46.02	63.90	18.47	38.60	57.87	20.43
	RFL-Self	62.10	-	16.95	61.13	63.20	17.06	58.92	59.90	17.38	58.14	61.80	17.40
	q-FFL+RFL-Self	62.30	-	16.74	61.07	63.00	16.88	59.40	60.40	17.52	58.17	60.33	17.19
	FairRFL	61.33	-	16.64	**61.31**	63.20	**16.22**	**59.60**	60.80	**15.71**	**60.91**	68.27	**15.57**
CIFAR-10	FedAvg	61.14	-	6.89	57.47	34.60	15.85	42.85	60.90	19.34	43.60	49.97	18.99
	q-FFL	60.44	-	6.70	51.49	74.60	17.83	56.58	63.40	9.55	59.23	62.93	8.54
	Dq-FFL	60.36	-	6.61	53.93	73.40	16.19	55.92	63.80	10.91	54.37	57.73	11.33
	Median	56.18	-	7.07	53.91	52.40	6.91	52.00	53.80	7.38	50.29	52.13	6.96
	FedMut	57.20	-	7.41	55.49	61.00	8.18	51.15	57.10	7.57	39.83	39.60	15.39
	FedCDA	58.28	-	10.10	41.69	56.60	21.45	40.98	51.40	18.49	37.49	41.93	17.19
	RFL-Self	59.64	-	7.42	59.19	54.40	7.24	59.12	60.00	6.70	58.11	59.67	6.86
	q-FFL+RFL-Self	59.22	-	7.05	59.71	54.00	6.81	59.02	59.50	6.68	57.97	59.33	7.75
	FairRFL	59.64	-	6.73	**59.72**	53.20	**6.67**	**59.58**	60.30	**6.62**	**59.20**	60.40	**6.52**

ACC_n: Average accuracy of normal clients, ACC_s: Average accuracy of selfish clients, STD: Standard deviation of the accuracy of all clients.

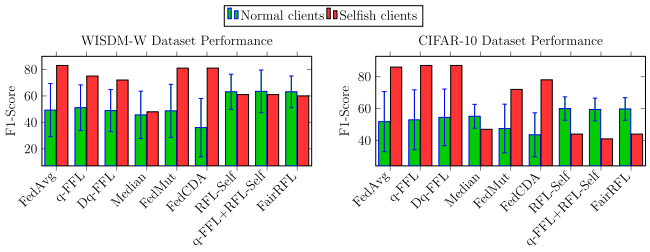

Figure 14. Performance of the global model on the two datasets under the different approaches for an FL setup with one selfish client and $\phi = 0.7$.

impact of heterogeneity (which is the underlying motivation for selfish behavior); and FedCDA [30], which employs a buffer of historical updates to mitigate the divergence between the current local updates and the overall global objective (which is the mechanism exploited by selfish clients). As also seen by the results, varying label skew, reducing the heterogeneity as done by FedMut, has a limited effect, comparable to FedAvg. FedCDA, by still including selfish updates in the aggregation, ends up interfering with the Γ estimation, leading to stronger selfish updates and a worse outcome for normal clients. These results confirm the need for a bespoke solution to the selfish client problem that can recover useful information from their updates, since simple mitigation of the underlying mechanisms exploited by selfish clients is not sufficient.

Lastly, we examine how the effectiveness of the evaluated strategies varies as the proportion of selfish clients increases, keeping ϕ constant at 0.7. From the results in Table III, it is evident that, with the notable exception of the underperforming median, most approaches achieve comparable accuracy when no selfish clients are present. However, the accuracy can get as low as 18.93% if no mitigation strategy is employed. Additionally, the standard deviation of the accuracy experienced by the clients reaches up to 20.57 for the FedAvg algorithm. Excluding FedAvg, in the presence of selfish clients, FedCDA experiences the highest increase in the standard deviation of the accuracy of normal clients.

RFL-Self ensures a balance between the accuracy of normal and selfish clients while preserving the overall system performance, regardless of the proportion of selfish clients. By itself, however, RFL-Self cannot tackle the fairness requirements. Naively combining RFL-Self with q-FFL does not achieve the desired fairness either. Indeed, in some occurrences, this combination experiences a standard deviation of accuracy up to 13% larger than that experienced by RFL-Self alone. The FairRFL-Self, instead, can effectively reduce the unfairness in the system, while maintaining a comparable accuracy to that of RFL-Self for both normal and selfish clients. Indeed, FairRFL outperformed on the CIFAR-10 dataset only in the absence of selfish clients, and the best-performing strategy in this case is Dq-FFL. On the WISDM-W dataset, in the 0% selfish clients setting FairRFL is outperformed by FedMut, albeit by a negligible difference of 0.09.

E. Ablation Study on Non-Persistent Selfish Clients

Finally, we investigate the performance of the considered strategies when the selfish clients are not persistent, i.e, when not all rounds are equally affected by selfish behavior. The aim of this analysis is three-fold: first, to assess the effectiveness of the proposed strategies when, to try to evade detection, the selfish clients are active only for a fraction of the FL rounds, and they behave normally for the remaining rounds; second, to understand if a selfish client that is not always selected for participation in the FL rounds can still have a significant impact on the global model; third, to assess whether the impact of selfish clients is more pronounced at specific stages of the FL process. We consider the same setup as in the previous section, with 50 clients and $\phi = 0.7$. We vary the percentage of FL rounds in which the selfish clients are active from 0%, where the selfish behavior is never active, to 100% .

From parts (b) and (d) of Figure 15, we can see that the non-persistent selfish clients can still cause a significant accuracy reduction for the normal clients and a high standard deviation of the accuracy if left unmitigated. Moreover, they are successful in their attempt at achieving a higher accuracy

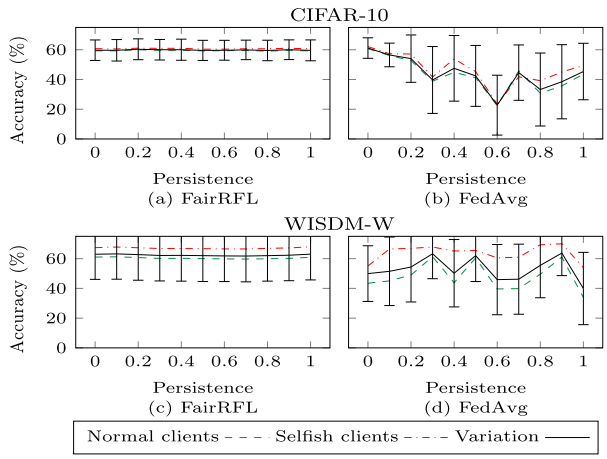

Figure 15. Impact of non-persistent selfish clients and effect of FairRFL on the CIFAR-10 and WISDM-W datasets.

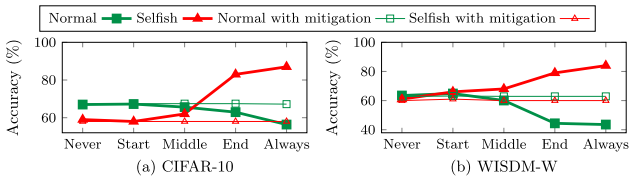

Figure 16. Impact of selfishness at different stages of the FL process on the two datasets. "Start refers to rounds 1-10, "Middle" to rounds 11-20, and "End" to rounds 21-30.

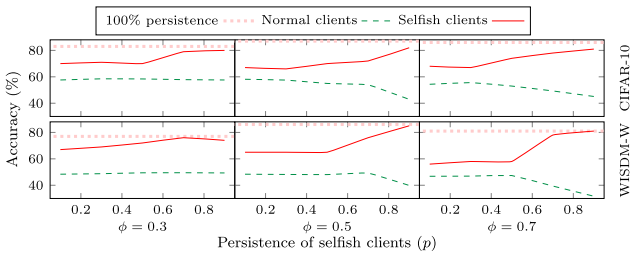

Figure 17. Impact of different selfishness levels with varying persistence with one selfish client on the two datasets.

than the normal clients. On the other hand, as shown in parts (a) and (c), the FairRFL approach can effectively mitigate the selfish clients, even when they are not persistent, maintaining almost the same accuracy for both classes of clients and a similar standard deviation irrespective of the percentage of FL rounds influenced by selfish clients.

Additionally, we investigate the impact of a selfish client restricting its selfish behavior to various stages of the FL process. We consider the same setup as in the previous section, with 50 clients and $\phi = 0.7$, and report the results in Figure 16. We observe that the impact of the selfish clients has mostly appeared in the late stages of the FL process, where the model is already close to convergence. This is likely because the selfish clients can exploit the model's proximity to convergence to better estimate the global updates and thus craft better updates. In all these instances, RFL-Self can effectively mitigate the selfish clients, ensuring stable performance.

Finally, Figure 17 shows that, without proper mitigation, a selfish client can manifest a significant negative impact even if it is not selected for most of the FL rounds.

VII. Conclusion

With the introduction of a novel notion of *selfish clients* who can deviate the overall FL training in their favor, we proposed FairRFL that incorporates: RFL-Self, a robust aggregation method to counteract the effects of selfish clients, and Dq-FFL, a fairness mechanism to ensure balanced performance across all clients. Through comprehensive analysis, we demonstrated that selfish clients could significantly disrupt the training process, potentially preventing convergence. In addition to the comprehensive theoretical analysis, we performed extensive experiments on two benchmark datasets across different levels of selfishness. The results demonstrated that FairRFL effectively mitigates selfish behavior without compromising the accuracy for normal clients, outperforming other standard approaches. This work opens up further research directions for investigating the collusion among selfish clients and their impact on performance disparities.

References

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016, *arXiv:1602.05629*.

[2] E. M. Campos, A. Gonzalez-Vidal, J. L. Hernández-Ramos, and A. Skarmeta, "FedRDF: A robust and dynamic aggregation function against poisoning attacks in federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 13, no. 1, pp. 48–67, First Quarter, 2024.

[3] Z. Luan, W. Li, M. Liu, and B. Chen, "Robust federated learning: Maximum correntropy aggregation against Byzantine attacks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 62–75, Jan. 2025.

[4] A. Gupta, T. Luo, M. V. Ngo, and S. K. Das, "Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning," in *Computer Security–ESORICS 2022*, V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, Eds. Cham, Switzerland: Springer, 2022, pp. 445–465.

[5] H. Wang et al., "Attack of the tails: Yes, you really can backdoor federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16070–16084.

[6] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 2545–2555.

[7] S. Awan, B. Luo, and F. Li, "CONTRA: Defending against poisoning attacks in federated learning," in *Proc. 26th Eur. Symp. Res. Comput. Secur.*, Darmstadt, Germany, 2021, pp. 455–475.

[8] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in Sybil settings," in *Proc. Int. Symp. Recent Adv. Intrusion Detection*, 2020, pp. 301–316.

[9] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proc. 35th Int. Conf. Machi. Learn.*, 2018, pp. 3521–3530. [Online]. Available: https://proceedings.mlr.press/v80/mhamdi18a.html

[10] X. Cao, J. Jia, and N. Z. Gong, "Provably secure federated learning against malicious clients," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6885–6893.

[11] Z. Qi, L. Meng, Z. Chen, H. Hu, H. Lin, and X. Meng, "Cross-silo prototypical calibration for federated learning with non-IID data," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 3099–3107.

[12] S. Zhao, T. Liao, L. Fu, C. Chen, J. Bian, and Z. Zheng, "Data-free knowledge distillation via generator-free data generation for non-IID federated learning," *Neural Netw.*, vol. 179, 2024, Art. no. 106627.

[13] A. Augello, A. Gupta, G. Lo Re, and S. K. Das, "Tackling selfish clients in federated learning," in *Proc. 27th Eur. Conf. Artif. Intell.*, 2024, pp. 1888–1895.

[14] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 27.
[15] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. Int. Conf. Netw. Distrib. Syst. Secur. Symp.*, 2021, pp. 18, doi: 10.14722/ndss.2021.24498.
[16] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proc. 29th USENIX Conf. Secur. Symp.*, 2020, pp. 1623–1640.
[17] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.
[18] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.
[19] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 1142–1154, 2022.
[20] A. Gupta, G. Markowsky, and S. K. Das, "Is performance fairness achievable in presence of attackers under federated learning?," in *Frontiers in Artificial Intelligence and Applications*. Amsterdam, Netherlands: IOS Press, 2023, pp. 948–955.
[21] Y. Shi, H. Yu, and C. Leung, "Towards fairness-aware federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11922–11938, Sep. 2024.
[22] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021, Art. no. 115.
[23] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proc. 3 rd Innovations Theor. Comput. Sci. Conf.*, 2012, pp. 214–226.
[24] M. Cong, H. Yu, X. Weng, and S. M. Yiu, "A game-theoretic framework for incentive mechanism design in federated learning," *Federated Learn.: Privacy Incentive*, vol. 12500, pp. 205–222, 2020.
[25] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4615–4625.
[26] G. Zhang, S. Malekmohammadi, X. Chen, and Y. Yu, "Proportional fairness in federated learning," *Trans. Mach. Learn. Res.*, 2023. [Online]. Available: https://openreview.net/forum?id=ryUHgEdWCQ
[27] V.-T. Tran, H.-H. Pham, and K.-S. Wong, "Personalized privacy-preserving framework for cross-silo federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 12, no. 4, pp. 1014–1024, Fourth Quarter 2024.
[28] A. Augello, G. Falzone, and G. L. Re, "DCFL: Dynamic clustered federated learning under differential privacy settings," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Affiliated Events*, 2023, pp. 614–619.
[29] L. Meng et al., "Improving global generalization and local personalization for federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 76–87, Jan. 2025.
[30] H. Wang, H. Xu, Y. Li, Y. Xu, R. Li, and T. Zhang, "FedCDA: Federated learning with cross-rounds divergence-aware aggregation," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 12.
[31] M. Hu et al., "FedMut: Generalized federated learning via stochastic mutation," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 12 528–12 537.
[32] B. Youssef, D. Jabir, and K. Abdellatif, "Impact of selfish nodes on federated learning performances," in *Proc. 9th Int. Conf. Wireless Netw. Mobile Commun.*, 2022, pp. 1–6.
[33] J. Zhou et al., "A differentially private federated learning model against poisoning attacks in edge computing," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 1941–1958, Jun./may 2023.
[34] A. Rahim et al., "Vehicular social networks: A survey," *Pervasive Mobile Comput.*, vol. 43, pp. 96–113, 2018.
[35] Y. Fraboni, R. Vidal, and M. Lorenzi, "Free-rider attacks on model aggregation in federated learning," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1846–1854.
[36] S. Arisdakessian, O. A. Wahab, A. Mourad, and H. Otrok, "Coalitional federated learning: Improving communication and training on non-IID data with selfish clients," *IEEE Trans. Serv. Comput.*, vol. 16, no. 4, pp. 2462–2476, Third Quarter, 2023.
[37] Y. E. Sagduyu, "Free-rider games for federated learning with selfish clients in NextG wireless networks," in *Proc. IEEE Conf. Commun. Netw. Secur.*, 2022, pp. 365–370.
[38] R. Luo, S. Hu, and L. Yu, "Rethinking client reweighting for selfish federated learning," 2021. [Online]. Available: https://openreview.net/forum?id=qfGcsAGhFbc
[39] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 634–643.
[40] P. J. Rousseeuw and C. Croux, "Alternatives to the median absolute deviation," *J. Amer. Statist. Assoc.*, vol. 88, no. 424, pp. 1273–1283, 1993.
[41] R. Garver, "Concerning the limits of a measure of skewness," *Ann. Math. Statist.*, vol. 3, no. 4, pp. 358–360, 1932.
[42] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta Mathematica*, vol. 30, no. 1, pp. 175–193, 1906.
[43] X. Chen, T. Chen, H. Sun, S. Z. Wu, and M. Hong, "Distributed training with heterogeneous data: Bridging median-and mean-based algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21616–21626.
[44] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009. [Online]. Available: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf
[45] S. Alam et al., "FedAIot: A federated learning benchmark for artificial intelligence of things," 2023, *arXiv:2310.00109*.

Andrea Augello received the bachelor's, master's, and PhD degrees in computer engineering from the University of Palermo, in 2019, 2021, and 2025, respectively. He is an assistant professor with the Department of Engineering, University of Palermo, Italy. His current research interests include cybersecurity, distributed systems, and artificial intelligence.

Ashish Gupta received the PhD degree in computer science and engineering from the Indian Institute of Technology (IIT) BHU, Varanasi, India. He is an assistant professor with the Department of Computer Science, BITS-Pilani Dubai campus, UAE. He has been a post doctoral fellow with CReWMaN Research Lab under the guidance of Prof. Sajal K. Das in Computer Science, Missouri University of Science and Technology, Rolla, USA, for 2 years. His current research interests include federated learning from a robustness and fairness perspective.

Giuseppe Lo Re (Senior Member, IEEE) received the laurea degree in computer science from the University of Pisa, in 1990, and the PhD degree in computer engineering from the University of Palermo, in 1999. He is a Full Professor of computer engineering with the University of Palermo, since 2019. In the 1991, he joined the Italian National Research Council (CNR), where he achieved the senior Researcher position. His current research interests include computer networks and distributed systems, focusing on reputation and security systems. He is a senior member of the IEEE Communication Society and the association for Computer Machinery.

Sajal K. Das (Fellow, IEEE) is a Curators' distinguished professor of computer science and Daniel St. clair endowed chair with the Missouri University of Science and Technology, USA. His research interests include cybersecurity, IoT, wireless sensor networks, mobile and pervasive computing, cyber-physical systems, and cloud computing. He is the editor-in-chief of Elsevier's Pervasive and Mobile Computing journal, and associate editor of *IEEE Transactions on Sustainable Computing*, *IEEE Transactions on Dependable and Secure Computing*, and *ACM Transactions on Sensor Networks*.

Open Access funding provided by 'Università degli Studi di Palermo' within the CRUI CARE Agreement