Model Stability Defense Against Model Poisoning in Federated Learning

Qi Guo, Di Wu, Yong Qi, Saiyu Qi, Qian Li, Minghao Yao, and Kaitai Liang, *Member, IEEE*

***Abstract*—Federated Learning (FL) exhibits susceptible to model poisoning attacks, which compromise the availability of the collaboratively trained model by introducing detrimental local updates during the training process. The predominant line of defense against such attacks has been to impose stringent restrictions on clients' model updates. However, this strategy raises new vulnerabilities where the global model can be infiltrated by meticulously crafted malicious perturbations. This vulnerability arises due to the model's inherent sensitivity to perturbations, making it exposed and fragile. In response, this work investigates a novel defensive paradigm centered on model stability—specifically, a model's resilience against perturbations within its parameter space. As a solution, we introduce a new method named Model Stability Defense for Federated Learning (MSDFL), designed to fortify the defense of FL systems against model poisoning attacks. MSDFL utilizes a minmax optimization framework, which is fundamentally linked to empirical risk for exploring the effects of model perturbations. The core aim of our approach is to minimize the norm of the model-output Jacobian matrix without compromising predictive performance, thereby establishing defense through enhanced model stability. Moreover, we propose a refined version of MSDFL, named Holistic Model Stability Defense for Federated Learning (HMSDFL), which considers model stability across all output dimensions of the logits to effectively eradicate the disparity in model convergence speed induced by MSDFL. Extensive experimental results fully demonstrate the fidelity, robustness, compatibility, and self-protection of our methods.**

***Index Terms*—Federated learning, model stability defense, model poisoning attack.**

I. Introduction

FEDERATED Learning (FL) represents a decentralized deep learning paradigm that enables multiple participants to jointly train a global model without centralizing their local data, thereby preserving individual data privacy [1]. Its appeal has surged due to its capacity to meet the escalating data demand, particularly in scenarios requiring stringent data privacy protection. However, the distributed characteristic of FL, combined with the server's inability to supervise and access the local training activities of the participants, renders FL susceptible to model poisoning attacks. In such attacks, malicious participants aim to undermine the global model by transmitting manipulated local updates that deliberately poison the collaborative learning process.

In light of the severe risks posed by model poisoning attacks, an extensive body of research has focused on developing mitigation strategies [2], [3], [4], [5], [6]. Since malicious model updates are typically distinctly different from benign ones, most existing methods leverage server-based robust aggregation to defend against such attacks. These methods typically involve filtering or revising potential malicious model updates (e.g., the model updates exhibit a significant geographic divergence from the center of all model updates), then aggregating the remaining model updates to form the global model. For instance, Trimean and Median [2] filter parameters in each dimension of the model parameters, thereby ensuring superior statistical performance within Byzantine robust FL systems. Krum [7] circumvents the issue of Byzantine nodes by excluding participants' model updates that significantly deviate from the center of all model updates. Bulyan [4], an evolution of Krum aggregation, integrates the strengths of both Krum and Trimean. As demonstrated by previous studies, these techniques can enhance the robustness of FL and mitigate the impact of attacks to a certain extent.

Nonetheless, recent studies have shown that even robust aggregation defenses are not immune to model poisoning attacks [8], [9]. In specific, these attacks could introduce minor perturbations into model updates, circumventing defense mechanisms and negatively influencing the global model. Unfortunately, current defenses can only defend against blatant, large-scale perturbations effectively, but are difficult to detect and neutralize subtle fine-tuned perturbations.

To elucidate this vulnerability, we analyze the dynamics between model poisoning attacks and robust aggregation defenses in the vicinity of model behavior boundaries within the parameter space (see Section III). By reframing our perspective and analyzing the entire defense process in terms of model perturbation, we can gain fresh insights to tackle model poisoning attacks more effectively.

In specific, we conceptualize model poisoning attacks as model perturbations towards the incorrect model behavior zones within the model parameter space. Traditional server-based

Received 30 December 2023; revised 26 July 2025; accepted 22 September 2025. Date of publication 7 October 2025; date of current version 14 January 2026. This work was supported in part by the Natural Science Basic Research Program of Shaanxi Program under Grant 2024JC-JCQN-67 and in part by Shaanxi Province QinChuangYuan "Scientist + Engineer" Team Building Project under Grant 2022KXJ-054. *(Qi Guo and Di Wu are contributed equally to this work.) (Corresponding author: Qi Guo.)*

Qi Guo is with the Air Traffic Control and Navigation School, Air Force Engineering University, Xi'an 710043, China (e-mail: qiguoqg@outlook.com).

Di Wu, Yong Qi, Saiyu Qi, Qian Li, and Minghao Yao are with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: ddiwu98@163.com; qiy@xjtu.edu.cn; saiyu-qi@xjtu.edu.cn; qianlix@xjtu.edu.cn; minghao_yao@stu.xjtu.edu.cn).

Kaitai Liang is with the Cybersecurity Group, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: Kaitai.Liang@tudelft.nl).

The source codes are maintained at https://github.com/qqoneone/MSDFL.

Digital Object Identifier 10.1109/TDSC.2025.3618769

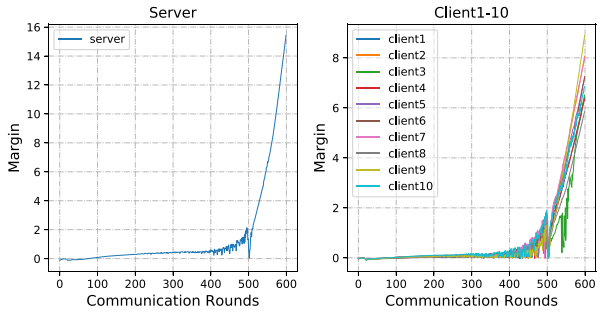

Fig. 1. Assessing the margin of the server's global and clients' local models using the FedAvg on the CIFAR10 dataset. The margin is the maximum distance between the model to the decision boundary.

robust aggregation defenses could then be conceptualized as restrictions on the global model. Given that the rationale of attack techniques and the percentage of attackers remain unknown to defenders, how to devise a perfect restriction is a considerable challenge. This uncertainty necessitates the implementation of broad and loose defensive restrictions, which unfortunately lack the capacity to filter out minor perturbations. Moreover, attackers can leverage the model's vulnerability to minor perturbations. By introducing subtle changes, they effectively steer the model towards erroneous behavior, which in turn triggers misclassification and impairs the global model's overall functionality. The above analysis indicates that relying solely on robust aggregations cannot provide sufficient defense against model poisoning attacks. Hence, we posit that addressing the model's stability in response to perturbations should be a central facet of any comprehensive defense strategy against attacks in FL, particularly those employing subtle perturbations. In this context, our goal shifts towards improving model stability against perturbations, thereby enhancing defense against model poisoning attacks.

However, bolstering the stability of the global model remains a challenging issue. Achieving a relatively stable global model through model updates' aggregation is notably challenging due to the uncertainty of the improvement direction, as well as the lack of server-side data to measure model stability against perturbations. Fortunately, we have identified a consistent relationship between the distances from the local model and the global model to the decision boundary, as presented in Fig. 1. This relationship suggests a potential solution: by improving the stability of each participant's model locally before aggregation, we can enhance the overall stability of the global model. Considering the fundamental role of the model-output Jacobian matrix in the Taylor expansion of model output in response to perturbations, we incorporate this Jacobian matrix into the analysis of the model stability to model perturbations. In this context, the magnitude of the Jacobian component serves as an indicator of the model stability: a larger Jacobian component implies a model that is more susceptible to instability in the face of perturbations.

Grounded in the preceding analysis, we introduce the Model Stability Defense for Federated Learning (MSDFL), a novel mechanism designed to counteract model poisoning attacks. The defense pivots on the principle of reducing the model-output Jacobian norm to enhance stability, all while maintaining the model's baseline accuracy. This is operationalized by augmenting the standard local loss function with a Jacobian regularization term, guiding the client-side gradient descent towards a more stable model. MSDFL would lead to a discrepancy in convergence speed between the correct and incorrect label categories during model training, affecting the convergence of the entire model slightly. To address this issue, we further propose a refined version of MSDFL named Holistic Model Stability Defense for Federated Learning (HMSDFL). HMSDFL takes into account the model stability across all logits output dimensions of the model, thereby eliminating the disparity in model convergence speed induced by MSDFL. As shown in Fig. 2, our methods are intrinsically designed for client-side deployment, thus facilitating the inherent self-protection capabilities of the client. Notably, our design of MSDFL (or HMSDFL) could serve as a plug-and-play fusion module, avoiding the need for total alterations to the existing standard FL framework. By enforcing enhanced stability in local models prior to their transmission (via MSDFL/HMSDFL), the resulting global model becomes inherently more stable. This improvement is achieved irrespective of the aggregation strategy employed by the server. Furthermore, we conduct a broad range of experiments to corroborate the superior performance of MSDFL (or HMSDFL).

Our main contributions in this work can be summarized as:

- We highlight the critical roles of both model stability and restrictions on model perturbations in defending against model poisoning attacks. Contrary to existing defenses that mainly focus on the latter while neglecting the former, our approach accentuates the essential role of model stability. To the best of our knowledge, our approach constitutes the inaugural effort to analyze FL defense from this novel perspective.
- We introduce a new method named Model Stability Defense for Federated Learning (MSDFL). MSDFL uses a minmax optimization framework closely tied to empirical risk, providing a systematic strategy to explore the impact of model perturbations.
- We propose a refined version of MSDFL named Holistic Model Stability Defense for Federated Learning (HMSDFL), which considers model stability across all output dimensions of the logits, effectively eradicating the disparity in model convergence speed induced by MSDFL.
- We validate the superior performance of MSDFL and HMSDFL regarding fidelity and robustness under different settings. Our methods can reduce the model's instability in the face of attacks by up to 99.93%. The experiments also show that our methods are compatible with existing server-based robust aggregation techniques, achieving an accuracy improvement of 0.15% to 9.83%, even under state-of-the-art model poisoning attacks.

II. Related Works

A. Robust Aggregation Defenses in Federated Learning

The robustness of Federated Learning (FL) against malicious attacks has garnered significant attention, prompting a surge

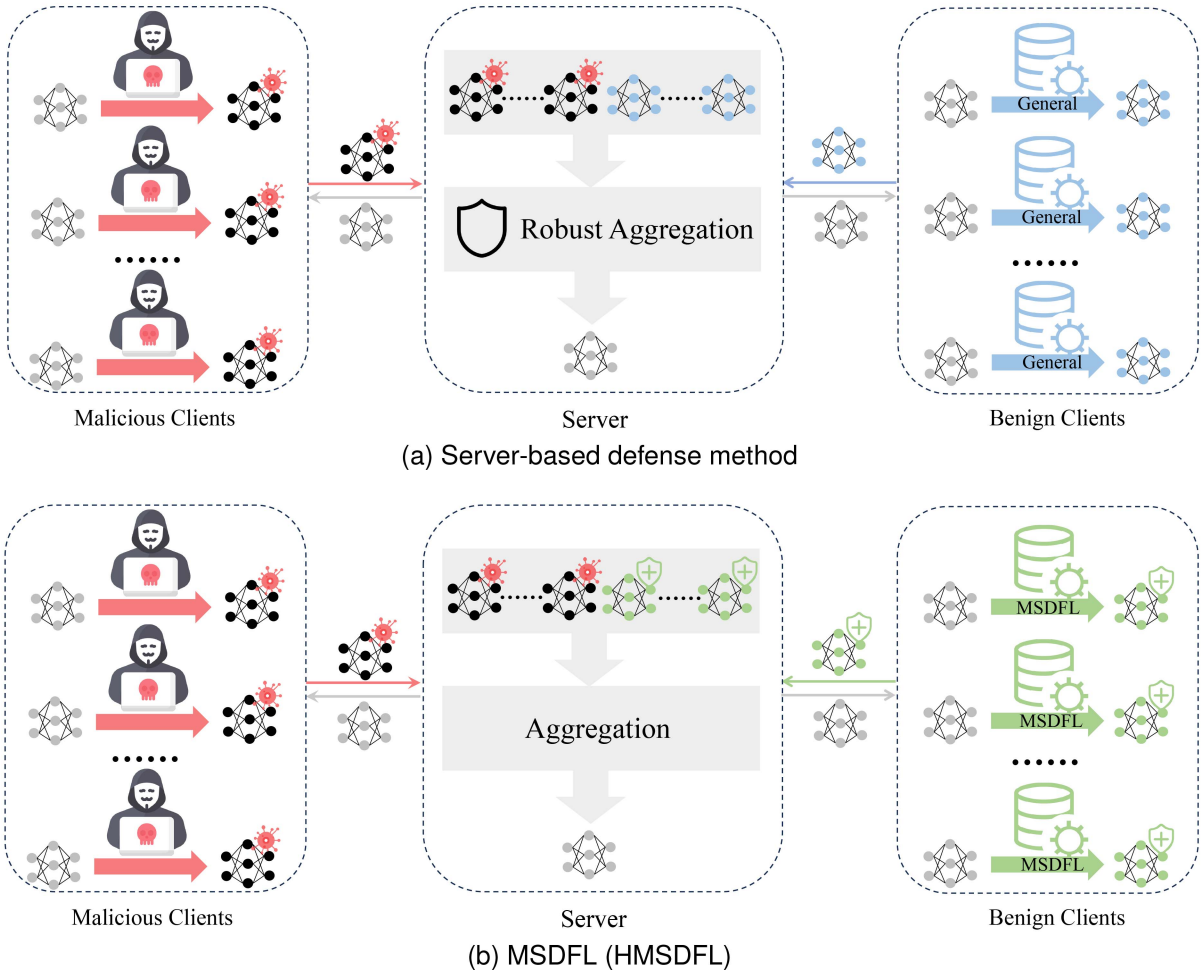

Fig. 2. A comparison of server-based defense method and MSDFL (HMSDFL).

in defensive work [2], [3], [4], [7]. The core strategy in these defenses is enhancing system robustness via robust aggregation methods on the server side.

Yin et al. [2] proposed two robust aggregation techniques, Trimean and Median, both of which independently apply coordinate-wise filtering operations to each dimension, thereby enhancing statistical analyses. Trimean, given a hyperparameter k, filters the top and bottom k values in each dimension before computing the remainder' mean. This method not only offers optimal statistical performance but also fortifies the resistance of FL against Byzantine attacks. Likewise, the Median technique computes each dimension's median value of the model parameters, constructing an aggregated result using these median values. This method stands out due to its efficiency, requiring only a few communication rounds. Additionally, it maintains robustness against Byzantine clients while also ensuring FL training convergence.

Krum [7] and Bulyan [4] offer an alternative strategy to enhance system robustness: eliminating anomalous models distantly situated from the majority of uploaded models. Krum functions by identifying the $(N - m - 2)$ models that are nearest to each given model, where N represents the total number of models and m stands for the number of malicious models. It then computes the sum of distances from these selected models to the current model, using this sum as a score for the current model. Following this process, each model receives a unique score. The model with the highest score is subsequently chosen as the output of the Krum aggregation process. Bulyan employs an iterative process that leverages the Krum method to establish a curated collection of potentially benign models. These models are then aggregated using the Trimean technique. In particular, the utilization of Trimean ensures that non individual malicious dimension goes unnoticed, enhancing the overall robustness of the process.

FLTrust [3] defense mechanism requires the server to maintain a root dataset. In each training round, the server first derives a reference model update by training the previous global model on this root dataset. Each client's submitted update is then compared to this reference via cosine similarity. The updates are subsequently scaled based on their similarity scores before a standard averaging aggregation is applied to form the new global model. While FLTrust has proven effective in both IID and NonIID settings, its performance is highly dependent on the quality and representativeness of the server's root dataset.

FLCert [10] provides a certified robustness framework for federated learning against poisoning attacks. FLCert leverages ensemble-based aggregation with theoretical guarantees. The method works by: (i) creating multiple sub-models through different aggregation rules; (ii) using majority voting among these sub-models to make predictions; and (iii) providing a certified radius within which the prediction remains unchanged regardless of poisoning attacks. FLCert proposes two variants based on client grouping: FLCert-P uses random sampling in each group, while FLCert-D deterministically partitions clients into disjoint groups. The certification is achieved through careful analysis of the voting mechanism and provides provable robustness guarantees.

FedREDefense [11] employs a reverse engineering approach to defend against model poisoning attacks. The key insight is that malicious model updates often exhibit distinct statistical patterns that can be detected by reconstructing the potential training data from gradients. Specifically, FedREDefense operates in three phases: (i) gradient collection, where the server collects model updates from all clients; (ii) data reconstruction, where the server attempts to reverse-engineer training samples from the received gradients using optimization techniques; and (iii) anomaly detection, where reconstructed data is analyzed to identify updates that deviate significantly from expected patterns. FedREDefense can defend against model poisoning attacks when the data distribution is highly NonIID and when there is a large number of malicious clients.

Beyond the aforementioned aggregation-based and detection-based defenses, researchers have also explored various alternative perspectives to enhance federated learning robustness [12], [13], [14], [15], [16], [17], [18], [19], incldung activation-based defenses [13], fairness-aware defenses [16], proactive defenses [17], truth discovery-based frameworks [18], etc.

B. Model Poisoning Attacks in Federated Learning

Model poisoning attack is a type of attack that compromises the overarching performance in FL, which is accomplished by delivering intricately crafted malicious models from participants [8], [20], [21]. This method is distinguishable from data poisoning attacks, which exploit mislabeled data to cultivate malicious models [22], [23], [24]. In contrast, model poisoning attacks concentrate on the direct formulation of the malicious model, aiming to exert a substantial threat on the broader FL.

By carefully defining the schema of local updates, model poisoning attacks can construct malicious model updates that differ substantially from benign model updates. This could result in the aggregated global model deviating considerably from its initial state when it was free from any attacks [8], [21]. Depending on different attack purposes, model poisoning attacks can be classified into two main categories: targeted and untargeted attacks. Targeted attacks seek to manipulate the global model into misclassifying data towards a predefined label [20], [25], [26], [27], [28]. In contrast, untargeted attacks pursue a more indiscriminate impact by seeking to degrade the global model's general performance [8], [9], [21]. In this study, we primarily investigate untargeted model poisoning attacks. These forms of attacks can be a greater risk to federated learning, as they have the potential to adversely affect the overall performance of FL across the complete dataset. Consequently, in this study, we concentrate on two extensively utilized untargeted attacks, namely the LIE attack [9] and the Fang attack [8].

Numerous defensive strategies aim to leverage statistical robustness to discern and eliminate instances where model updates exhibit significant deviations from the population mean. In the LIE attack [9], the authors highlight that when there is a high empirical variance among model updates, an attacker can exploit this vulnerability and launch attacks within the population variance. The LIE attack, an untargeted model poisoning method, crafts malicious updates by injecting minor noise into each dimension of the mean of benign updates. In addition, the knowledge of aggregation rules is not required to perform the LIE attack. Following the attack categorization criteria [21], LIE could also be classified as a partial knowledge attack when information from all benign client updates is utilized. Conversely, if only information from malicious client updates is employed, LIE could be categorized as an agnostic attack.

Fang attack [8] can be characterized as an untargeted model poisoning attack, with its primary goal being to inhibit the global model from reaching convergence. This is accomplished by crafting malicious local model updates on compromised participants. This strategy is conceptualized as an optimization problem where the objective is to maximize the deviation of the aggregate global model. Specifically, it is directed towards the inverse of the trajectory that the global model would naturally follow in the absence of any attacks. According to the classification rule [29], the Fang attack falls under the category of omniscient attacks as it necessitates updated information and aggregation rules from benign clients to successfully execute the attack.

Min-Max and Min-Sum can operate without knowledge of the server's aggregation algorithm [29]. Min-Max constrains malicious updates such that their maximum distance from any benign gradient does not exceed the maximum pairwise distance between benign gradients. Min-Sum employs a stricter constraint based on the sum of distances. Both attacks perturb benign aggregates in opposing directions, yet within benign statistical bounds, to evade detection.

III. Motivation

To gain insight into the susceptibility of robust aggregation defenses to model poisoning attacks, we investigate how attacks and defenses manifest in the parameter space around model behavior boundaries. As Fig. 3 depicts, model poisoning attacks can be perceived as intentional parametric perturbations that steer the global model astray, while robust aggregations are formalized as mechanisms that enforce restrictions to maintain the model within predetermined boundaries.

Given the unavailability of concrete information regarding the attack techniques and the attacker count, creating an all-encompassing defensive restriction for all attacks remains a significant challenge. Each robust aggregation, equipped with its distinct strategy, has the capacity to fend off certain attacks

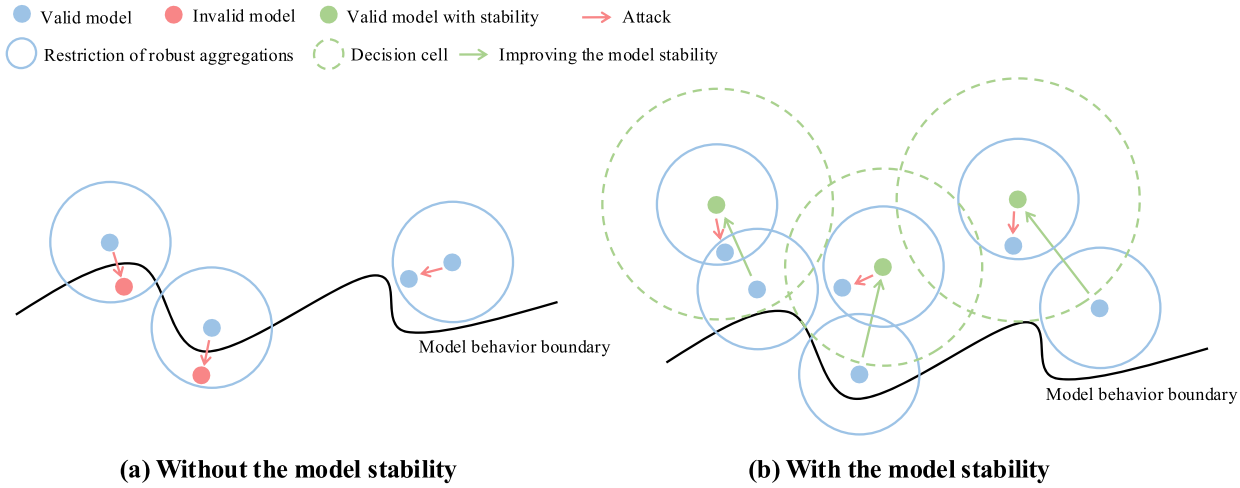

Fig. 3. A conceptual illustration of model behavior regions in parameter space. Each point represents a model with specific parameters, and boundaries indicate transitions between regions where models exhibit different classification performance (e.g., correct vs. incorrect classification behavior). Attacks aim to perturb models across these boundaries.

under particular situations. Despite the advancements in these methods, the robustness of the resulting models has not been adequately addressed, leaving them susceptible to subtle adversarial perturbations. These perturbations, despite being minuscule yet meticulously crafted, can bypass robust aggregations, compromising the global model [9]. Moreover, the restrictions introduced by robust aggregations could potentially interfere with the global model's overall convergence, causing a degradation in performance. This explains why the sole application of certain defensive techniques, even in the absence of attacks, might lead to degraded performance of the final model, as confirmed by the experimental results in Section VII-B.

The absence of the rationale of attack techniques and the proportion of attackers hampers the effectiveness of robust aggregations in mitigating model poisoning attacks. Accordingly, to effectively prevent such attacks, the optimal strategy would involve implementing defensive restrictions that successfully mitigate substantial malicious perturbations, while simultaneously guaranteeing that the model remains resilient to minor ones. As shown in Fig. 3, models located near the model behavior boundary, denoted by the blue points, are intrinsically unstable. Even with model restrictions, attackers can still induce malicious perturbations, consequently leading the model towards incorrect model behavior areas. Conversely, the green points representing models distant from the model behavior boundary exhibit considerable stability against perturbations. Upon application of defensive restrictions, it becomes unfeasible for attackers to misdirect these models into erroneous regions through malicious perturbations. These findings underscore the paramount importance of two complementary elements: ensuring model stability against parameter-space perturbations and implementing defensive restrictions to mitigate poisoning attacks. The prevalent one-sided emphasis in robust aggregation methods—prioritizing restrictions while overlooking the cultivation of model stability—constitutes a fundamental vulnerability that attackers can readily exploit. Therefore, we explore a novel perspective to mitigate model poisoning attacks, with a specific focus on enhancing model stability against perturbations within the parameter space.

Beyond robust aggregation methods, recent advances have introduced valuable alternative defense paradigms. Detection-based approaches such as FedREDefense [11] effectively identify malicious updates through innovative gradient analysis and data reconstruction techniques, demonstrating strong performance against various attacks. Certification-based methods like FLCert [10] make significant contributions by providing theoretical robustness guarantees through ensemble aggregation, offering provable security bounds that are crucial for high-stakes applications. These methods have substantially advanced the defense landscape and provide important protection mechanisms from different perspectives. However, the client-side enhancement of model stability against perturbations remains relatively underexplored. This observation motivates our complementary perspective: by enhancing the model's intrinsic stability at the client side, we can work synergistically with existing server-side defenses to create a more comprehensive defense framework. Our approach aims to make models inherently more resilient to perturbations, thereby providing an additional layer of protection that strengthens the overall robustness of federated learning systems when combined with existing methods.

IV. Problem Setup

In this section, we formally set up our problem from the threat model and defense goals.

A. Threat Model

We consider a threat model similar to those utilized in prior research [3], [4], [5]. The goal, capabilities, and knowledge of the attacker are presented as follows.

- *Attacker's goal:* The attacker's objective is to strategically manipulate malicious clients to undermine the performance of the global model indiscriminately.

- *Attacker's capabilities:* The attacker can compromise legitimate clients to make them malicious or directly inject malicious clients into the FL system. During each round of the FL training process, malicious clients can send arbitrary local model updates to the server. The attacker has the ability to tamper with data and models on these malicious clients while also accessing global model parameters distributed in each round.
- *Attacker's knowledge:* Attackers can be classified into three categories based on their knowledge about the server's aggregation rules and benign clients' model updates: agnostic attacks, partial knowledge attacks, and omniscient attacks. In agnostic attacks, the attackers require neither knowledge, making them the least powerful type of attacker. Partial knowledge attackers, who possess a moderate level of capability, are aware of either the server's aggregation rules or the model updates of benign clients. Lastly, omniscient attackers are the most potent ones, as they know both the server's aggregation rules and the model updates of benign clients, enabling them to execute a wide range of powerful assaults. In this study, we mainly consider two types of omniscient attacks: the LIE attack and the Fang attack.

B. Defense Goals

We aim to design a Byzantine-robust federated learning method that effectively mitigates the impact of malicious clients. We assume that the majority of clients are benign and willing to protect themselves. While this assumption limits our defense against scenarios with overwhelming malicious majorities, our approach can be used to complement server-side defenses (e.g., FedREDDefense) that can handle such extreme cases, creating a comprehensive defense system. The method should ensure fidelity, robustness, and compatibility while maintaining simplicity and client autonomy. Importantly, the method should neither disrupt the standard federated learning framework nor require additional validation sets on the server. In other words, we consider the lowest setting regarding the defender's knowledge and abilities. In particular, our method intends to achieve the following defense goals:

- *Fidelity:* The method should not substantially compromise the global model's accuracy in the absence of poisoning attacks, thereby preserving the model's utility under normal conditions.
- *Robustness:* The method should guarantee satisfactory performance of the global model even under strong model poisoning attacks, i.e., omniscient attacks.
- *Compatibility:* The method should be compatible with prevalent defense techniques, such as Trimean, Median, Bulyan, and FLTrust. This compatibility allows for seamless integration with server-based defenses, further enhancing the overall security guarantee.
- *Self-protection:* The method should be client-driven, allowing participants in the federated learning process to independently deploy the method locally, thereby achieving self-protection.

V. Model Stability Defense for Federated Learning

To enhance the robustness of the model, we probe the model's stability under perturbations, utilizing the lens of empirical risk. Specifically, we employ a minmax optimization framework tied to empirical risk, providing a structured approach to scrutinize the effects of perturbations on the model. By addressing this optimization problem, we endeavor to enhance the robustness and thus fortify its defense mechanisms. Furthermore, we provide a detailed description of the algorithm developed to implement this approach.

A. The Minmax Problem

In the context of FL, we consider a configuration of N client nodes, each with its own unique dataset denoted as $\mathcal{D}_i$. The network model parameters are denoted as $\boldsymbol{\theta} \in \mathbb{R}^M$, whereas the model perturbations are represented by $\boldsymbol{\varepsilon} \in \mathbb{R}^M$. The optimization goal of conventional FL can be formulated as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^M} \mathcal{F}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\theta}), \tag{1}$$

where $\mathcal{F}$ represents the overall optimization objective function of conventional FL, and F_i is the specific optimization objective function corresponding to the i-th client. Distinct from this conventional FL objective, our defense-oriented goal within the FL framework is to bolster model robustness by improving its stability in response to perturbations. We construct this goal as a minmax optimization problem as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^M} \max_{\boldsymbol{\varepsilon}\in\mathbb{R}^M} \mathcal{F}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) = \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\theta}+\boldsymbol{\varepsilon}). \tag{2}$$

This formulation represents finding model parameters $\boldsymbol{\theta}$ that remain stable even under worst-case perturbations $\boldsymbol{\varepsilon}$. It captures the defender's goal of achieving robustness against unknown attacks.

In supervised FL, the cross entropy function $H(\cdot)$ is employed to match the distribution learned by the model with the distribution of real data. Consequently, the minmax optimization problem can be expressed as follows

$$\begin{aligned}
&\min_{\boldsymbol{\theta}\in\mathbb{R}^M} \max_{\boldsymbol{\varepsilon}\in\mathbb{R}^M} \mathcal{F}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) \\
&= \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|\mathcal{D}_i|} \sum_{n=0}^{|\mathcal{D}_i|-1} H(y_n, \hat{f}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}, x_n)) \\
&= -\frac{1}{N}\sum_{i=1}^{N} \frac{1}{|\mathcal{D}_i|} \sum_{n=0}^{|\mathcal{D}_i|-1} \log \hat{f}_{c^*}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}, x_n),
\end{aligned} \tag{3}$$

where $\hat{f}$ denotes the output of the sample processed by the model, which is the probability output obtained after applying the softmax layer. Besides $\hat{f}_{c^*}$ represents the value of the dimension corresponding to the label of the sample in $\hat{f}$.

To further analyze the above problem, we introduce the Taylor expansion, allowing us to express the problem as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^M}\max_{\boldsymbol{\varepsilon}\in\mathbb{R}^M}\mathcal{F}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{D}_i|}\sum_{n=0}^{|\mathcal{D}_i|-1}\log(\hat{f}_{c^*}(\boldsymbol{\theta})+\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\cdot\boldsymbol{\varepsilon}+\mathrm{O}(\boldsymbol{\varepsilon}^2)), \tag{4}$$

where the Taylor expansion $\hat{f}_{c^*}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) = \hat{f}_{c^*}(\boldsymbol{\theta})+\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\cdot\boldsymbol{\varepsilon}+\mathrm{O}(\boldsymbol{\varepsilon}^2)$ can be explained as linearizing the model's response to perturbations. The Jacobian matrix $J_{c^*}(\boldsymbol{\theta}) = \nabla\hat{f}_{c^*}(\boldsymbol{\theta})$ naturally emerges from the Taylor expansion when analyzing the model's response to perturbations. It quantifies the rate of change of the model output with respect to parameter perturbations. Considering $\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\cdot\boldsymbol{\varepsilon}\geq-|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})||\boldsymbol{\varepsilon}|$ and $|\boldsymbol{\varepsilon}|\leq\sqrt{L}$ (L is a constant), we can easily derive the following inequality

$$\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\cdot\boldsymbol{\varepsilon}\geq-\left|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\right||\boldsymbol{\varepsilon}|\geq-\left|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\right|\sqrt{L}, \tag{5}$$

where the inequality $\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\cdot\boldsymbol{\varepsilon}\geq-|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})||\boldsymbol{\varepsilon}|$ is considered as bounding the worst-case impact of perturbations. The magnitude of the Jacobian directly influences the supremum of the minmax problem. A larger $|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})|$ indicates greater sensitivity to perturbations, making the model more vulnerable to attacks. By minimizing the Jacobian norm, we reduce this sensitivity, thereby enhancing model stability.

Utilizing inequality (5), we can derive the formulation from problem (4) as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^M}\max_{\boldsymbol{\varepsilon}\in\mathbb{R}^M}\mathcal{F}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) \leq-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{D}_i|}\sum_{n=0}^{|\mathcal{D}_i|-1}\log(\hat{f}_{c^*}(\boldsymbol{\theta})-\sqrt{L}\left|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})\right|+\mathrm{O}(\boldsymbol{\varepsilon}^2)). \tag{6}$$

The optimization strategy referenced above in the minmax problem is designed to boost the stability in response to model perturbations. We interpret these perturbations as attacks, which invariably degrade the model's performance. Our defense mechanism aims to improve the model's stability by broadening its perturbation tolerance, thus providing an effective countermeasure. By minimizing the supremum of the entire minmax problem, we can further enhance the model stability, ultimately bolstering our defensive posture.

B. The Model Stability Defense via the Minmax Problem

To fortify model stability against perturbations, it is necessary to diminish the supremum of the overarching optimization problem. In other words, we aim to decrease the value on the right-hand side of inequality (6). The elements influencing this value are $\hat{f}_{c^*}(\boldsymbol{\theta})$, $-\sqrt{L}|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})|$, and $\mathrm{O}(\boldsymbol{\varepsilon}^2)$. Elevating $\hat{f}_{c^*}(\boldsymbol{\theta})$ corresponds to the fundamental objective of each benign local client in FL, which can be achieved via the cross entropy function. Given the analytical nature of function $\hat{f}$, the higher-order terms can be typically negligible for relatively minor perturbations, $\boldsymbol{\epsilon}$. The negative sign preceding $-\sqrt{L}|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})|$ indicates that increasing $-\sqrt{L}|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})|$ is equivalent to reducing $|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})|$. To streamline the analysis of model stability, we introduce the model-output Jacobian matrix as follows

$$J_{c^*}(\boldsymbol{\theta})\equiv\nabla\hat{f}_{c^*}(\boldsymbol{\theta}), \tag{7}$$

where the larger Jacobian value $|J_{c^*}(\boldsymbol{\theta})|$ indicates higher sensitivity to perturbations.

Upon analyzing the minmax problem, our defense strategy is chiefly tailored to accomplish two principal objectives:

- Elevating $\hat{f}_{c^*}(\boldsymbol{\theta})$ aims to preserve the performance of the essential tasks carried out by each local client. Consequently, the loss of local tasks, denoted by $\mathcal{L}_{\text{local}}$, for each benign client should be minimized.
- Reducing $|\nabla\hat{f}_{c^*}(\boldsymbol{\theta})|$ is intended to enhance the robustness by fortifying the model's stability against perturbations. As such, the model-output Jacobian regularization, represented as $|J_{c^*}(\boldsymbol{\theta})|$, should be minimized.

Therefore, the model stability defense for FL can be formulated as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^M}\mathcal{G}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(\boldsymbol{\theta}), \tag{8}$$

where

$$\mathcal{L}_i(\boldsymbol{\theta}) = \mathcal{L}_{\text{local}}(\boldsymbol{\theta})+\lambda_{\text{MSD}}|J_{c^*}(\boldsymbol{\theta})|. \tag{9}$$

The Jacobian regularization term $|J_{c^*}(\boldsymbol{\theta})|$ in (9) serves as a stability regularizer that complements the local loss function, creating a dual objective that maintains model accuracy while improving robustness against perturbations.

In each iteration, an optimized loss function, denoted as $\mathcal{L}_i$, incorporates the model Jacobian regularizer. This function is optimized over a mini-batch $\mathcal{B}$, which is a collection of labeled instances $\{\boldsymbol{x}^\alpha,\boldsymbol{y}^\alpha\}_{\alpha\in\mathcal{B}}$. Specifically, the optimization is conducted via stochastic gradient descent (SGD), aiming to minimize the composite loss

$$\mathcal{L}_i^{\mathcal{B}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{local}}^{\mathcal{B}}(\boldsymbol{\theta})+\lambda_{\text{MSD}}\left[\frac{1}{|\mathcal{B}|}\sum_{\alpha\in\mathcal{B}}|J_{c^*}(\boldsymbol{\theta})|\right]. \tag{10}$$

Algorithm 1: The Model Stability Defense for Federated Learning (MSDFL).

Input: the global model $\boldsymbol{\theta}^t$, the local dataset $\mathcal{D}_i$, mini-batch size $|\mathcal{B}|$.
Output: the updated local model $\boldsymbol{\theta}_i^{t+1}$

$\boldsymbol{\theta}_{i,0} = \boldsymbol{\theta}^t$
$R = |\mathcal{D}_i|/|\mathcal{B}|$
for $r = 1$ *to* R **do**
  $\mathcal{B}\leftarrow$ sample a mini-batch with size $|\mathcal{B}|$
  model outputs $\boldsymbol{p}^\alpha = \hat{f}(\boldsymbol{\theta}_{i,r-1},\{\boldsymbol{x}^\alpha,\boldsymbol{y}^\alpha\}_{\alpha\in\mathcal{B}})$
  $J_{c^*}^\alpha(\boldsymbol{\theta}) = \partial\boldsymbol{p}^\alpha/\partial\boldsymbol{\theta}^\alpha$
  $\mathcal{L}_i^{\mathcal{B}} = \mathcal{L}_{\text{local}}^{\mathcal{B}}+\lambda_{\text{MSD}}\frac{1}{\mathcal{B}}\sum_{\alpha}|J_{c^*}(\boldsymbol{\theta})|$
  $\boldsymbol{\theta}_{i,r} = \boldsymbol{\theta}_{i,r-1}-\nabla_{\boldsymbol{\theta}}\mathcal{L}_i^{\mathcal{B}}(\boldsymbol{\theta}_{i,r-1})$
end
$\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}_{i,R}$

For MSDFL, the additional computational overhead arises from computing the model-output Jacobian $J_{c^*}(\boldsymbol{\theta}) = \nabla \hat{f}_{c^*}(\boldsymbol{\theta})$ for the correct label. This computation requires one additional backward pass per mini-batch to calculate gradients with respect to model parameters. It also incurs a memory overhead of $\mathcal{O}(|\boldsymbol{\theta}|)$ for storing the Jacobian values. A comprehensive presentation of the model stability defense for federated learning is illustrated in Algorithm 1.

VI. Holistic Model Stability Defense for Federated Learning

In the previous section, we fortified the model stability against perturbations using the minmax problem to enhance the model's robustness. However, although this method bolstered the model stability, it had the inadvertent side effect of slightly hindering the model's convergence. This is because MSDFL solely reduced the model-output Jacobian matrix norm for the correct label category. As a result, when the model is trained on the correct label category of data, convergence occurs at a slower pace. In contrast, the rate of change remains unaffected when training on incorrect label categories. This leads to a discrepancy in convergence speed between the correct and incorrect label categories during model training, affecting the convergence of the entire model slightly.

To address this issue, we propose a refined version of MSDFL named holistic model stability defense for federated learning (HMSDFL). HMSDFL takes into account the model stability across all logits output dimensions of the model, thereby eliminating the disparity in model convergence speed induced by the previous method. Consequently, it not only improves the generalization performance of the model but also provides enhanced defense capabilities.

This section formalizes the model-logits Jacobian based on the pre-softmax logits output. To enhance robustness, we improve the model's tolerance to parameter perturbations by minimizing the Frobenius norm of this Jacobian. Additionally, an efficient algorithm is introduced for practical computation of the corresponding regularizer.

A. Model-Logits Jacobian and Stability Analysis

Given the set of classification functions, denoted as $\boldsymbol{f}$, our main objective lies in learning this classification function via a neural network equipped with model parameters $\boldsymbol{\theta} \in \mathbb{R}^M$. Here, the input vector is represented as $\boldsymbol{x} \in \mathbb{R}^I$, while the output is a score vector, $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}^C$. Each element, z_c, signifies the likelihood of the input pertaining to the category, c. The vector $\boldsymbol{z}$ corresponds to the logits prior to the application of a softmax layer. The softmax function calculates the probabilistic output p_c in relation to z_c, using $p_c \equiv \frac{e^{z_c/T}}{\sum_{c'} e^{z_{c'}/T}}$ at temperature T, typically set at unity.

As the model parameters $\boldsymbol{\theta}$ are utilized for each input vector $\boldsymbol{x}$, for the sake of simplicity, we exclude the explicit dependency on $\boldsymbol{x}$, hence $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{\theta})$. We then consider a small perturbation, $\boldsymbol{\varepsilon} \in \mathbb{R}^M$, which shares the same dimensionality as the model parameter.

In the course of stability analysis of the logits output in the face of model perturbations, the model-logits Jacobian matrix naturally comes into play. For a perturbed model with $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + \boldsymbol{\varepsilon}$, the corresponding values shift as follows:

$$\begin{aligned}\widetilde{z}_c = f_c(\boldsymbol{\theta} + \boldsymbol{\varepsilon}) &= f_c(\boldsymbol{\theta}) + \sum_{i=1}^{M} \varepsilon_i \cdot \frac{\partial f_c}{\partial \theta_i}(\boldsymbol{\theta}) + O\left(\boldsymbol{\varepsilon}^2\right) \\ &= z_c + \sum_{i=1}^{M} J_{c;i}(\boldsymbol{\theta}) \cdot \epsilon_i + O\left(\boldsymbol{\varepsilon}^2\right), \end{aligned} \tag{11}$$

where the function in the second equality is a Taylor expansion with respect to the model perturbation, $\boldsymbol{\varepsilon}$, while the model-logits Jacobian matrix,

$$J_{c;i}(\boldsymbol{\theta}) \equiv \frac{\partial f_c}{\partial \theta_i}(\boldsymbol{\theta}) \tag{12}$$

is introduced in the third equality. Equation (11) reveals how individual parameter perturbations ϵ_i propagate through the model to affect the logit output, where $J_{c;i}(\boldsymbol{\theta})$ quantifies the sensitivity of class c's logit to the i-th parameter. By minimizing the Frobenius norm $\|J(\boldsymbol{\theta})\|_{\mathrm{F}}^2$, we effectively reduce the model's sensitivity to parameter perturbations across all output dimensions simultaneously, thereby enhancing the model's robustness against manipulations in the parameter space.

Unlike the conventional input-output Jacobian for the sample [30], we focus on the model-logits Jacobian for model parameters. For sufficiently small $\boldsymbol{\varepsilon}$, higher-order terms $O(\boldsymbol{\varepsilon}^2)$ can be neglected, and the stability of the model is dominantly governed by the Jacobian matrix. Therefore, controlling the magnitude of the Jacobian directly increases the model's tolerance to parameter perturbations and enhances its overall stability.

B. The Model Stability Defense via Minimizing Model-Logits Jacobian

As indicated by (11), larger Jacobian components amplify the variation in logits outputs under parameter perturbations, thereby increasing model instability. To suppress this effect, we adopt a direct strategy of controlling the magnitude of the Jacobian components. This is achieved by minimizing the squared Frobenius norm of the model-logits Jacobian,

$$\|J(\boldsymbol{\theta})\|_{\mathrm{F}}^2 \equiv \left\{ \sum_{i,c} \left[J_{c;i}(\boldsymbol{\theta})\right]^2 \right\}. \tag{13}$$

This Frobenius norm (i) captures the overall sensitivity of all logits to parameter perturbations, not just the correct label; (ii) eliminates the convergence speed disparity between correct and incorrect label categories observed in MSDFL; (iii) provides a differentiable objective that can be efficiently optimized. This holistic approach provides stronger guarantees for model stability compared to the single-class focus of MSDFL.

Consequently, the local training process for benign clients is reframed to pursue dual objectives: maintaining the primary learning objective by minimizing the standard local loss, while

simultaneously incorporating the model-logits Jacobian regularization as an auxiliary term to bolster robustness through improved model stability.

The Jacobian regularizer as delineated in (13) can seamlessly integrate with the client's loss function. Given N clients with respective datasets $\mathcal{D}_1, \ldots, \mathcal{D}_N$, we then construct the holistic model stability defense for federated learning (HMSDFL) as follows:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathcal{G}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(\boldsymbol{\theta}), \tag{14}$$

where

$$\mathcal{L}_i(\boldsymbol{\theta}) = \mathcal{L}_{\text{local}}(\boldsymbol{\theta}) + \lambda_{\text{HMSD}} \left\| J\left(\boldsymbol{\theta}\right) \right\|_{\text{F}}^2 \tag{15}$$

signifies the ith client's loss function, $\mathcal{L}_{\text{local}}$ refers to an inherent local loss function, and λ_{HMSD} serves as a hyperparameter that dictates the extent to which the model Jacobian regularizer influences the model. With an appropriate choice of λ_{HMSD}, we anticipate the model to learn in a manner that is both correct and robust by minimizing the combined loss function.

During each training iteration, the enhanced objective function $\mathcal{L}_i$ is optimized using Stochastic Gradient Descent (SGD) over the model parameters. For a mini-batch $\mathcal{B}$ containing labeled examples $\{\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha\}_{\alpha \in \mathcal{B}}$, this optimization corresponds to the minimization of the following composite objective:

$$\mathcal{L}_i^{\mathcal{B}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{local}}^{\mathcal{B}}\left(\boldsymbol{\theta}\right) + \lambda_{\text{HMSD}} \left[\frac{1}{|\mathcal{B}|} \sum_{\alpha \in \mathcal{B}} \left\| J\left(\boldsymbol{\theta}\right) \right\|_{\text{F}}^2 \right]. \tag{16}$$

It is necessary to clarify the fundamental distinction between our model stability defense and traditional regularization-based methods in federated learning proposed to address the NonIID issues. While both approaches involve regularization, they address entirely different problems. Traditional regularization methods (e.g., FedProx with L_2 penalties) primarily target NonIID data distribution issues or aim to maintain model similarity across clients for better convergence. In contrast, our Jacobian-based regularization $\|J(\boldsymbol{\theta})\|$ specifically measures and reduces the model's sensitivity to adversarial perturbations in parameter space. This is derived from our minmax optimization framework that analyzes how small parameter perturbations can bypass robust aggregations. The key insight is that models near boundaries are inherently unstable, making them vulnerable to carefully crafted attacks even when robust aggregations are employed. By minimizing the model-output Jacobian, we enhance the model's stability against such perturbations, addressing a security vulnerability that traditional regularization methods were not designed to handle.

C. Algorithm

As established previously, enhancing model stability serves as an effective strategy to improve the robustness of FL. This is realized in client-side training by minimizing the Frobenius norm of the model-logits Jacobian. A primary challenge, therefore, lies in the efficient computation and integration of this Jacobian regularizer into diverse, pre-existing FL frameworks.

Algorithm 2: Holistic Model Stability Defense for Federated Learning (HMSDFL).

Input: the global model $\boldsymbol{\theta}^t$, the local dataset $\mathcal{D}_i$, mini-batch size $|\mathcal{B}|$, and number of projections m_{proj} .
Output: the updated local model $\boldsymbol{\theta}_i^{t+1}$

$\boldsymbol{\theta}_{i,0} = \boldsymbol{\theta}^t$
$R = |\mathcal{D}_i| / |\mathcal{B}|$
for $r = 1$ *to* R **do**
  $\mathcal{J}_F = 0$
  $\mathcal{B} \leftarrow$ sample a mini-batch with size $|\mathcal{B}|$
  model outputs $\boldsymbol{z}^\alpha = f(\boldsymbol{\theta}_{i,r-1}, \{\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha\}_{\alpha \in \mathcal{B}})$
  for $k = 1$ *to* m_{proj} **do**
    $\{q_c^\alpha\} \sim \mathcal{N}(0, \mathbb{I})$; // $(|\mathcal{B}|, C)$-dim
    $\hat{\boldsymbol{q}}^\alpha = \boldsymbol{q}^\alpha / \|\boldsymbol{q}^\alpha\|$; // Uniform sampling from the unit sphere
    $\boldsymbol{z}_{\text{flat}} = \text{Flatten}(\{\boldsymbol{z}^\alpha\})$, $\boldsymbol{q}_{\text{flat}} = \text{Flatten}(\{\hat{\boldsymbol{q}}^\alpha\})$
    $Jq = \partial(\boldsymbol{z}_{\text{flat}} \cdot \boldsymbol{q}_{\text{flat}}) / \partial \boldsymbol{\theta}^\alpha$
    $\mathcal{J}_F += C\|Jq\|^2 / (m_{\text{proj}} |\mathcal{B}|)$
  end
  $\mathcal{L}_i^{\mathcal{B}} = \mathcal{L}_{\text{local}}^{\mathcal{B}} + \lambda_{\text{HMSD}} \mathcal{J}_F$
  $\boldsymbol{\theta}_{i,r} = \boldsymbol{\theta}_{i,r-1} - \nabla_{\boldsymbol{\theta}} \mathcal{L}_i^{\mathcal{B}}(\boldsymbol{\theta}_{i,r-1})$
end
$\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}_{i,R}$

The exact computation of the Frobenius norm for the model-logits Jacobian involves leveraging a complete orthonormal basis in conjunction with automatic differentiation. For each basis vector $\{\boldsymbol{e}\}$, the derivative elements within the model-logits Jacobian can be calculated efficiently by differentiating the product $\boldsymbol{e} \cdot \boldsymbol{z}$ with respect to the model parameters, $\boldsymbol{\theta}$,

$$\|J(\boldsymbol{\theta})\|_{\text{F}}^2 = \text{Tr}\left(JJ^{\text{T}}\right) = \sum_{\{\boldsymbol{e}\}} \boldsymbol{e} JJ^{\text{T}} \boldsymbol{e}^{\text{T}} = \sum_{\{\boldsymbol{e}\}} \left[\frac{\partial(\boldsymbol{e} \cdot \boldsymbol{z})}{\partial \boldsymbol{\theta}} \right]^2, \tag{17}$$

where a constant orthonormal basis, $\{\boldsymbol{e}\}$, is embedded within the C-dimensional output space. Equation (17) encapsulates an exact computation that requires C times of backpropagating gradients through the model, corresponding to C orthonormal basis vectors $\{\boldsymbol{e}\}$. Nonetheless, the computational cost, which is linear with respect to the output category C, becomes impracticable and overwhelmingly costly for numerous large-scale FL applications.

To effectively implement the model Jacobian regularizer, we employ a random projection method to calculate the Frobenius norm of the model-logits Jacobian [31]. Consequently, (17) is reformulated in terms of the expectation of an unbiased estimator,

$$\|J(\boldsymbol{\theta})\|_{\text{F}}^2 = C\mathbb{E}_{\hat{\boldsymbol{q}} \sim S^{C-1}} \left[\|\hat{\boldsymbol{q}} \cdot J\|^2 \right], \tag{18}$$

where the random vector $\hat{\boldsymbol{q}}$ originates from the $(C-1)$-dimensional unit sphere S^{C-1}, with each of its elements sampled from a standard normal distribution. Equation (18) efficiently estimates the full Jacobian using random sampling, which could

reduce computational complexity to make it practical for applications. As per this formulation, the square of the Frobenius norm of the model-logits Jacobian can be estimated using samples of m_{proj} random vectors $\hat{\boldsymbol{q}}^m$ as

$$\|J(\boldsymbol{\theta})\|_{\text{F}}^2 \approx \frac{1}{m_{\text{proj}}} \sum_{m=1}^{m_{\text{proj}}} \left[\frac{\partial \left(\hat{\boldsymbol{q}}^m \cdot \boldsymbol{z} \right)}{\partial \boldsymbol{\theta}} \right]^2, \tag{19}$$

which converges to the true value as $O(m_{\text{proj}}^{-1/2})$ [30]. Furthermore, we derive $\|J(x)\|_{\text{F}}^2$ by taking an average over a mini-batch of samples of size $|\mathcal{B}|$. We foresee the fluctuation of our estimator to be further suppressed by $\sim 1/\sqrt{|\mathcal{B}|}$ due to compensations within the mini-batch. Given that the error associated with nearly independent and identically distributed samples in a mini-batch is projected to be of order $(m_{\text{proj}}|\mathcal{B}|)^{-1/2}$ [30], our method can perform commendably in practice even with $m_{\text{proj}} = 1$. For HMSDFL, the additional computational overhead stems from computing the model-output Jacobian $J_{c;i}(\boldsymbol{\theta}) \equiv \frac{\partial f_c}{\partial \theta_i}(\boldsymbol{\theta})$. This process necessitates m_{proj} additional backward passes per mini-batch to compute gradients with respect to model parameters, resulting in a memory overhead of $\mathcal{O}(|\boldsymbol{\theta}| \times m_{\text{proj}})$. While the computational overhead scales linearly with m_{proj}, empirical evidence suggests that setting $m_{\text{proj}} = 1$ is often sufficient for practical applications. A detailed description of the HMSDFL is provided in Algorithm 2.

For resource-constrained clients in FL settings, we further investigate appropriate training method selection. To accommodate the heterogeneous computational capabilities of participating clients, we can define three resource thresholds—high, low, and minimum—that categorize clients based on their available resources in practical scenarios. For clients with abundant resources (exceeding the high threshold), the strategy employs HMSDFL with full projection dimension ($m_{\text{proj}} = C$) to maximize model expressiveness. Clients with moderate resources (between the low and high thresholds) utilize HMSDFL with adaptive projection dimensions ($m_{\text{proj}} \in (1, C)$) to achieve an optimal trade-off between computational overhead and learning performance. When resources fall between the minimum and low thresholds, clients can flexibly choose between HMSDFL with minimal projection ($m_{\text{proj}} = 1$) or MSDFL depending on their specific scenarios. For severely resource-limited clients (below the minimum threshold), the strategy defaults to standard FL to ensure participation without overwhelming their computational capacity. This hierarchical approach enables efficient integration of resource-constrained clients into the FL ecosystem while maintaining overall system performance.

VII. Experiment

A. Experiment Setup

1) Datasets and Network Model Architectures: Our experimental study employed four distinct datasets: CIFAR10 [32], MNIST [33], SVHN [34], and Fashion-MNIST [35]. The CIFAR10 and SVHN datasets, composed of 50,000 training and 10,000 test samples each, were utilized for 10-class color image and color-digit image classification tasks, respectively. On the other hand, MNIST and Fashion-MNIST, both consisting of 60,000 training and 10,000 test samples, were designated for 10-class digit image and fashion image classification tasks, respectively.

In terms of network model architecture, we deployed the AlexNet [36] for the CIFAR10 and SVHN datasets, and the LeNet [33] for the MNIST and Fashion-MNIST datasets.

2) Experimental Scenarios: We sought to build a FL system in both Independent and Identically Distributed (IID) and NonIID scenarios for each classification task. We set the total number of participating clients to 50 for both scenarios, while considering a threat model where up to 20% of these clients could act maliciously.

In the simulation of IID environments, each of the 50 clients was assigned a local dataset containing 1,000 samples for CIFAR10 and SVHN, and 1,200 for MNIST and Fashion-MNIST, all encompassing the complete range of class labels.

To simulate Non-IID scenarios, we sorted the dataset by labels and evenly distributed it among clients. In particular, each client was assigned an individual dataset that contained 1,000 samples for CIFAR10 and SVHN, and 1,200 samples for MNIST and Fashion-MNIST, each with just 2 out of the 10 possible labels.

3) Optimization and Training: Throughout all classification tasks, we used the Stochastic Gradient Descent (SGD) optimizer, with a learning rate fixed to 0.1. However, the quantity of communication rounds was tailored according to the specific dataset and model in use.

For the CIFAR10 and SVHN tasks, which employed the AlexNet model, we assigned the number of communication rounds to 600. In scenarios where the FLTrust technique was applied, this number was increased to 1,000 to ensure model convergence.

For tasks involving the MNIST and Fashion-MNIST datasets that utilized the LeNet model, the number of communication rounds was set to 500. Setting the number of communication rounds adjustably caters to the divergent complexities and requirements presented by the various datasets and network models.

4) Robust Aggregation Methods: We implement seven widely recognized aggregation methods: FedAvg [1], Trimean [2], Median [2], Bulyan [4], FLTrust [3], FLCert [10], and FedREDefense [11]. FedAvg directly averages the updates from all clients, consequently formulating a new global model for the next iteration. Trimean and Median employ coordinate-wise aggregation techniques, performing separately on each dimension for robust aggregation. Bulyan integrates Krum and Trimean to provide better robustness to Byzantine clients. FLTrust is a defense mechanism introduced recently under the assumption that the server has access to a meticulously chosen root dataset for assisting aggregation. FLCert proposes an ensemble FL that provides provable security guarantees against poisoning attacks from malicious clients. FedREDefense identifies and filters out malicious clients based on the discrepancies in their model update reconstruction errors.

5) Model Poisoning Attacks: We implement two eminent model poisoning attacks: LIE attack [9] and Fang attack [8]. LIE attack constructs the malicious update based on the mean and standard deviation of the benign updates, as well as the

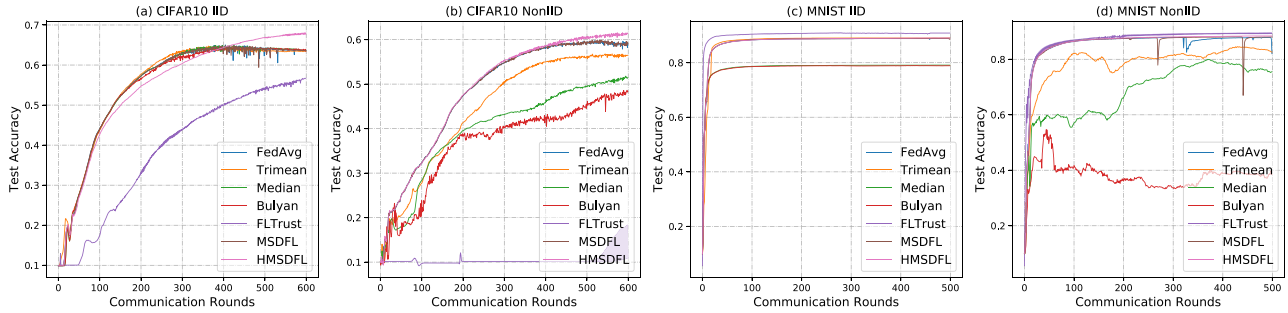

Fig. 4. The fidelity of different robust aggregation methods on the CIFAR10 and MNIST datasets.

TABLE I
THE HIGHEST ACCURACY VALUES RECORDED BY DIFFERENT ROBUST AGGREGATION METHODS ON THE CIFAR10 AND MNIST DATASETS WITH THE IID AND NONIID SETTINGS

Method		FedAvg	Trimean	Median	Bulyan	FLTrust	MSDFL	HMSDFL
CIFAR10	IID	65.03%	64.75%	64.95%	64.26%	56.70%	64.81%	68.06%
	NonIID	59.74%	56.81%	51.79%	48.61%	18.33%	59.92%	61.54%
MNIST	IID	88.94%	89.05%	79.12%	78.94%	90.97%	88.90%	89.06%
	NonIID	87.89%	84.59%	80.00%	54.68%	89.58%	88.14%	88.45%

TABLE II
THE IMPACT OF INCORPORATING HMSDFL ON THE GENERALIZATION PERFORMANCE OF FEDAVG, FEDPROX, TRIMEAN, MEDIAN, AND BULYAN UNDER THE CIFAR10 AND MNIST DATASETS. "Δ" REPRESENTS THE CHANGE VALUES OF THE VARIANTS AFTER INTRODUCING HMSDFL COMPARED WITH THE ORIGINAL METHODS.

Method			FedAvg	Fedprox	Trimean	Median	Bulyan
CIFAR10	IID	Standard	65.03%	65.01%	64.75%	64.95%	64.26%
		HMSDFL	68.17%	68.04%	67.87%	67.66%	64.92%
		Δ	3.14%↑	3.03%↑	3.12%↑	2.71%↑	0.66%↑
	NonIID	Standard	59.74%	59.79%	56.81%	51.79%	48.61%
		HMSDFL	62.07%	61.92%	60.33%	52.18%	48.35%
		Δ	2.33%↑	2.13%↑	3.52%↑	0.39%↑	-0.26%↓
MNIST	IID	Standard	88.94%	98.80%	89.05%	79.12%	78.94%
		HMSDFL	88.89%	98.90%	79.22%	79.18%	79.06%
		Δ	-0.05%↓	0.10%↑	0.17%↑	0.06%↑	0.12%↑
	NonIID	Standard	87.89%	88.12%	84.59%	80.00%	54.68%
		HMSDFL	88.29%	88.32%	91.59%	82.74%	65.07%
		Δ	0.40%↑	0.20%↑	7%↑	2.74%↑	10.39%↑

ratio of benign to malicious clients. Fang attack constructs malicious updates by solving an optimization problem where the attack's essence is to drive the global model to deviate maximally towards the inverse of the benign direction through the upload of meticulously crafted malicious updates.

B. Fidelity

Previous defense methods tend to overlook the impact on accuracy in scenarios without attacks. As such, we first focus on demonstrating the fidelity of our proposed methods in a standard, attack-free scenario. We employ MSDFL and HMSDFL with FedAvg, comparing them with the standard FedAvg to scrutinize their fidelity. We also compare our methods with the current robust aggregation methods, including Trimean, Median, Bulyan, and FLTrust.

Our fidelity evaluation results are demonstrated in Fig. 4. To facilitate the direct comparison of peak accuracy reached by diverse methods, we provide the highest accuracy values recorded by all methods under both IID and NonIID settings across various datasets, as delineated in Table I. From the table and figure, we can see that both MSDFL and HMSDFL display competitive performance in comparison to FedAvg under both IID and NonIID conditions across different datasets. Specifically, HMSDFL exhibits superior accuracy in the CIFAR10 and MNIST datasets under both settings, while MSDFL presents higher accuracy in NonIID scenarios and slightly less accuracy in IID scenarios across these datasets. These results emphasize the minimal impact on the accuracy by our methods. Notably, FLTrust shows the best generalization performance on the MNIST dataset and the poorest performance on the CIFAR10 dataset. This variance is due to its strong dependence on the quality of the assumed server's root dataset. The closer the alignment between the root dataset and the global data distribution of all clients, and the simpler the global data distribution, the better FLTrust performs, and vice versa. In comparison with other robust aggregation methods, our methods show overwhelming superiority. Existing robust aggregation methods tend to deliver sub-optimal performance under IID situations, and some of which even produce unstable convergence results under NonIID circumstances. In the CIFAR10 dataset, MSDFL maintains its performance as seen with the MNIST dataset, while HMSDFL further enhances FedAvg's generalization. In summary, these experimental results suggest that our methods' fidelity surpasses baselines.

The enhanced generalization capacity of HMSDFL was further assessed through comparative analysis with FedAvg, FedProx, Trimean, Median, and Bulyan. These five methods span three categories: (i) the standard FL paradigm (FedAvg), (ii) the client-side regular training method (FedProx), and (iii) server-side robust aggregation methods (Trimean, Median, and Bulyan). As illustrated in Table II, HMSDFL improves the generalization performance of the global model in the vast majority of cases. The reason for this improvement is that the innate heterogeneity of FL can be perceived as a perturbation in the model parameter space. Due to the distributed nature of FL, clients are unlikely to possess identical datasets even under the IID scenarios, which leads to fundamental heterogeneity. NonIID conditions further intensify this heterogeneity. The impact of this heterogeneity can appear as perturbations to the parameter space. As HMSDFL fosters model stability against perturbations, it implicitly helps FL systems to tackle heterogeneity, thereby improving the global model's generalization.

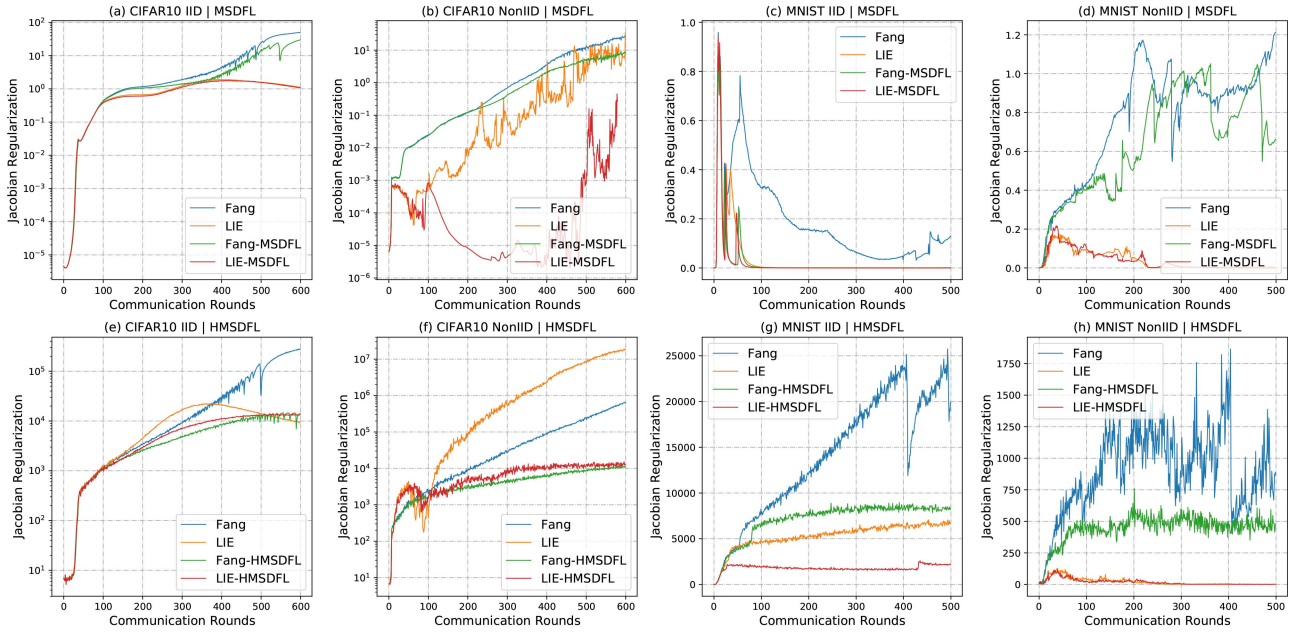

Fig. 5. The Jacobian regularization of different attacks under Trimean incorporated with MSDFL or HMSDFL on the CIFAR10 and MNIST datasets.

C. Robustness

The robustness of MSDFL and HMSDFL against model poisoning attacks is examined in this section. Specifically, we merge MSDFL and HMSDFL with the Trimean defense method to evaluate their defense potential against LIE and Fang attacks. Fig. 6 exhibits the accuracy performance of our methods under two distinct scenarios, IID and NonIID, tested on the CIFAR10 and MNIST datasets. These experiments are conducted to provide a practical illustration of our methods' utility in defensing model poisoning attacks. Fig. 5 depicts the variations in the Jacobian regularization values that correlate with MSDFL and HMSDFL under both the IID and NonIID settings on the CIFAR10 and MNIST datasets. This illustration serves to showcase the contribution of our methods to both model stability and resistance against poisoning attacks.

As shown in Fig. 6, HMSDFL consistently improves defense performance against both LIE and Fang attacks across diverse datasets. For instance, HMSDFL significantly bolsters accuracy under both IID and NonIID conditions on the CIFAR10 dataset, substantiating its versatility. Despite a breached defense and continuous performance degradation in the NonIID setting on the MNIST dataset due to LIE attacks, HMSDFL still manages to bolster model performance and diminish the attack's impact. The IID cases on the MNIST dataset present a similar performance across all methods, due to an already established sufficient model functionality. In this scenario, model poisoning attacks minimally affect the global model performance, making it challenging to evaluate our method's robustness based on experimental results. With the exception of its performance under the NonIID setting on the MNIST dataset, MSDFL marginally improves accuracy performance under LIE and Fang attacks. Disregarding the influence of dataset variations on MSDFL, the performance

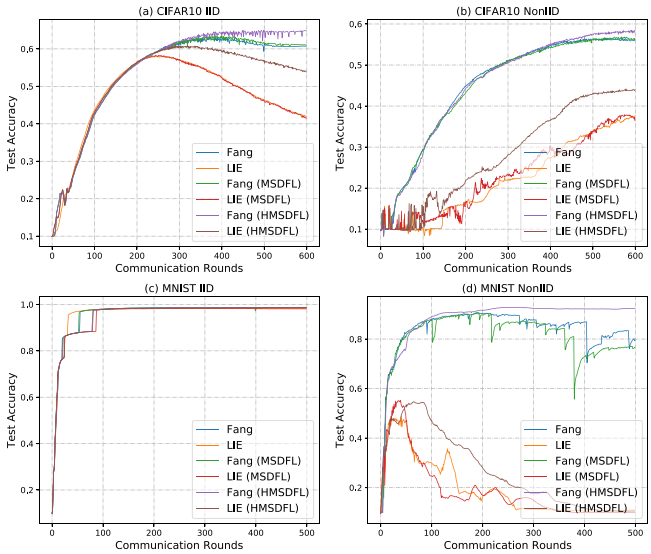

Fig. 6. The accuracy of different attacks under Trimean incorporated with MSDFL or HMSDFL on the CIFAR10 and MNIST datasets.

of MSDFL is somewhat inferior to HMSDFL. This is primarily due to the differing convergence speeds of correct and incorrect label categories during model training when using MSDFL, which might slightly impede model convergence during defense.

To further demonstrate on our methods' robustness in defending against model poisoning attacks, we evaluate the Jacobian regularization related to each method, highlighting its intrinsic influence on the model. Fig. 5 illustrates the experimental results of Jacobian regularization for MSDFL and HMSDFL under LIE and Fang attacks across different datasets with both

TABLE III
COMPARISON OF THE CLIENT WITHOUT AND WITH MSDFL (HMSDFL) OVER VARIOUS AGGREGATION METHODS UNDER LIE ATTACK AND FANG ATTACK ON CIFAR10, MNIST, SVHN, AND FASHION-MNIST DATASETS. (THE BEST RESULT IN EACH SET OF EXPERIMENTS IS MARKED IN BOLD; CASES WHERE BOTH MSDFL AND HMSDFL COMPLETELY FAIL TO IMPROVE DEFENSIVE PERFORMANCE ARE MARKED IN RED.)

Dataset	Aggregation	IID						NonIID					
		LIE Attack			Fang Attack			LIE Attack			Fang Attack		
		Standard	MSDFL	HMSDFL	Standard	MSDFL	HMSDFL	Standard	MSDFL	HMSDFL	Standard	MSDFL	HMSDFL
CIFAR10	FedAvg	60.81%	60.79%	**67.22%**	63.51%	63.24%	**66.84%**	55.56%	55.84%	**57.14%**	59.36%	57.14%	**59.98%**
	Trimean	56.12%	59.36%	**65.95%**	63.00%	63.58%	**66.16%**	37.61%	37.88%	**44.19%**	56.44%	56.84%	**58.47%**
	Median	56.79%	57.76%	**64.70%**	60.73%	61.56%	**61.93%**	11.07%	14.52%	**14.87%**	49.93%	50.00%	**50.04%**
	Bulyan	46.95%	47.03%	**54.19%**	50.46%	**51.44%**	51.12%	11.60%	12.64%	**13.66%**	20.16%	**21.24%**	21.12%
	FLTrust	59.40%	56.56%	**61.49%**	60.54%	56.53%	**60.97%**	16.38%	**16.49%**	16.38%	11.46%	**11.50%**	11.47%
	FLCert	56.03%	58.40%	**62.15%**	63.27%	63.70%	**66.53%**	35.61%	38.22%	**42.10%**	52.44%	53.21%	**53.96%**
	FedREDefense	58.90%	59.25%	**64.42%**	64.45%	66.90%	**67.25%**	52.11%	52.83%	**53.56%**	60.48%	61.33%	**61.70%**
MNIST	FedAvg	88.71%	88.74%	**88.89%**	98.65%	98.60%	**98.79%**	87.78%	**97.18%**	88.16%	97.00%	97.21%	**97.40%**
	Trimean	98.28%	98.31%	**98.62%**	98.73%	98.71%	**98.78%**	50.09%	54.41%	**58.43%**	90.09%	90.46%	**93.81%**
	Median	78.79%	78.94%	**79.12%**	78.90%	78.92%	**78.99%**	39.65%	40.51%	**44.50%**	84.82%	**86.94%**	86.66%
	Bulyan	87.44%	87.45%	**87.76%**	78.19%	78.23%	**78.56%**	30.39%	**37.47%**	34.86%	10.05%	10.05%	10.05%
	FLTrust	90.99%	91.00%	**91.65%**	91.00%	91.10%	**91.71%**	87.32%	**89.29%**	88.55%	87.15%	**89.13%**	88.58%
	FLCert	98.35%	98.61%	**98.74%**	97.67%	97.75%	**97.89%**	52.43%	56.78%	**60.92%**	91.84%	92.15%	**92.26%**
	FedREDefense	98.65%	98.83%	**98.87%**	97.90%	97.74%	**98.25%**	87.25%	90.35%	**88.48%**	92.12%	92.58%	**93.03%**
SVHN	FedAvg	87.24%	87.21%	**88.02%**	87.87%	87.38%	**88.21%**	82.21%	82.76%	**84.43%**	83.50%	**83.91%**	83.51%
	Trimean	86.09%	86.53%	**87.74%**	87.91%	87.96%	**88.00%**	11.21%	**11.38%**	11.35%	83.76%	**84.81%**	**84.81%**
	Median	81.64%	81.76%	**86.62%**	84.86%	**85.01%**	84.89%	10.42%	10.44%	**11.10%**	68.49%	**71.14%**	68.83%
	Bulyan	70.43%	70.53%	**81.17%**	68.37%	68.90%	**73.63%**	10.76%	**23.62%**	11.10%	17.46%	23.62%	**28.85%**
	FLTrust	85.03%	84.21%	**85.84%**	85.27%	83.57%	**85.70%**	10.08%	10.08%	10.08%	10.08%	10.08%	10.08%
	FLCert	86.34%	86.59%	**87.75%**	87.91%	**88.04%**	88.01%	11.32%	11.40%	11.47%	82.36%	**83.75%**	83.67%
	FedREDefense	88.29%	88.70%	**89.16%**	88.42%	88.90%	**89.26%**	10.53%	10.38%	10.72%	83.72%	**85.06%**	84.96%
Fashion-MNIST	FedAvg	88.46%	88.52%	**89.03%**	88.54%	88.60%	**89.26%**	79.39%	79.56%	**80.01%**	82.81%	**83.50%**	82.86%
	Trimean	75.51%	75.53%	**75.78%**	66.71%	66.78%	**67.10%**	30.52%	33.82%	**37.75%**	59.06%	**62.76%**	60.82%
	Median	86.91%	86.95%	**87.51%**	87.60%	87.80%	**88.78%**	34.16%	**41.52%**	35.94%	55.37%	59.83%	**63.95%**
	Bulyan	82.72%	83.71%	**85.98%**	81.91%	82.20%	**82.63%**	25.51%	**32.12%**	28.66%	10.29%	**16.67%**	15.39%
	FLTrust	83.71%	83.72%	**84.95%**	83.68%	83.72%	**84.93%**	79.60%	80.33%	**81.01%**	79.84%	80.47%	**81.02%**
	FLCert	78.35%	78.60%	**79.81%**	71.41%	71.86%	**72.15%**	32.26%	34.07%	**40.64%**	69.45%	**71.80%**	70.90%
	FedREDefense	85.10%	85.37%	**85.52%**	89.63%	89.72%	**89.75%**	75.42%	78.06%	**79.51%**	83.03%	83.17%	**83.42%**

IID and NonIID settings. These experimental results clearly indicate that both MSDFL and HMSDFL markedly diminish the model's Jacobian regularization during the training process. They can reduce the model's instability in the face of attacks by up to 99.93%. Without MSDFL and HMSDFL, the Jacobian regularization would persistently expand, maintaining this trend even once the model accuracy has stabilized. The large Jacobian regularization signifies global model instability and potentially creates an exploitable vulnerability for malicious entities. Conversely, the integration of MSDFL or HMSDFL curtails the Jacobian regularization within a set range, securing model stability and reinforcing the model's robustness against model poisoning attacks.

Consequently, our empirical investigations corroborate the robustness of both MSDFL and HMSDFL, as evidenced by their exceptional accuracy performance and attenuated Jacobian regularization. Essentially, by limiting Jacobian regularization, MSDFL and HMSDFL safeguard the global model's stability, effectively neutralizing the harmful repercussions of model poisoning attacks.

D. Compatibility

To evaluate the compatibility of MSDFL and HMSDFL, we integrate them with the most widely recognized server-side robust aggregation methods, and evaluate the enhancements they brought to robust aggregation methods. Our experiments evaluate the defensive performance of robust aggregation methods against LIE and Fang attacks, conducted on CIFAR10, MNIST, SVHN, and Fashion-MNIST datasets, under both IID and NonIID settings.

As shown in Table III, the incorporation of MSDFL and HMSDFL consistently improves the defensive performance of robust aggregation methods, which permeates across distinct datasets, various robust aggregation defenses, and different attacks. For instance, HMSDFL outperforms FedAvg, Trimean, Median, Bulyan, and FLTrust under the IID setting on the CIFAR10 dataset, with improvements of 6.41%, 9.83%, 7.91%, 7.24%, and 2.09% respectively against the LIE attack, and 3.33%, 3.16%, 1.20%, 0.66%, and 0.43% respectively against the Fang attack. MSDFL and HMSDFL are compatible with existing server-based robust aggregation methods, achieving an accuracy improvement of 0.15% to 9.83% on four datasets, even under LIE and Fang attacks. It's noteworthy that, HMSDFL often exhibits a superior accuracy improvement over MSDFL, as it accounts for model stability across all logits output dimensions, thereby circumventing the divergence in convergence speed induced by MSDFL.

The results highlighted in red in Table III indicate scenarios where both MSDFL and HMSDFL failed to further improve the defensive performance of robust aggregation methods. This failure can be attributed to the complete compromise of models in these instances, resulting in a loss of convergence capability. Consequently, any attempts to improve model stability are rendered ineffective; in these circumstances, convergence remediation measures are more critical and effective than introducing regularization.

Contrasting with Trimean, Median, Bulyan, and FLTrust, FedAvg proves to be more effective in certain scenarios. Specifically, on the CIFAR10 dataset, FedAvg demonstrates superior defense performance against both LIE and Fang attacks in IID and NonIID settings. This aligns with the empirical evaluation [9], as attacks utilized in the experiments take into consideration the existence of server-side robust aggregation. These attacks craft malicious updates specifically to evade robust aggregation and contaminate the global model. Consequently,

they overlook the inherent impact of basic aggregation brought by FedAvg, thereby enabling FedAvg to display strong defense capabilities against these attacks in certain circumstances. However, it is important to highlight FedAvg's vulnerability to attacks, due to its lack of mechanisms to distinguish or filter out malicious updates.

The defensive performance of FLTrust fluctuates significantly under different settings on various datasets. For instance, for Fang attacks under the NonIID setting, FLTrust exhibits the weakest defense performance on the CIFAR10 dataset but outperformed Trimean, Median, and Bulyan on the Fashion-MNIST dataset. This inconsistency arises from FLTrust's requirement for a server-side root validation dataset. FLTrust's defense strategy relies on the similarity between root global and local model updates in each round, inevitably making its defense performance contingent on the quality of the root dataset distribution. Notably, on CIFAR10 with the NonIID setting, FLTrust accuracy under LIE (18.33%) differs significantly from the ∼80% reported in the original work. This discrepancy stems from the experimental setup: (i) we used a 100-sample root dataset that may inadequately represent the global distribution in our extreme NonIID setting; (ii) we evaluated against strong omniscient attacks (LIE and Fang) rather than the simpler attacks used in the original work. Despite this, FLTrust demonstrates excellent performance on simpler datasets, achieving 87.32% on MNIST and 79.60% on Fashion-MNIST under NonIID conditions—substantially outperforming other robust aggregation methods. These results highlight that while FLTrust provides strong defense when the root dataset aligns well with the global distribution, our client-side stability approach offers complementary protection that does not rely on such alignment, making it particularly valuable in scenarios with complex or unknown data distributions.

Our experimental results also reveal notable differences in defense effectiveness between LIE and Fang attacks. HMSDFL demonstrates stronger improvement against LIE attacks (e.g., 6.41% improvement for FedAvg on CIFAR10 with the IID setting) compared to Fang attacks (3.33% improvement). This disparity stems from the fundamental differences in attack mechanisms: LIE attacks inject noise based on statistical variance of benign updates, creating irregular perturbations that our Jacobian regularization effectively dampens. In contrast, Fang attacks optimize for maximal deviation in specific directions, making them more structured and harder to mitigate. This effectiveness gap widens under NonIID settings, where LIE attacks can better camouflage within naturally high variance, while Fang attacks maintain their distinctive optimization patterns. These findings suggest that variance-based attacks are more susceptible to our stability-oriented defense mechanism than optimization-based attacks.

Both FLCert and FedREDefense benefit significantly from our stability enhancement methods. While FedREDefense performs well in IID settings (98.65% on MNIST), both methods struggle in NonIID scenarios. When combined with HMSDFL, we observe consistent improvements: up to 5.52% for FedREDefense and 6.12% for FLCert on CIFAR10 (IID), with even larger gains in NonIID settings (8.49% improvement for FLCert on MNIST). These results confirm that client-side stability enhancement effectively strengthens both detection-based and certification-based defenses, particularly in challenging NonIID environments.

In summary, our empirical evaluation substantiates the effectiveness and compatibility of MSDFL and HMSDFL, which demonstrates the feasibility of defending model poisoning attacks from the client side. Importantly, these client-side defensive methods do not conflict with server-side defense strategies and can be easily combined with them to work together.

E. Discussion on Privacy Considerations

While our MSDFL/HMSDFL methods compute Jacobian matrices locally without transmitting them to the server, we acknowledge the importance of co-designing privacy-preserving mechanisms with our stability defense. The Jacobian regularization could be naturally integrated with differential privacy by applying noise injection after stability-regularized training: $\boldsymbol{\theta}_{\text{private}} = \boldsymbol{\theta} + \mathcal{N}(0, \sigma^2\mathbf{I})$, where σ is calibrated according to the privacy budget (ϵ, δ). Notably, our stability regularization may offer synergistic benefits with differential privacy—by minimizing the model-output Jacobian and ensuring $\|\nabla f(\boldsymbol{\theta})\| \leq L$, we inherently reduce the model's sensitivity to perturbations, which could translate to tighter sensitivity bounds and improved privacy-utility tradeoffs. A comprehensive privacy-preserving variant could incorporate differential privacy directly into the optimization: $\mathcal{L}_{\text{private}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{local}}(\boldsymbol{\theta}) + \lambda\|J_{c^*}(\boldsymbol{\theta})\| + \mathcal{P}_{\text{DP}}(\boldsymbol{\theta})$ implements gradient clipping and noise addition following the DP-SGD framework. The co-design of privacy-preserving mechanisms with stability defense would provide dual protection against both model poisoning attacks and privacy breaches, offering a more comprehensive security framework for federated learning. This integration addresses two critical FL concerns simultaneously, where the reduced sensitivity from stability regularization enables stronger privacy guarantees with minimal utility loss, enhancing FL's applicability in privacy-sensitive domains.

F. Impact of the Hyperparameter

Finally, we explore the impact of the key hyperparameter λ (i.e., λ_{MSD} and λ_{HMSD}) on our methods via multiple experiments. We conduct these experiments on two datasets: CIFAR10 and MNIST under the NonIID setting. For the aggregation method, we used Trimean, while the LIE technique was employed for the attack.

The evaluation result is illustrated in Fig. 7. With a high value of λ, the contribution of the model Jacobian regularization significantly increases, thereby dominating the training objective. While this higher contribution can bolster model stability defense, it also curbs model convergence excessively, resulting in a decline in the model's performance. On the other hand, when λ is minimal or verges on 0, the Jacobian regularization's influence diminishes. Consequently, MSDFL and HMSDFL are unable to provide an effective defense against attacks in the FL system. The selection of λ is pivotal in maintaining a balance between the model's robustness and overall performance. Hence, in a specific scenario, λ must be carefully fine-tuned to achieve an optimal trade-off between accuracy and robustness.

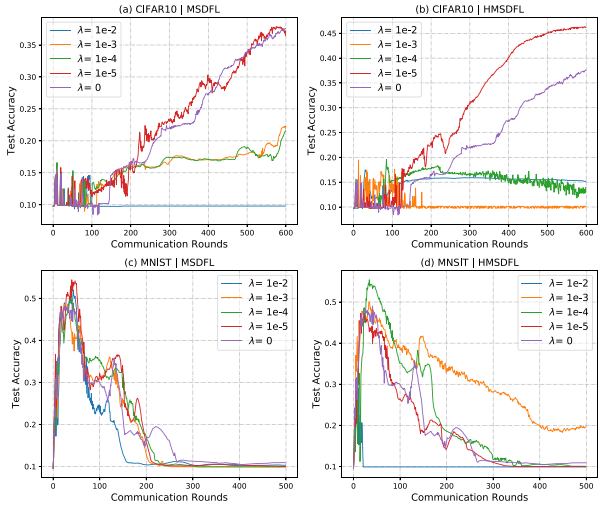

Fig. 7. The impact of the hyperparameter λ (i.e., λ_{MSD} and λ_{HMSD}) on MSDFL and HMSDFL.

An adaptive or automatic method for selecting the hyperparameter λ would strengthen the practical applicability. There are several mature adaptive or automatic parameter selection methods in the literature that could be adapted to hyperparameter λ: Grid search [37], Random search [37], Bayesian optimization [38], Gradient-based optimization [39], Evolutionary algorithms [40]. Taking grid search as an illustrative example, this method systematically evaluates all candidate values within a predefined parameter space, making it well-suited for automatic λ selection. To enhance the practical applicability of hyperparameter optimization in FL, we suggest considering an efficient two-stage coarse-to-fine strategy to balance computational efficiency with optimization effectiveness. For instance, when applying HMSDFL to the NonIID MNIST dataset, we first conduct a coarse search using a sparse grid (e.g., $\lambda \in \{10^{-6}, 10^{-2}, 10^{2}\}$). Based on the coarse search results, we generate a denser grid around the optimal region (e.g., if $\lambda^* = 10^{-2}$, we search $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$). Finally, we could automatically get the optimal the hyperparameter value (e.g., $\lambda^* = 10^{-3}$).

It's noteworthy that in Fig. 7(c) and (d), the accuracy begins to decline after attaining a certain peak. This observation is due to the fact that, post a successful attack, a persistent assault can escalate the attack's impact to such an extent that the model loses its functionality, indicated by a plunge in the accuracy rate to approximately 10%. Intriguingly, Fig. 7(d) illustrates that an appropriately selected λ (i.e., $1e-3$) can alleviate the effects of poisoning attacks, even under severe conditions. This ability to dampen the impact of poisoning attacks ultimately bolsters the model's utility, as demonstrated by the recuperation of the accuracy rate to approximately 20%.

G. Applicability to More Attacks and Defenses

To demonstrate the broader applicability of our methods to various attacks and defenses, we further evaluate our method against additional attack strategies and recent defense mechanisms on MNIST under NonIID setting. We consider three additional attacks beyond LIE and Fang: (1) Min-Max [29]: maximizes model deviation through optimization; (2) Min-Sum [29]: minimizes distance to benign updates while maintaining attack impact; (3) Jacobian-Aware Adaptive (JAA) attack: an adaptive attack we design where adversaries craft updates that execute LIE attack while reducing Jacobian norm to evade our defense. The JAA objective is: $\min_{\theta_m} \mathcal{L}_{\text{LIE}}(\theta_m) + \alpha\|\mathbf{J}(\theta_m)\|_F^2 + \beta\|\theta_m - \theta_{\text{target}}\|^2$, where $\theta_{\text{target}} = \mu + z\sigma$ follows LIE, and α, β balance stealth and effectiveness. We also evaluate compatibility with recent defenses: FLCert [10] and FedREDefense [11].

TABLE IV
ACCURACY (%) COMPARISON OF VARIOUS DEFENSE METHODS AGAINST MULTIPLE ATTACKS WITH AND WITHOUT MODEL STABILITY DEFENSE ON MNIST DATASET UNDER NONIID SETTING

Attack	Stability	FedAvg	Trimean	Median	Bulyan	FLTrust	FLCert	FedREDefense
LIE	standard	87.78	50.09	39.65	30.39	87.32	52.43	87.25
	MSDFL	97.18	54.41	40.51	37.47	89.29	56.78	90.35
	HMSDFL	88.16	58.43	44.50	34.86	88.55	60.92	88.48
Fang	standard	97.00	90.09	84.82	10.05	87.15	91.84	92.12
	MSDFL	97.21	90.46	86.94	10.05	89.13	92.15	92.58
	HMSDFL	97.40	93.81	86.66	10.05	88.58	92.26	93.03
Min-Max	standard	77.85	42.36	36.82	18.74	78.93	44.68	80.12
	MSDFL	82.54	46.92	38.65	24.36	81.27	49.17	84.76
	HMSDFL	79.68	51.28	41.87	21.52	80.45	53.42	85.92
Min-Sum	standard	83.12	46.87	40.32	22.15	82.65	49.21	85.38
	MSDFL	87.46	51.23	42.78	28.93	84.92	53.58	88.62
	HMSDFL	84.93	55.67	45.94	26.47	84.18	57.85	86.27
JAA	standard	84.92	47.35	36.48	27.86	85.64	49.67	86.53
	MSDFL	91.87	51.78	37.92	35.14	88.23	54.12	93.15
	HMSDFL	87.48	56.15	42.36	32.58	87.89	58.43	90.12

Table IV shows that our methods maintain effectiveness against the adaptive attack. Despite JAA being specifically designed to evade our defense, MSDFL with FedREDefense achieves 93.15% accuracy, demonstrating that stability enhancement cannot be easily circumvented. Adaptive attackers face inherent trade-offs between evading detection (statistical similarity), bypassing our defense (low Jacobian norm), and maintaining attack effectiveness (model degradation). These conflicting objectives significantly restrict the feasible attack space. Across all attacks, our methods consistently improve accuracy (e.g., HMSDFL improves FedREDefense by 5.80% under Min-Max). The better performance with FLCert and FedREDefense confirms compatibility with recent defenses, validating that model stability provides a fundamental defense layer complementary to existing approaches.

H. Applicability to More Datasets

To evaluate the applicability of our methods on more diverse datasets, we tested various defenses on CIFAR100 [32] and AG News [41] under NonIID settings with LIE attacks. CIFAR100 is an image classification dataset with 100 categories, while AG News is a widely-used text classification benchmark dataset in natural language processing. We adopt the default FL settings and use a Dirichlet distribution with coefficient 0.5 to generate NonIID datasets. We use Wide-ResNet-16-2 [42] for CIFAR100 and ALBERT-base [43] for AG News. As shown in Table V, both MSDFL and HMSDFL consistently improve defense performance across different data modalities. On CIFAR100, HMSDFL demonstrates substantial improvements across all methods, with gains ranging from 0.09% (FLTrust) to 5.22% (Trimean). The most significant improvements occur for coordinate-wise aggregation methods: Trimean (33.54% to 38.76%), Median

TABLE V
ACCURACY (%) OF DIFFERENT DEFENSES ON CIFAR100 AND AG NEWS DATASETS

Method	CIFAR100			AG News		
	standard	MSDFL	HMSDFL	standard	MSDFL	HMSDFL
FedAvg	37.27	37.51	39.84	87.75	87.92	88.43
Trimean	33.54	34.12	38.76	75.41	76.83	79.25
Median	34.65	36.47	37.23	73.92	74.68	76.14
Bulyan	27.11	28.35	29.67	68.31	70.42	71.86
FLTrust	37.32	37.48	37.41	91.07	91.35	91.82
FlCert	33.26	33.89	36.54	87.63	87.95	87.71
FedREDefense	38.64	39.15	41.23	89.79	90.24	90.86

(34.65% to 37.23%), and distance-based methods like FlCert (33.26% to 36.54%). Even advanced defenses like FedREDefense benefit from our approach, improving from 38.64% to 41.23%. MSDFL also provides consistent but more modest gains. For AG News, our methods successfully enhance text classification robustness across all aggregation techniques. HMSDFL achieves improvements ranging from 0.08% to 3.84%, with Trimean (75.41% to 79.25%) and Bulyan (68.31% to 71.86%) showing the most substantial gains. Notably, methods that initially performed poorly under attacks benefit most from our stability enhancement, suggesting that model stability is crucial for maintaining robustness in heterogeneous federated settings. The consistent improvements across both vision and language tasks, despite their fundamental differences in data representation and model architectures, validate the universal applicability of our model stability principle.

VIII. CONCLUSION

In this work, we introduce model stability as a novel defense against poisoning attacks in FL. Unlike existing robust aggregation methods, we propose MSDFL to enhance robustness by improving model stability against parameter perturbations. Its refined version, HMSDFL, ensures uniform stability across all logits dimensions, addressing convergence disparities in MSDFL, and incorporates an efficient algorithm to reduce Jacobian regularization overhead. Extensive experiments on four datasets validate the defense's fidelity, robustness, compatibility, and self-protection.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[2] 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.

[3] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," in *Proc. 28th Annu. Netw. Distrib. Syst. Secur. Symp.*, The Internet Society, 2021, pp. 1–18. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/fltrust-byzantine-robust-federated-learning-via-trust-bootstrapping/

[4] R. Guerraoui et al., "The hidden vulnerability of distributed learning in Byzantium," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.

[5] A. Panda, S. Mahloujifar, A. N. Bhagoji, S. Chakraborty, and P. Mittal, "SparseFed: Mitigating model poisoning attacks in federated learning with sparsification," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 7587–7624.

[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] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 118–128.

[8] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1605–1622.

[9] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8635–8645.

[10] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong, "FLCert: Provably secure federated learning against poisoning attacks," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3691–3705, 2022, doi: 10.1109/TIFS.2022.3212174.

[11] Y. Xie, M. Fang, and N. Z. Gong, "FedREDefense: Defending against model poisoning attacks for federated learning using model update reconstruction error," in *Proc. 41st Int. Conf. Mach. Learn.*, Vienna, Austria, 2024, pp. 54460–54474. [Online]. Available: https://openreview.net/forum?id=Wjq2bS7fTK

[12] Y. Liu, C. Chen, L. Lyu, Y. Jin, and G. Chen, "Exploit gradient skewness to circumvent Byzantine defenses for federated learning," in *Proc. Sponsored Assoc. Advance. Artif. Intell.*, Philadelphia, PA, USA, 2025, pp. 19024–19032, doi: 10.1609/aaai.v39i18.34094.

[13] M. B. Ghali, R. Bellafqira, and G. Coatrieux, "FEDCLEAN: Byzantine defense by clustering errors of activation maps in non-IID federated learning environments," 2025, *arXiv:2501.12123*.

[14] L. Yang et al., "Enhanced model poisoning attack and multi-strategy defense in federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 3877–3892, 2025, doi: 10.1109/TIFS.2025.3555193.

[15] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8726–8746, Jul. 2024, doi: 10.1109/TNNLS.2022.3216981.

[16] X. Sheng, Z. Yang, and W. Bao, "FairGuard: A fairness attack and defense framework in federated learning," *IEEE Trans. Dependable Secur. Comput.*, vol. 22, no. 3, pp. 2519–2532, May/Jun. 2025, doi: 10.1109/TDSC.2024.3520134.

[17] H. Yan et al., "A proactive defense against model poisoning attacks in federated learning," *IEEE Trans. Dependable Secur. Comput.*, vol. 22, no. 4, pp. 3529–3543, Jul./Aug. 2025, doi: 10.1109/TDSC.2025.3533029.

[18] M. Wu, B. Zhao, Y. Xiao, C. Deng, Y. Liu, and X. Liu, "MODEL: A model poisoning defense framework for federated learning via truth discovery," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 8747–8759, 2024, doi: 10.1109/TIFS.2024.3461449.

[19] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, and Y. Liu, "LoMar: A local defense against poisoning attack on federated learning," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 1, pp. 437–450, Jan./Feb. 2023, doi: 10.1109/TDSC.2021.3135422.

[20] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2020, pp. 2938–2948.

[21] V. Shejwalkar and A. Houmansadr, "Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. 28th Annu. Netw. Distrib. Syst. Secur. Symp.*, The Internet Society, 2021, pp. 1–19.

[22] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 19–35.

[23] A. Shafahi et al., "Poison frogs! Targeted clean-label poisoning attacks on neural networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6106–6116.

[24] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning FAIL? Generalized transferability for evasion and poisoning attacks," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 1299–1316.

[25] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?," 2019, *arXiv:1911.07963*.

[26] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–19.

[27] 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.

[28] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 634–643.

[29] V. Shejwalkar and A. Houmansadr, "Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. 28th NDSS*, The Internet Society, 2021.
[30] J. Hoffman, D. A. Roberts, and S. Yaida, "Robust learning with Jacobian regularization," 2019, *arXiv:1908.02729*.
[31] D. Varga, A. Csiszárik, and Z. Zombori, "Gradient regularization improves accuracy of discriminative models," *CoRR*, vol. abs/1712.09936, 2018.
[32] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Master's Thesis, Dept. of Comput. Sci., Univ. of Toronto, 2009.
[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS workshop on deep learn. unsupervised feature learn.*, no. 5, 2011.
[35] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[37] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012, doi: 10.5555/2503308.2188395.
[38] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 2960–2968. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html
[39] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 2113–2122. [Online]. Available: http://proceedings.mlr.press/v37/maclaurin15.html
[40] S. R. Young, D. C. Rose, T. P. Karnowski, S. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *Proc. Workshop Mach. Learn. High-Perform. Comput. Environ.*, Austin, TX, USA, ACM, 2015, pp. 4:1–4:5, doi: 10.1145/2834892.2834896.
[41] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.
[42] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, BMVA Press, Sep. 2016 pp. 87.1–87.12.
[43] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. 8th Int. Conf. Learn. Representations*, Addis Ababa, Ethiopia, Apr. 26-30, 2020.

Qi Guo received the BS and MS degrees in communication and information systems from Northwestern Polytechnical University, China, in 2017 and 2020, respectively, and the PhD degree in computer science and technology from Xi'an Jiaotong University, China, in 2025. His research interests include federated learning, artificial intelligence security, and large language model.

Di Wu received the BS degree in computer science in 2019 from Xi'an Jiaotong University, Xi'an, China, where he is currently working toward the PhD degree in computer science. His research interests include privacy-preserving machine learning and federated learning, with a particular focus on Byzantine-robust federated learning.

Yong Qi received the PhD degree from Xi'an Jiaotong University, Xi'an, China. He is currently a professor with the Department of Computer Science and Technology, Xi'an Jiaotong University, and the director with the Institute of Computer Software and Theory. He is also the Deputy Director with the System Software Committee and also with Information System Committee, China Computer Federation. He has authored or coauthored many papers in *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Reliability*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Smart Grid*, *IEEE Transactions on Computers*, and *IEEE Transactions on Mobile Computing*, JPDC, and USENIX ATC. His research interests include distributed systems, cloud computing, mobile computing, and federated learning.

Saiyu Qi received the BS degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2008, and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2014. He is currently an associate professor with the School of Computer Science and Technology, Xi'an Jiaotong University. He has authored or coauthored papers in *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Dependable and Secure Computing*, IJCAI, and CCS. His research interests include applied cryptography, cloud security, and distributed systems.

Qian Li received the PhD degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2021. He is currently an assistant professor with the School of Cyber Science and Engineering, Xi'an Jiaotong University. His research interests include deep adversarial learning, artificial intelligence security, and optimization of theory.

Minghao Yao received the MS degree from Northeastern University, China in 2021. He is currently working toward the PhD degree in computer science and technology with Xi'an Jiaotong University, Xi'an, China. His research interests include distributed systems and artificial intelligence security.

Kaitai Liang (Member, IEEE) received the PhD degree in computer science from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014. He is currently an assistant professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. His research interests include applied cryptography and information security, particularly encryption, blockchain, postquantum crypto, privacy-enhancing technology, and security in cloud computing.