# An Exploratory Analysis on Users' Contributions in Federated Learning

Huang, Jiyue; Talbi, Rania; Zhao, Zilong; Boucchenak, Sara; Chen, Lydia Y.; Roos, Stefanie

**DOI**
10.1109/TPS-ISA50397.2020.00014

**Publication date**
2020

**Document Version**
Accepted author manuscript

**Published in**
Proceedings - 2020 2nd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2020

# An Exploratory Analysis on Users' Contributions in Federated Learning


Jiyue Huang*
Delft University of Technology
Netherlands
Email: J.Huang-4@tudelft.nl

Rania Talbi
INSA-Lyon
France
Email: rania.talbi@insa-lyon.fr

Zilong Zhao
Delft University of Technology
Netherlands
Email: Z.Zhao-8@tudelft.nl

Sara Boucchenak
INSA-Lyon
France
Email: Sara.Bouchenak@insa-lyon.fr

Lydia Y. Chen*
Delft University of Technology
Netherlands
Email: lydiaychen@ieee.org

Stefanie Roos*
Delft University of Technology
Netherlands
Email: s.roos@tudelft.nl



***Abstract*—Federated Learning is an emerging distributed collaborative learning paradigm adopted by many of today's applications, e.g., keyboard prediction and object recognition. Its core principle is to learn from large amount of users data while preserving data privacy by design as collaborative users only need to share the machine learning models and keep data locally. The main challenge for such systems is to provide incentives to users to contribute high-quality models trained from their local data. In this paper, we aim to answer how well incentives recognize (in)accurate local models from honest and malicious users, and perceive their impacts on the model accuracy of federated learning systems. We first present a thorough survey on two contrasting perspectives: incentive mechanisms to measure the contribution of local models by honest users, and malicious users to deliberately degrade the overall model. We conduct simulation experiments to empirically demonstrate if existing contribution measurement schemes can disclose low-quality models from malicious users. Our results show there exists a clear tradeoff among measurement schemes in terms of the computational efficiency and effectiveness to distill the impact of malicious participants. We conclude this paper by discussing the research directions to design resilient contribution incentives.**



***Keywords:*** Federated Learning, Contribution Measurement, Adversarial Behavior, Incentive Mechanisms.


## I. Introduction

The increasing capabilities of ubiquitous sensors and smart devices, whether in terms of computation, storage, or connectivity resources, are driving services from the cloud side to the edge of the networks [1]. Popular machine learning (ML) services are no exception to this trend. Another critical reason behind this trend is the privacy concern [2] of user data that is often sensed and collected on edge devices. Users increasingly ask for on-device learning so as to minimize sharing the data with the cloud.

Federated Learning (FL) [3] is the emerging paradigm that empower ML-tasks on edge devices in a privacy-preserving manner. FL systems enable collaborative training of a machine learning model across distributed users by local model sharing, instead of direct data exchange with the untrusted service providers. Figure 1 illustrates a simplified federated system, where there are multiple users and one federator, the light-weight central server to measure the contribution and provide the rewards[1]. Users rely on their local data to train a common model and periodically exchange their updates of model parameters with the federator, e.g., the weights of neural networks, until the common model converges.

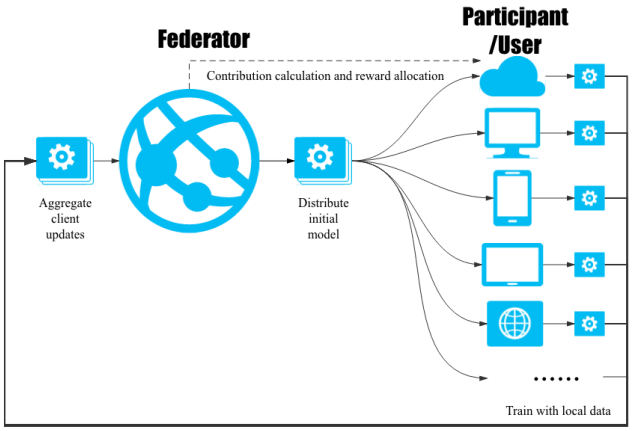


Fig. 1: An illustration of federated system: federator and multiple users/participants.

As local data never leaves the users' devices in federated learning systems, personalized applications that also benefit from other collaborative users thrive, e.g. text prediction [4], voice recognition [5], and self-driving cars [6]. However, in collaborative systems, it is more a norm than rarity that there exist malicious users who either purposely deteriorate the model quality or take advantage of the system without producing real contributions (free-riders).

In this paper, we study the impact of incentive mechanisms on the model quality of federated learning systems considering two type of participants: i) honest participants with varying update quality and ii) malicious participants who deliberately

[1]This is one of the most common configurations of federated systems [3]

send low-quality updates. We show how incentives mechanisms characterize contributions made by these two types of participants and to survey the-state-of-the-art incentive mechanisms that lead to maximal model accuracy.

The specific contributions of this paper are summarized as follows:

- We provide an exploratory analysis of contribution measurement and incentive mechanisms in the presence of honest participants (Section III).
- We characterize malicious behaviors that has been shown to deteriorate model accuracy (Section IV).
- We experimentally evaluate existing contribution metrics in the presence of malicious participants (Section V).
- We provide future research directions to better assess users' contribution and hence handle honest and malicious participants (Section VI) .

## II. Background and Preliminary Notions

***Federated Learning*** is a machine learning setting where multiple participants collaborate in solving a machine learning problem, under the coordination of a central server or service provider called ***federator***. Each participant's raw data is stored locally and is not exchanged or transferred; instead, model updates, e.g., weights of intended for immediate aggregation are used to achieve the learning objective [7].

The federator plays the role of an orchestrator. It starts the training process by assigning learning tasks to the participants, initializes the ***global model***, and aggregates the ***updates*** submitted by participants in each training round. These updates can be either neural network weights or gradients in existing studies.

***Participants*** (or *Users*), on the other hand, locally own data relevant to these specific training tasks. It is important that participants have sufficient computation capability, data, and network resources to be involved in the training process. They use their local training data to update the global model sent by the Federator to build their own *local models*.

Federated learning is an iterative learning procedure composed of five steps that are summarized in Figure 2. These steps are the following: **1. Initialization:** The federator defines a specific machine learning task and initialize the global model. **2. Participant Selection:** To maximize the model quality and for the sake of fault tolerance, the Federator chooses participants with a good network connection and battery level to take part in the training process at a given round, where one round refers to one iteration of local training and global aggregation along with reward allocation. **3. Local Training:** Selected participants receive the initial model from the federator and train local models using their own data. **4. Secure Aggregation:** The federator averages the model updates uploaded from partcipants without access to their local data. **5. Reward Allocation:** The federator distributes rewards to participants based on their own contribution. All steps but Initialization are iterated until the global model achieves a desired performance.

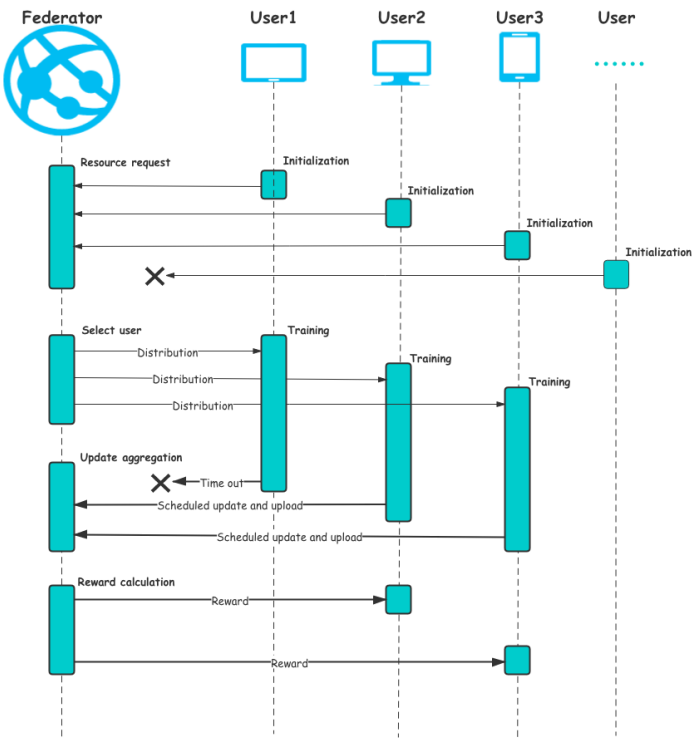


Fig. 2: Protocol of Federated Learning

In real-world applications, participants can either be honest, whose submitted updates are genuinely trained locally with varying data quality, or malicious. Malicious participants misbehave to gain more profits from the offered service or even aim at deteriorating the whole FL ecosystem. In order to characterize the behavior of both types of participants, we would, however, focus on different peculiarity in response to their presences. First, for honest users, fair reward distribution mechanism surely encourage users' participation, especially those with high data quality and willing to contribute more computational power. Designing feasible contribution measurement strategies in federated learning is indispensable but challenging since directly assessing the quality of a user's local data is not possible for the other participants and the federator. Accordingly, there are a number of contribution measurement strategies and corresponding reward systems (see Section III). In contrast, for malicious users, it is essential to identify the malicious nodes and the type of misbehavior. Based on different classes of attacks, defences need to be designed accordingly. In this paper, we present a thorough classification of both attacks and defences.

## III. Assessing Contribution for Honest Participants

For honest Users in federated learning, Federators are supposed to recruit sufficient participants to complete the large-scale tasks with high quality. Participants are more willing to provide high-quality data and resources if they receive rewards. The value of the reward should relate to a participant's level of contribution, i.e., participants who contribute more,

by some measures, should receive a bigger reward. Yet, a major challenge for contribution measurement of FL systems is data isolation caused by the fact that users keep their raw data secret. Local updates reveal information about their performance indirectly, since parameters of neural networks are deep mapped features and do not carry direct information. As a result, FL systems can measure contribution based on updates, without requiring access to the raw data.

### A. Contribution Evaluation Taxonomy

In this section, we summarize three major taxonomy contribution measurement strategies applied in existing federated learning systems. They are of evidently differences in detecting accuracy and transmission complexity but could be suitable for various of application scenarios.

*1)* ***Test /Self-Reported Based Contribution Evaluation****:* The most straight-forward way to measure contribution is to have participants self-report their score, as they have access to their local data and can hence conduct the measurements. Theoretically, self-reported contribution is not a measurement strategy, so we would not discuss it specifically in this paper.

There are multiple ways to define the quality of data in the context of self-reporting. The first one is just the size of the data [8], without knowing their distribution. So in this paper, the model owner (federator) negotiates with the mobile devices (users) about the size of their training data. In return, each mobile device receives the revenue. Alternatively, revenues can depend on the accuracy of the solution to the local sub-problems [9]

Prior to formally define the measure of users' contributions, we first introduce the notations and assumptions. We assume there is a linearly decreasing valuation function $v(\theta_k)$ (which is negatively related to reward portion) for user $k$ depending on the relative accuracy $\theta_k$ attained for the local sub-problem. The protocol, however, requires a trusted third party to ensure uniform pricing as basis and leaves it open how such a trusted party would be realized in practice.

*2)* ***Marginal Loss Based Contribution Evaluation****:* The marginal loss strategies determine the benefit that a participant deserves according to the marginal loss that it brings withdrawing from the alliance. It is widely adopted in Profit Distribution Games [10], which refers to designing reasonable profit distribution strategies among multiple contributors, such as reward allocation for users in federated learning. We note that computing marginal loss requires a central party, which could be either the federator or a different trusted third party with access to the global model. Based on the idea of marginal loss, Richardson et al. [11] show how a payment structure can be designed to measure contributions of different data owners for linear regression models in a crowd-sourcing scenario as well as assigning incentives. It determines the influence that data points have on the loss function of the model to calculate the decrease without a specific user owning these data points. However, the paper merely focuses on linear regression and hence is not of general interest. Furthermore, [12] designs a deletion method to measure contribution of horizontal FL, which means users hold data with same feature space and different ID. In contrast, Shapley Value [13] has been introduced for vertical FL, referring to users holding data with different feature space and same ID. While the Shapley Value can be seen as a marginal loss-based contribution measurement, its main idea relates to game-based incentives, so that we defer to the respective section for a detailed explanation.

*3)* ***Similarity Based Contribution Evaluation****:* Marginal loss-based strategies require the federator or a third party to implement contribution evaluation. However, there are also studies [14] that focus on pairwise measurement, i.e., participants evaluate each other. In this manner, the system reduces both the trust in and the load on the central party. Having a distributed contribution measurement further enhances robustness to the central failure. Kang et. al [14] accomplishes the pairwise contribution qualification by introducing reputation. users apply a multi-weight subjective logic model [15] to obtain reputation of each other. A participant gains higher reputation by providing more positive actions that are recorded in a blockchain for transparency. Besides the pairwise direct reputation by users, there are also indirect reputation designed in this model using the records of multiple federators. Lyu et. al propose FPPDL [16] and demonstrate similarity-based qualification by differential privacy generative adversarial networks (DPGAN) [17]. In FPPDL, data provider generates artificial samples with DPGAN, and data verifier uses its local model to implement cross-user labeling. Then, the verifier computes the contribution measure by the label similarity between the data provider and verifier.

### B. Incentive Mechanisms as Reaction

Here, we introduce incentive mechanism that rewards and reacts to honest participants with different quality based on contribution measurement. Firstly, we rigorously define FL incentives to give a clear understanding. Then, the various goals of incentive design are provided. Moreover, we also survey game theory that is widely adopted incentive design.

*1)* ***Definition of Incentive Mechanism in FL****:* Incorporating the ideas from a multitude of studies on incentive mechanism of federated learning, we propose the first formal definition of incentives for FL.

*Incentive Mechanism of Federated Learning:* An incentive mechanism in FL system consists of a set of rewards $R$ and three functions $v$, $c$, and $r$. The function $v\colon R \to \mathbb{R}$ assigns each reward a value. For a set of participants $P$, the function $c : P \to \mathbb{R}$ assigns each participant a score that measures their contribution to the system. We discussed different contribution measurement strategies in the previous subsection. Last, the function $r : \mathbb{R} \to R$ assigns a reward based on the score that $c$ provides. The reward function $r$ offers rewards of monotonously increasing value, i.e., if $x > y$, then $(r(x)) \geq (r(y))$.

From this definition, we see that the incentive design of federated learning include two main procedure: 1. *Contribution Measurement*, which is discussed above; and 2. *Rewards (punishments) Allocation*. The FL systems deliver rewards

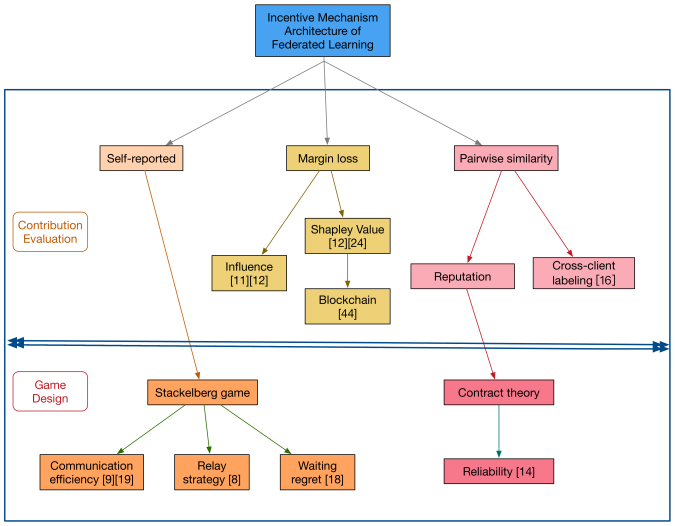

Fig. 3: Recent studies on incentive mechanism of federated learning.

based on the contribution using profit distribution methods including game theory and blockchain. Rewards could be monetary reward, generally, or other schemes such as biased information.

*2) **Goals of Designing Incentive Schemes**:* Based on the definition above, we examine that the incentive mechanism designs for federated learning attempt to encourage desirable behavior in users. More specifically, goals of an incentive scheme generally include two main factors:

**Attract Users of High Local Quality.** The aggregated results of federated learning highly depend on the quality of participated users, including local data size and computation resources. An incentive scheme should attract users of high quality to join, such that the global model achieves good performance. On the other hand, data owners of low quality are supposed to be discouraged from joining due to the low revenue the incentive mechanism offers.

**Attract Users with Good Networking Resources.** Network transmission condition of users and the federator, or between users are also supposed to be taken into consideration while designing incentive mechanism, since both effectiveness and efficiency are imperative for the system performance. Additionally, some systems apply incentive mechanism to enhance some specific characteristics according to the objectives of these distributed systems. For instance, [18] solves the issues of costs and temporary mismatch between contributions and rewards to model users' regret user. Other examples ( [19] , [9]) focus on improving the communication efficiency of federated learning systems through involving transmission time as a highly weighted factor in the utility function of incentive mechanism.

*3) **Incentive Design**:* When participating in FL, users aim to maximize their rewards through incentives in comparison to the data and resources they provide to the system. Given their specific local situation, each participant hence has a utility function they aim to maximize. In order to determine the best way of action, participants consider possible action plans for themselves and the other participants. From a network resource perspective, the overall goal is to maximize collective utility. As a consequence, game theory is a useful methodology to design and analyze FL incentives. In the following, we discuss the different assumptions about participants and their relations in the context of the resulting games and incentives.

**Stackelberg Equilibrium in Non-cooperative Game.** Stackelberg games are of use if one of the players is in a leader position while the others are followers. Thus, they are quite suitable for FL as the federator can be seen as the leading party. In a Stackelberg game, the followers usually first observe the behavior of the leading party before deciding their own actions. Concretely, the leader decides an output, and then the followers can observe this to determine their own output factors such as resource inputs. A limitation of the game is the assumption that the leader should be able to fully apprehend the behavior of the followers and thus needs to be aware of their local utility functions. Thus, the output determined by the leader is a profit maximization constrained by the utility function of the followers. In this strategy, the non-cooperative framework assumes all participants act separately. .

As the data of the participants in FL is not available to the federator, the federator does not know the utility functions with regard to data. Thus, Stackelberg games are only relevant when incentivizing the contribution of network resources. So, they can be applied to mitigate the delays in completion of each training batch by analytically obtaining equilibrium solution of a Stackelberg game [20].

Another Stackelberg game-based approach [9] handles the communication efficiency of users implementing an uncoordinated computation strategy during model aggregation. Specifically, it models a two-stage Stackelberg game by establishing a communication-efficient cost model for users and a reward rate for the federator.

Resources are particularly important in the context of edge and IoT due to the restricted capacity of the devices. Here, Stackelberg games have been suggested for user utility functions depending on the number of local iterations, i.e., local computation power [19]. In contrast, the federator aims at maximizing its utility in terms of the global model, trying to, e.g., minimize the number of communication rounds needed to reach a desirable global accuracy. However, there is not any concrete utility function in this work.

Other studies focus on very specific scenarios for FL. In the absence of direct communication between all participants, incentives for adapting a relay network can be modeled as Stackelberg games [8]. However, in a cooperative relay network design, a larger training data set can result in a lower probability to be relayed due to its higher bandwidth use. As a result, the learning service pricing and cooperative relaying should be considered jointly. Moreover, [18] makes the assumption that the rewards can only be paid once the federation has made a gain from their model. It thus studies the payoff-sharing scheme on costs and temporary mismatch between contributions and rewards of FL, focusing on waiting

time fairness. Their proposed scheme FLI maximizes the overall effectiveness of the data alliance, and at the same time minimizes the imbalance of regret between users of delays caused by the training and commercialization time.

**Contract Theory Application.** Contract theory is an economical theory that regards all transactions and institutions as a kind of contract. It then designs the optimal contract to reduce the moral hazard, adverse selection, and extortion of the parties under the condition of asymmetric information, so as to ultimately improve social welfare. Contract theory can either deal with complete contracts [21], meaning that the predefined contract specifies the legal consequences of every possible state, or incomplete contracts, which includes consideration of the incentive effects of parties' inability to make complete contingent contracts [22].

In federated learning systems, complete contract theory [23] has been applied due to its clear decision tree of responsibilities and obligations. The federator determines the contract items and users choose appropriate contract types based on their own resources to maximize profits on each side. Thus, contract theory is a type of Unbalanced Stackelberg game, with the federator as the leader and dominant the optimization objective of the federated learning system. However, the federator provides multiple optional contract classes for contract theory-based incentives, which is not possible using Stackelberg games to enhance rewarding efficiency.

Incentive schemes based on contract theory are more robust than Stackelberg game, in terms of computational complexity. They allow to simulate data market transactions more realistically and avoid some unnecessary fine-grained operations to enhance efficiency of the federated learning systems. Concretely, contract theory allows the user to select the function that maximizes its own utility based on the evaluation of the quantity, quality, computing resources, and communication capacity of the local data. To maximize its global profit, the federator takes the computation and communication efficiency and model accuracy of the uploaded gradient by users into account. However, verifying the authenticity and quality of the uploaded updates provided by the users remains difficult.

For incentive studies based on contract theory in federated learning, Kang et al [14] address the challenges of incentive mechanisms for participating in training and worker selection schemes for reliable federated learning. It introduces reputation as the metric to measure the reliability and trustworthiness of the mobile devices and combine contract theory to motivate high-reputation mobile devices with high-quality data.

**Shapley Value in Cooperative Game.** The above games are from the perspective of the federator and are based on leadership competition or non-cooperative games. An alternative approach is given by cooperative games: the profit of at least one party increases without reducing the profits of other parties. Thus, the total utility increases with the participating of multiple members. The key methodology here is the Shapley Value [13], which evaluates the contribution of a participant as loss experienced by the participant leaving. In this manner, the Shapley value is independent of the order in which participants join. It assigns a unique distribution among the parties of a total surplus generated by the coalition of all members. Furthermore, the Shapley Value allows using a combination of desirable properties to define a participant's contribution rather than focusing on one property.

Formally, we denote the data federation $F =< Users, v >$ has been contributed by several users as $Users = \{U_1, U_2, ..., U_i\}$, where $v$ is the contribution value function of this system. In federated learning scenarios, it could be the aggregated model accuracy. The Shapley Value define the contribution of $U_i$ to join $Users$ in $F$ as a margin loss despite the joining sequence as:

$$\delta U_i(users) = v\{users \cup U_i\} - v\{users\} \tag{1}$$

Since the Shapley Value makes a fair distribution regardless of the joining order, there are $|Users|!$ joining sequences with corresponding probabilities. The probabilities of each sequence (or coalition) $S$ containing $User_i$ could easily be obtained by $|S|!(|F| - |S| - 1)!/|F|!$ Thus, the contribution of $C_i$ by Shapley Value is:

$$SV(F, C_i) = \sum_{S \subseteq F \setminus \{C_i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \delta C_i(F) \tag{2}$$

There are a number of studies that use Shapley Value for their incentive design. In vertical FL, it has been used to calculate the grouped feature importance since features are grouped to join data federation by multiple users [12]. Although Shapley Value also works for horizontal FL, the reason why the authors apply Influence function is that we need to note is that Shapley Value based distribution solution often takes exponential time to compute with a huge complexity of $O(n!)$, where $n$ denotes the user size. Nevertheless, this method also sheds light on the researches in model contribution using Shapley Value in the context of federated learning.

The key challenge of computing the Shapley Value lies on the need for extra training to compute the marginal contribution of a user. A contribution index that reconstructs the approximate models on different combinations of the datasets through the intermediate results during the training process replaces the exact Shapley value [24]. In this manner, efficient contribution measurement becomes possible.

Last, Shapley Value has been used in combination with a blockchain network due to its fairness and high computation overhead. The party who can decide on a new block is selected based on their Shapley Value [12].

## IV. Malicious User Updates: How to Detect and Limit the Damage

In FL frameworks, machine learning tasks are massively distributed among participants. Ideally, this large-scale distribution helps ML-service providers reach more diversified data sources and thus build stronger models. Nonetheless, in the basic design principals of Federated Learning, user selection

is mainly based on users' data availability, their computational power, and network resources, without any solid guarantees on user reliability or trustworthiness [25]. As a consequence, Federated Learning can be subject to various client-side attacks with different objectives.

It has been shown in prior art [] that participants might deviate from the intended FL protocol and try to bring damage to the ecosystem. This malicious activity varies from simple selfish user behavior to intentionally sending faulty contributions to tamper with the federated model.

In the following, we characterize types of malicious user contributions that might intentionally deteriorate model quality and survey existing detection and prevention mechanisms that protect against them.

### *A. Malicious Behaviour Characterization Criteria*

Multiple state-of-the-art works have been proposed to demonstrate the damage caused by malicious participants in Federated Learning. It is worth mentioning that the attacks discussed in this section are carried out during training time either by insider malicious participants or by outsider adversaries that take over honest participants' devices. Threats are characterized according to the following criteria.

**Adversarial Goal.** Participants can maliciously contribute to FL frameworks for a myriad of goals ranging from provoking arbitrary damage to the system to targeted causative violations. Offenders might try to prevent model convergence, deteriorate model accuracy, incorporate backdoors in the model, miss-classify a certain type of inputs, or even have access to the model without actually participating in the training process.

**Number of Offenders.** Adversarial behavior can be carried by individual participants separately or multiple participants simultaneously. The latter can either be controlled by the same malicious party in order to bring more damage to the system (Sybil Attacks) or can collude to achieve a common adversarial goal.

**Participants' Background Knowledge.** The background knowledge of the attacker is a deterministic factor of the attack severeness. For instance, they may know other honest participants' training data or their training parameters. They can be aware of the mechanism applied by the federator to detect malicious activity or of the global data distribution, and so on.

**Attack Duration.** Some FL malicious behavior may require to be carried out continuously through multiple rounds to take effect. In this case, the attack is said to be stealthy. On the other hand, some adversarial goals are more straightforward to achieve and thus the attack can be carried out in a single round.

### *B. Characterizing Malicious Behaviour in FL*

In the following, we characterize three possible malicious participant contributions (summarized in Table I), that might negatively impact model quality in the FL ecosystem. We describe these attack categories according to the criteria defined above and survey the existing state-of-the-art works that study them.

***Targeted Poisoning.*** In this type of malicious behaviour, an attacker tries to inject a backdoor task of his interest in the global model along with the main task that was initially trained without deteriorating the model's accuracy.This adversarial goal can be achieved in two possible ways. The first one is generating poisonous data locally, carrying out local training on the malicious participant side using this faulty data, and then sending the resulting poisonous updates to the federator for aggregation. Generating poisonous data can be done by simply flipping labels or by injecting naturally occurring or artificial patterns in the feature space that is associated with the backdoor. This malicious behavior is referred to as data poisoning attacks in the state-of-the-art [27], [29], [30], [34]. The second way is model poisoning where the attackers carefully craft poisonous updates that efficiently inject the backdoor task in the model [6], [26], [28], [31]. Both of these attacks can be done by a single participant individually or by multiple sibyls collaboratively [6], [29], [30], [32], [34], [35]. To achieve model poisoning, malicious participants might send faulty contributions over multiple training rounds till the damage is done while the most severe attacks can successfully inject the backdoor in a single round [6].

***Untargeted Poisoning.*** Unlike targeted poisoning, in this category of malicious behaviour, the attacker's goal is to cause a high miss-classification rate indiscriminately for testing samples. As a consequence, the learned model is unusable and hence the attack is essentially a denial-of-service attack. Generally, the malicious participant does not need to carry out data poisoning but can simply craft model updates that provoke severe accuracy drop. Concrete instantiations of this type of attack in the federated learning setting include [31], [32]. The impact on model accuracy can be even more aggressive when the attacker is aware of the detection mechanism used on the federator's side [31] since it can adapt the pace of sending malicious contributions to remain undetected (up to 78% accuracy drop [31]).

***Free-rider.*** In this category of malicious behaviour, self-interested participants want to take advantage of the federated learning service without actually participating in it due to the lack of data, lack of computing resources, or even for privacy concerns. To do that, free-riding participants craft fake updates via simple random generation or based on previous versions of the model to pretend that they participate in the learning process. Even though this kind of behavior has been widely explored in the case of peer-to-peer systems, there is only one state-of-the-art work that explores how it applies to federated learning [33]. Although the presence of free-riders in FL-based frameworks might seem harmless, the behaviour of this category of participants is opposite to the main purpose of federated learning which consists of doing large scale distribution of ML-based tasks to have access to more diversified and rich data sources. Free-riders can either have no novel contributions to the system or in worse scenarios send arbitrary updates that might negatively impact the trained

| Attack Category | Attack | Adversarial Goal | Number of Offenders | Participants' Background Knowledge | Offense Duration |
|---|---|---|---|---|---|
| Targeted Poisoning Attacks | [26] | Provoke targeted misclassification and negate the combined effect of benign agents | Single attacker | White-box access to the model, Access to training data | Stealthy |
| | [27] | Assign an attacker-chosen label to input data with a specific trigger | Sybil attack | White-box access to the model, Access to training data, Access to a portion of a subset of the feature space | Stealthy |
| | [28] | Introduce a persistent change in a joint meta-learning model such that, when a user adapts it for a new classification task, targeted misclassification occurs | Single attacker | White-box access to the model, Access to training data | One-shot |
| | [29] | Provoke high testing errors for particular subset of classes | Sybil attack | White-box access to the model, Access to training data | Stealthy |
| | [6] | Inject a backdoor task in the model | Single attacker | White-box access to the model, Access to training Data, Knowledge regarding the detection mechanism used by the federator | One-shot |
| | [30] | Provoke high testing errors for particular subset of classes | Sybil attack | White-box access to the model, `Access to training Data | Stealthy |
| Untargeted Poisoning Attacks | [31] | Cause a high miss-classification rate | Sybil attack | White-box access to the model, Access to training Data, | Stealthy |
| | [32] | Degrade the overall model performance | Sybil attack | White-box accessto the model | Stealthy |
| Free-Rider Attacks | [33] | Have access to the model without participating in the training | Single attacker | White-box access to the model, Knowledge of how normal updates look like | Stealthy |

TABLE I: Characterization of malicious behavior in Federated Learning

model's accuracy.

### C. Defense Mechanisms Against Malicious Contributions

There are two possible ways to protect against malicious contributions in Federated Learning. On one hand, the federator can implement detection mechanisms and punish attackers once he suspects an anomaly. He can either react by reducing their learning rate gradually or directly evict them from the system. On the other hand, the basic Federated Learning protocol can be enhanced by prevention mechanisms that stop malicious behavior from occurring in the first place. We present below some state-of-the-art mechanisms that were proposed to detect and prevent malicious contributions in FL frameworks.

***Gradient Auditing.*** The purpose of this kind of protection mechanism is to detect and punish malicious behaviour such as model poisoning or free-riding. In this case, the federator is assumed to be trusted and he monitors statistical changes in model updates. The latter tries to point out suspicious updates, and exclude them from the aggregation process or reduce their weights. Examples of such approaches are FoolsGold [36] and Gradient Norm Bounding [37].

***Trusted Execution Environments.*** This a hardware-based protection mechanism that is mostly adapted to cross-silo [2] federated learning ecosystems where the local training code on the participants-side is implemented in a Trusted Execution Environment (TEE) such as Intel-SGX (e.g., [38]). This way, the code run by participants is certified by the federator to make sure that the updates they send are not malicious. Thus, trusted execution environments prevent any attempt at deviation from the intended FL protocol.

***Gradient Sparsification.*** This protection mechanism limits the effect of causative attacks in federated learning by pruning gradients that have small magnitude, this is also referred to as gradient compression. It has been shown in [39] that gradients can be compressed up to a factor of 300, while maintaining the same model accuracy. This approach was initially proposed to reduce communication bandwidth in distributed learning but was proved in [27] to be an effective way to protect against targeted poisoning with a reasonable accuracy-loss/protection-level tradeoff.

***Differential Privacy*** Initially, differentially-private FL was proposed to reduce information leakage about local users' data [40]. However, since adding noise to user updates bounds their influence over the joint model, some state-of-the-art works [6], [27] considered using differential privacy as a protection mechanism to limit the damage caused by poisoning attacks. This approach works by first clipping amplified and potentially malicious updates, then adding Gaussian or Laplacian noise to them. This simply reduces the impact of causative attacks but does not entirely eliminate them. Also, adding user-level noise potentially reduces the accuracy of the trained models.

[2]Cross-silo Federated Learning is an FL setting that involves a small number of relatively reliable clients, for example multiple organizations collaborating to train a model.

## V. Empirical Analysis

Here, we aim to quantify how the existing contribution measurement strategies could recognize attackers and their stability under attackers. Specifically, we consider a scenario of federated training image classifier with benign and malicious users. We implement three popular strategies against the attack of flipping labels.

### A. Experiment Setup

The Federated Learning system under evaluation consists of one federator and 4 users. The model to be trained is a VGG-type [41] convolutional neural network (CNN). Each user possesses 6000 unique data samples randomly selected from the CIFAR10 dataset [42]. Original CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.

Some of the users are malicious and perform a data poisoning attack. When the attackers train their local models, they inject data noise by flipping the label with a probability $p$. The label is flipped with one of the other 9 labels randomly.

The flipping probability $p$ is varied between 10%, 30%, 50% and 100%. The number of attackers is varied between 0 and 3.

In the following, we evaluate the user data contribution to the global model with three mechanisms:

- **Influence**: The classic notion of Influence means to measure the effects on global accuracy of individual data points [43]. Denote the global aggregated model as $\hat{\theta}$ and the global model $\hat{\theta}_{/i}$ without the user $U_i$ as $\hat{\theta}_{/i}$. The contribution for a data set $T$ is then quantified as the difference in accuracy between the two models, i.e., $inf(U_i, T, \theta) = Acc(T, \hat{\theta}) - Acc(T, \hat{\theta}_{/i})$.
- **Reputation**: Similar to Influence, Reputation quantifies the influence of each user. However, the score assigned is binary with 1 indicating that the involvement of user $C = U_i$ improves global accuracy. Reputation considers several time slots (similar to global rounds). In our experiment, there are $ts = 5$ time slots and we average the contribution measurement of user $U$ as $Rep(U_i, T, \theta) = \frac{1}{ts} \sum_{ts} H(Acc(T, \hat{\theta}) - Acc(T, \hat{\theta}_{/i}))$, where H(x) is the heaviside unit step function.
- **Shapley**: In the settings of Shapley Value, we follow the definition and calculation of Equation 2. Four users join this training process and the federator determines their contributions by sequential deletion of marginal loss. The Shapley Value could see the impact on joining order of different users in federated learning.

The reason why we present the evaluation details of Shapley in Section 3 while the others above is that Influence and Reputation are relatively straight forward and we just need to specify some parameters. However, Shapley evaluation is also a solution to Cooperative Game whose algorithm is well defined in existing studies. Note that all three mechanisms are marginal loss-based, as the other types of approaches like self-reporting are obviously unable to deal with attacks.

The experiments are conducted with library Keras-2.3 based on Tensorflow-2.2, and executed on Dell Alienware Aurona (20 CPUs with 32G RAM) equipped with one RTX 2018 Ti GPU.

### B. Experimental Evaluation

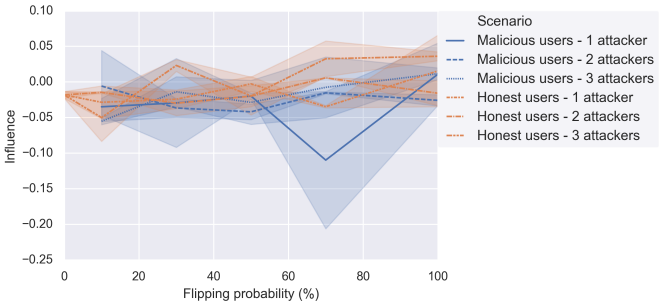


Fig. 4: Influence mechanism under combinations of users and attackers. A higher value indicates a higher contribution.

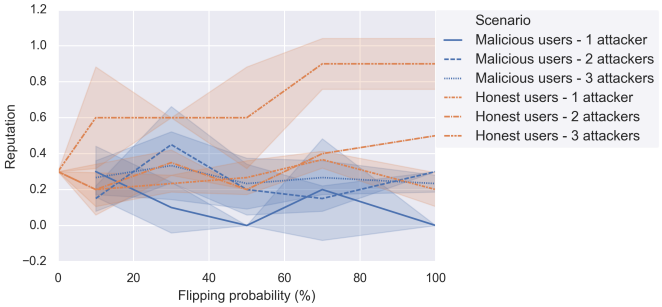


Fig. 5: Reputation mechanism under combinations of users and attackers. Higher the value, better the reputation.

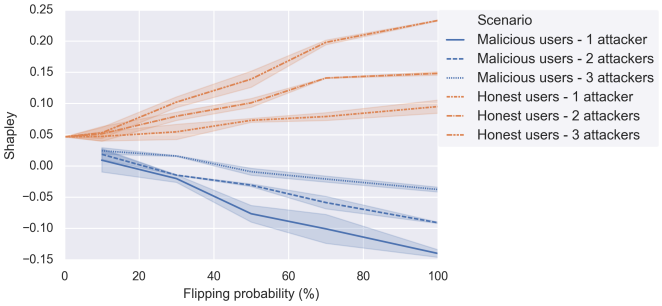


Fig. 6: Average Shapley Value under combinations of users and attackers. Higher the value, better the contribution.

Figure 4 - 6 display the measured user's data contribution with respectively Influence, Reputation, and Shapley. In our experiments, we vary the number of attackers, and we vary the flipping probability $p$. Here, the values of Reputation have been normalized into $[0, 1]$. Thus, generally, all three strategies succeed in recognizing attackers since we could see from Figure 4 - 6 that the mean values of averaging honest users are larger than those of attackers. It demonstrates the

effectiveness of contribution measurement approaches based on marginal loss and similarity. Additionally, overall results also show that for malicious users, higher flipping rates may result in lower measured contribution, which is more stable in Figure 6 while there are fluctuations in Figure 4 and 5. And if we consider a given flipping rate, e.g., 3 attackers with a flipping rate of 50%, the Influence in Figure 4 of honest users and the Influence of malicious users are almost the same with 3 attackers with a flipping rate of 10%. This exhibits the fact that such techniques (similar in Figure 5) to quantify user contribution is not pertinent in the case of malicious users.

Comparing the three figures, Shapley measurement in Figure 6 shares the highest capability while Influence in Figure 4 finds difficulty in recognizing attackers. This is reasonable since Reputation in Figure 5 qualifies and sums up influence values in multiple rounds, which also indicates the potency after multiple global iterations of both strategies. We could also observe that especially in Figure 6 and Figure 5, the average value on honest and malicious users share opposite trends on the value with increasing flipping ratio. The diversity indicates the implicit relativity between the contribution of the honest and the malicious users since they are all based on marginal loss. As for Shapley, the significant difference to Influence illustrates the importance of the impact of joining sequences in federated learning. In addition, similarly, the variety on different flipping level shows more discrepancies and conforms most to our theoretical prospective on Shapely than Influence and Reputation.

## VI. Research Directions

We have seen that incentives in federated learning require consideration of malicious behavior as they are not necessarily able to detect such behavior. In this section, we outline research directions to investigate this research gap which we believe are promising.

### A. Novel Attack-Aware Incentives

As indicated by the results in Section V, designing new incentive mechanisms should consider attacks. One possible solution may be introducing blockchain-based contribution measurements with transmitted parameter records on chain [44]. Indeed, such a incentive mechanism can possibly be used to detect attackers as those users achieve low scores in the contribution measurement. After attack detection, malicious users can be evicted from the system to prevent future harm.

### B. Alternative Contribution Measurements and Alternative Attacks

Our experimental evaluation in this paper considered merely label flipping attacks and three contribution measurement approaches. Future studies should extend these results to other attacks and contribution measurement mechanisms. In Section IV-B, we already identified untargeted poisoning and Free-riding attacks as potential threats that require further consideration in the context of incentives. An example for a future study related to Free-riding is to evaluate whether cross-user labeling recognizes attackers whose adversarial goal is to have access to the model without participating in the training. In particular, as all users just transmit and verify generated data based on their own data, an attacker can generate new data based on other submissions to appear as if they contribute.

We can also evaluate these contribution measurement strategies in the presence of other non-causative active attacks that aim at inferring sensitive information about participants' data such as class-representatives [45], data distribution [46], etc. Although these attacks do not specifically target model quality, they may indirectly have an influence on it.

### C. New Attacks Targeting Incentives

In this paper, we primarily focused on the impact of attacks on model accuracy. Yet, Free-riding does not primarily target model accuracy but rather deals with parties that gain something without contributing appropriately. As stated in Section IV, Free-riding attacks are not yet fully explored in the context of Federated Learning. The work presented in [33] considers an adversarial model where lazy participants aim at using the federated model without actually being engaged in the training process. We believe that it could be interesting to explore other adversarial strategies for this attack category in the presence of incentive mechanisms. In the context of incentives, adversaries want to maximize the profit they gain out of the deployed incentive mechanism and simultaneously minimize the computational effort they have to invest into gaining from the mechanism. Concretely, a self-interested participant carefully crafts model updates that seemingly have high quality without doing actual local training. This is a contrasting view of attack-aware incentive design in terms of adversarial goals that are equally undesirable as participants are less likely to be incentivized to contribute honestly if incentives can be gamed.

## VII. Conclusion

Motivated by the increasing threat of malicious users on federated learning systems, we presented exploratory analysis on how contribution measurement strategy of incentive mechanisms can characterize attackers. We surveyed existing attacks on model accuracy and highlighted that they can have a detrimental impact on incentive measures. Through federatedly training a deep image classifier, we evaluated how simple label flipping attacks can degrade the performance of the state-of-the-art incentive measures. Based on empirical evaluation and observations, we discuss future research directions. Specifically, it is imperative to design new incentive mechanisms that are resilient to novel attacks circumventing the detection of incorrect data. We also highlight how free-rider attacks with the goal of gaining unjustified rewards is a largely unexplored but critical threat.

## VIII. Acknowledgment

This work has been partly funded by the Swiss National Science Foundation NRP75 project 407540_167266.

## References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 19–38.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

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

[5] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.

[6] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 2938–2948.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, and A. N. B. et.al, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019. [Online]. Available: http://arxiv.org/abs/1912.04977

[8] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y. Liang, "Joint service pricing and cooperative relay communication for federated learning," in *iThings/GreenCom/CPSCom/SmartData 2019, Atlanta, GA, USA, July 14-17, 2019*. IEEE, 2019, pp. 815–820. [Online]. Available: https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00148

[9] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wireless Communications*, vol. 19, no. 5, pp. 3241–3256, 2020. [Online]. Available: https://doi.org/10.1109/TWC.2020.2971981

[10] Y. Wang, X. Ma, Z. Li, Y. Liu, M. Xu, and Y. Wang, "Profit distribution in collaborative multiple centers vehicle routing problem," *Journal of cleaner production*, vol. 144, pp. 203–219, 2017.

[11] A. Richardson, A. Filos-Ratsikas, and B. Faltings, "Rewarding high-quality data via influence functions," 2019.

[12] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*. IEEE, 2019, pp. 2597–2604. [Online]. Available: https://doi.org/10.1109/BigData47090.2019.9006179

[13] A. E. Roth and L. S. Shapley, "The shapley value: Essays in honor of lloyd s. shapley," *Economic Journal*, vol. 101, no. 406, pp. 235–264, 1988.

[14] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 700–10 714, 2019. [Online]. Available: https://doi.org/10.1109/JIOT.2019.2940820

[15] Y. Liu, K. Li, Y. Jin, Y. Zhang, and W. Qu, "A novel reputation computation model based on subjective logic for mobile ad hoc networks," *Future Gener. Comput. Syst.*, vol. 27, no. 5, pp. 547–554, 2011. [Online]. Available: https://doi.org/10.1016/j.future.2010.03.006

[16] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng, "Towards fair and privacy-preserving federated deep models," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, 2019.

[17] P. Lu and C. Yu, "POSTER: A unified framework of differentially private synthetic data release with generative adversarial network," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 2547–2549. [Online]. Available: https://doi.org/10.1145/3133956.3138823

[18] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*, A. N. Markham, J. Powles, T. Walsh, and A. L. Washington, Eds. ACM, 2020, pp. 393–399. [Online]. Available: https://doi.org/10.1145/3375627.3375840

[19] L. U. Khan, N. H. Tran, S. R. Pandey, W. Saad, Z. Han, M. N. H. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *CoRR*, vol. abs/1911.05642, 2019. [Online]. Available: http://arxiv.org/abs/1911.05642

[20] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *IEEE Networking Letters*, 2019.

[21] B. Holmstrom and P. Milgrom, "Multi-task principal-agent analyses: Incentive contracts, asset ownership and job design," *Journal of Law Economics & Organization*, vol. 7, pp. 232–244, 2012.

[22] O. D. Hart and J. Moore, "Incomplete contracts and renegotiation," *Econometrica*, vol. 56, 1988.

[23] B. Holmstrom and P. Milgrom, "Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design," *JL Econ. & Org.*, vol. 7, p. 24, 1991.

[24] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2577–2586, 2019.

[25] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[26] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, 2019, pp. 634–643.

[27] Y. Liu, Z. Yi, and T. Chen, "Backdoor attacks and defenses in feature-partitioned collaborative learning," *arXiv preprint arXiv:2007.03608*, 2020.

[28] C.-L. Chen, L. Golubchik, and M. Paolieri, "Backdoor attacks on federated meta-learning," *arXiv preprint arXiv:2006.07026*, 2020.

[29] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," *arXiv preprint arXiv:2007.08432*, 2020.

[30] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding distributed poisoning attack in federated learning," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 233–239.

[31] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," *arXiv preprint arXiv:1911.11815*, 2019.

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

[33] J. Lin, M. Du, and J. Liu, "Free-riders in federated learning: Attacks and defenses," *arXiv preprint arXiv:1911.12560*, 2019.

[34] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings."

[35] G. Sun, Y. Cong, J. Dong, Q. Wang, and J. Liu, "Data poisoning attacks on federated machine learning," *arXiv preprint arXiv:2004.10020*, 2020.

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

[37] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[38] D. Lie and P. Maniatis, "Glimmers: Resolving the privacy/trust quagmire," in *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, 2017, pp. 94–99.

[39] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[40] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[42] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[43] R. D. Cook and S. Weisberg, "Characterizations of an empirical influence function for detecting influential cases in regression," *Technometrics*, vol. 22, no. 4, pp. 495–508, 1980.

[44] Y. Liu, S. Sun, Z. Ai, S. Zhang, Z. Liu, and H. Yu, "Fedcoin: A peer-to-peer payment system for federated learning," *CoRR*, vol. abs/2002.11711, 2020. [Online]. Available: https://arxiv.org/abs/2002.11711

[45] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.

[46] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Eavesdrop the composition proportion of training labels in federated learning," *arXiv preprint arXiv:1910.06044*, 2019.