[1]Dr. Jyoti Yadav
[2]Swati Jadhav
[3]Dr. Vilas Kharat
[4]Dr. A. D. Shaligram

Strengthening Federated Learning: Addressing Model Poisoning Attack and Defense Methods

JES
Journal of Electrical Systems

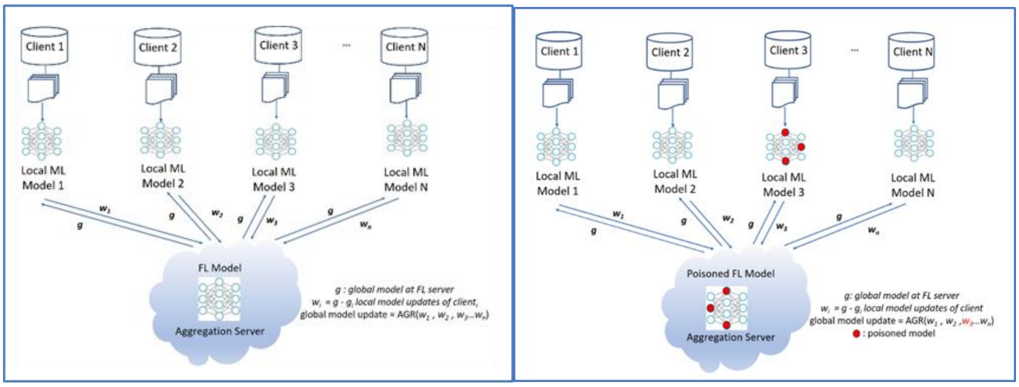
Abstract: - Federated Learning (FL) is a machine learning technique that enables multiple devices to train a shared global model collaboratively without compromising privacy. In FL, data remains on each device, and models are trained locally using that data. The local model updates are then aggregated to update a global model, representing all devices. Since only model updates are shared with the server, privacy is maintained. FL offers numerous benefits, including privacy-preserving, lower communication costs, better scalability, and more. It can be applied in various applications such as natural language processing, computer vision, personalized recommendations, etc. However, FL also poses challenges, particularly model poisoning attacks. Model poisoning occurs because of FL's decentralized and privacy-preserving nature, where the central server does not have direct access to the participants' data and updates, making it difficult to detect when malicious participants send manipulated updates that can degrade or maliciously influence the global model. This vulnerability is especially concerning as FL is increasingly adopted in sensitive fields like medicine and finance. Thus, understanding various model poisoning attacks and their impact on the global model's performance is critical. This paper highlights the need for a new robust aggregation method to handle a wider range of attacks by analyzing various model poisoning attacks and countermeasures.

Keywords: Federated Learning (FL), Model Poisoning attacks, Defense, Global Model

I. Introduction

In the era of global digitization, huge amounts of data are produced every day. Different machine learning techniques are developed to extract valuable insights from this data. This vast data may be spread over multiple devices and comprise important private data, making central machine learning difficult. To solve this issue, a new machine learning technique has been proposed called FL, in which an initial global model is broadcasted to all client devices.

A central server frequently collects updates (weights or gradients) that the clients' compute by training this model using their local private data. These gradients from multiple clients are aggregated using different aggregation rules (AGR) and are used to train the shared global model jointly trained FL model. This updated server's model is sent to all clients participating in the FL training [1]. This entire process is repeated until there is no change in the global model's performance. This FL architecture is shown in fig. 1(a).


Client 1
Client 2
Client 3
...
Client N
Local ML Model 1
Local ML Model 2
Local ML Model 3
Local ML Model N
w_1
w_2
w_3
w_n
g
FL Model
Aggregation Server
g : global model at FL server
$w_i = g - g_i$ local model updates of client,
global model update = AGR(w_1, w_2, w_3...w_n)
Poisoned FL Model
g: global model at FL server
$w_i = g - g_i$ local model updates of client
global model update = AGR(w_1, w_2, w_3...w_n)
● : poisoned model

(a) (b)

Fig. 1. (a) FL Architecture (b) Model Poisoning in FL

[1]Department of Computer Science, Savitribai Phule Pune University, Pune, India. yadav.jyo@gmail.com

[2]*Department of Computer Science, Savitribai Phule Pune University, Pune, India. swati311278@gmail.com

[3]Department of Computer Science, Savitribai Phule Pune University, Pune, India. laddoo1@yahoo.com

[4]Department of Electronic Science, Savitribai Phule Pune University, Pune, India. adshaligram@gmail.com

This collaborative approach can also make FL vulnerable to various attacks like backdoor attacks, data reconstruction attacks, model stealing attacks, model inversion attacks, differential privacy attacks, Sybil attacks, poisoning attacks, etc. Poisoning attacks include Data poisoning attacks, where attackers inject malicious or invalid data into the training data to corrupt the local model's learning, and Model poisoning attacks, where attackers make efforts to degrade the effectiveness of the resulting server's model by manipulating updates (malicious inputs) to the global model through the FL training, as shown in fig. 1(b). The model poisoning attacks are of two types: untargeted attacks and targeted attacks. In untargeted attacks [2][3][4][5], the attacker's objective is to decline the accuracy of the global model on any test data, and in targeted attacks [6][7], the attacker's objective is to decline the accuracy on specific test data.

In model poisoning attacks [2][3][4][6][7][8], the attackers can change the gradients directly on compromised devices and then send them to the server in each iteration. With data poisoning attacks [9][10], the attacker indirectly manipulates the gradients of compromised clients by poisoning data used for training. As a direct change in the gradients is possible in model poisoning attacks, it strongly impacts FL. So, to understand the risk severity in FL, we focus on the untargeted model poisoning attacks and their remedies.

The remaining paper is organized as follows: Section 2 describes existing methods for untargeted model poisoning attacks, followed by the countermeasures against them, and Section 3 analyses and compares different untargeted model poisoning attack and defense methods. Finally, Section 4, summarizes the work and provides some promising future research directions.

II. Related Work

A. Model Poisoning Attacks

They are categorized into targeted attacks and untargeted attacks. Different methods in the literature that can induce untargeted poisoning attacks in FL are studied, as shown in fig. 2. Here, the unauthorized user aims to decrease the global model's overall performance.

a) **LIE** [3] (Little is Enough): Here, the adversary computes and sends the malicious updates using the average and standard deviation of known benign gradients without any knowledge of AGR at the server. The small but well-crafted changes to the data or model parameters are sufficient to compromise the accuracy of the shared trained model.

b) **Additive Noise attack** [17]: This attack adds Gaussian noise to local model updates to make it malicious and then sends it to the FL server during the training [18][19].

c) **Sign-flipping attacks** [18]: It flips the sign of malicious model updates without knowing other benign client updates [19][20]. Hard-thresholding-based defense fails since the magnitude of the local model updates remains the same [17][21].

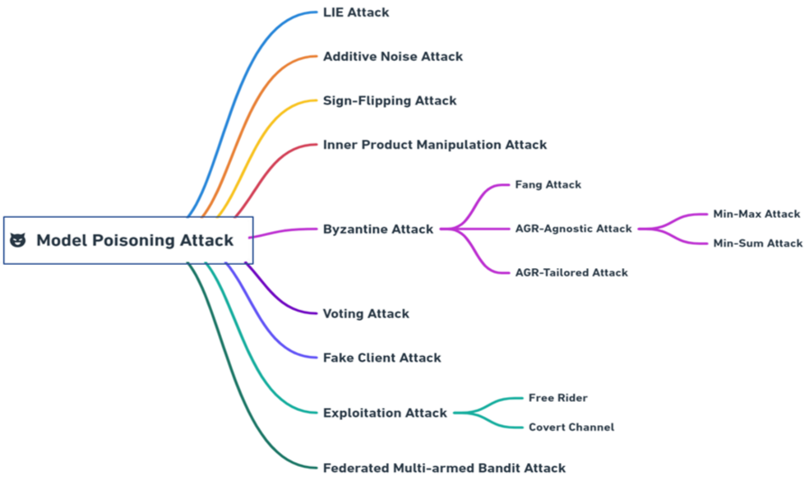


Fig. 2. Untargeted Model poisoning attacks

d) Exploitation Untargeted Poisoning Attack: Poisons the FL model so that it can be used for illegal purposes.

- **Free rider attack** [22]: Here attacker masks as a benign client and receives the model from the server free of cost without making any changes to the gradient. This does not impact the effectiveness of the model, but hampers model convergence speed.

- **Covert Channel attack** [23]: This attack successfully established a secret communication of 1-bit among two clients with the help of poisoning attacks by turning FL systems into covert channels to execute a stealth communication arrangement. A malicious sender can poison the global model by submitting purposely crafted examples. Although this attack has less impact on the aggregation model and other participants, it can be observed by a malicious receiver and used to transmit the data.

e) **Inner Production manipulation attack** [24]: Here, the gradient descent algorithm is made effective during the training, so that the direction of the true gradient should be the same as that of the direction of the aggregation vector i.e., their inner product must be non-negative.

f) **Model Poisoning Attack based on Fake Clients** (MPAF) [25]: The attacker uses free software [26][27] or open-source projects [28] to generate fake clients. In each epoch, these clients calculate the fake model update's direction from the current server's model and the randomly initialized (base) model and scale it up.

g) **Voting Attack** [29]: It compresses the local model updates using SignSGD (a gradient compression technique that uses the sign of each coordinate of the stochastic gradient vector) before sending them to the FL server. The FL server calculates the position-wise summation of all input gradients, and the summation's sign is used to decide the global model updates.

h) Byzantine Attacks: Here, some clients can be malicious, They manipulate the parameters (gradients or weights), and then send them to the FL server to reduce the global model's training accuracy.

- **Fang attack** [4]: It generates a corrupted local model to diverge the parameters of a global model in the reverse direction of the updates of a global model. The attack has control over less than 50% of the devices in the FL system and has complete information about the server's AGR, client dataset, client model, and ML algorithm used for training. Attack impact is greater on Krum and Bulyan AGR, but less on Mkrum, Trimmed-mean, and Median AGR [1].

- **Manipulating the Byzantine** [1]: It uses the information about some benign client updates. The attacker evaluates the reference benign updates and uses it to generate a malicious update to divert benign aggregate in the malicious direction.

➢ **AGR-tailored attacks** are the attacks, where various optimization techniques are applied to break the state-of-the-art aggregation rules. It increases the perturbation to a reference benign update so that it will not be detected by robust AGRs.

➢ **AGR-agnostic attacks** use l2-norm space within which all the benign gradients lie and decide the upper bound on perturbation. These attacks search for malicious gradients based on the distance calculations of benign inputs.

o **Min-Max attack** calculates the distance between the benign inputs. The highest and lowest distance among benign inputs defines the malicious input. All malicious inputs should be closer to maximize the attack impact [1][29].

o **Min-Sum attack** ensures that the sum of squared distances of the malicious inputs from all the benign inputs is the maximal sum of squared distances among any benign inputs. As a result, malicious inputs and benign gradients lie closer.

i) **Federated Multi-Armed Bandit (FMBA) attack** [30]: Here a group of learners have various models locally and play a multi-armed bandit game. The users utilize and send their gathered demerits to the service provider to study more about the global feedback model. If more benign members are in the group, then that is easy to make more accumulation unable to do. To minimize the risk from Byzantine attackers, it uses robust statistics and presents a defensive method called Fed-MoM-UCB using an estimator based on the median.

B. *Defense against Untargeted Model Poisoning Attacks*

In literature, different defense methods are proposed to mitigate the effect of model poisoning attacks. These methods are categorized as statistical-based aggregation and criteria-based aggregation methods as shown in fig. 3.

a) Statistical-based aggregation: These aggregation methods reduce the influence of statistical outliers among the local model updates which helps to reduce the impact of model poisoning attacks.

i) FedAvg [31] finds the dimension-wise average of all weights received from the clients and then applies it to the global model, effectively updating it based on the knowledge gained from the local models across the devices. The performance of FedAvg under the AGR-agnostic attack on the MNIST dataset is better as compared to the AGR-tailored attack (AGR-updates) on the CIFAR10 [1] dataset.

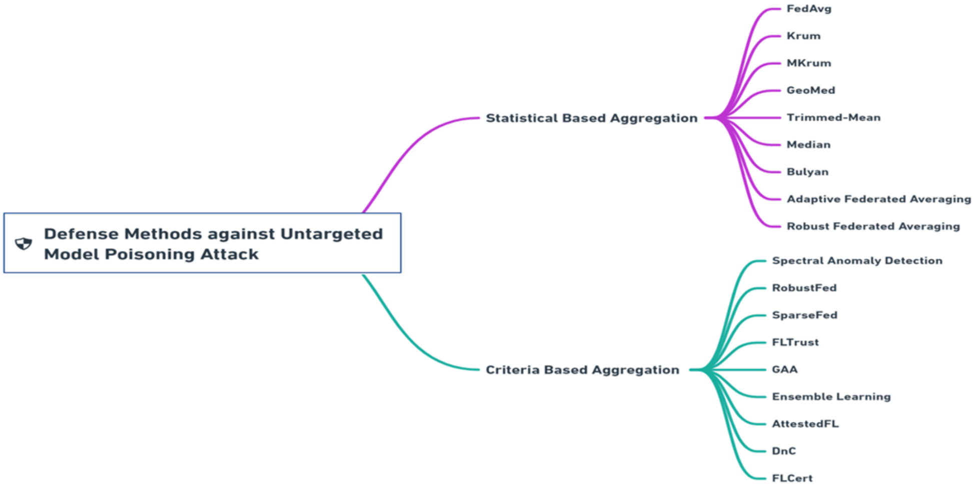

Defense Methods against Untargeted Model Poisoning Attack
Statistical Based Aggregation
FedAvg
Krum
MKrum
GeoMed
Trimmed-Mean
Median
Bulyan
Adaptive Federated Averaging
Robust Federated Averaging
Criteria Based Aggregation
Spectral Anomaly Detection
RobustFed
SparseFed
FLTrust
GAA
Ensemble Learning
AttestedFL
DnC
FLCert

Fig. 3. Defense methods against untargeted model poisoning attacks in FL

ii) Krum [11] selects the gradient from the input set that is close to its n - 2 - m adjacent gradients in the squared Euclidean norm space, where n denotes the number of client devices and m is the number of attackers. It may be affected by some abnormal model parameters since individual local model updates may have an impact on the Euclidean distance between two local models [2].

iii) MKrum [11] applies Krum to select gradients from all gradients received from clients and adds them to a set of selected gradients. Selects k gradients by applying Krum such that $(n – k) > (2m + 2)$, and then the average of all gradients is input to the global model. MKrum increases global model accuracy much better than Krum, but under an AGR-tailored attack, performance is very poor [1].

iv) GeoMed [17] calculates the geometric median of all model updates received, which might include malicious updates. The geometric median may not match any local model update. GeoMed AGR is strong against the additive noise attack. With a sign-flipping attack, the global updates generated deviate from benign updates. It also fails with the backdoor attack [20].

v) Trimmed-Mean [5] sorts each dimension of all input gradients, removes m highest and lowest values, and takes the median of each dimension, where m is the number of attackers.

vi) Median [13] aggregates all input by taking dimension-wise median. This method returns the median for every coordinate of the local updates. The median is stronger than the mean AGR, the resultant model is less inclined by malicious clients.

vii) Bulyan [2] selects k gradients from all input gradients such that $k <= n - 2$ using MKrum and then calculates the Trimmed-mean of these gradients.

viii) **Adaptive Federated Averaging (AFA)** [32] calculates cosine similarity between each of the input model updates and the weighted average. It discards the gradients based on mean, median, or standard deviation of the similarities.

ix) **Robust Federated Averaging (RFA)** [33] proposed the aggregation built using the geometric median based on the Weiszfeld method. RFA beats the classical aggregation methods in terms of robustness, but it is completely uncertain to the actual corruption level. RFA preserves privacy as compared to other robust methods.

b) **Criteria-based aggregation**: Here, the evaluation of local model updates is done based on some criteria like trust score [34], reliability score [35], error rate [4], etc. In the aggregation process, criteria are used as the weights of local model updates. All local models, that do not satisfy criteria are detected as malicious inputs and removed from the aggregation process, so these methods help to detect poisonous model updates.

i) **Spectral Anomaly Detection with Variational Autoencoder** [17] detects attackers by embedding updates from local models into their low-dimensional latent space using an encoder that removes noisy and redundant features from the dataset, which helps to identify normal or abnormal data. Then the decoder module uses these embeddings to reconstruct the original data and find reconstruction errors. In each FL round, the dynamic threshold is computed by taking the mean of entire the rebuilt errors. Remove model updates that have a high dynamic threshold as they are malicious, and the remaining benign updates are used in the aggregation process. To optimize the parameters of the encoder-decoder, reconstruction error is used model until convergence. Due to the dynamic thresholding policy, this defense is stronger.

ii) **FLTrust** [34] allocates a Trust Score (TS) to the client’s update depending upon its similarity in direction to the server’s model update in each round. A TS of a client is a similarity of the ReLU-truncated cosine similarity to the model updates at the server. The server stores a small data set corresponding to the learning algorithm for validation purposes. A smaller TS value indicates a larger inclination of the client model. The server normalizes each local model update and takes their average as weighted by their trust score. FLTrust outperforms statistics-based aggregation methods [37].

iii) **Robust-Fed** [35] evaluates the reliability score for all model updates using an optimization-based truth inference method that minimizes the deviation of weight from the true aggregated parameters. Based on the model updates submitted in the current and earlier rounds, the reliability score is calculated. Clients with low reliability scores are treated as attackers and removed from the aggregation process.

iv) **Sparse-Fed** [36], the server calculates the aggregation of only high top_k magnitude elements. It decreases the certified radius. The certified radius is the maximum distance between a benign model and a poisoned model. This reduces the impact of an attacker on the model. This defense mitigates attacks where multiple clients participate in the training phase.

v) **Ensemble Learning** [4], the server randomly forms groups of all clients. Each group trains a model. A server records predictions made by each model. It removes the local model that has negative effects on the error rate - Error Rate-based Rejection (ERR) and loss - Loss Function-based Rejection (LFR) for the validation dataset of the global model. Finally, the server selects models whose predictions are the same as those of benign models and uses them as input to a global shared model. With 1,000 clients on the MNIST dataset, model accuracy is greater than 80% with less than 40 malicious clients.

vi) **Gradient Aggregation Agent (GAA)** [38] on the server receives the updated model inputs from clients, and based on this, the server assigns a credit score to each client, which determines the weight of each client. Based on these weights, global model parameters are computed. Client with high scores indicates high-performing clients. The validation dataset helps in identifying benign or poisoned clients. Model accuracy with GAA is 80% when the Byzantine ratio is 0.5, which is near 84.5% when the system is under no attack.

vii) **Attested-FL** [39] uses various methods to determine if a client is benign or not. Firstly, remove clients whose model looks untrained using their history. Secondly, the server calculates the cosine similarity of successive local updates and removes an abnormal client’s input that is not exhibiting correlation over time. Finally, with the help of the validation dataset, the server identifies the effect of model updates on the rate of error, and clients with a high negative impact are removed. Clients with a smaller error rate difference in successive iterations are treated as reliable clients.

viii) **DnC** [1] picks up a subset of a few indices from all dimensions of input updates and sorts each dimension. The dimension-wise mean is computed to get a centered subsampled set. Then compute the centered gradient's projections along with the top right singular eigenvector, evaluate the outlier score vector, and eliminate the gradient with a high score. All remaining gradients are considered valid and averaged. DnC increases model accuracy under Adaptive attack, Fang, LIE, AGR-tailored, and AGR-agnostic attacks on the CIFAR10 and MINST datasets.

ix) **FLCert** [40] trains the global model on multiple groups, which consist of multiple clients. The grouping strategies FLCert-P and FLCert-D are proposed, utilizing random grouping and hashing based on client IDs, respectively. Based on the majority vote, the labels of the test inputs are predicted. If the number of attackers is controlled, then the output of the global model will not change under any attack. FLCert effectively defends against poisoning attacks but cannot identify attackers.

III. Results and Discussion

A. Comparative analysis of Untargeted attacks

Different model poisoning attack methods are studied and analyzed based on several parameters, like the percentage of attackers, the machine learning model used, the dataset used, and the effectiveness of the attack, as mentioned in Table 1. The impact of the attack is illustrated by comparing the model's accuracy before and after the attack.

Table 1: Untargeted Model Poisoning Attacks

Attack Method	#Clients/ Attackers	ML Model/Dataset	Attack Effectiveness
LIE [1][3]	51, 10-12	FC/MNIST, CNN/CIFAR10	MNIST→ Krum (88.6% vs 76.2), Trimmed-mean-(96.2% vs 89.8%), Bulyan (95.4% vs 86.2%), MKrum (96.1% vs 90%) CIFAR10→Krum (59.60% vs 35.50%), Trimmed-mean (75.50% vs 47.10%), Bulyan (75% vs 38.60%), MKrum (75.40% vs 51.80%)
Sign-flipping [18]	70, 20	DNN/MNIST	Mean→SGD (97.0% vs 0.11%), BSGD (98.6% vs 0.16%), and SAGA (96.5% vs 0.12%) GeoMed→BSGD (98.0% vs 90.3%), SAGA (96.3% vs 86.4%)
Additive Noise [17]	70, 20	CNN/MNIST	Mean→SGD/BSGD (98.6% vs 36.3%), SAGA (96.5% vs 14.5%) GeoMed→SGD (92.5% vs 92.3%), BSGD (98.0% vs 98.0%), SAGA (96.3% vs 96.4%)
Fang [1] [4]	100, 20	DNN, AlexNet, FC, CNN, CIFAR10, MNIST, FEMNIST, Purchase,	MNIST/DNN→Krum (89% vs 25%), LR (86% vs 28%) CIFAR10/AlexNet→Krum (53.5% vs 31.7%). Purchase/FC→Bulyan (91.3% vs 70.4%). MNIST/DNN →Trimmed-mean (94% vs 86%), Median (87% vs 81%) Purchase/FC→Median (87.4% vs 87.2%). CIFAR10/VGG11→Mkrum (75.4% vs 66.9%).
AGR-tailored and AGR-agnostic (Min-Max and Min-Sum attack) [1]	50-100, 10-20	Alexnet/CIFAR10, VGG11/CIFAR10, CNN/FEMNIST, FC/MNIST, FC/Purchase	CIFAR10/AlexNet→ Bulyan AGR-tailored attack (66.9% vs 21.3%), AGR-agnostic attack: Min-Sum (66.9% vs 22.4%), CIFAR10/AlexNet → Trimmed-Mean AGR-tailored attack (67.7% vs 21.9%), AGR-agnostic attack: Min-Max (67.7% vs 26.1%), For Cross-device setting: CIFAR10/AlexNet→ Krum (53.9% vs 44.4%),

			MKrum (64.5% vs 49.9%),
MPAF [25]	1000, 100 (Fake Clients)	CNN/MNIST, CNN/FEMNIST, FC/Purchase	Purchase/FC→Trimmed-mean (85% vs 68% for 10% Fake clients), (85% vs 51% for 25% Fake clients),
Voting Attack [29]	100, 50	CNN/CIFAR10, CNN/FEMNIST	CIFAR10→ FedAvg (51% vs 30%). FEMNIST→ FedAvg (90% vs 85%).
DNY-OPT [41]	24, 8	CNN/FEMNIST, Alexnet/CIFAR10	FEMNIST→ FedDet (52% vs 41%)
RL-based attack [43]	100, 10	RL, MNIST, Fashion MNIST, EMNIST, and CIFAR-10	MNIST→Krum 50%
Poisoned FL [44]	200, 1-40	MNIST, FEMNIST, Purchase, CIFAR-10 and FEMNIST	Purchase→ TrMean (91.65% vs 46.14%) FEMNIST→FLDetector `(69.98% vs 8.75%)
VGAE-MP attack [47]	100-300, 100	MNIST, FEMNIST and CIFAR-10	FEMNIST→32.3% MNIST→27.4% CIFAR-10→24.9%

FC: Fully Connected Network, CNN: Convolution Neural Network, DNN: Deep neural networks,

LR: Multi-class logistic regression, SGD: Stochastic Gradient Descent, SAGA: Stochastic Average Gradient Algorithm, BSGD: Byzantine attack resilient distributed SGD, RL-Reinforcement learning,

VGAE-MP Adversarial variational graph autoencoder model poisoning attack

The impact of the LIE [3] attack is more significant with Krum [11], Trimmed-mean [5], and Bulyan [13] aggregation methods on CIFAR10 [14] as compared to MNIST [15], but it is weak against MKrum [11]. Adding noise helps to preserve data privacy, but with more noise, model performance decreases [17]. Additive noise attack is effective against Mean aggregation, but GeoMed [17] and Spectral Anomaly detection methods are robust against Additive noise attack. Sign-flipping [18] attack is strong against Mean aggregation, but it is weak against GeoMed. Sign-flipping attack is stronger than Additive noise attack [17].

With AGR-tailored attack and AGR-agnostic attack, model accuracy drop is high on Bulyan AGR with CIFAR10 as compared to Fang [4] and LIE [3] (Fang-11.8%, LIE-30.0%, AGR-tailored - 45.6% and Min-Sum AGR-agnostic - 44.5%). The same is true for Trimmed-mean aggregation, but their impact is low in cross-device settings and more in cross-silo settings. The attack impact reduces with Krum and MKrum aggregation by 9.5% and 14.6%, respectively [1]. Hence, for Bulyan, Trimmed-Mean, and Krum AGR, AGR-agnostic attacks reduce model accuracy much more than Fang and LIE attacks.

For MPAF [25], the test accuracy drop is very high for Trimmed-mean (32% to 49%) with the Purchase dataset as compared to Gaussian noise attack (4%). Under norm clipping, when the norm threshold is around 100, the test accuracy drop is 17% (85% vs 68%), but as the norm threshold decreases, the effect of an attack is small since more fake local model updates are clipped. Without knowing the hyperparameters of FL, it reduces model accuracy significantly under Trimmed-mean and norm-clipping defense. MPAF does not work with defense methods against targeted poisoning attacks.

The impact of the Voting attack on the model test accuracy on the CIFAR10 dataset with FedAvg AGR is high i.e., 21% in the presence of 20% malicious clients whereas with FEMNIST it is only 5%. The decrease in model accuracy with Fang, LIE, rescaling attack, Min-sum, and Min-Max attacks is negligible when model compression is applied using SignSGD based on quantization [29].

Under DNY-OPT attack, FedDet is more robust than the Trimmed-Mean aggregation method and outperforms Krum aggregation [41]. Under a Poisoned FL [44] attack, it is harder for an attacker to compromise genuine clients or inject fake clients into such a cross-silo FL system. The malicious models are easily detected and eliminated under VGAE-MP attack [47].

B. Comparative analysis of Defense methods against Untargeted attacks

Various defense or aggregation methods have been studied and analyzed based on different parameters, such as the technique used, type of attack defended, machine learning model used, the dataset used, and performance evaluation. The performance of the AGR is assessed based on the effect of the attack on the model accuracy, comparing values before and after the attack. Table 2 represents an analysis of Statistical-based aggregation methods; Table 3 represents an analysis of criteria-based aggregation methods.

Krum is a strong aggregation method against Min-Max AGR-Agnostic attack as the model accuracy decreases only by 0.7%, but weak against AGR-tailored attack as the accuracy drops is 33.9% for MNIST dataset [1]. MKrum is a strong aggregation method against LIE attack (accuracy drops by 3.3%) and weak against AGR-tailored (accuracy drops by 18.6% for MNIST and 36.8 % for CIFAR10 dataset) and Min-Max AGR agnostic attack (accuracy drops by 31.7% for CIFAR10 dataset) [1].

Trimmed Mean aggregation is strong against Fang and LIE attacks as the attack impact is only 1.6% and 1.9% respectively for the Purchase dataset [1][5]. Under an AGR-agnostic attack, the performance of Trimmed Mean is better as compared to an AGR-tailored attack (AGR - updates) [1]. The Median is less affected by outliers and performs better than the mean, but its performance is poor under an AGR-tailored attack [13]. Under the AGR-agnostic attack, Median performs better than AGR-tailored attack (AGR - updates) [1].

Bulyan aggregation is strong against Min-Max AGR agnostic attack (accuracy drops by 4.8%) but weak against AGR tailored attack (accuracy drops by 8.2%) and LIE attacks (accuracy drops by 9.2%) for MNIST dataset. It is also weak for CIFAR10 dataset as the model accuracy drops by 53% for AGR-tailored attack and 44.5% for AGR -agnostic attack. Thus, after the AGR-tailored attack on MNIST, the performance of Bulyan is better but not with the CIFAR10 dataset under the AGR-tailored (AGR-updates) attacks [1]. Bulyan performs Krum multiple times, making it non-scalable [1][2]. Adaptive Federated Averaging (AFA) performs better with the AGR-agnostic attack on MNIST, but not with the AGR-tailored attack (AGR - updates) on FEMNIST [1].

The efficiency of Krum, MKrum, and Bulyan algorithms is limited by the need for the server to calculate pairwise distances among each client, which becomes computationally expensive with many clients.

Table 2. Defense Methods: Statistical-based Aggregation Methods

Defense Method	Aggregation Strategy	Attack Mitigated	ML Model /Dataset	Performance Evaluation
Krum [1][11]	Euclidean Distance	Byzantine, Backdoor, AGR-Agnostic Min-Max	ANN, FC and CNN MNIST, FEMNIST	MNIST/FC → Min-Max attack (88.6% vs 87.9%) FEMNIST/CNN → AGR-tailored attack (agr-only) (69.3% vs 66.4%) MNIST/FC → AGR-tailored (agr-updates) attack (88.6% vs 54.7%) FEMNIST/ CNN AGR-tailored (agr-updates) attack (69.3% vs 39.3%)
MKrum [11]	Select *k* input gradients using Krum and their average	Byzantine, Backdoor, LIE	ANN, FC and CNN CIFAR10, MNIST, FEMNIST	MNIST/FC → LIE attack (96.1% vs 92.8%) MNIST/FC → AGR-tailored attack (agr-updates) (96.1% vs 77.5%) CIFAR10/AlexNet→AGR-tailored attack (agr-updates) (67.6% vs 30.8%)
Trimmed Mean [5]	Remove *m* largest and smallest gradients after sorting each dimension and avg	Fang, LIE	FC/Purchase/ MNIST, Alexnet/ CIFAR 10	Purchase/FC→Fang attack (92.0% to 90.4%) Purchase/FC→LIE attack (92.0% to 90.1%) MNIST/FC → Min-Max AGR-agnostic attack (96.2% vs 84.6%) CIFAR10/Alexnet → AGR-tailored (agr-updates) attack (67.7% to 21.9%)

Median [13]	Median of values of each dimension for all input gradients.	Fang and LIE -FC/Purchase [1]	FC/Purchase/ MNIST, Alexnet/ CIFAR10	MNIST/FC → AGR-agnostic attack (93.2% vs 89.8%) CIFAR10/Alexnet → AGR-tailored attack (agr - updates) (65.5% vs 24.6%) FEMNIST/CNN → AGR-tailored attack (agr - updates) (77.1% vs 46.9%)
Bulyan [2]	Selects k gradients same as MKrum followed by Trimmed-mean	Min-Max attack	FC/MNIST, CNN/ CIFAR10	MNIST/FC → Min-Max AGR-agnostic attack (updates-only) (95.4% vs 90.4%) MNIST/FC → AGR-tailored (agr-updates) (95.4% vs 87.2%) MNIST/FC → LIE (updates-only) (95.4% vs 86.2%) CIFAR10/ VGG11→ AGR-tailored (agr-updates) attack (75% vs 22%)
AFA [32]	Cosine similarity and range function (mean, median, and standard deviation)	Fang – MNIST/FC	FC/MNIST, CNN/ FEMNIST	MNIST/FC → AGR-agnostic attack (96.5% to 94.9%) FEMNIST/ CNN → AGR-tailored (AGR - updates) attack (84.6% Vs 7.6%)
RFA [33]	Geometric median, computed using the Weiszfeld algorithm	-	CNN/FMNIST, Shakespeare/ LSTM, Sent140/LR	FEMNIST/ FedAvg → Data poisoning (52.8% vs. 41.2%) FEMNIST/ FedAvg → No Data poisoning (64.3% vs. 62.9%) Sent140 / FedAvg → No Data poisoning (65.0% vs. 64.7%)
LoMAR [45]	Two-phase defense algorithm	Stealthy model poisoning attack and Label flipping attack	MNIST, KDD, Amazon and VGGface2	LoMAR / Amazon → Label Flipping attack (96.0% to 98.8%.)
FLAIR [46]	Reputation-based scheme and FL	Directed deviation attack	MNIST, CIFAR-10, FEMNIST and Shakespeare	MNIST/DNN→Full Krum attack (92.52% vs 87.73%) MNIST/DNN→Full Trim attack (92.52% vs 90.55%) FEMNIST/DNN→Full Krum attack (83.58% vs 80.19%) FEMINIST/DNN→Full Trim attack (83.58% vs 82.51%) CIFAR10/AlexNet→Full Krum attack (66.92% vs 61.26%)

Full-Krum Attack is tailored to deceive the Krum algorithm (and transferable to Bulyan).

Full-Trim Attack is tailored to deceive the Trimmed Mean aggregation (and transferable to the Median).

RFA is more robust than the median, norm clipping, and trimmed mean, but MKrum is more robust than RFA. The communication cost of RFA is higher due to multiple Weiszfeld iterations, although its accuracy is comparable to FedAvg [33]. FLAIR [46] is more robust to Full-Trim attack (accuracy drops by 1.97%) and Full-Krum attack (accuracy drops by 5.66%) for MNIST and FMNIST datasets, but for CIFAR10 dataset its performance is poor (accuracy is 66.92% without attack). It offers a malicious client percentage of 45%, which gives byzantine robustness. The model is limited to the synchronous setting without gradient encryption, which is less relevant in cross-device because of its computational overhead.

As shown in Table 3, the Spectral Anomaly Detection method performs better against additive noise attacks and sign-flipping attacks compared to backdoor attacks. FLTrust converges quickly, similar to FedAvg, and provides better model accuracy with 60% malicious clients [37]. However, a clean validation dataset is required on the server. RobustFed is robust against label flipping, noisy data, and Byzantine attacks on the MNIST and FEMNIST datasets, but not on the CIFAR10 dataset [35]. SparseFed outperforms other defense methods, but Trimmed-mean performs better with 20% attackers on the CIFAR dataset [36].

Table 3. Defense Methods: Criteria-based Aggregation Methods

Defense Method	Aggregation Strategy	Attack Mitigated	ML Model Type/Dataset	Performance Evaluation
Spectral Anomaly	The mean value of all the	Additive noise attack, Sign-	LR/MNIST, CNN/	MNIST→ Sign-flipping attack (f1 score: 0.97), Backdoor attack (f1 score: 1),

Detection [17]	reconstruction errors	flipping attacks, Backdoor attack	FEMNIST, RNN/ Sentiment140	additive noise attack (f1 score: 1) FEMNIST → Sign-flipping attack (f1 score: 0.99), Backdoor attack (f1 score: 0.87), additive noise attack (f1 score: 1) Sentiment140→ Sign-flipping attack (f1 score: 1), Backdoor attack (f1 score: 0.93), additive noise attack (f1 score: 1)
FLTrust [34]	cosine similarity score clipped using ReLU and then find an average of weights	Fang attack, Adaptive attack	CNN/MNIST, ResNet20/ FEMNIST, ResNet20/ CIFAR10,	MNIST/CNN→Krum attack: FLTrust (96% vs 96%), FedAvg (96% vs 90%), Krum (90% vs 10%), Trim-Mean/Median (94% vs 93%) CIFAR10/ ResNet20→Krum attack: FLTrust (82% vs 82%), FedAvg (84% vs 76%), Krum (46% vs 10%), Trim-Mean (76% vs 48%), Median (75% vs 36%),
RobustFed [35]	Reliability score	Byzantine attack, Data poisoning attack- label flipping and noisy data	CNN, VGG-11, MNIST, FMNIST, CIFAR10	MNIST→ Byzantine attack (99.32% vs 98.34%), label flipping (99.32% vs 96.34%), noisy data attack (99.32% vs 96.82%) CIFAR10→ Byzantine attack (69.75% vs 54.67%), label flipping (99.32% vs 51.10%)
Ensemble FL [4]	Train model on k groups of clients. Removes model that negatively impacts ERR, LFR and their union (ERR +LFR)	Fang attack	DNN, ResNet20 CH-MNIST-, Multiclass LR, MNIST, and FMNIST, Breast Cancer Wisconsin	MNIST/DNN→ Trimmed mean with LFR (88%), Trimmed mean with ERR (79%) MNIST/LR→ Krum with union (LFR+ ERR) (52%), Krum with LFR (42%),
GAA [38]	Credit points and each client's updated parameters	Static attack, Randomized attack, Byzantine attack	CNN, MNIST, CIFAR-10, Yelp reviews	MNIST/CNN→Randomized attacks (96.4% vs 90%) Yelp/CNN→Randomized attacks (84.5% to 83%)
DnC [1]	singular feature vector's projection in the direction of the gradient features and average	Fang, LIE Byzantine attack, AGR-agnostic, and AGR-tailored attack, Adaptive attack	CNN- AlexNet, VGG11, FC, MNIST, CIFAR10, FEMNIST, Purchase	MNIST/FC→ Adaptive attack (96.2% vs 94.3%) CIFAR10/Alexnet→ AGR-tailored attack (67.6% vs 32.5%) in cross-silo FL. CIFAR10/Alexnet→ AGR tailored attack (64.6% vs. 61.2%) in cross-device FL.
FLCert [40]	Ensemble learning and majority voting	Untargeted attacks	CNN, MNIST, DNN, Human Activity Recognition (HAR)	MNIST→ 88% vs 83% (with 1% malicious clients)

Krum attack: Untargeted local model poisoning attack optimized for the Krum aggregation rule, ERR: Error Rate-based

Rejection, LFR: Loss Function based Rejection, LR: Multi-class logistic regression, GAA: Gradient Aggregation Agent

In Ensemble learning [4], LFR is weaker than the union of ERR and LFR when Krum AGR is used, and ERR defense is weak compared to LFR with Trimmed mean. Therefore, none of the methods outperforms all AGR. This approach assumes a small percentage of malicious clients among all clients. It maintains the model accuracy greater than 80% with malicious clients less than 4% on MNIST. For more than 4% of malicious clients, this defense is weak. Additionally, maintaining validation datasets at the server may not be feasible with data-sensitive applications.

For Gradient Aggregation Agent (GAA) [38], the model accuracy drops by 6.4% on the MNIST dataset and by 1.5% on the Yelp dataset, but under randomized attack for the MNIST dataset, with Byzantine ratio(β) =26/50, it assigns rewards to both Byzantine and benign workers, so they cannot be distinguished. Also, GAA has computation overheads.

In AttestedFL [39] AGR, the model accuracy increases by an average of 50%, and the model convergence is increased by 12% to 58%. AttestedFL has no upper bound on the malicious nodes and has a high success rate, but it has more computation complexity.

With DnC, the maximum impact of the attack is limited to 12.7% and the highest accuracy achieved by the global model is 73.9%, but the model accuracy in cross-device FL is less [1]. In comparison, Krum, which is considered the most robust among existing aggregation algorithms, achieves a maximum accuracy of only 66.4%.

In FLCert [40], the certified accuracy of the ensemble FedAvg drops to 0% when up to 6.1% and 30% of the clients are malicious in the MNIST and HAR datasets, respectively. It does not require any clean training data and can train too many global models in practice, but it does not provide a provable security guarantee against fake clients. It is unable to identify the attackers [42].

IV. Conclusion

Among the aggregation methods studied, FedAvg, Krum and MKrum are popular choices as they are Byzantine-robust, efficient for execution, and relatively easy to implement compared to other aggregation methods. However, these methods are not without their limitations. They are vulnerable to certain types of Byzantine attacks, and new attack methods to poison the FL model are constantly being developed.

The need for stronger aggregation methods is evident for several reasons. Firstly, the existing aggregation methods are not perfect, and there is a demand for more robust defense mechanisms against evolving attack techniques. Secondly, as the FL is being used in sensitive applications, a new robust aggregation method is in demand which can handle various types of attacks.

Finally, the computational complexity of the defense methods is an important consideration. There is a demand for strong defense methods that can handle model poisoning attacks with less computational overhead. As a result, more advanced aggregation techniques are required to address these challenges and ensure the security and reliability of Federated Learning systems.

Acknowledgment

The research has not received funding from any Agency.

References

[1] Shejwalkar, Virat and A. Houmansadr. (2021). Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning, Network and Distributed Systems Security (NDSS) Symposium, 2021: 21-25, Available from: https://dx.doi.org/10.14722/ndss.2021.24498

[2] El Mahdi El Mhamdi, Rachid Guerraoui, and S´ebastien Rouault. (2018). The hidden vulnerability of distributed learning in byzantine.In International Conference on Machine Learning, pages 3518–3527.

[3] Moran Baruch, Baruch Gilad, and Yoav Goldberg. (2019). A little is enough: Circumventing defenses for distributed learning. Advances in Neural Information Processing Systems.

[4] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. (2020). Local model poisoning attacks to byzantine-robust federated learning.In 29th USENIX Security Symposium (USENIX Security 20), Boston, MA, USENIX Association, 1605–1622.

[5] Cong Xie, OluwasanmiKoyejo, and Indranil Gupta. (2018). Generalized byzantine-tolerant sgd. arXiv preprint:1802.10116.

[6] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. (2019). Analyzing federated learning through an adversarial lens. In International Conference on Machine Learning, pages 634–643.

[7] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. (2018). How to backdoor federated learning. arXiv preprint arXiv:1807.00459.

[8] Cong Xie, SanmiKoyejo, and Indranil Gupta. (2019). Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation.arXiv preprint arXiv:1903.03936.

[9] Matthew Jagielski, Aline Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. (2018). Manipulating machine learning: Poisoning attacks and countermeasures against regression learning. 39th IEEE Symposium on Security and Privacy.

[10] Luis Mu˜noz-Gonz´alez, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. (2017). Towards poisoning of deep learning algorithms with back-gradient optimization. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pages 27–38. ACM.

[11] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems, pages 119–129.

[12] Junchuan Liang, Rong Wang. A Survey on Federated Learning PoisoningAttacks and Defenses (2022), PREPRINT (Version 1) available at Research Square, https://doi.org/10.21203/rs.3.rs-1900743/v1

[13] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the 35th International Conference on Machine Learning.

[14] Alex Krizhevsky and Geoffrey Hinton. (2009). Learning multiple layers of features from tiny images..

[15] Yann LeCun, L´eonBottou, Yoshua Bengio, Patrick Haffner, et al. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324.

[16] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. (2017). Emnist: Extending mnist to handwritten letters. In International Joint Conference on Neural Networks (IJCNN), pages 2921–2926. IEEE.

[17] Li, S., Cheng, Y., Wang, W., Liu, Y., & Chen, T. (2020). Learning to Detect Malicious Clients for Robust Federated Learning. arxiv, abs/2002.00211.

[18] Zhaoxian Wu, Qing Ling, Tianyi Chen, and et al. (2019). Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. arXiv preprint arXiv:1912.12716v1.

[19] Liping Li, Wei Xu, Tianyi Chen, and et al. (2019). RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In Proceedings of AAAI.

[20] Li, Shenghui& Ngai, Cheuk & Voigt, Thiemo. (2022). An Experimental Study of Byzantine-Robust Aggregation Schemes in Federated Learning. 10.36227/techrxiv.19560325.v2.

[21] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. (2019). Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963.

[22] Lin, J., Du, M., Liu, J. :Free-riders in federated learning. (2019). Attacks and defenses. arXiv preprint arXiv:1911.12560.

[23] Costa, G., Pinelli, F., Soderi, S., Tolomei, G. (2021). Covert channel attack to federated learning systems. arXiv preprint arXiv:2104.10561.

[24] Xie, C., Koyejo, O., Gupta, I. (2020). Fall of empires: Breaking byzantine to lerant sgd by inner product manipulation. In: Uncertainty in Artificial Intelligence, pp. 261–270, PMLR.

[25] Cao, Xiaoyu, and Gong, Neil Zhenqiang. (2022). MPAF: Model Poisoning Attacks to Federated Learning Based on Fake Clients. Retrieved from https://par.nsf.gov/biblio/10352089. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. Web. doi:10.1109/CVPRW56347.2022.00383.

[26] Noxplayer.: The perfect android emulator to play mobile games on pc.https://www.bignox.com/.

[27] “The world’s first cloud-based android gaming platform.” https://www.bluestacks.com/.

[28] “Android-x86 run android on your pc.” https://www.androidx86.org/.

[29] X. Ma, L. Wei, B. Zhang, Y. Wang, C. Zhang and Y. Li. (2022). A Voting-Based Poisoning Attack to Federated Learning with Quantization, 5th International Conference on Hot Information-Centric Networking (HotICN), Guangzhou, China, pp. 125-131.

[30] Demirel, Ilker & Yildirim, Yigit & Tekin, Cem. (2022). Federated Multi-Armed Bandits Under Byzantine Attacks..

[31] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Ag¨uera y Arcas. (2017). Communication efficient learning of deep networks from decentralized data. In AISTATS.

[32] Luis Mu˜noz-Gonz´alez, Kenneth T Co, and Emil C Lupu. (2019). Byzantine robust federated machine learning through adaptive model averaging. arXiv preprint arXiv:1909.05125.

[33] K. Pillutla, S. M. Kakade, and Z. Harchaoui. (2023). Robust Aggregation for Federated Learning, IEEE Transactions on Signal Processing.

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

[35] Farnaz Tahmasebian, Jian Lou, and Li Xiong. (2021). Robustfed: a truth inference approach for robust federated learning. arXiv preprint arXiv:2107.08402.

[36] Panda, A., Mahloujifar, S., Nitin Bhagoji, A., Chakraborty, S. &amp; Mittal, P. (2022). SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification. Proceedings of 25th International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research 151:7587-7624

[37] Huili Chen and Farinaz Koushanfar. (2023). Tutorial: Toward Robust Deep Learning against Poisoning Attacks. ACM Trans. Embed. Comput. Syst. 22, 3, Article 42.

[38] Pan, X., Zhang, M., Wu, D., Xiao, Q., Ji, S., Yang, Z. (2020) Justinian's gaavernor: Robust distributed learning with gradient aggregation agent. In: 29th USENIX Security Symposium, pp. 1641–1658.

[39] Mallah, R.A., Lopez, D., Farooq, B. (2021). Untargeted poisoning attack detection in federated learning via behavior attestation. arXiv preprint arXiv:2101.10904.

[40] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong. (2022). FLCert: Provably secure federated learning against poisoning attacks. IEEE Trans. Inf. Forensics Security, vol. 17, pp. 3691–3705.

[41] Yang, Han, Dongbing Gu, and Jianhua He. (2024). A Robust and Efficient Federated Learning Algorithm Against Adaptive Model Poisoning Attacks. IEEE Internet of Things Journal.

[42] G. Xia, J. Chen, C. Yu and J. Ma (2023). Poisoning Attacks in Federated Learning: A Survey. IEEE Access, vol. 11, pp. 10708-10722.

[43] Li, Henger, Xiaolin Sun, and Zizhan Zheng. (2022). Learning to attack federated learning: A model-based reinforcement learning attack framework. Advances in Neural Information Processing Systems 35: 35007-35020.

[44] Xie, Yueqi, Minghong Fang, and Neil Zhenqiang Gong. (2024). PoisonedFL: Model Poisoning Attacks to Federated Learning via Multi-Round Consistency. arXiv preprint arXiv:2404.15611.

[45] Li, Xingyu, Zhe Qu, Shangqing Zhao, Bo Tang, Zhuo Lu, and Yao Liu. (2021). Lomar: A local defense against poisoning attack on federated learning. IEEE Transactions on Dependable and Secure Computing 20, no. 1: 437-450.

[46] Sharma, Atul, Wei Chen, Joshua Zhao, Qiang Qiu, Saurabh Bagchi, and Somali Chaterji. (2023) Flair: Defense against model poisoning attack in federated learning. In Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security, pp. 553-566.

[47] Li, Kai, Xin Yuan, Jingjing Zheng, Wei Ni, Falko Dressler, and Abbas Jamalipour. (2024). Leverage Variational Graph Representation for Model Poisoning on Federated Learning. IEEE Transactions on Neural Networks and Learning Systems.