

# A Survey of Security Issues in Federated Learning

1<sup>st</sup> Yunhao Feng

National University of Defense and Technology  
Changsha, China  
fengyunhaonudt@nudt.edu.cn

2<sup>nd</sup> Yinjian Hou

National University of Defense and Technology  
Changsha, China  
houyinjian18@nudt.edu.cn

**Abstract**—As people’s awareness of the importance of personal privacy protection has grown, there has been a surge of interest in federated learning, which is a machine learning paradigm that enables training without requiring access to users’ private data.

## I. Introduction

The rapid development of digital technology has made the diversification, informationization, and diversity of digital data the main topics of the current era. Meanwhile, deep learning (DL) has demonstrated tremendous success in multiple fields, including computer vision, natural language processing, and graphic networks. Clearly, using diverse data in deep learning models can effectively improve their ability. However, there is also a growing interest in data privacy protection, such as the General Data Protection Regulation (GDPR) [1]. On the other hand, data sources may encounter the challenge of distributed storage, as is the case with data from mobile smart devices or Internet of Things (IoT) scenarios [2], [3]. Therefore, utilizing these data to train models requires overcoming limitations related to distribution and privacy [4].

To solve these problems, federated learning (FL) is a machine learning paradigm proposed as a possible response to these challenges [5]. FL enables collaborative model building among distributed members while ensuring sensitive data remains within each participant’s control [6]. Specifically, federated learning allows two or more participants to collaboratively train a shared global DL model while keeping their training datasets locally. Each participant trains the shared model on its own training data and exchanges and updates model parameters with other participants. Federated learning can improve the training speed and the performance of the shared model while protecting privacy of the participants’ training datasets [7]. Thus, it is a promising technique for the scenarios where the training data is sensitive (e.g., medical records, personally identifiable information, etc.) [8], [9].

Federated learning can be classified based on whether the participating datasets are the same, resulting in two types: homogeneous federated learning and heterogeneous federated learning [10], [11]. In homogeneous federated learning, all participants have datasets with the same characteristics and data distribution, whereas in heterogeneous federated learning, participants’ datasets may differ in their characteristics and data distribution. The second

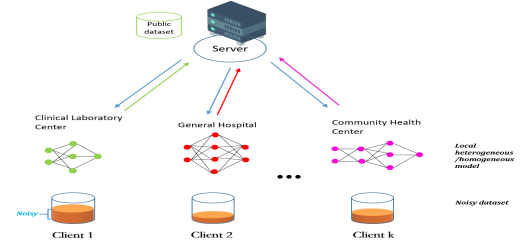


Fig. 1. A schematic of federated learning.

classification of federated learning is based on whether the models involved are the same, resulting in two types: horizontal federated learning and vertical federated learning [12], [13]. In horizontal federated learning, all participants have the same model architecture, but may have different local data [14], while in vertical federated learning, each participant has a different model architecture but they collaborate on processing the same set of data together [15]. The third way to classify federated learning is based on the type of task involved, resulting in several types such as federated learning for clustering [16], [17], federated learning for classification [18], [19], federated learning for regression [20], among others. The fourth way to classify federated learning is based on the optimization approach used between the participants, resulting in several types such as federated averaging [21], [22], federated learning optimization, federated meta-learning [23], and so on.

Federated learning methods currently face significant challenges related to their robustness. This article focuses on three main attacks, including backdoor attacks [24], [25], [26], [27], [28], adversarial attacks [31], [32], [33], [34], and Byzantine attacks [29], [30]. A backdoor attack involves a malicious participant in the federated learning process adding a backdoor to the model being trained, which can be triggered by a specific input pattern, allowing the attacker to control the output of the model in a targeted way. Adversarial attacks, on the other hand, entail adding small, carefully crafted perturbations to the input data to deceive the model and cause it to make incorrect predictions [31], [32].

And adversarial attacks can occur in federated learning when a malicious participant intentionally sends adversarial examples to the central server in an attempt to

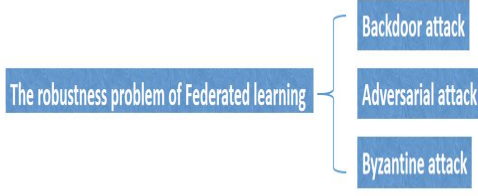


Fig. 2. The robust threat to federal learning

bias the model towards their own interests. This can be particularly problematic in applications such as personalized advertising or credit scoring, where the malicious participant may be motivated to gain an unfair advantage. Finally, Byzantine attacks involve one or more malicious participants in the federated learning process sending incorrect or misleading updates to the central server to disrupt the training process [35].

While federated learning can be vulnerable to certain types of attacks, there are techniques and approaches that can be used to improve the robustness and security of the process. It is important to carefully consider these issues when designing and implementing federated learning systems [38], [39]. For instance, knowledge distillation is a technique that can mitigate backdoor attacks by training a smaller [36], distilled model using the output of the original model as the target labels. This can help remove any backdoor triggers that may have been added to the original model, as the smaller model won't be able to identify them. Another technique to mitigate backdoor attacks is model erasure [44], where the model is trained to ignore specific input patterns that may be associated with the backdoor.

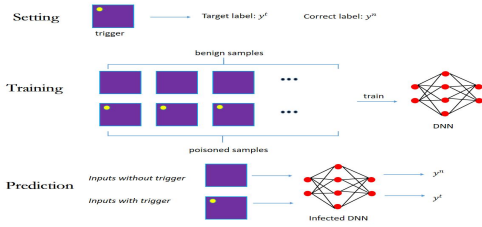


Fig. 3. Backdoor Attack

Adversarial training is a technique that involves explicitly training the model to resist adversarial examples by adding adversarial perturbations to the training data [31], [32]. This can improve the model's ability to detect and resist adversarial attacks in federated learning settings. Clustering can be used to identify malicious clients in federated learning systems subject to Byzantine attacks [35], [36]. The idea is to group participating clients based on the similarity of their updates, and to identify any

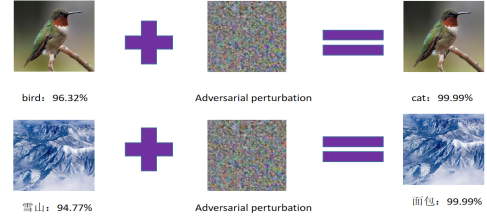


Fig. 4. Adversarial Attack

clients whose updates are significantly different from the others. These clients can then be excluded from the training process, or their updates can be treated with greater suspicion to minimize the impact of their malicious behavior [37]. This paper provides an overview

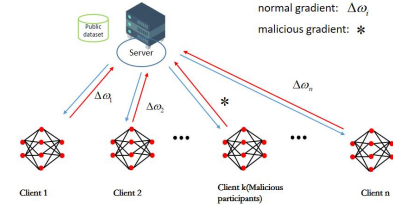


Fig. 5. Byzantine Attack

of methods to increase the robustness of federated learning models, with the aim of enhancing the credibility and security of federated learning. While previous work has addressed the security of federated learning [39], [40], [41], [42], [43], it has primarily focused on privacy leakage or backdoor attacks, with relatively few studies and reports on adversarial attacks. Building on prior work, this paper summarizes the attacks and defense methods of adversarial, backdoor, and Byzantine attacks in federated learning. A new classification method is proposed, supplementing the deficiencies of previous work on adversarial attacks. Moreover, this paper investigates a multi-level defense system against these attacks, and identifies open problems and future research directions for improving the robustness of federated learning.

## II. Thraet Model

Prior to delving into the details of the threats to federal learning, it is essential to establish the connections between these threats based on different criteria. Specifically, we can categorize these threats into two main stages: the training phase and the inference phase. Additionally, we can differentiate between untargeted attacks and targeted attacks based on whether a specific target is present or not. [38], [39], [40], [41].

### A. Training Phase and Inference Phase

1) Training Phase: Attacks that occur during the model training process are intended to either disrupt or impact the federated learning model itself. Backdoors are inserted into the model during the training phase to influence the resulting model outcomes [45], [46]. On the other hand, Byzantine attacks disrupt the convergence of the model by utilizing malicious clients or servers [29].

2) Inference Phase: Attacks that occur during the reasoning phase are typically intended to alter the model's reasoning outcomes and deceive it into generating incorrect outputs [47]. During the training stage, backdoor attacks involve the insertion of a backdoor into the model, whereas input deception models with triggers are utilized during the reasoning stage to cause the model to generate incorrect results. Adversarial attacks, on the other hand, leverage the model's vulnerability to disturbances and utilize samples with adversarial perturbations as input to the model, causing it to produce erroneous outcomes.

### B. Untargeted and Targeted

1) Untargeted attack: Untargeted attacks are designed to compromise the integrity of the target model in an arbitrary manner. Byzantine attack is one form of an untargeted attack that involves uploading malicious gradients to the server in an arbitrary manner, with the goal of causing the global model to fail [48], [49], [50], [51].

2) Targeted Attack: A targeted attack is executed with the aim of inducing the model to produce the target label specified by the adversary for specific testing examples, while keeping the testing error for other testing examples unaffected [51].

## III. Backdoor Attack

A backdoor attack on deep neural networks entails surreptitiously implanting a malicious backdoor within the model. This enables the model to function normally when processing benign inputs, but triggers a pre-defined malicious behavior when presented with a specific malicious trigger. The first neural backdoor in centralized settings can be traced back to 2013 [52], [53].

Due to the unique nature of federated learning, whereby the model is trained on individual clients, it is more susceptible to backdoor attacks compared to the general centralized training model. These types of attacks in federated learning can be divided into two categories based on the different stages at which the adversary inserts the backdoor into the training pipeline: data poisoning attacks and model poisoning attacks.

### A. Data Poisoning Attack

In data poisoning attacks, it is assumed that the adversary has full control over the training data collection process of compromised clients. Then the poisoned dataset typically consists of a combination of clean data with ground-truth labels and data with backdoor triggers that have targeted labels.

1) Invisible Poisoning: In this context, the term "invisible" denotes the user's ability to execute a backdoor attack on a sample without requiring any additional actions to be performed on the sample itself.

Label flipping is a widely recognized attack in centralized machine learning (ML), as demonstrated in previous research studies [54], [55]. In addition, it is also a suitable method for the federated learning (FL) scenario, given its adversarial goal and capabilities [56]. As shown in the picture, some of the samples labeled "dog" are flipped to "bird" and the samples of "cat" are flipped to "bread".

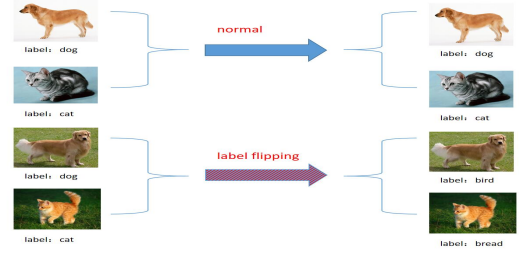


Fig. 6. Label Flipping

Nevertheless, the label flipping technique has its limitations since it necessitates modifying the label of the sample, making it less practical. Thus, an attack technique that is more covert and can deceive manual inspection would be more appealing in this scenario. A clean-label attack preserves the label of the poisoned data, and the manipulated image still appears to be a benign sample [57], [58]. This type of attack leverages feature collision, where the crafted poison examples continue to resemble the source instance in visual appearance, while being closer to the targeted instance in the latent feature space.

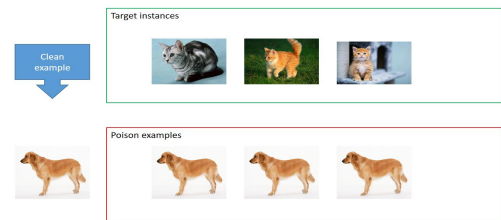


Fig. 7. Clean Label Attack

2) Visible Poisoning: Early methods of backdoor attacks in federated learning can rely on a single trigger, meaning that all corrupted clients inject the same trigger into their local training dataset. The triggers employed in this method are usually predetermined, such as a square located at redundant pixels in the image. During reasoning, the inserted trigger is used on a malicious client to activate the aggregation model [24], [27]. While the effectiveness of inserting the backdoor has been shown to be significant, the aforementioned approach merely trans-

fers backdoor attacks from centralized learning directly to federated learning, without fully leveraging the distributed nature of the latter. This is because the same trigger is embedded in all adversarial clients.

Xie et al. [59] proposed a distributed backdoor attack (DBA) that decomposes a global trigger pattern, similar to a centralized attack, into local patterns and embeds them into different malicious clients. Compared to traditional methods that insert the same trigger, DBA is more efficient and covert due to its hidden local trigger mode, making it easier to bypass robust aggregation rules.

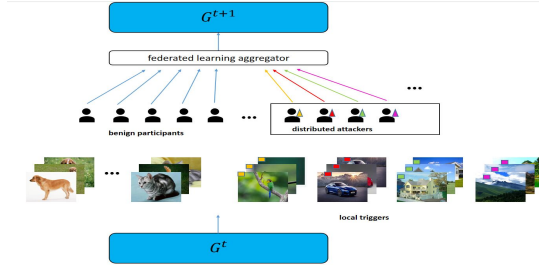


Fig. 8. Distributed Backdoor Attack

In order to enhance the effectiveness of data poisoning attacks, dynamic triggers have been proposed and applied. Salem et al [60] conducted a clear and systematic study on the feasibility of dynamic flip-flops, which can facilitate the backdoor attacks by generating antagonistic network algorithms to create triggers. This way, the same tags can be hijacked with different trigger patterns that share similar potential representations and positions. Et al [61] also conducted a direct investigation of dynamic triggers. These triggers can be flexibly produced during the physical attack phase, as they maintain their effectiveness even under substantial changes. The results of the study show that dynamically triggered backdoor attacks are more powerful, and they require new techniques to be defeated because they break the static trigger hypothesis of most current defense systems.

Despite the potential threat of data poisoning attacks in federated learning, they face many practical limitations due to the unique distributed characteristics of this approach [25]. This is because the data distribution and model aggregation steps in federated learning tend to neutralize most of the contributions of the backdoor model, leading to rapid forgetting of the backdoor by the global model. In light of this situation, Wang et al. [25] proposed selecting poisoning samples from edge data to reduce the forgetting effect caused by model updating. Dai et al. [64] proposed a new backdoor attack called Champeleon, which enables attackers to create more persistent visual backdoors by adapting to peer-to-peer images. The durability of the backdoor largely depends on the existence of two types of identical benign images that are closely related to the toxic image: 1) interferers, which are images that share the same original tag as the toxic

image, and 2) facilitators, which are images with the target back tag. Interferers can cause update conflicts between toxic updates and benign updates, which may reduce the accuracy of the backdoor. Conversely, facilitators can help reintroduce backdoor information into the federated learning model and mitigate the catastrophic forgetting effect after the attacker leaves the training process. Inspired by these observations, Champeleon is designed to amplify these effects and enhance the durability of the backdoor.

## B. Model Poisoning Attack

To address the limitations of data poisoning attacks, we can not only focus on improving the data poisoning technology itself, but also explore the potential of model poisoning techniques. Since the average method is the most widely-used approach for aggregating local updates from clients, a simple way to amplify the backdoor effect is to prioritize updates from adversarial clients over those from benign clients.

Bagdasaryan et al. [24] proposed the first backdoor attack against federated learning. Their approach involves training a backdoor model that closely resembles the global model, which is then used to replace the latest global model. To improve the effectiveness of this replacement, they slow down the learning rate to extend the lifespan of the backdoor model, and add an anomaly detection term to the loss function to avoid detection. This strategy requires careful evaluation of the global parameters and performs better when the global model is close to convergence.

Zhou et al. [63] proposed an optimization-based model poisoning attack that involves injecting adversarial neurons into the redundant space of a neural network to maintain the attack's concealment and persistence. To identify the redundant space, the Hessian matrix is used to measure the update distance and direction of each neuron's main task (i.e. "importance"). An additional term is then added to the loss function to prevent poisoned neurons from being injected in locations that are particularly relevant to the main task. In a similar vein, Zhang et al. [62] proposed a persistent backdoor attack called Neurotoxin. This method relies on the empirical observation that the norm of a stochastic gradient is primarily concentrated in a small number of "heavy hitter" coordinates. Neurotoxin identifies these heavy hitters using the top-k heuristic and avoids them. By avoiding directions that are most likely to receive large updates from benign devices, the chance of the backdoor being erased is mitigated.

Sun et al. [65] proposed a distance-aware attack (ADA), which enhances poisoning attacks by identifying optimized target classes in the feature space. They addressed the challenge of limited prior knowledge of customer data that competitors may face. To overcome this problem, ADA infers the pairwise distances between different categories in the potential feature space from the shared model



parameters using backward error analysis. They conducted an extensive empirical evaluation of ADA by varying attack frequency in three different image classification tasks. As a result, ADA successfully improved the attack performance by 1.8 times in the most challenging cases with attack frequency of 0.01x.

### C. Summary Of Federated Backdoor Attack

The primary challenge of backdoor attacks in federated learning is how to leverage the distributed nature of this approach, evade detector checks, and ensure the persistence of the backdoors during multiple rounds of update iterations. We identify three research directions for federated backdoor attacks: using multiple malicious clients to insert backdoor attacks, combining generation technology with backdoor attacks, and conducting research on persistent backdoor attacks in federated learning.

## IV. Defenses against Backdoor Attack

To mitigate the problem of backdoor attacks in federated learning, various defensive techniques have been proposed. Given that we previously categorized backdoor attacks as data poisoning attacks and model poisoning attacks, we will now discuss defensive strategies for each of these attack types.

### A. Defense Against Data Poisoning

The simplest approach is to filter out poisoned data samples, which aims to remove the poisoned samples from the training dataset. After the filtering process, only benign samples or purified poisoned samples are used during the training process, thus eliminating the creation of a backdoor from the source.

Tran et al. [67] were the first to investigate methods for filtering out malicious samples from the training set. They demonstrated that poisoned samples tend to leave detectable traces within the covariance range of the feature representation. Exploiting this insight, it is possible to filter out poisoned samples from the training set. Zeng et al. [68] revealed that poisoned samples of existing attacks had some high-frequency artifacts even if their trigger patterns are invisible in the input space. Based on this observation, they designed a simple yet effective filtering method based on those artifacts. Similarly, Chen et al. [69] proposed a two-stage filtering approach. In the first stage, activation values of samples in each class are clustered into two groups, and in the second stage, it is determined which clusters correspond to poisoned samples.

As is well known, backdoors are triggered during the inference stage. Filtering out malicious samples from the testing samples during the inference stage can prevent the backdoor from being activated. Gao et al. [71] proposed to filter attacked samples by overlaying various image patterns on suspicious samples. The smaller the randomness of the input perturbation prediction, the higher the

probability that the suspicious sample is attacked. In [72], Subedar et al. used model uncertainty to distinguish between benign and attacked samples. Later, Du et al. [73] regarded filtering as outlier detection and proposed a differential privacy-based filtering method. Recently, [74] proposed a lightweight method that can filter attacked samples or prior hypotheses of trigger patterns without labeled samples. However, Tang et al. [70] demonstrated that simple target contamination can result in malicious and benign samples being indistinguishable in the feature representation space. Therefore, most sample filtering techniques are ineffective.

Based on data-driven defense methods, in addition to filtering out samples, it is also possible to consider directly preprocessing the samples, specifically by erasing any backdoors within them to prevent them from being embedded in the model. Liu et al. [75] proposed a pre-trained autoencoder as a preprocessor to prevent malicious samples from embedding backdoors by preprocessing the input samples without affecting data classification accuracy.

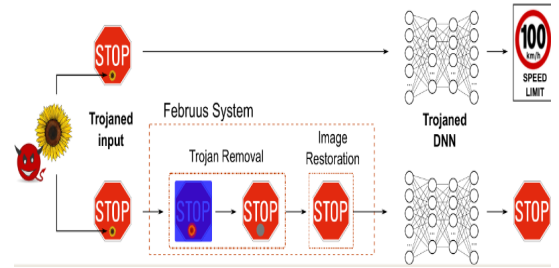


Fig. 9. Overview of the Februs System.

Doan et al. [76] proposed a two-stage image processing method called Februs, in which, in the first stage, Februs uses GradCAM to identify influential regions, which are then removed and replaced with a neutral color frame. Subsequently, Februs uses a GAN-based repair method to reconstruct the masked regions to mitigate their adverse effects (such as benign accuracy reduction). Li et al. [77] discussed the properties of existing poisoning-based static trigger mode attacks. They demonstrated that the attack performance may sharply decrease if the appearance or location of the trigger is slightly changed. Based on this observation, they recommended using spatial transformations (such as contraction, flipping) for defense. Compared to previous methods, this method is more efficient because it requires almost no additional computational cost.

### B. Defense Against Model Poisoning

1) Filtering: Similar to defense methods against poisoned data, defense methods against poisoned models also begin with model filtering. Fung et al. proposed FoolsGold [78], which checks for and eliminates suspicious updates during local updates. FoolsGold is based on the fact that when a global model is trained by a group of attackers,

they will submit updates with the same backdoor objectives throughout the training process, resulting in similar behavior.

However, this similarity does not occur among honest participants because each user’s training dataset is unique and not shared with others. Therefore, malicious attackers can be separated from benign attackers through gradient updates. After detecting such anomalies, FoolGold maintains the learning rate of benign users (submitting only unique gradient updates) and reduces the learning rate of malicious users (repeatedly uploading similar gradient updates) to mitigate backdoor attacks. However, experimental results show that FoolsGold cannot defend against adaptive attacks. Li et al. [79] proposed a spectral anomaly detection framework for a central aggregator that detects and erases backdoor model updates through strong detection model detection. The key idea of spectral anomaly detection is that there is a significant difference between the embedding of benign updates and backdoor updates in a low-dimensional latent space. One practical method for approximating low-dimensional embedding is to build a model using an encoder-decoder structure, where the encoder takes in the raw update and returns a low-dimensional embedding, and the decoder is fed the embedding and outputs the generation error. After training the encoder-decoder model on benign updates, it can be used to identify backdoor updates, generating errors much higher than benign errors; malicious updates will be excluded from the aggregation process. However, this defense method cannot handle multi-trigger backdoor attacks, i.e., injecting various backdoors simultaneously.

Nguyen et al. proposed FLGuard [80], a two-layer defense method that checks for locally updated updates with clear backdoor effects and eliminates residual backdoors through pruning, smoothing, and noise addition. Unlike FoolsGold [78], it also applies to multi-trigger backdoor attacks while maintaining high prediction accuracy for benign primary tasks. Additionally, FLDetector [81] proposed a method to detect malicious clients by checking the consistency of model updates. Essentially, the server predicts the client’s model update based on past updates in each iteration. If the model update received by the client differs significantly from the predicted update over multiple iterations, the client is marked as malicious. Overall, the focus of these methods is mainly divided into two types: the first is to remove toxic updates from malicious clients before model aggregation, and the second is to reduce the impact of malicious clients on the aggregated model, such as reducing the learning rate of suspicious clients.

2) Robust Training: After filtering techniques, another class of techniques aims to directly mitigate backdoor attacks during model training through robust joint training. Differential privacy algorithms have been shown to be effective against backdoors [82], but they may compromise model performance under data imbalance commonly found

in federated learning [83]. DP-FedAvg [84] (Central-DP) is a differential private aggregation strategy that eliminates outliers by clipping the norm of model updates and adding Gaussian noise, but the required amount of noise significantly reduces task accuracy. Sun et al. [27] proposed weak DP, which adds sufficient Gaussian noise to defeat backdoors while maintaining task accuracy, but it is ineffective against constraint-based backdoor attacks [25]. Additionally, DP-based defenses can potentially impact the benign performance of the global model, as the clipping factor also changes the weights of benign model updates.

In addition to DP-based defenses, Andreina et al. [85] proposed Feedback-based Federated Learning (BaFFle) to eliminate backdoors. The key idea of BaFFle is to leverage participants to verify the global model. BaFFle includes a super-digit verification process for each round of federated learning. Specifically, each selected participant checks the current global model by computing a verification function on their secret data and reports to the central server whether the model is backdoored or not. The central server then decides whether to accept or reject the current global model based on feedback from all users. The verification function compares the error rate of a specific class of the current global model with the error rate of previously accepted global models. If the error rate is significantly different, the central server rejects the current global model as it may be backdoored and issues an alert. Unlike anomaly detection, BaFFle is compliant with secure aggregation.

Considering that all of the above defense works lack robustness certification, Xie et al. [86] proposed the first general defense framework CRFL for training certifiably robust FL models against backdoor attacks. CRFL controls the model smoothness through pruning and smoothing of model parameters and generates sample robustness certification against amplitude-limited backdoor attacks. Smoothness and perturbation methods are also used as an additional component to constrain the L2 norm of individual updates to enhance defense performance [87]. In addition, the FL-WBC [88] method aims to identify the fragile parameter space in FL and perturb it during client training. FL-WBC also provides robustness guarantees against backdoor attacks and convergence guarantees for FedAvg. These developments demonstrate promising steps towards improving FL’s robustness against backdoor attacks. In FLARE [89], a trust evaluation method is proposed that calculates a trust score for each model update based on the difference between all model updates and their penultimate layer representation values. FLARE assumes that most clients are trustworthy and assigns low scores to updates that are far from benign update clusters. The model updates are then aggregated with their trust scores as weights and the global model is updated accordingly.

In [98], the concept of anti-backdoor learning is in-

roduced, which involves training a clean model given infected data, dividing the overall learning task into a dual task of learning the clean part of the data and the backdoor part. The article leverages two inherent weaknesses of backdoor attacks: models learn backdoor data faster than clean data, and the stronger the attack, the faster the model converges on the backdoor data. Additionally, backdoor tasks are mutually associated with specific classes. Based on these two weaknesses, a generic learning scheme is proposed to automatically prevent backdoor attacks during training. A two-stage gradient ascent mechanism is introduced to isolate and separate backdoor samples from the target class during the early training stage and break the association between backdoor samples and the target class during the later training stage.

3) Model Reconstruction: The model reconstruction-based approach aims to eliminate hidden backdoors in infected models by directly modifying suspicious models. Therefore, even if triggers are included in the attack samples, the reconstructed model will still correctly predict them because the hidden backdoors have been removed.

As mentioned earlier, the forgetting mechanism of federated backdoor attacks means that as training and model aggregation progress, the backdoor will be forgotten in successive iterations. As a defense, this forgetting mechanism can also be utilized to create many defensive methods.

Zeng et al. [90] defined multiple training as a min-max problem and used implicit hyper-gradients to explain the interdependence between internal and external optimization. In [91], Zhao et al. showed that hidden backdoors infecting DNNs can be repaired using pattern connectivity techniques with a certain number of benign samples. Yoshida et al. and Li et al. [92] perturbed backdoor-related neurons based on the distillation process, reconstructed (infected) DNNs using knowledge distillation techniques, and thus removed hidden backdoors. In addition to directly eliminating hidden backdoors, defense based on trigger synthesis first synthesizes backdoor triggers, and then in the second phase, eliminates hidden backdoors by suppressing the influence of triggers. These defenses have some similarities in the second phase with reconstruction-based defenses. For example, pruning and retraining are commonly used techniques to remove hidden backdoors in both defenses. However, compared to reconstruction-based defenses, the trigger-based defenses' trigger information makes the removal process more effective and efficient.

A GAN-based method was proposed in [93] to synthesize trigger distributions. In [94], they showed that the detection process used in [95] to determine synthesized triggers has several failure modes and proposed a new defense method based on this observation. Additionally, Cheng et al. [96] revealed that the  $\ell_\infty$  norm of activation values can be used to distinguish backdoor-related neurons based on synthesized triggers. Therefore, they propose performing

an  $\ell_\infty$ -based neuron pruning to remove neurons with high activation values in response to triggers. Similarly, Aiken et al. [97] also propose removing hidden backdoors by pruning DNNs based on synthesized triggers from another perspective.

## V. Byzantine Attack

The first type of attack is the Sybil attack. The Sybil attack is a type of network security attack where the attacker creates multiple fake identities or nodes to undermine the network's trust mechanism and disrupt its normal operation. The Sybil attack was first proposed by John Douceur in 2002 (1). The principle behind this attack is that the attacker creates a large number of fake identities, nodes, or accounts that appear independent and genuine to the network but are actually controlled by the attacker. The attacker can use various means to create these fake entities, including using forged IP addresses, anonymous proxy servers, virtual machines, and more. This attack was initially introduced in peer-to-peer networks. The Sybil attack mainly targets systems that rely on trust and identity verification mechanisms, such as peer-to-peer networks, social networks, and blockchains. Attackers can use a large number of fake identities to control the resources of the system, deceive other users, and disrupt consensus mechanisms. For example, in a peer-to-peer network, attackers can create numerous fake nodes to control the distribution process of files, leading to uneven resource distribution or network performance degradation. Since its introduction, the Sybil attack has posed significant threats to various network security systems. Bhagoji (2) explored some attack strategies against deep neural networks by enhancing malicious agent updates and employing alternating minimization strategies for stealth and adversarial targets. They demonstrated the possibility of effective and covert model poisoning attacks. Jun (3) and others were the first to explore the Freerider attack in federated learning and proposed an incremental weight attack method that can evade most defense monitoring methods at that time. They also introduced a new high-dimensional anomaly detection method called STD-DAGMM, which is particularly useful for detecting freeriders. This method has potential applications for detecting other models' weight anomalies as well, which will be discussed in the next chapter. In recent years, Sybil attacks have undergone significant development. Attackers employ more advanced techniques to create counterfeit identities, making Sybil nodes more realistic and difficult to distinguish. For instance, they might use virtualization technology to create multiple independent virtual machines, each with its own IP address and network identifier. Attackers can also utilize social engineering tactics and information acquisition methods to disguise themselves as genuine users or obtain additional fake identities through the authorization of legitimate users. Furthermore, attackers might construct more complex and

realistic Sybil networks by leveraging shared information in social networks. Haonan Yang (4) focused on the problem of vehicular ad hoc networks and criticized previous studies for only considering the security threats and impacts of Sybil attacks in VANETs without analyzing the potential threats and impacts in VFCs (Vehicular Fog Computing). The author summarized four types of Sybil attacks that could affect VFCs in areas such as routing, vehicle decision-making, voting, and reputation systems. With the proliferation of mobile devices and the widespread use of wireless networks, Sybil attacks have also started to increase in these environments. Attackers can jeopardize the security of location services, social networks, and mobile applications by utilizing false identities and location information on mobile devices. In the future, Sybil attacks can be more extensively applied to blockchain technology, enabling attackers to manipulate consensus algorithms, launch double-spending attacks, or undermine the fairness of blockchain networks by controlling a large number of counterfeit identities. The second type of attack is based on the transmission process. Byzantine attacks can occur during the communication process between participants. Attackers can tamper with, delete, or insert malicious messages to disrupt the transmission of model updates. Linyuan Zhang (5) proposed a centralized related probability small-scale attack (CDPS), which utilizes additional information. For example, fusion rules in fault diagnosis (6) can be known as prior knowledge available for optimizing attack strategies. One type of CDPS attack is collaborative attacks among malicious users who communicate with each other. Additionally, information exchange can guide malicious users in designing effective attack strategies. Generally, colluding malicious users (7)(8) first exchange measurement values to ensure more accurate decisions on the licensed channel state and then report the coordinated forged results to enhance attack power. Moreover, Omid Fatemeh introduced another attack model called the full-knowledge attack (9). Cooperative spectrum sensing (CSS) is considered a promising method for identifying available spectrum. However, it not only requires a significant amount of communication resources but also introduces vulnerabilities to Byzantine attacks. Jun Wu (10) mentioned that attackers keep reporting false information to the fusion center consistently. In fact, attackers may attack intermittently while appearing normal during rest periods. The author proposed a low-complexity sequential 0/1 (S0/1) method for CSS in the presence of strategic Byzantine attacks. It does not require strong assumptions or any prior knowledge, and this method will be discussed in the next chapter. With the proliferation of IoT devices and enhanced interconnectivity, attackers may exploit communication vulnerabilities and weaknesses between IoT devices to launch future attacks. Attackers may also utilize high-speed, low-latency 5G/6G networks to carry out larger-scale distributed denial-of-service (DDoS)

attacks, bypass security measures using network slicing and virtualization technologies, or exploit vulnerabilities in mobile communication protocols to invade devices and systems. The third type of attack is known as attack model aggregation. In this type of attack, Byzantine attackers manipulate the model parameters of the participants in order to influence the aggregation results. As discussed by El Mhamdi (11), these attacks by Byzantine workers may not prevent the model from converging but can instead lead to convergence on suboptimal solutions. Another common type of attack is called Gamber, as described by Cong Xie (12). In this attack, an attacker can modify some of the data communicated between participants. The attacker randomly selects data and makes malicious changes to them. A different type of attack is called omniscient. The attacker needs to have knowledge of the gradients sent by all the staff members. The attacker then replaces some gradient vectors by taking the sum of all gradients, scaled by large negative values. The objective is to mislead the Stochastic Gradient Descent (SGD) algorithm into moving in the opposite direction with larger step sizes. Weaker attacks, such as Gaussian attacks, are also possible. In these attacks, some gradient vectors are replaced with random vectors sampled from a Gaussian distribution that has a large variance. Such attackers do not require any information from the staff members. In many applications, multiple sources can provide descriptions of the same object or event, which can inevitably lead to conflicts in data or information. Resolving conflicts in data is an important challenge in achieving convergence in federated learning. Qi Li (13), Yanna Jiang (14), and Harry Hsu (15) have all explored questions related to data differences and conflicts. Byzantine attackers have the ability to exploit these issues in order to manipulate the convergence of the model. In future attacks, attackers may attempt to achieve their objectives by making modifications to the trained model. This can involve incorporating backdoors into the model, which would produce incorrect outputs under specific conditions, or tampering with the model parameters to disrupt its robustness. In order to evade detection, future attack strategies could involve the development of more advanced and covert techniques to camouflage malicious modifications. Additionally, malicious insiders who have access to the model parameters may exploit this privilege to launch attacks. They could manipulate the parameters to compromise the model's performance or even embed sensitive information within the model itself. In the coming years, attackers may explore more sophisticated techniques and algorithms to bypass security monitoring and detection mechanisms. The aim would be to implement internal personnel attacks more efficiently. These advancements would allow attackers to circumvent existing safeguards and carry out their malicious activities with greater precision and effectiveness.



## VI. Defenses against Byzantine Attack

Based on the different types of Byzantine attacks discussed in the previous chapter, we categorize the Byzantine defense methods into the following classes: For Sybil attacks, the defense methods include trust-based evaluation methods and machine learning models for detecting fake nodes. Stephanie Gil (1) proposed a novel algorithm that analyzes received wireless signals to detect the presence of deceptive clients created by adversaries. It does not require specialized hardware or encrypted key exchanges; commercial Wi-Fi cards and software are sufficient. It utilizes the physical characteristics of wireless signals to "perceive" deceivers. The author conducted experiments using the AscTec quadcopter server team and iRobot Create ground clients, demonstrating a deception detection rate exceeding 96%. Blanchard et al. (2) introduced Krum, which detects and removes outliers in gradient aggregation, allowing convergence even in the presence of multiple Byzantine attackers, with low complexity. Luis Muñoz-González (3) proposed a novel algorithm for robust federated learning called Adaptive Federated Averaging, aiming to detect faults, attacks, and malicious updates provided by participants in a collaborative model. The author also proposed a Hidden Markov model to simulate and learn the quality of model updates provided by each participant during training. Defense methods against Sybil attacks in robot networks have also seen rapid development. In Frederik Mallmann-Trenn's work (4), a combination of wireless signal analysis and observations from sociological learning was used to demonstrate rapid convergence of correct trust values for all robots in the team when facing attacks. All robots develop their own opinions about network trust by observing messages sent through the network. They then compare their opinions with those of their neighbors to reach a consensus on whether the robots can be trusted. By leveraging their neighbors' opinions, each robot effectively increases the number of observations available to them about the network, thereby eliminating messages with a high probability of coming from malicious robots through cross-validation. Experiments showed that in a limited number of communication rounds, all robots agreed on the global consensus of trustworthiness in their neighborhood. Sybil attacks are not easily solved because attackers can flexibly forge multiple identities, making it challenging to detect and differentiate false identities. Therefore, designing secure systems requires considering multiple defense mechanisms and selecting the most suitable solutions based on specific application scenarios. Future prospects for such defense methods mainly focus on the following aspects: improving identity authentication mechanisms for distinguishing genuine users from Sybil nodes and establishing more robust trust mechanisms to help identify and filter out Sybil nodes with malicious behavior. The application of blockchain technology in recent years pro-

vides a decentralized, tamper-resistant, and secure shared ledger that can be used to defend against Sybil attacks. By integrating identity authentication and trust mechanisms with blockchain technology, the detection and prevention capabilities of Sybil nodes can be enhanced. Defense methods against attacks during the transmission process include designing secure communication protocols and encryption mechanisms. Previous malicious detection and suppression algorithms focused only on simple "always attack" scenarios, where attackers continuously report false information to the central entity. In reality, attackers may intermittently attack while behaving normally at other times. Under the assumption of simple attack strategies, Ruiliang Chen (5) proposed sequential hypothesis testing to defend against Byzantine attacks, but it requires significant computation and prior knowledge. Therefore, Jun Wu (6) introduced S0/1 using support vector regression (SVR) to offset strategic Byzantine attack defense. Even in a blind scenario, this approach greatly reduces the sample size while providing higher correct sensing ratios. Luis Muñoz-González (3) also proposed a robust aggregation rule that detects and discards malicious or poor local model updates during each training iteration. It includes mechanisms for blocking unwanted participants, thereby enhancing computational and communication efficiency. In traditional federated learning, the performance of models may deteriorate due to variations in data distributions across devices. To address this issue, Hesham Mostafa (7) proposed a technique called "representation matching." This technique improves model performance by matching features between the global model and local models. For clients with homogeneous data distributions, this approach consistently improves accuracy. For heterogeneous clients, in addition to improving accuracy, this approach enhances training robustness and avoids catastrophic training failures without the need for manual hyperparameter tuning for each task. In the future, these defense methods can be further developed by constructing more secure and robust encryption protocols and algorithms. Additionally, more effective identity authentication technologies can be developed to ensure the trustworthiness of communication parties. Continual improvement and enhancement of transport layer protocols, such as SSL/TLS, can provide stronger security and defense capabilities, including the ability to resist man-in-the-middle attacks, data tampering, and replay attacks. The development of real-time threat detection systems capable of detecting and responding to attacks during the transmission process can also be explored. With the advancement of quantum computing, traditional encryption algorithms may be threatened. Therefore, researching and developing quantum-safe communication protocols and algorithms is important to ensure security during the transmission process and protect data confidentiality in a quantum computing environment. Defense methods against model aggregation attacks include designing robust aggregation

algorithms and using differential privacy techniques to protect model parameters. Qi Li (8) modeled the problem using an optimization framework where ground truth and source reliability are defined as two sets of unknown variables. The objective is to minimize the overall weighted discrepancy between the ground truth and multiple-source observations, with each source weighted by its reliability. This framework can incorporate different loss functions to identify features of various data types and has developed efficient computing methods that effectively identify the true information among conflicting data sources. El Mahdi El Mhamdi (9) argues that mere convergence of the Byzantine resilient rules is not sufficient. It is possible that Byzantine workers' attacks can lead to convergence to the worst suboptimal solution. To address this, he proposes a generic enhancement method called Bulyan. This method significantly reduces the leeway for Byzantine workers, constraining them within narrow boundaries. For commonly used batch sizes, Bulyan achieves performance comparable to average speed. Cong Xie (10) introduces three robust aggregation rules for stochastic gradient descent (SGD) with low computational costs. These rules are the first to be theoretically and empirically studied under the non-convex setting, based on median aggregation. Harry Hsu (11) presents a method for synthesizing datasets with continuous identical ranges and provides performance metrics for the federated averaging algorithm. Experimental results demonstrate that performance deteriorates with distribution differences, prompting the introduction of a mitigation strategy using server momentum. It has been shown that using momentum on top of SGD for non-iid problems has achieved significant success in accelerating network training. The proposed method suppresses oscillations through the accumulation of gradient history. Amit Portnoy (12) proposes a novel method called Byzantine Robust Client Weighting (BRCW) to minimize the impact of malicious clients in federated learning. BRCW assigns different weights to each client based on their credibility and contribution to the overall model accuracy. During training, the weights are dynamically updated according to each client's performance and behavior. Unlike previous work, this study considers the sample size provided by clients as untrustworthy, as it may come from malicious clients. BRCW exhibits robustness against various forms of attacks, including communication disruptions and data poisoning. Krishna Pillutla (13) designs a novel robust aggregation oracle based on classical geometric medians and demonstrates its robustness in federated learning with limited heterogeneity, even when up to half of the devices are faulty. The proposed RFA algorithm outperforms the standard FedAvg (14) in high damage scenarios and nearly matches FedAvg's performance in low damage scenarios, but with 1-3 times higher communication costs. Yanna Jiang (15) introduces the combination of handling imbalanced data and Byzantine attacks in the context of federated learning for the first time. The concept of

an intelligence pool is proposed to separate the task of judging the shared local model values from the aggregator, avoiding the limitations of a single criterion through two-layer verification. Future prospects for such defense methods will focus on more stringent authentication and authorization mechanisms, as well as parameter integrity protection. Mechanisms such as hash algorithms or digital signatures can be adopted to safeguard the integrity of parameters, preventing tampering during transmission and storage. Detection of malicious parameter modifications and corresponding response measures are also required. By comprehensively applying these methods, the defense capability against malicious parameter modification attacks can be enhanced, ensuring the security and integrity of systems and applications.

## VII. Adversarial Attack

Both adversarial attacks and backdoor attacks are techniques used to modify benign testing samples in order to make models misbehave during the inference process. While adversarial perturbations are sample-agnostic in universal adversarial attacks, these attacks can appear similar to backdoor attacks. Consequently, researchers who are unfamiliar with backdoor attacks may question their significance, as these attacks require additional controls on the training process to some extent. However, despite certain similarities, these attacks have essential differences.

Firstly, adversarial attackers need to control the inference process to a certain degree but not the training process of models. They must query the model results or even gradients multiple times to generate adversarial perturbations by optimization, given a fixed targeted model. On the other hand, backdoor attackers require modifying certain training stages, such as data collection and model training, without any additional requirements in the inference process. Secondly, from the perspective of attacked samples, backdoor attackers use known, non-optimized perturbations, whereas adversarial attackers require obtaining them through the optimization process based on the model output. This optimization in adversarial attacks requires multiple queries, making them unable to be real-time in many cases. Finally, the mechanisms of these attacks are fundamentally different. Adversarial vulnerability results from the differences in behaviors of models and humans, while backdoor attackers exploit the excessive learning ability of deep neural networks (DNNs) to establish a latent connection between trigger patterns and target labels.

In federated learning, because the data is not shared among participants, attackers can use this to generate antagonistic samples locally, then inject them into the data sets of participants, and train the model with the data of other participants during the training process.

So, how should attacks targeting federated learning be designed? In centralized learning, FGSM [99] was

initially used to generate adversarial examples. Let  $\theta$  be the parameters of a model,  $x$  the input to the model,  $\eta$  the perturbation to original input and  $\|\eta\|_\infty \leq \epsilon$ ,  $y$  the targets associated with  $x$  (for machine learning tasks that have targets) and  $J(\theta, x, y)$  be the cost used to train the neural network. We can linearize the cost function around the current value of  $\theta$ , obtaining an optimal max-norm constrained perturbation of:

$$\eta = \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \quad (1)$$

In federated learning, attackers can use FGSM locally to generate adversarial samples and then inject them into the datasets of some participants to affect the performance of the entire model.

PGD [100] is a more powerful adversarial sample generation method, which generates adversarial samples by iteratively perturbing the input. Specifically, PGD first randomly generates an initial disturbance  $\delta_0$ , and then iteratively updates the disturbance  $\delta_t$ , to meet the constraint, that is,  $|\delta_t|_\infty \leq \epsilon_{\max}$ , where  $\epsilon_{\max}$  is the maximum value of the disturbance. After each update, PGD also projects the disturbance  $\delta_t$  back into the constraint space to ensure that it meets the constraint conditions. Specifically, the update formula for PGD is as follows:

$$x^{t+1} = \text{Clip}(x + \text{sign}(\nabla_x L(\theta, x', y)) \cdot \epsilon_{\max}, x - \epsilon_{\max}, x + \epsilon_{\max}), \quad (2)$$

Where Clip means to cut  $x$  into the interval, and  $\epsilon_{\max}$  is the maximum value of the disturbance. Unlike FGSM, PGD requires multiple iterations to generate adversarial examples, so attackers need to use the local dataset and model parameters of each participant to iteratively update the perturbation.

In federated learning, using PGD (Projected Gradient Descent) or FGSM (Fast Gradient Sign Method) to generate adversarial examples and perform adversarial attacks is a common method. However, compared to centralized learning, there are some considerations and techniques that need to be taken into account.

1) Data privacy and distribution heterogeneity: : In federated learning, data is typically distributed across various edge devices, making it slightly more difficult to generate adversarial examples and perform attacks. Therefore, attackers need more complex strategies to adapt to the heterogeneity of the data distribution. In this case, the generation of adversarial examples should consider the data distribution and usage on each device. In federated learning, the amount and quality of data from each participant may vary, leading to imbalanced training data. This may affect the effectiveness of adversarial attacks, as attackers may not be able to generate sufficiently accurate adversarial examples to attack all participants' models.

2) Network latency and bandwidth limitations: : In federated learning, devices communicate through a network, which may result in network latency or bandwidth limitations. Therefore, when generating adversarial examples and performing attacks, these factors need to

be taken into consideration. For example, it may be necessary to optimize the size and generation speed of adversarial examples to adapt to network limitations. And in federated learning, the training steps of each participant are asynchronous, which may lead to synchronization issues. For example, attackers may not be able to perform attacks on all participants' models because they may be at different training steps or states.

3) Update of attack strategies: : Due to the dynamic nature of federated learning, attackers need to constantly update their attack strategies to adapt to changes in the learning model and data distribution. This may require designing more complex attack strategies and using more efficient optimization algorithms.

4) Attack detection and defense: : In federated learning, adversarial attacks may be easier to detect due to the distribution and usage of data. Therefore, it is necessary to design adversarial examples that are more difficult to detect or use more complex attack strategies to avoid detection.

Overall, using PGD or FGSM to generate adversarial examples and perform adversarial attacks in federated learning requires considering the characteristics and limitations of federated learning, including the heterogeneity of data distribution, network latency and bandwidth limitations, update of attack strategies, and attack detection and defense. This may require designing more complex and efficient attack strategies and adversarial example generation methods.

## VIII. Defenses against Adversarial Attack

Overall, progress has been made in defending against adversarial samples. As adversarial samples share similarities with the previously discussed backdoor samples, defense methods involving sample filtering can be employed. However, in this section, we will refrain from discussing filtering methods and instead focus on the defense method of adversarial training.

Adversarial training is a defense method against adversarial attacks. Its basic idea is to incorporate adversarial examples into the training process, enabling the model to better withstand unknown adversarial attacks. Specifically, adversarial training combines the original training dataset with adversarial samples generated to target the model, and retrain the model accordingly. This way, the model encounters adversarial examples continuously during the learning process, thereby enhancing its robustness and generalization capability.

FAT (Federated Adversarial Training) is a method proposed in [31] that combines federated learning and adversarial training to reduce evasion threats during inference while preserving data privacy during training. In the FAT protocol, each device trains a local model and generates adversarial samples, which are mixed with the original data for training. The models are then uploaded to the server for aggregation. This helps to improve the

robustness of the model against adversarial attacks. The authors note that the protocol does not work out of the box and requires careful tuning of the optimization parameters to achieve good results. Additionally, the authors acknowledge that the experiments were conducted on idealized federated settings and that further research is needed to evaluate the effectiveness of FAT in more realistic scenarios.

From the perspective of variance-bias decomposition, Chen et al. [108] decomposed the loss function into a combination of bias and variance, generated adversarial samples on the server and returned them to each client for adversarial training. Then, they used any model aggregation algorithm to improve the global model's adversarial robustness.

However, there are also several challenges with implementing adversarial training in federated learning, such as heterogeneous clients caused by distribution, certified guarantees, privacy leakage, slow convergence rate.

1) heterogeneous clients caused by distribution: The paper [101] proposes a novel method called FedRobust to address the problem of distribution shifts in federated learning. FedRobust is a gradient descent ascent (GDA) algorithm that solves the minimax robust optimization problem and can be efficiently implemented in a federated setting. This paper shows that FedRobust, which alternates between the perturbation and parameter model variables, will converge to a stationary point in the minimax objective that satisfies the Polyak-Łojasiewicz (PL) [102] condition. This optimization method can be used to address the issue of distribution shifts in federated learning, which can significantly impact the performance of the trained model.

Zhang et al. [34] found that the accuracy of the federated learning model with adversarial training decreases on clean data, especially when the data of different clients are non-independent and identically distributed, so they propose an algorithm based on decision boundary, which uses local reweighting and global regularization to improve the accuracy and countermeasure robustness in federated countermeasure training. Chen et al. [32] studies the skewed labels problem in federation learning and proposes a CalFAT framework to calculate the logits of each class. Finally, the models between different clients are isomorphic to enhance the robustness and accuracy of the global model.

Hong et al. [105] have studied the problems caused by clients with different computing resources. During federated learning, some clients can support adversarial training, but some clients with lack of computing resources can not support it. This paper studies how to spread resistance robustness from resource-rich users who can support confrontation training to those users with poor resources. This paper shows that an effective robust propagation algorithm is proposed by using BN layer technology correctly.

2) Certified Guarantees: Chen et al. [104] proposes incorporating randomized smoothing techniques into federated adversarial training to enable data-private distributed learning with certifiable robustness to test-time adversarial perturbations. The experiments conducted in the paper show that this approach can deliver models as robust as those trained by centralized training, and can enable provably-robust classifiers to 2-bounded adversarial perturbations in a distributed setup.

3) privacy leakage: In [106], it is proved that the model of confrontation training is more prone to privacy disclosure than the model of normal training. Zhou et al. [?] argues that conventional federated learning frameworks are vulnerable to strong adversarial attacks, even if adversarial training using locally generated adversarial examples is performed on each client. To address this problem, the paper proposes a new framework called FedBVA (Federated Learning with Bias-Variance Analysis) that provides a tiny amount of bias-variance perturbed data from the central server to the clients through asymmetrical communication. This approach dramatically improves the robustness of the training model under various settings, without violating the clients' privacy.

4) slow convergence rate: In [107], a method was proposed to solve this problem. Assuming the local iteration number of federated learning is  $E$ , this article believes that a larger  $E$  will increase the drift between models and affect the adversarial robustness, but will converge quickly. A smaller  $E$  will increase the adversarial robustness but decrease the convergence efficiency. Therefore, this article proposes a dynamic mechanism for adjusting  $E$ , and uses FedCurv instead of FedAvg as the global aggregation algorithm, and proposes a new algorithm called FedDynAT to simultaneously improve the convergence speed and adversarial robustness.

## IX. Hybrid Defenses

Recently, some works have studied the latent connection between adversarial attacks and backdoor attacks. For example, Weng et al. [66] empirically demonstrated that defending against adversarial attacks via adversarial training may increase the risks of backdoor attacks.

## X. Advanced Research and Problems

## XI. Conclusion

## References

- [1] M. Goddard, The EU General Data Protection Regulation (GDPR): European regulation that has a global impact, *International Journal of Market Research* 59 (2017) 703-705.
- [2] O. Gómez-Carmona, D. Casado-Mansilla, F. A. Kraemer, D. L. de Ipiña, J. GarcíaZubia, Exploring the computational cost of machine learning at the edge for human-centric internet of things, *Future Generation Computer Systems* 112 (2020) 670-683.
- [3] J. Zhang, D. Tao, Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things, *IEEE Internet of Things Journal* 8 (2021) 7789-7817.

- [4] C. Ma, J. Konečný, M. Jaggi, V. Smith, M. Jordan, P. Richtárik, M. Takáč, Distributed optimization with arbitrary local solvers, *Optimization Methods and Software* 32 (2017) 813–848.
- [5] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.
- [6] GlobalFederatedLearningMarketbyApplication (Drug Discovery, Industrial IoT, Risk Management), Vertical (Healthcare Life Sciences, BFSI, Manufacturing, Automotive Transportation, Energy Utilities), and Region - Forecast to 2028, "researchandmarkets.com", Accessed date: May 12, 2023.
- [7] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, H. Yu, Federated Learning, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2019.
- [8] M. Househ, E. Borycki, and A. Kushniruk. *Multiple Perspectives on Artificial Intelligence in Healthcare*. Springer, 2021.
- [9] R. Rau, R. Wardrop, and L. Zingales. *The Palgrave Handbook of Technological Finance*. Springer, 2021.
- [10] Fang, Xiuwen, and Mang Ye. "Robust federated learning with noisy and heterogeneous clients." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- [11] Tang, Zhenheng, et al. "Virtual homogeneity learning: Defending against data heterogeneity in federated learning." *International Conference on Machine Learning*. PMLR, 2022.
- [12] Kairouz, Peter, et al. "Advances and open problems in federated learning." *Foundations and Trends® in Machine Learning* 14.1 2 (2021): 1-210.
- [13] Yang, Qiang, et al. "Federated machine learning: Concept and applications." *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019): 1-19.
- [14] Huang, Wei, et al. "Fairness and accuracy in horizontal federated learning." *Information Sciences* 589 (2022): 170-185.
- [15] Liu, Yang, et al. "Vertical federated learning." *arXiv preprint arXiv:2211.12814* (2022).
- [16] Ghosh, Avishek, et al. "An efficient framework for clustered federated learning." *Advances in Neural Information Processing Systems* 33 (2020): 19586-19597.
- [17] Briggs, Christopher, Zhong Fan, and Peter Andras. "Federated learning with hierarchical clustering of local updates to improve training on non-IID data." *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020.
- [18] Hsu, Tzu-Ming Harry, Hang Qi, and Matthew Brown. "Federated visual classification with real-world data distribution." *Computer Vision ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X* 16. Springer International Publishing, 2020.
- [19] Wahab, Omar Abdel, et al. "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems." *IEEE Communications Surveys and Tutorials* 23.2 (2021): 1342-1397.
- [20] Yang, Shengwen, et al. "Parallel distributed logistic regression for vertical federated learning without third-party coordinator." *arXiv preprint arXiv:1911.09824* (2019).
- [21] Deng, Yuyang, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. "Distributionally robust federated averaging." *Advances in neural information processing systems* 33 (2020): 15111-15122.
- [22] Sun, Tao, Dongsheng Li, and Bao Wang. "Decentralized federated averaging." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.4 (2022): 4289-4301.
- [23] Fallah, Alireza, Aryan Mokhtari, and Asuman Ozdaglar. "Personalized federated learning: A meta-learning approach." *arXiv preprint arXiv:2002.07948* (2020).
- [24] Bagdasaryan, Eugene, et al. "How to backdoor federated learning." *International conference on artificial intelligence and statistics*. PMLR, 2020.
- [25] Wang, Hongyi, et al. "Attack of the tails: Yes, you really can backdoor federated learning." *Advances in Neural Information Processing Systems* 33 (2020): 16070-16084.
- [26] Gong, Xueluan, et al. "Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions." *IEEE Wireless Communications* (2022).
- [27] Sun, Ziteng, et al. "Can you really backdoor federated learning?." *arXiv preprint arXiv:1911.07963* (2019).
- [28] Ozdayi, Mustafa Safa, Murat Kantarcioglu, and Yulia R. Gel. "Defending against backdoors in federated learning with robust learning rate." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 10. 2021.
- [29] Fang, Minghong, et al. "Local model poisoning attacks to Byzantine-Robust federated learning." *29th USENIX security symposium (USENIX Security 20)*. 2020.
- [30] Guo, Shangwei, et al. Byzantine-Resilient Decentralized Stochastic Gradient Descent.
- [31] Zizzo, Giulio, et al. "Fat: Federated adversarial training." *arXiv preprint arXiv:2012.01791* (2020).
- [32] Chen, Chen, et al. "Calfat: Calibrated federated adversarial training with label skewness." *Advances in Neural Information Processing Systems* 35 (2022): 3569-3581.
- [33] Li, Xiaoxiao, Zhao Song, and Jiaming Yang. "Federated adversarial learning: A framework with convergence analysis." *International Conference on Machine Learning*. PMLR, 2023.
- [34] Zhang, Jie, et al. "Delving into the adversarial robustness of federated learning." *arXiv preprint arXiv:2302.09479* (2023).
- [35] Prakash, Saurav, and Amir Salman Avestimehr. "Mitigating byzantine attacks in federated learning." *arXiv preprint arXiv:2010.07541* (2020).
- [36] Huang, Hanxun, et al. "Distilling Cognitive Backdoor Patterns within an Image." *arXiv preprint arXiv:2301.10908* (2023).
- [37] Blanchard, Peva, et al. "Machine learning with adversaries: Byzantine tolerant gradient descent." *Advances in neural information processing systems* 30 (2017).
- [38] Lyu, Lingjuan, et al. "Privacy and robustness in federated learning: Attacks and defenses." *IEEE transactions on neural networks and learning systems* (2022).
- [39] Guo, Shangwei, et al. "Robust and privacy-preserving collaborative learning: A comprehensive survey." *arXiv preprint arXiv:2112.10183* (2021).
- [40] Enthoven, David, and Zaid Al-Ars. "An overview of federated deep learning privacy attacks and defensive strategies." *Federated Learning Systems: Towards Next-Generation AI* (2021): 173-196.
- [41] Rodríguez-Barroso, Nuria, et al. "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges." *Information Fusion* 90 (2023): 148-173.
- [42] Tariq, Asadullah, et al. "Trustworthy Federated Learning: A Survey." *arXiv preprint arXiv:2305.11537* (2023).
- [43] Zhang, Yifei, et al. "A Survey of Trustworthy Federated Learning with Perspectives on Security, Robustness, and Privacy." *arXiv preprint arXiv:2302.10637* (2023).
- [44] Dal Fabbro, Nicolò, Aritra Mitra, and George J. Pappas. "Federated TD Learning over Finite-Rate Erasure Channels: Linear Speedup under Markovian Sampling." *IEEE Control Systems Letters* (2023).
- [45] Miao, Chenglin, et al. "Towards data poisoning attacks in crowd sensing systems." *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2018.
- [46] Zhang, Hengtong, et al. "Data poisoning attack against knowledge graph embedding." *arXiv preprint arXiv:1904.12052* (2019).
- [47] Barreno, Marco, et al. "Can machine learning be secure?." *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. 2006.
- [48] Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine generals problem." *Concurrency: the works of leslie lamport*. 2019. 203-226.
- [49] Xie, Cong, Oluwasanmi Koyejo, and Indranil Gupta. "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation." *Uncertainty in Artificial Intelligence*. PMLR, 2020.



- [50] Bernstein, Jeremy, et al. "signSGD with majority vote is communication efficient and fault tolerant." arXiv preprint arXiv:1810.05291 (2018).
- [51] Damaskinos, Georgios, et al. "Aggregathor: Byzantine machine learning via robust gradient aggregation." *Proceedings of Machine Learning and Systems* 1 (2019): 81-106.
- [52] Geigel, Arturo. "Neural network trojan." *Journal of Computer Security* 21.2 (2013): 191-232.
- [53] Gu, Tianyu, Brendan Dolan-Gavitt, and Siddharth Garg. "Badnets: Identifying vulnerabilities in the machine learning model supply chain." arXiv preprint arXiv:1708.06733 (2017).
- [54] Shen, Shiqi, Shruti Tople, and Prateek Saxena. "Auror: Defending against poisoning attacks in collaborative deep learning systems." *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 2016.
- [55] Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." *Advances in neural information processing systems* 30 (2017).
- [56] Tolpegin, Vale, et al. "Data poisoning attacks against federated learning systems." *Computer Security ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I* 25. Springer International Publishing, 2020.
- [57] Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." *Advances in neural information processing systems* 31 (2018).
- [58] Zhu, Chen, et al. "Transferable clean-label poisoning attacks on deep neural nets." *International Conference on Machine Learning*. PMLR, 2019.
- [59] Xie, Chulin, et al. "DBA: Distributed Backdoor Attacks against Federated Learning." *International Conference on Learning Representations, International Conference on Learning Representations*, Apr. 2020.
- [60] Salem, Ahmed, et al. "Dynamic backdoor attacks against machine learning models." *2022 IEEE 7th European Symposium on Security and Privacy (EuroSP)*. IEEE, 2022.
- [61] Li, Yiming, et al. "Rethinking the Trigger of Backdoor Attack." arXiv: Cryptography and Security, arXiv: Cryptography and Security, May 2021.
- [62] Zhang, Zhengming, et al. "Neurotoxin: Durable backdoors in federated learning." *International Conference on Machine Learning*. PMLR, 2022.
- [63] Zhou, Xingchen, et al. "Deep model poisoning attack on federated learning." *Future Internet* 13.3 (2021): 73.
- [64] Dai, Yanbo, and Songze Li. "Chameleon: Adapting to Peer Images for Planting Durable Backdoors in Federated Learning." arXiv preprint arXiv:2304.12961 (2023).
- [65] Sun, Yuwei, Hideya Ochiai, and Jun Sakuma. "Semi-targeted model poisoning attack on federated learning via backward error analysis." *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022.
- [66] Weng, Cheng-Hsin, Yan-Ting Lee, and Shan-Hung Brandon Wu. "On the trade-off between adversarial and backdoor robustness." *Advances in Neural Information Processing Systems* 33 (2020): 11973-11983.
- [67] Tran, Brandon, Jerry Li, and Aleksander Madry. "Spectral signatures in backdoor attacks." *Advances in neural information processing systems* 31 (2018).
- [68] Zeng, Yi, et al. "Rethinking the backdoor attacks' triggers: A frequency perspective." *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.
- [69] Chen, Bryant, et al. "Detecting backdoor attacks on deep neural networks by activation clustering." arXiv preprint arXiv:1811.03728 (2018).
- [70] Tang, Di, et al. "Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection." *30th USENIX Security Symposium (USENIX Security 21)*. 2021.
- [71] Gao, Yansong, et al. "STRIP: A Defence Against Trojan Attacks on Deep Neural Networks." *Cornell University - arXiv, Cornell University - arXiv*, Feb. 2019.
- [72] M. Subedar, N. Ahuja, R. Krishnan, I. J. Ndiour, and O. Tickoo, "Deep probabilistic models to detect data poisoning attacks," in *NeurIPS Workshop*, 2019.
- [73] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," in *ICLR*, 2020.
- [74] M. Javaheripi, M. Samragh, G. Fields, T. Javidi, and F. Koushanfar, "Cleann: Accelerated trojan shield for embedded neural networks," in *ICCAD*, 2020.
- [75] Liu, Yuntao, Yang Xie, and Ankur Srivastava. "Neural trojans." *2017 IEEE International Conference on Computer Design (ICCD)*. IEEE, 2017.
- [76] Doan, Bao Gia, et al. "Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems." *Annual Computer Security Applications Conference*, 2020, <https://doi.org/10.1145/3427228.3427264>.
- [77] Li, Yiming, et al. "Backdoor Attack in the Physical World." arXiv: Cryptography and Security, arXiv: Cryptography and Security, Apr. 2021.
- [78] Fung, Clement, Chris JM Yoon, and Ivan Beschastnikh. "Mitigating sybils in federated learning poisoning." arXiv preprint arXiv:1808.04866 (2018).
- [79] Li, Suyi, et al. "Learning to detect malicious clients for robust federated learning." arXiv preprint arXiv:2002.00211 (2020).
- [80] Nguyen, ThienDuc, et al. "FLGUARD: Secure and Private Federated Learning." arXiv: Cryptography and Security, arXiv: Cryptography and Security, Jan. 2021.
- [81] Zhang, Zaixi, et al. FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients. July 2022.
- [82] Naseri, MohammadKazemGharib, et al. "Toward Robustness and Privacy in Federated Learning: Experimenting with Local and Central Differential Privacy." *Cornell University - arXiv, Cornell University - arXiv*, Sept. 2020.
- [83] Bagdasaryan, Eugene, et al. "Differential Privacy Has Disparate Impact on Model Accuracy." *Neural Information Processing Systems, Neural Information Processing Systems*, May 2019.
- [84] McMahan, H. Brendan, et al. "Learning Differentially Private Recurrent Language Models." arXiv: Learning, arXiv: Learning, Oct. 2017.
- [85] Andreina, Sebastien, et al. "BaFFLe: Backdoor Detection via Feedback-Based Federated Learning." *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021, <https://doi.org/10.1109/icdcs51616.2021.00086>.
- [86] Xie, Chulin, et al. "CRFL: Certifiably Robust Federated Learning against Backdoor Attacks." arXiv: Learning, arXiv: Learning, June 2021.
- [87] Nguyen, ThienDuc, et al. FLAME: Taming Backdoors in Federated Learning.
- [88] Sun, Jingwei, et al. "FL-WBC: Enhancing Robustness against Model Poisoning Attacks in Federated Learning from a Client Perspective." *Neural Information Processing Systems, Neural Information Processing Systems*, Oct. 2021.
- [89] Wang, Ning, et al. FLARE: Defending Federated Learning against Model Poisoning Attacks via Latent Space Representations.
- [90] Zeng, YiXin, et al. "Adversarial Unlearning of Backdoors via Implicit Hypergradient." *Cornell University - arXiv, Cornell University - arXiv*, Oct. 2021.
- [91] Zhao, Pu, et al. "Bridging Mode Connectivity in Loss Landscapes and Adversarial Robustness." *International Conference on Learning Representations, International Conference on Learning Representations*, Apr. 2020.
- [92] Yoshida, Kota, and Takeshi Fujino. "Disabling Backdoor and Identifying Poison Data by Using Knowledge Distillation in Backdoor Attacks on Deep Neural Networks." *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*, 2020, <https://doi.org/10.1145/3411508.3421375>.
- [93] Zhu, Liwan, et al. "GangSweep: Sweep out Neural Backdoors by GAN." *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, <https://doi.org/10.1145/3394171.3413546>.
- [94] Guo, Wenbo, et al. "Towards Inspecting and Eliminating Trojan Backdoors in Deep Neural Networks." *2020 IEEE International Conference on Data Mining (ICDM)*, 2020, <https://doi.org/10.1109/icdm50108.2020.00025>.

- [95] Wang, Bolun, et al. "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks." 2019 IEEE Symposium on Security and Privacy (SP), 2019, <https://doi.org/10.1109/sp.2019.00031>.
- [96] Cheng, Hao, et al. "Defending against Backdoor Attack on Deep Neural Networks." arXiv: Cryptography and Security, arXiv: Cryptography and Security, Feb. 2020.
- [97] Aiken, William, et al. "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks." *Computers Security* 106 (2021): 102277.
- [98] Li, Yige, et al. "Anti-backdoor learning: Training clean models on poisoned data." *Advances in Neural Information Processing Systems* 34 (2021): 14900-14912.
- [99] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
- [100] Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." arXiv preprint arXiv:1706.06083 (2017).
- [101] Reiszadeh, Amirhossein, et al. "Robust federated learning: The case of affine distribution shifts." *Advances in Neural Information Processing Systems* 33 (2020): 21554-21565.
- [102] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963. (Cited on page 5.)
- [103] Zhou, Yao, Jun Wu, and Jingrui He. "Adversarially robust federated learning for neural networks." (2020).
- [104] Chen, Cheng, et al. "Certifiably-robust federated adversarial learning via randomized smoothing." 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS). IEEE, 2021.
- [105] Hong, Junyuan, et al. "Federated robustness propagation: Sharing robustness in heterogeneous federated learning." arXiv preprint arXiv:2106.10196 (2021).
- [106] Zhang, Jingyang, Yiran Chen, and Hai Li. "Privacy Leakage of Adversarial Training Models in Federated Learning Systems." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- [107] Shah, Devansh, et al. "Adversarial training in communication constrained federated learning." arXiv preprint arXiv:2103.01319 (2021).
- [108] Zhou, Yao, et al. "Adversarial robustness through bias variance decomposition: A new perspective for federated learning." *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 2022.