



Robust and privacy-preserving collaborative training: a comprehensive survey

Fei Yang¹ · Xu Zhang² · Shangwei Guo² · Daiyuan Chen¹ · Yan Gan² · Tao Xiang² · Yang Liu³

Accepted: 6 May 2024 / Published online: 20 June 2024
© The Author(s) 2024

Abstract

Increasing numbers of artificial intelligence systems are employing collaborative machine learning techniques, such as federated learning, to build a shared powerful deep model among participants, while keeping their training data locally. However, concerns about integrity and privacy in such systems have significantly hindered the use of collaborative learning systems. Therefore, numerous efforts have been presented to preserve the model's integrity and reduce the privacy leakage of training data throughout the training phase of various collaborative learning systems. This survey seeks to provide a systematic and comprehensive evaluation of security and privacy studies in collaborative training, in contrast to prior surveys that only focus on one single collaborative learning system. Our survey begins with an overview of collaborative learning systems from various perspectives. Then, we systematically summarize the integrity and privacy risks of collaborative learning systems. In particular, we describe state-of-the-art integrity attacks (e.g., Byzantine, backdoor, and adversarial attacks) and privacy attacks (e.g., membership, property, and sample inference attacks), as well as the associated countermeasures. We additionally provide an analysis of open problems to motivate possible future studies.

Keywords Collaborative learning · Federated learning · Byzantine · Backdoor · Privacy

1 Introduction

Deep learning (DL) has shown remarkable success in numerous disciplines, such as computer vision, natural language processing, bioinformatics, and even board game programs. DL systems adopt deep neural networks (DNNs) to learn autonomously from massive training datasets (Krizhevsky et al. 2012a; Szegedy et al. 2015; Devlin et al. 2018). A learning system relies primarily on two components to efficiently train a DL model: a large number of high-quality training samples and high-performance Graphics Processing Units (GPUs). Nevertheless, the training datasets and GPUs may be dispersed among numerous parties for various reasons. Consider the following two examples (Litjens et al. 2017; Gawali et al. 2021; Hard et al. 2018):

Fei Yang and Xu Zhang have contributed equally to this work.

Extended author information available on the last page of the article

Medical image classification A hospital needs to acquire a lung cancer detector model to assist its doctors in identifying lung cancer patients from their computed tomography (CT) images. Due to the hospital's limited experience with lung cancer patients, it is difficult for it to develop a highly accurate model. To ensure the accuracy of the diagnosis, the hospital unites with other hospitals to collaboratively learn a shared model together. Taking patient confidentiality into account, all hospitals must store CT images locally.

Mobile keyboard prediction Gboard, the Google keyboard, intends to give dependable and quick mobile input techniques, such as next-word predictions, as more users migrate to mobile devices. Although publicly accessible datasets can be utilized for such tasks, their distribution rarely matches that of users. Thus, Gboard requires user-generated texts for improved performance without making users feel uneasy about the gathering and remote storage of their personal information.

Collaborative learning has gained popularity as a possible option for such application scenarios in recent years (Dean et al. 2012; Peteiro-Barral and Guijarro-Berdiñas 2013; Leroy et al. 2019; He et al. 2020; Zheng et al. 2020; Lim et al. 2020; Aledhari et al. 2020). Specifically, collaborative learning enables two or more participants to collaboratively train a shared global DL model while maintaining individual training datasets locally. Each participant trains the shared model using his own training data and exchanges and updates model parameters with others. Collaborative learning can increase the training speed and performance of the shared model while maintaining the confidentiality of the training datasets of the participants. Therefore, it is a paradigm for cases in which training data is sensitive (e.g., medical records, personally identifiable information, etc.). Such a paradigm is not just a theoretical construct; it addresses the real-world challenges faced by organizations and individuals alike. In an age where data is abundant yet siloed due to privacy concerns and regulatory constraints, collaborative learning provides a bridge, allowing diverse stakeholders to benefit from collective intelligence without compromising on data confidentiality. Several learning architectures have been proposed for collaborative learning: with and without a central server, with different means of aggregating models (Li et al. 2014; Moritz et al. 2015; Liu et al. 2019b; Sun et al. 2021d; Sahu et al. 2018; Reddi et al. 2020; Wang et al. 2020c; Lu and De Sa 2021). Federated learning is an essential branch of collaborative learning (Li et al. 2021b) that enables participants such as mobile phones to collaboratively learn a shared prediction model while retaining all the training data on the device, decoupling machine learning from the requirement to store data in the cloud.

Although each participant stores his training dataset locally and only shares the updates of the global model at each iteration, adversaries can still conduct attacks during the training process to compromise model integrity and data privacy (Guerraoui et al. 2018; Bhagoji et al. 2019; Zhu et al. 2019b; Zhang et al. 2018a). One of the most severe threats is model integrity, which can be undermined easily if some participants are unreliable (Blanchard et al. 2017; Guo et al. 2021b). For example, malicious participants may poison their training datasets with carefully crafted malicious triggers. Then, at each iteration, they generate malicious updates containing the triggers and gradually inject such triggers as backdoors into the global model by spreading the malicious updates in order to generate further profit or expand their advantages (Bagdasaryan et al. 2020; Wang et al. 2020b). In addition to disguising themselves as participants, adversaries can damage the collaborative learning process by delivering malicious updates to their neighborhoods or parameter servers (Muñoz-González et al. 2017; Bhagoji et al. 2019; Baruch et al. 2019). Blanchard et al. (2017) and Guo et al. (2021b) demonstrate that a single malicious participant can dominate the entire collaborative learning process.

Aside from risks to model integrity, the protection of each participant's data privacy is a key challenge. Despite the fact that participants do not share the raw training samples with others, it has been established that the shared updates are created from the samples and indirectly leak information about the training datasets. For instance, Melis et al. (2019) discovered that it is possible to capture the membership and accidental feature leaking from shared gradients throughout the training procedure. More seriously, Zhu et al. (2019b) proposed an optimization approach that can reconstruct training samples from the corresponding updates.

To address the above integrity and privacy threats, numerous strategies are recommended to defend against these attacks (Blanchard et al. 2017; Cao and Lai 2019; Guerraoui et al. 2018; Muñoz-González et al. 2019; Pan et al. 2020a; Shejwalkar and Houmansadr 2021; Xie et al. 2019b, 2020; Yin et al. 2018; Tran et al. 2018; Chen et al. 2018, 2019; Chan and Ong 2019; Chou et al. 2018; Gao et al. 2019; Truong et al. 2020; Ma and Liu 2019; Liu et al. 2019c, 2020; Wang et al. 2019a; Huang et al. 2019; Sun et al. 2019; Zhao et al. 2020b; Zhu et al. 2019b; Ozdayi et al. 2020; Chaudhuri et al. 2011; Abadi et al. 2016; Zhang et al. 2018b; Li et al. 2018, 2020d; Yu et al. 2019a; Jayaraman and Evans 2019; Aono et al. 2016; Kim et al. 2018; Bonawitz et al. 2017). For instance, to achieve byzantine-resilient collaborative learning, Blanchard et al. (2017) use statistic tools to analyze the updates of participants at each iteration and discard potentially malicious updates when aggregating updates. In terms of privacy protection, Gao et al. (2021) proposed searching for privacy-preserving transformation functions and pre-processing training samples with these functions in order to defend against reconstruction attacks as well as preserve the accuracy of the trained DL models. Multiple defenses (Ma et al. 2022b; Grama et al. 2020; Naseri et al. 2020; Qi et al. 2021; Liu et al. 2021) also proposed robust and privacy-preserving defenses to protect against attacks to both integrity and privacy.

A number of surveys (Lyu et al. 2020a, b; Mothukuri et al. 2021; Zhang et al. 2018a; Liu et al. 2019a; Vepakomma et al. 2018; Kairouz et al. 2019; Enthoven and Al-Ars 2020; Yang et al. 2020) have compiled some of the threats and defenses associated with collaborative learning. However, as indicated in Table 1 they have a number of drawbacks. Firstly, the majority of them exclusively investigate certain subfields of collaborative learning and lack a complete and systematic investigation of other collaborative learning systems. Several studies, for instance, Lyu et al. (2020a, b), Enthoven and Al-Ars (2020) focus primarily on the threats and defenses in federated learning. Vepakomma et al. (2018) provide an overview of the privacy issues and countermeasures in distributed learning systems. Secondly, present surveys do not focus on the training process of collaborative learning systems (the most crucial stage) and selectively introduce existing threats and defenses, rendering them incapable of adequately summarizing cutting-edge techniques.

This work endeavors to fill existing knowledge lacunae in the collaborative learning domain. Our exhaustive exploration and systematic assessment of security and privacy impediments offer a fresh vantage point, transcending the scope of preceding surveys. We anticipate this survey to act as a touchstone for academicians and industry experts alike, aiding them in unraveling the intricacies of collaborative learning and assuring the secure and efficient application of AI models in tangible settings.

This survey provides a systematic and comprehensive evaluation of security and privacy studies in collaborative training, contrasting with prior surveys that focus on a single collaborative learning system. Our contributions are as follows:

Table 1 Comparison of our survey with other existing surveys

Survey	System	Integrity		Privacy		Stages	
		Byzantine	Backdoor	Membership	Property	Sample	Training Inference
Lyu et al. (2020a)	●	●	●	●	●	●	●
Lyu et al. (2020b)	●	●	●	●	●	●	●
Mothukuri et al. (2021)	●	●	●	●	●	●	●
Zhang et al. (2018a)	●	○	○	●	●	●	●
Kairouz et al. (2019)	●	●	●	●	●	●	●
Enthoven and Al-Ars (2020)	●	●	●	●	●	●	●
Yang et al. (2020)	●	●	○	○	○	○	○
Ours	●	●	●	●	●	●	●
● Federated only ● Distributed only		● Threat only	● Weak defense				

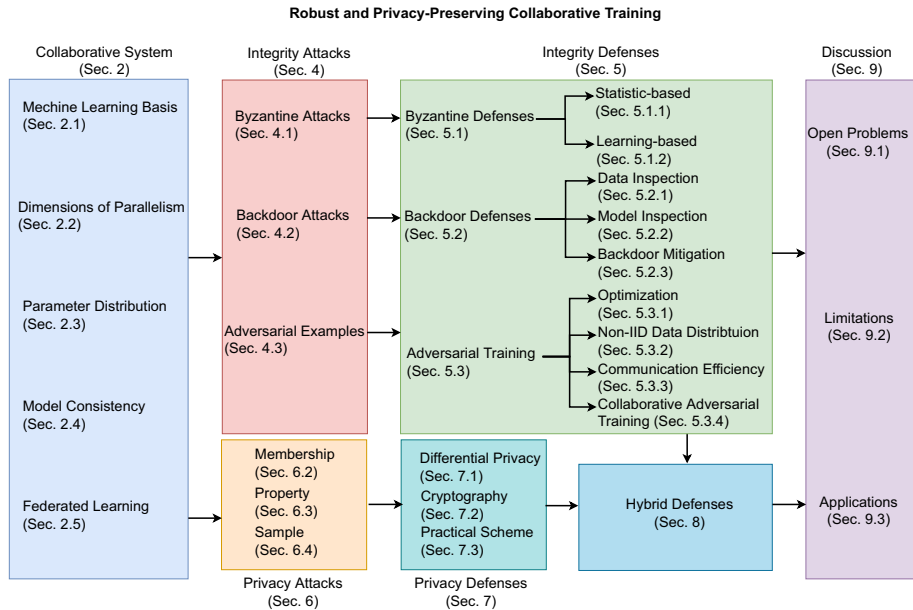


Fig. 1 Overview of threats and defenses in collaborative learning

- We provide an exhaustive exploration and systematic assessment of security and privacy impediments in collaborative learning, transcending the scope of preceding surveys.
- We summarize the integrity and privacy risks of collaborative learning systems, describing state-of-the-art integrity attacks (e.g., Byzantine, backdoor, and adversarial attacks) and privacy attacks (e.g., membership, property, and sample inference attacks), as well as the associated countermeasures.
- By shedding light on prospective challenges and their solutions, we chart a path towards a fortified, privacy-centric, and inclusive collaborative AI future.

In this paper, we examine the integrity and privacy attacks and defenses during the training process of collaborative learning, as well as the state-of-the-art remedies. The overview of threats and defenses in collaborative learning is presented in Fig. 1. Specifically, in Sect. 2, we systematically introduce different forms of collaborative learning systems from distinct perspectives. Then, in Sect. 3, we describe the privacy and integrity threats in collaborative learning. On the one hand, we exhibit existing integrity attacks and the corresponding defenses in Sects. 4 and 5, respectively. On the other hand, we show the state-of-the-art privacy attacks and the corresponding defenses in Sects. 6 and 7, respectively. We present a summary of hybrid defense approaches for achieving robust and privacy-preserving collaborative learning in Sect. 8. We highlight various open problems and prospective solutions in collaborative learning in Sect. 9.1. We also include limitations and applications of the proposed work in Sects. 9.2, 9.3, followed by Sect. 10 that concludes this paper.

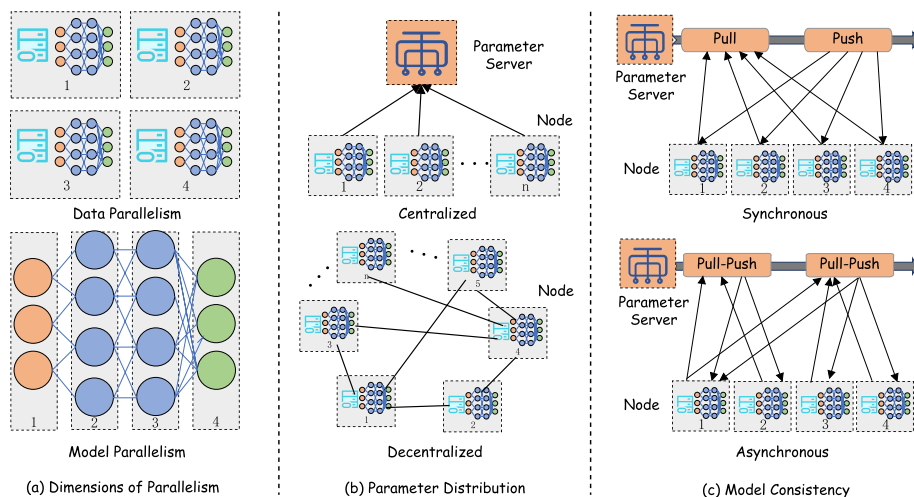


Fig. 2 Collaborative learning systems

2 System overview

2.1 Machine learning basis

We use a *dataset* $\mathcal{D}$ to denote a probability distribution of data; $z \sim \mathcal{D}$ denotes a random sampled variable z from $\mathcal{D}$, and $\mathbb{E}_{z \sim \mathcal{D}}[f(\xi)]$ denotes the expected value of $f(\xi)$ for a random variable ξ . For a deep learning model, $w \in \mathbb{R}^d$ is the d -dimensional parameter vector to estimate the model; $L_{\mathcal{D}}(f)$ be the loss calculated by f on dataset $\mathcal{D}$; l is the loss function of a single sample. Therefore, we represent the goal of machine learning as the following optimization problem:

$$w^* = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} L_{\mathcal{D}}(f_w) = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \mathbb{E}_{\xi \sim \mathcal{D}} [l(w, \xi)]. \quad (1)$$

There are numerous techniques (Battiti 1992) for minimizing the loss function, including gradient descent, second-order methods, evolutionary algorithms, etc. In machine learning, optimization is majored performed via gradient descent. We can apply *Stochastic Gradient Descent* (SGD) (Goyal et al. 2017) by sampling data at random in each iteration to optimize Eq. 1.

2.2 Dimensions of parallelism

Due to the increase in the number of models and datasets, machine learning has expanded fast in the past decade. Thanks to increasingly complex models and larger and larger datasets, machine learning algorithms have progressed significantly. Therefore, parallelism is used to give machine learning algorithms with scalability. As illustrated in Fig. 2, parallel training enables users to distribute data and calculation activities over various processing

resources, such as cores and devices. By the dimensions of parallelism, there are four major partitioning strategies: data parallelism, model parallelism, pipelining, and hybrid parallelism.

2.2.1 Data parallelism

As depicted in the upper figure of Fig. 2a, the technique for data parallelism (Krizhevsky et al. 2012b) is to partition the samples from the dataset among multiple computational resources (cores or devices). This approach is the predominant training strategy for distributed deep neural networks. An illustrative example of data parallelism is in training large-scale image classification models, such as ResNet (He et al. 2016). Different sets of images are distributed across multiple GPUs, allowing for simultaneous processing, thus accelerating the overall training time.

2.2.2 Model parallelism

Data parallelism (the top figure of Fig. 2a) can be rendered difficult or inefficient by extremely large models due to the memory required to store parameters and activations and the time required to synchronize parameters. Model parallelism (Dean et al. 2012) is introduced to address the aforementioned issues. Model parallelism involves dividing the model into various computational resources. It splits computational work based on the number of neurons present in each layer. In addition, the sample minibatch is replicated on all processors, and distinct portions of the model are executed on each processor. For instance, the training of Transformer architectures, especially those with extensive layers such as GPT series (Brown et al. 2020; Ouyang et al. 2022), leverages model parallelism. Certain layers or tensors might be offloaded onto one GPU while others are processed by a separate GPU. Such distribution not only alleviates memory constraints but also harnesses the concurrent processing power of multiple devices.

2.2.3 Pipelining

In machine learning, pipelining can refer to either overlapping calculations between layers or splitting the DNN models according to depth and assigning layers to individual processors. Therefore, pipelining is both a form of data parallelism as samples are processed by the network in parallel, and a form of model parallelism as models are partitioned by layers.

The forward evaluation, backpropagation, and weight-updating operations can be overlapped using a standard pipelining approach, which minimizes the idle time of the processor. Pipelining can alternatively be viewed as layer partitioning; each processor handles a certain layer, and the data flow is predetermined throughout the entire procedure. A practical use case of pipelining can be observed in deep learning frameworks, such as PipeDream (Narayanan et al. 2019), where layers of a neural network, especially those with different computational complexities, are designated to distinct

processors. By overlapping forward passes, backpropagation, and weight updates, the system ensures efficient utilization of resources, thereby mitigating any potential idle times.

2.2.4 Hybrid parallelism

Hybrid parallelism combines multiple parallelism schemes. In AlexNet, for instance, it is effective to apply data parallelism to the convolutional layer, where the majority of calculations are conducted, and model parallelism to the fully connected layer, where the majority of parameters are maintained. Another notable instance is the training of language models like Megatron-LM (Shoeybi et al. 2019). Given the extensive computations required in the recurrent layers and the substantial parameters in the embedding layers, hybrid parallelism can effectively distribute these tasks by implementing data parallelism for recurrent computations and model parallelism for embeddings.

2.3 Parameter distribution

In the following, unless otherwise specified, we will always refer to data parallelism in this study, as it is the most prevalent and frequently discussed parallelization method for collaborative learning. Figure 2b shows the types of communication topology between devices, including centralized and decentralized.

2.3.1 Centralized

Most distributed learning systems use centralized topology. A typical centralized architecture is *Parameter Server* (PS) (Li et al. 2014). In a PS architecture, there could be single or multiple master nodes and multiple worker nodes. Each worker node stores a duplicate of the model and a portion of the dataset. Within a training iteration, the master node distributes the weights of the model to the workers, then every worker node randomly samples a batch of data from its data partition and calculates the gradient of the weights upon the samples. Finally, all workers send their computed results to the master, and the master updates the weights of the model based on the aggregated gradients before moving on to the subsequent iteration. We illustrate the centralized distributed learning in Algorithm 1, where $\mathcal{L}(x, y, w)$ denotes the prediction error loss, η represents the learning rate, and $\Omega(w)$ signifies the regularizer implemented to mitigate model complexity. Real-world applications of centralized learning systems include distributed training of machine translation models, where language datasets are enormous, and centralized control can help streamline the learning process (Wu et al. 2016).

Algorithm 1 Distributed subgradient descent

Task Scheduler:
 | issue LoadData () to all workers
 | **for** iteration $t=0, \dots, T$ **do**
 | | issue WorkerIterate(t) to all workers

Worker $r=1, \dots, m$:
 | **Function** LoadData ():
 | | load a part of training data $\{y_{ik}, x_{ik}\}_{k=1}^{n_r}$
 | | pull the working set $w_r^{(0)}$ from servers
 | **Function** WorkerIterate (t):
 | | gradient $g_r^{(t)} \leftarrow \sum_{k=1}^{n_r} \partial \ell(x_{ik}, y_{ik}, w_r^{(t)})$
 | | push $g_r^{(t)}$ to servers
 | | pull $w_r^{(t+1)}$ from servers

Servers:
 | **Function** ServerIterate (t):
 | | aggregate $g^{(t)} \leftarrow \sum_{r=1}^m g_r^{(t)}$
 | | $w^{(t+1)} \leftarrow w^{(t)} - \eta(g^{(t)} + \partial \Omega(w^{(t)}))$

Output $w^{(T+1)}$

2.3.2 Decentralized

Due to a communication bottleneck on the master node, the scalability of a centralized distributed learning architecture is constrained. A decentralized network topology is presented as a solution for this issue. Here, we classify the prevalent decentralized approach to ring topology and decentralized topology in general.

Baidu¹ introduces ring topology to decentralized distributed learning to execute the all-reduce operation, which is inspired by the ring all-reduce algorithm from the network community. Later, Nvidia successfully implements ring all-reduce in their GPU collective communication library (NCCL).²

General decentralized topology may be demonstrated with a weighted undirected graph (V, W) , where $V = \{1, 2, \dots, n\}$ represents the set of nodes, with $W \in \mathbb{R}^{n \times n}$, satisfying $w_{i,j} \in [0, 1]$, $w_{ij} = w_{ji}$ and $\sum_j w_{ij} = 1$. The decentralized learning process can be viewed as an optimization problem that minimizes the average expectation of the loss function over all nodes, as follows:

$$\operatorname{argmin}_{w \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(x; \xi). \quad (2)$$

Decentralized parallel stochastic gradient descent (D-PSGD) (Lian et al. 2017) is the most widely utilized algorithm in decentralized distributed learning. Here, we illustrate the D-PSGD in Algorithm 2. Decentralized systems are particularly advantageous in IoT environments. For example, a network of sensors across a city for monitoring air quality might

¹ <https://github.com/baidu-research/baidu-allreduce>.

² <https://github.com/NVIDIA/nccl>.

employ decentralized learning to train local models on each device while ensuring overall model coherence (Shi et al. 2016).

Algorithm 2 Decentralized parallel stochastic gradient descent on the i -th node

Input: initial point $x_{0,i} = x_0$, step length η , weight matrix W , and the number of iterations K

1 **for** $t = 0, \dots, K - 1$ **do**

2 Randomly sample $\xi_{t,i}$ from local data of the i -th node

$\forall i$ Compute the local stochastic gradient $\partial \ell(x_{it}, y_{it}, w_r^{(t)})$

 Compute the neighborhood weighted average by fetching optimization variables from neighbors: $x_{t+\frac{1}{2},i} = \sum_{j=1}^n w_{ij} x_{t,j}$

 Update the local optimization variable $x_{t+1,i} \leftarrow x_{t+\frac{1}{2},i} - \eta \partial \ell(x_{it}, y_{it}, w_r^{(t)})$.

Output: $\frac{1}{n} \sum_{i=1}^n x_{K,i}$

2.4 Model consistency

The objective of collaborative learning is to train a single copy of model parameter w from multiple participants. However, as demonstrated in Fig. 2c, due to the possibility of multiple instances of SGD running independently on separate nodes, the model parameter is updated simultaneously by numerous nodes. Therefore, several strategies are applied to ensure the consistency of the model.

2.4.1 Synchronous

A straightforward method for updating the model is to employ a *synchronized* strategy. For each training iteration, every participant synchronizes their parameters. For instance, in Spark (Moritz et al. 2016), a master node aggregates the parameters after all the worker nodes complete the calculation for one batch of data. This strategy ensures a strong consistency of the model, however, it results in a low utilization of processing power because a node that finishes early must wait until all other nodes complete their computations. Therefore, synchronous strategies are commonly employed in controlled environments such as data centers, where uniform computational capability and network latency can be guaranteed. This approach is prevalent in scenarios like training very deep neural networks where consistency across iterations is critical (Goyal et al. 2017).

2.4.2 Asynchronous

An *asynchronized* model updating strategy maximizes the usage of the computational resources. For instance, in Parameter Server (Li et al. 2014), a worker node pushes its result to the server and pulls the current parameter without waiting for other nodes. Consequently, the strategy eliminates the waiting time of a node. Asynchronous models are suitable for environments with variable computational capabilities, like a mix of edge devices and cloud servers. Applications in mobile health monitoring, where devices like smartwatches and smartphones collect data and contribute to model training, often employ asynchronous strategies for efficiency (Konečný et al. 2016).

2.5 Federated learning

Federated learning (Li et al. 2021b) is a rapidly growing research area in recent years. It is a strategy for machine learning that trains an algorithm on numerous centralized or decentralized edge devices or servers keeping local data samples without exchanging data. Data are not expected to be uploaded to servers in federated learning, nor are local data samples considered to be uniformly distributed. Federated learning enables numerous nodes to construct a unified, robust machine learning model without sharing data, hence addressing crucial concerns such as data privacy, data security, data access rights, and heterogeneous data.

The most general federated learning training procedure is FedAvg (McMahan et al. 2017), which unites enormous clients with one central server. Each training iteration consists of four components: (1) the server first selects a subset of clients before distributing the weights of the global model to these clients; (2) selected clients receive model weights from the server and update the local model with their dataset; (3) all clients send their model weights to server; (4) the server aggregates model weights and updates the global model. The detail is illustrated in Algorithm 3.

Algorithm 3 FedAvg training procedure at the t -th iteration

Input: central server S , K clients are indexed by k , number of samples n , fraction C , local batch size B , local epochs E , and learning rate η

```

3  $m \leftarrow \max(C * K, 1)$ 
   $S_t \leftarrow$  randomly sample  $n$  clients
  for  $k \in S_t$  do
4    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$ 
5  $w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$ 
  Output:  $w_{t+1}$ 

6 ClientUpdate ( $k, w$ ):
7   for  $i = 0, \dots, E - 1$  do
8     for  $b = 0, \dots, \lfloor \frac{n_k}{B} \rfloor$  do
9        $w \leftarrow w - \eta \nabla \ell(w; \{x_i, y_i\}_{i=1}^b)$ 
10  return  $w$  to server

```

Since federated learning inherits the architecture of collaborative learning, it inevitably inherits the same security vulnerabilities. In later parts, we will also elaborate on the security threats, privacy issues, attack and defense methods for federated learning.

3 Threats in collaborative training

The complexity of the learning system and the unreliability of participants or parameter servers pose severe security and privacy threats for collaborative learning, notwithstanding its impressive results in a variety of domains. The underlying enemies in thousands of participants are more difficult to detect and defend against, making these security issues worse than those of standalone learning systems. We classify existing threats into two categories based to the objective of adversaries: integrity and privacy threats.

3.1 Integrity threats

Model integrity necessitates the accuracy and completeness of trained models, which offers the challenges of modifying or manipulating the models. It is the fundamental prerequisite during training and the implementation of deep learning in practice. Recent studies have shown, however, that in collaborative learning scenarios a single malicious participant can affect or even control the entire model training procedure (Blanchard et al. 2017; Guo et al. 2021b).

Compromise vs. backdoor vs. adversarial examples According to the associated adversarial goals, attacks for subverting the integrity of collaborative learning may be divided into three categories: compromise, backdoor, and adversarial examples. The objective of a compromising attack is to degrade or destroy the trained model's performance by modifying model parameters, which generally prevents the shared model from convergent to satisfactory during the training phase. It can also be caused by system problems such as system failures, network congestion, however in the following sections, we will solely discuss adversarial manipulations.

Byzantine attacks (Blanchard et al. 2017; Baruch et al. 2019; Bhagoji et al. 2019; Fang et al. 2020; Shejwalkar and Houmansadr 2021) can achieve such adversarial goals, in which some participants within the collaborative learning system engage in inappropriate behaviors and propagate false information, leading to the failure of the learning system. To illustrate a Byzantine attack with a real-world analogy, consider a team collaborating online to produce a research paper. If a member intentionally spreads misinformation or makes conflicting edits, it disrupts the collective effort. Similarly, in collaborative learning, Byzantine attackers can provide misleading data or model updates, making the aggregation process problematic. For instance, in distributed systems, a few compromised nodes might provide false system metrics, leading the entire network to make inefficient or harmful decisions. In machine learning, such behavior can prevent models from converging or lead them to make incorrect predictions (Shi et al. 2022).

Backdoor attacks attempt to inject predefined malicious training samples, i.e., backdoors, into a victim model while preserving the performance of the primary task (Gu et al. 2017; Huang et al. 2020b; Ji et al. 2017; Liu et al. 2018; Nguyen et al. 2020; Shafahi et al. 2018; Sun et al. 2020; Tolpegin et al. 2020; Wang et al. 2020b; Xie et al. 2019a; Zhao et al. 2020d). If an input sample includes the injected triggers, the backdoors would be activated. Due to the secrecy of triggers, it is challenging to recognize backdoor attacks, as a backdoored model behaves normally on regular data. Nevertheless, backdoors can cause catastrophic damage, such as allowing a model to forecast incorrectly on important samples. For a practical perspective, consider the scenario where a facial recognition system is compromised due to backdoor attacks. An adversary might introduce a trigger pattern that causes the system to recognize an irrelevant person as a specific individual, say, a celebrity or a president. In a real-world application, such misidentification could lead to unauthorized access to secured areas or false accusations (Zelenkova et al. 2022).

Adversarial examples refer to samples prepared by deliberately introducing adversarial perturbations to benign samples, which causes a victim model to provide an inaccurate class prediction with high confidence. Notably, the adversarial perturbation is typically a minor and imperceptible signal resembling additive noise; therefore, synthesized adversarial examples resemble original clean samples in appearance. In contrast to backdoor attacks that affect only a single victim model, adversarial examples can

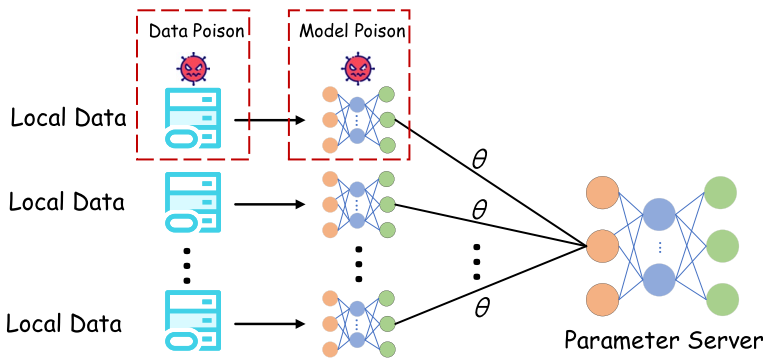


Fig. 3 Two types of attacks: data and model poisoning

be generalized to similar training objectives, such as image classification. In addition, backdoor attacks emphasize the stealth of their attacks, whereas adversarial examples emphasize their efficacy. In a real-world example, adversarial perturbations on road signs were shown to fool autonomous driving systems into misinterpreting them, posing serious safety concerns (Li et al. 2020c).

Data poisoning vs. model poisoning Two types of adversarial attacks against collaborative learning systems are data and model poisoning. In data poisoning, attackers might use carefully crafted triggers to introduce malicious samples into the training datasets of some participants (Sun et al. 2019). For instance, backdoor attacks for the image classification task contaminate training datasets with trigger-attached photos with false labels, from which the collaborative learning system learns a shortcut from the triggers to the labels. Thus, photos containing the injected triggers would be categorized according to predetermined labels. For model poisoning, attackers compromise certain participants and exert complete behavioral control over them throughout training. Then, attackers might directly alter the local model updates in order to affect the global model (Fang et al. 2020). Figure 3 depicts the two types of poisoning.

3.2 Privacy threats

A significant advantage of collaborative learning over standalone learning systems is that the instruct participant only communicates the local model update to the parameter server to ensure the privacy of training data. However, because updates are derived from training samples, they continue to convey sensitive information, making collaborative learning systems susceptible to a variety of inference attacks. For example, attackers can recover pixel-wise accuracy for images and token-wise matching for texts by analyzing the gradients transmitted at each iteration (Zhu et al. 2019b).

Membership vs. property vs. sample According to different attack goals, we can classify existing attacks into three categories: membership, property, and sample inference attacks. A membership inference attack determines, given a data record and black-box access to a model or updates, whether the record is in the model's training dataset (Guo et al. 2021a). With membership inference, an attacker can infer the presence of a specific data sample in a training dataset, which poses a severe privacy risk, particularly when the training dataset contains sensitive samples. For instance, if multiple

hospitals collaborate to train a shared model on the medical records of patients with a particular disease, a participant or the parameter server can launch a membership inference attack to infer a specific patient's health condition, which directly affects the patient's privacy (Pedarla et al. 2023).

Property inference attacks in collaborative learning (Hitaj et al. 2017; Melis et al. 2019; Wang et al. 2019b) aim to infer properties of participants' training data that are class representatives or properties that characterize the training classes. Some attacks even allow an attacker to infer when a property appears and disappears in the dataset during the training process (Melis et al. 2019). Consider a real-world scenario where multiple hospitals are collaboratively training a model on patient data. While individual patient details might be hidden, a property inference attack can determine whether a majority of the patients in a particular hospital's dataset suffer from a specific condition, such as diabetes or heart disease. This could inadvertently reveal sensitive health trends specific to a locality or community (Naveed et al. 2015).

Sample inference attacks (Geiping et al. 2020; Lam et al. 2021) attempt to extract both the training data and their labels when attackers obtain model updates during the training phase. Recent research first generates a dummy sample, then uses an optimization method to gradually reduce the distance between the dummy sample and the grand truth (Zhu et al. 2019b; Zhao et al. 2020a). To provide a tangible example of a sample inference attack, suppose a malicious entity gains access to the model updates during this collaborative process. Leveraging sample inference techniques, this entity could potentially reconstruct a patient's medical profile, extracting detailed features such as medical history, lab results, and even genetic information. This exposure would be a significant breach of patient confidentiality and could lead to various ethical and legal implications (Jagannatha et al. 2021).

Passive vs. active According to the basis of the behavior of adversaries, we classify privacy attacks in collaborative learning into two categories: passive and active attacks. In passive mode, the attacker can only witness the authentic calculations performed by the training algorithm and the model, observe the updates, and execute the aggregation operator without affecting the collaborative training method. In the active model, the attacker is permitted to perform any action during training. As a participant, for instance, the attacker can maliciously alter his parameter uploads. In order to boost his weights during aggregation, he may also send false information to the parameter server(s) or his neighborhoods. A global attacker (a parameter server) can manipulate the update participants at each iteration and modify the aggregate parameters supplied to the target participant(s) using an adversarial assault. Active attackers can be further categorised based on whether or not they have accomplices: single attackers conduct attacks alone, whereas byzantine attackers interact and share information with their accomplices. Byzantine attackers can coordinate to execute the most effective strikes. The attackers may be participants with shared interests or a hostile enemy in charge.

4 Integrity attacks

In this section, we summarize the collaborative learning attacks that compromise the integrity of trained global models. We elaborate on Byzantine and backdoor, two typical forms of attacks. We include the most prevalent integrity attack algorithms in Table 2.

Table 2 Taxonomy of byzantine and backdoor attacks

Methods	Parallelism		Consistency		Poisoning		Framework
	Data	Model	Sync	Async	Data poisoning	Model poisoning	
Byzantine							
Jagielski et al. (2018)	–	–	–	–	✓		Standalone
Muñoz-González et al. (2017)	–	–	–	–	✓		Standalone
Blanchard et al. (2017)	✓		✓			✓	Centralized
Baruch et al. (2019)	✓		✓			✓	Centralized
Shejwalkar and Houmansadr (2021)	✓		✓			✓	Federated
Backdoor							
Gu et al. (2017)	–	–	–	–	Unclean		Standalone
Shafahi et al. (2018)	–	–	–	–	Clean		Standalone
Zhu et al. (2019a)	–	–	–	–	Clean		Standalone
Huang et al. (2020b)	–	–	–	–	Clean		Standalone
Turner et al. (2018)	–	–	–	–	Clean		Standalone
Zhao et al. (2020d)	–	–	–	–	Clean		Standalone
Tao et al. (2022)	–	–	–	–	Unclean		Standalone
Zhao et al. (2022b)	–	–	–	–	Unclean		Standalone
Nguyen et al. (2020), Wang et al. (2020b)	✓		✓		Clean		Federated
Tolpegin et al. (2020)	✓		✓		Unclean		Federated
Xie et al. (2019a)	✓		✓		Unclean		Federated
Sun et al. (2020)	✓		✓		Unclean		Federated
Sun et al. (2019)	✓		✓			✓	Federated
Bagdasaryan et al. (2020)	✓		✓			✓	Federated
Fang et al. (2020)	✓		✓			✓	Federated
Bhagoji et al. (2019)	✓		✓			✓	Federated

Table 3 Symbol description

Global	
S	Central server
G^t	Global model at iteration t
η	Global learning rate
K	Number of clients
w^t	Weights of G^t
n	Number of all data samples
C	Selected fraction
$\mathcal{F}$	Aggregate rule
V_a^t	Aggregated updates at iteration t
$V_{i,j}^t$	The value of the j -th dimension of V_i^t
$V_{:j}^t$	The value list of the j -th dimension of $V_{[n]}^t$
$U(w^t, V_a^t, \rho)$	Weights update function
Local	
D_k	Local dataset on k -th client
E	Local epochs
B	Local batch size
ℓ	Loss function
n_k	Dataset size of D_k
$\{x_i, y_i\}_k$	Samples of the dataset on client k
V_i^t	Update from client i at iteration t
V_{mal}	Malicious update
V_{ben}	Benign update

4.1 Byzantine attacks

Although data poisoning has demonstrated a significant impact on stand-alone model training systems (Muñoz-González et al. 2017; Jagielski et al. 2018), recent studies show that model poisoning is much more effective than data poisoning against Byzantine attacks in collaborative learning scenarios (Bhagoji et al. 2019; Baruch et al. 2019). Intuitively, model poisoning and data poisoning are both hypothesized to try to change the weights of local models. Clearly, the former has a greater immediate effect.

Byzantine attacks presuppose that the attacker has the authorization to view and modify updates from multiple participants in a collaborative learning system. We refer to the modified updates as malicious updates. For illustrative purposes, the symbol description is provided in Table 3. It is straightforward to implement a Denial-Of-Service attack in the average collaborative learning method by transmitting a linear mix of a malicious update and other benign updates (Blanchard et al. 2017). As shown in Eq. 3, where $\mathcal{F}$ is weight sum and λ_i 's are non-zero scalars, a single Byzantine attacker with knowledge of all updates from other clients may force the averaged update to be replaced by an arbitrary vector $U \in \mathbb{R}^d$.

$$V_{mal} = V_n = \frac{1}{\lambda_n} \cdot U - \sum_{i=1}^{n-1} \frac{\lambda_i}{\lambda_n} V_i$$

$$\mathcal{F}(V_1, \dots, V_n) = \sum_{i=1}^n \lambda_i \cdot V_i. \quad (3)$$

Nevertheless, this basic attack might be easily filtered out, as the magnitude of the linear combination frequently differs from that of benign ones. Alternately, given that model updates form a high-dimensional vector, it is possible to generate malicious updates by drifting innocuous updates with a constrained value. The overall procedure can be described by Eq. 4. Attackers attempt to add the largest scale perturbation under specific constraints to the benign update statistics. $\mathcal{H}$ represents statistical functions such as mean or median. For adversaries with just partial knowledge of benign updates, he could estimate the statistics of whole benign updates using the original updates on malicious clients.

$$V_{mal} = \tilde{V}_{ben} + \text{Max}\{\text{Constrain}(P)\}, \tilde{V}_{ben} = \mathcal{H}(V_{ben}). \quad (4)$$

Baruch et al. (2019) demonstrate that slight perturbations are sufficient to circumvent magnitude-based defense policies. In Eq. 5, it use Cumulative Standard Normal Function ϕ to limit the size of factor perturbation z , where n, f, μ_j, σ_j are the total clients number, the Byzantine clients number, the benign updates mean and the standard deviation of the j dimension. During their experiment, it shows a nearly 50% accuracy decline with one-fifth of malicious clients.

$$V_{mal,j} = \mu_j - z^{max} \cdot \sigma_j$$

$$z^{max} = \max_z \left(\phi(z) < \frac{n - 2f - \lfloor \frac{n}{2} + 1 \rfloor}{n - f} \right). \quad (5)$$

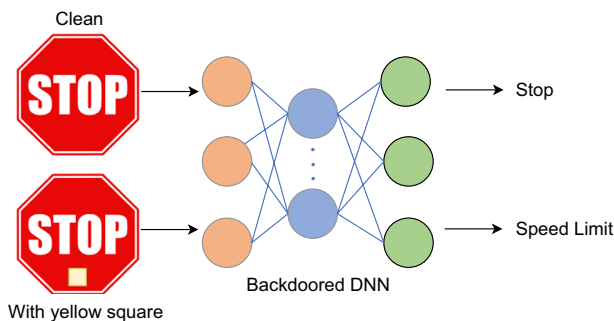
In a more relaxed setting, attackers could launch a more damaging version of updates if they know the aggregation rules of the server (Fang et al. 2020; Shejwalkar and Houmansadr 2021). This setting is reasonable in various scenarios, for example, the provider of the server may make the aggregation rule public for attracting potential participants (McMahan et al. 2017).

$$\underset{\lambda}{\operatorname{argmax}} V_{mal} = \mathcal{F}(V_{mal_1}, \dots, V_{mal_f}, V_{f+1}, \dots, V_n)$$

$$V_{mal} = V_{mal_1} = \dots = V_{mal_f} = \mu - \lambda \cdot \text{sign}(\sigma). \quad (6)$$

Equation 6 shows the defense-specific Byzantine attack for Krum (Blanchard et al. 2017) presented by Fang et al. (2020). It constructs malicious updates by deviating the mean of the benign updates along the sign of the standard deviation. λ is initialized with a big value and decreased iteratively by a constant factor until the Byzantine-robust aggregation rule selects a malicious update. Furthermore, attacks on other defenses follow the same iterative process, albeit malicious update building may vary. Shejwalkar and Houmansadr (2021) strengthened this attack by locating an approximate maximum of λ , which achieves a slightly more severe accuracy decline but usually incurs dozens of extra computation costs.

Fig. 4 Visible backdoor trigger on traffic sign (Gu et al. 2017). (Color figure online)



4.2 Backdoor attacks

4.2.1 Data poisoning

We first introduce data poisoning in stand-alone backdoor attacks. A backdoor could be embedded in the neural networks trained by a compromised dataset (Ji et al. 2017; Liu et al. 2018). The methods for injecting backdoors through data poisoning presume that the attacker has control over a substantial portion of the training data. Consequently, backdoor attacks alter the behaviour of the model only on specific attacker-chosen inputs via data poisoning (Liu et al. 2018; Gu et al. 2017). These techniques could be categorized into two classes: unclean and clean label stand-alone backdoors.

$$\min_w \sum_{(x,y) \in \mathcal{D}_c} \alpha \ell(G(x), y) + \sum_{(x,y) \in \mathcal{D}_p} \beta \ell(G(T_p(x)), y_t) \quad (7)$$

$$T_p(x) = x + p$$

The process of unclean label stand-alone backdoor could be illustrated by Eq. 7. T_p is the backdoor injection function that generates the poison sample by introducing a certain perturbation p into the clean sample. The adversary introduces some poison samples with modified target labels y_t into the original dataset. Therefore, the optimization objective function of model training covers the performance on both clean dataset $\mathcal{D}_c$ and poison dataset $\mathcal{D}_p$.

For example, Gu et al. (2017) proposed the BadNets model, which injects a visible trigger pattern into a collection of randomly chosen training images. As demonstrated in Fig. 4, the stop sign with a yellow square patch would be misclassified as a speed-limit sign. Most studies use an optimization-based method to progressively build an imperceptible trigger, despite explicitly attaching a visible trigger to clean samples. In particular, they employ similarity measure approaches to restrict the difference between the clean sample and the poison sample. Therefore, the creation of a trigger could be described in Eq. 8:

$$p = \min_p \sum_{(x,y) \in \mathcal{D}_p} \ell(G(T_p(x)), y_t) + d(T_p(x), x). \quad (8)$$

Wang et al. (2019a) expressed poison samples as $T_p(x) = (1 - m) \cdot x + m \cdot p$, where m donate the mask. The l_1 -norm of the mask was then used to measure the magnitude of modification, $d(T_p(x), x) = |m|$. Zhao et al. (2022b) used l_2 -norm distance on image-pixel space ($d(T_p(x), x) = \|T_p(x) - x\|_2$) and introduced extra latent feature constraint in model

training to strengthen the backdoor embedding. Tao et al. (2022) decomposed perturbation on each pixel into positive and negative changes via $\tanh$ function: $T_p(x) = \text{clip}(x + \frac{1}{2}(\tanh(p_{pos}) - \tanh(p_{neg})) \cdot \text{maxp})$. $p_{pos}, p_{neg} \in (-\infty, +\infty)$ donate positive and negative perturbation, respectively, and maxp donates the maximum pixel value. In accordance with Wang et al. (2019a), the l_1 -norm of mask equals $\sum_{h,w} (\frac{1}{2}(\tanh(b_{neg}) + 1)) + (\frac{1}{2}(\tanh(b_{neg}) + 1))$. In addition to extra constraints, the invisible trigger could be generated through a DNN model. For example, Li et al. (2021d) added sample-specific noise into the selected images using DNN-based image steganography (Baluja 2017; Zhu et al. 2018; Tancik et al. 2020). The image steganography model consists of an encoder and decoder. The encoder is trained to embed a specific string into the input image in a non-perceptible way and the decoder is trained to recover the string information from the embedded image. They trained such a network on clean samples or directly adopted a pre-trained encoder to embed target labels into clean samples.

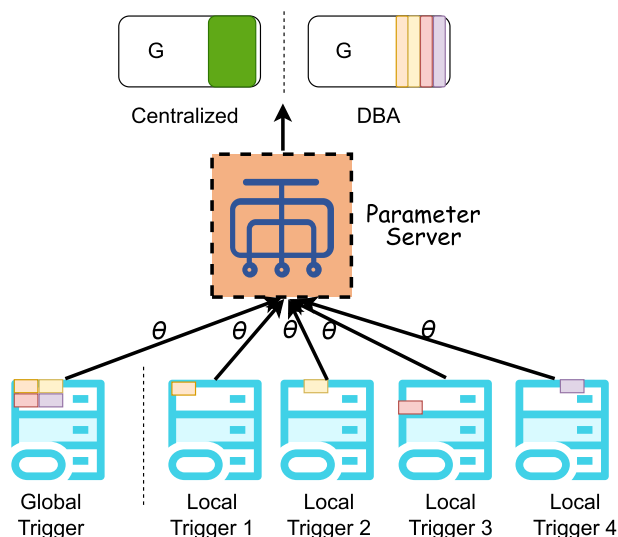
Since the poisoned images are mislabeled, unclear label attacks can be easily detected by simple data filtering or human inspection (Zhao et al. 2020d). The clean label stand-alone backdoor is therefore offered. It assumes the adversary cannot alter the labels of any training samples and keeps the labels of poisoned samples. Visually, the tampered samples are comparable to the beginning ones. For example, Shafahi et al. (2018) explored poisoning attacks on neural networks and presented an optimization-based feature collision attack method for crafting poisons. Concretely, the poison sample has the same appearance as the clean sample, while it collides in feature space with the target class sample. The generation process of the poison example is depicted below.

$$\begin{aligned}\hat{x}_i &= x_{i-1} - \eta \nabla_x \mathcal{L}(G(x_{i-1}), G(x_i)) \\ x_i &= (\hat{x}_i + \eta \beta x_c) / (1 + \beta \eta) \\ \mathcal{L}(x_1, x_2) &= \|x_1 - x_2\|_2, \quad x_0 = x_c,\end{aligned}\tag{9}$$

where x_t, x_c donate the sample of the target class and the clean sample, respectively; β controls the similarity between the poison sample and clean sample. After model training, samples with target class features might be misclassified into the corresponding class of the clean sample. Experiments demonstrate that a single poison image can alter the behaviour of a classifier using transfer learning. However, the method proposed by Shafahi et al. (2018) requires complete or query access to the victim model. Then, Zhu et al. (2019a) assumed the victim model is not accessible to the attacker and proposed a new convex polytope attack in which poison images are designed to surround the targeted image in the feature space.

Soon afterward, Huang et al. (2020b) demonstrated that feature collision and convex polytopes attacks only work on fine-tuning and transfer learning pipelines, however, they fail when the victim trains their model from scratch. Furthermore, they are not general-purpose, meaning that an attacker may have objectives beyond a limited number of targets. To solve these difficulties, Huang et al. (2020b) proposed a MetaPoison algorithm for crafting poison images that manipulate the victim's training pipeline in order to achieve arbitrary model behaviors. It is a bi-level optimization problem, where the inner level corresponds to training a network on a poisoned dataset and the outer level corresponds to updating those poisons to achieve a desired behavior on the trained model. In addition, Turner et al. (2018) introduced two techniques to strengthen the backdoor attack, including latent space interpolation using GANs and adversarial perturbations bounded by l_p -norm.

Fig. 5 Centralized trigger and distributed trigger comparison (Xie et al. 2019a). The green square is utilized to signify a global model that has been backdoored with the single global trigger in a centralized backdoor attack, where all adversaries employ the same trigger. DBA breaks down the global trigger into unique local patterns, represented by squares of different colors. (Color figure online)



Data poisoning in collaborative learning systems follows the attacks in the stand-alone setting. Tolpegin et al. (2020) investigated targeted data poisoning attacks against collaborative learning systems, in which a malicious subset of the participants aim to poison the global model by sending model updates derived from mislabeled data. However, Bagdasaryan et al. (2020) pointed out that these attacks in the stand-alone setting are not effective against collaborative learning, where the malicious model is aggregated with hundreds or thousands of benign models. In order to implement a backdoor attack in collaborative learning systems, a constrain-and-scale technique to inject a backdoor in collaborative learning is proposed (Bagdasaryan et al. 2020). Compared with previous backdoor attacks, in collaborative learning, the attacker controls the entire training process, though only for one or a few participants.

Based on the above assumption, Nguyen et al. (2020) determined that the collaborative learning based IoT intrusion detection systems are vulnerable to backdoor attacks and developed a data poisoning attack method. The core concept of this method is that it allows an adversary to implant a backdoor into the aggregated detection model to incorrectly classify malicious traffic as benign traffic. Furthermore, an adversary can gradually poison the detection model by only using compromised IoT devices to inject small amounts of malicious data into the training process. From another perspective, Wang et al. (2020b) focused on attacking algorithms that leverage data from the tail of the input data distribution. Then, they established in theory that, if a model is vulnerable to adversarial attacks, under mild conditions, backdoor attacks are unavoidable. When properly built, backdoors are difficult to detect.

Although the previously reported backdoor attacks for collaborative learning systems have good performance, they do not fully exploit the distributed learning methodology of collaborative learning since they embed the same global trigger pattern for all adversarial parties (Xie et al. 2019a). In order to take full advantage of the distributive nature of collaborative learning, Xie et al. (2019a) suggested a distributed backdoor attacking (DBA) method. As depicted in Fig. 5, DBA decomposes a global trigger pattern into distinct local patterns and embeds them into the corresponding training sets of antagonistic parties.

4.2.2 Model poisoning

In model poisoning, the training process is performed on local devices. Therefore, fully compromised clients are able to entirely alter the local model update, thereby affecting the global model. Bhagoji et al. (2019) proposed a model poisoning method, which is executed by an adversary who controls a limited number of malicious agents (often a single agent) and aims to cause the global model to misclassify a set of selected inputs with high confidence. They employed the local model weights to estimate the global weights and adopted an explicit boosting coefficient λ to strengthen the attack effect. The modified objective function of local model training is as follows, including the trigger performance in the global weights estimation, the main task performance and the stealthiness of malicious updates.

$$\underset{V_{mal}^t}{\operatorname{argmin}} \sum_{(x,y) \in \mathcal{D}_{aux}} \lambda \ell_{\hat{w}^t}(x, y_t) + \sum_{(x,y) \in \mathcal{D}_k} \ell_{w_{mal}^t}(x, y) + \rho \|V_{mal}^t - \bar{V}_{ben}^{t-1}\|. \quad (10)$$

In contrast to boosting the objective function, Bagdasaryan et al. (2020) directly scaled the malicious updates to achieve model replacement. The malicious could increase $C \cdot n$ times to cover other benign updates under the average aggregation rule. Inspired by this idea, data poison might be effectively combined with model poison. For instance, Wang et al. (2020b) first employed a PGD data poison backdoor attack to train a local malicious model and then scaled malicious updates to enhance the success rate and lasting effect of triggers.

4.3 Adversarial examples

In order to manage the perceptibility of adversarial examples, the additive adversarial perturbation $\delta \in \mathbb{R}^{h \times w \times c}$ is generally constrained by a budget ϵ . Here, h , w , c represent image height, width, and color channel, respectively. In the context of image classification, $f(x; w)$ denotes an image classifier that maps a clean image x to a discrete category label y , whereas w denotes the model parameters of the classifier. Hence, δ is optimized as follows:

$$\delta_i^* = \underset{\|\delta_i\|_p \leq \epsilon}{\operatorname{argmax}} \ell(f(x_i + \delta_i; w), y_i), \quad (11)$$

where $\ell(\cdot, \cdot)$ is the training loss function, and norm-bounded p can be 0, 1, 2 and ∞ . A specific adversarial sample x'_i of x_i is expressed as:

$$x'_i = x_i + \delta_i^*. \quad (12)$$

The formulation above underpins the most prevailing understanding of adversarial examples. Recently, unrestricted adversarial examples (Qiu et al. 2020) are proposed, which are neither conditioned to manipulate the original image nor limited to the perturbation norm budget. Nevertheless, these unrestricted adversarial examples are still perceived as clean by humans to produce the same label as benign images but fool the victim classifier.

4.3.1 Knowledge assumption

The attacking methods can be categorized based on information the adversary needs to acquire. Such knowledge acquisition often involves query access to the victim model, the

model's architecture, and its trained parameters. Therefore, these attacks can be generally categorized as white-box and black-box attacks. The white-box attack assumes that the adversary has comprehensive knowledge of the target model. In addition, if the adversary has only limited knowledge of the training process and parameters, we refer to this as a restricted knowledge white-box attack, also known as a gray-box attack. The black-box attack, on the other hand, assumes no prior knowledge of the target model, which is a stricter instance. The adversary is solely aware of the model's predictions, which may be a single or many labels with or without a single or multiple confidence score(s).

In the standalone learning system, classical white-box attacks, including the Fast Gradient Sign Method (FGSM) (Goodfellow et al. 2014), Projected Gradient Descent (PGD) (Madry et al. 2017) and Carlini & Wager (CW) (Carlini and Wagner 2018) to name a few, have access to trained parameters of the victim model, allowing the adversary to perturb the back-propagation process. In order to generate indirectly adversarial samples, certain black-box attacks conduct a huge number of queries as a result of the limited knowledge they possess. Specifically, with predicted labels output of querying, the main concept of black-box attacks (Chen and Gu 2020) is to find the classification boundary between labels, by estimating gradients according to the input data using a binary querying method, so that the adversary is able to manipulate the back-propagation process similar to white-box attacks. Other black-box attacks utilize adversarial transferability between white-box surrogate models and the target victim model to boost attack performance. Feng et al. (2022) intended to address surrogate biases by transferring partial parameters of the conditional adversarial distribution of surrogate models and then learning the remaining parameters depending on user queries. All of these attacks are also applicable to collaborative systems.

4.3.2 Evasion attack

In both standalone and collaborative learning systems, the evasion attack (Kwon et al. 2019) is launched at test time. This approach combines adversarial examples with clean test data to alter the prediction from a correct category label to a random or determined one, so destroying the original dataset's integrity. From the black-box attack vantage point, the adversary is solely aware of the dataset type and output predictions of the model. Kwon et al. (2019) attempted to generate selective audio adversarial examples, by minimizing the probability of incorrect classification by the protected classifier and that of correct classification by the victim classifier. These elaborate adversarial samples are used in the speech recognition task during the test process. Consequently, this audio attack achieves a 91.67% attack success rate measured by analyzing protected classifier accuracy. For the deep face recognition task, Hu et al. (2022) proposed an adversarial makeup transfer method, called AMT-GAN, to preserve stronger black-box transferability and better visual quality simultaneously. AMT-GAN is designed to adopt a novel regularization module to reconcile the conflicts between the adversarial noises and the visual consistency, achieving the trade-off between the attack success rate, visual changes, and identity preservation.

During white-box attacks, the adversary can recognize more important information. Checking input data for intrinsic context consistency has recently been shown to be resistant to adversarial examples. Yin et al. (2022) aimed to evade such examination, by formulating a joint optimization problem and solving three sub-optimization problems in a pipeline to generate more adaptive adversarial examples. As a result, two attack objectives are simultaneously achieved: deceiving the object detector and escaping the check system.

5 Integrity defenses

5.1 Byzantine defenses

Byzantine defense seeks to filter out malicious participants utilizing experience from updates, which may be the mean or median of updates and interaction history. Therefore, we classify known Byzantine-tolerant algorithms into two categories: learning-based and statistics-based, as summarized in Table 4.

5.1.1 Statistic-based inspection

In each iteration of training, statistically-based inspection employs anomaly detection on participants. For example, updates that deviate significantly from the average could be flagged as potential attacks. Existing research focuses mostly on two criteria: magnitude and performance. We summarize the equations of Byzantine defenses that leverage the magnitude of updates in Table 5 and $\text{sort}(\cdot)$ denotes the sorting algorithm with increasing order. Blanchard et al. (2017) proposed Krum to compute updates similarity using the Euclidean distance. It first calculates the Euclidean distance of each update from other updates and then selects the one that has the minimum sum of the distances with its closest $n - f - 2$ updates. Krum could effectively remove the malicious updates that are less than $\frac{n}{2} - 1$ and far from the benign updates. However, Krum endures a high computational overhead when computing distances of high-dimensional vectors. Hence, Yin et al. (2018) used the mean of dimensions to replace the Euclidean distance, called Trimmed Mean. It treats each update independently, sorts each dimension of updates, and removes β largest and smallest items, then calculates the mean of the remaining values as the global update. Cronus (Chang et al. 2019) and FedDF (Lin et al. 2020) shared predictions of local models on the public data to reduce updates' dimensions. In addition, Krum is not capable of malicious update that has a similar overall magnitude with benign updates but a certain dimension that varies greatly. Therefore, Guerraoui et al. (2018) proposed Bulyan, a combination of Krum and Trimmed Mean. It first runs Krum for several iterations to select a certain number of candidates, then it applies a variant of Trimmed Mean to calculate the global update. Moreover, there are also many median-based updates estimators, such as geometric median (Feng et al. 2014; Chen et al. 2017), marginal median, mean around median (Xie et al. 2018), median of means (MOM) (Tu et al. 2021) and mean of median (Fan et al. 2021). Furthermore, some researchers applied more sophisticated statistics techniques to compute update similarity. Muñoz-González et al. (2019) computed the weighted average of all updates and compute the cosine similarities between the averaged update to each update. Then, it removes updates with similarities out of a certain threshold. The threshold function $T(\cdot)$ could be the function of mean, median, and standard deviation. Shejwalkar and Houmansadr (2021) presented Dnc, which uses Singular value decomposition (SVD) and dimensionality reduction to discard outliers. It first randomly samples b dimensions from total d dimensions for dimensionality reduction and then computes the top right singular eigenvector v of centered updates V^c . An outlier score is used to filter malicious updates, which is defined as the inner product between V_i and v .

All the aforementioned magnitude methods concentrate on the scenario in which less than half of the participants are compromised. Some researchers expect to break through

Table 4 Taxonomy of Byzantine defenses

Methods	Parallelism		Consistency		Methodology	Data Distribution		Framework
	Data	Model	Sync	Async		IID	Non-IID	
Statistic								
Chen et al. (2017), Tu et al. (2021)	✓		✓		Median	✓		Centralized
Blanchard et al. (2017)	✓		✓		Euclidean	✓		Centralized
Yin et al. (2018)	✓		✓		Mean	✓		Centralized
Guerraoui et al. (2018)	✓		✓		Euclidean + median		✓	Centralized
Cao and Lai (2019)	✓		✓		Euclidean		✓	Centralized
Xie et al. (2019b)	✓		✓		Loss		✓	Centralized
Xie et al. (2020)	✓			✓	Loss		✓	Centralized
Shejwalkar and Houmansadr (2021)	✓		✓		SVD		✓	Federated
Deng et al. (2021)	✓		✓		Loss	✓		Federated
Yang and Li (2021)	✓			✓	Median/mean		✓	Centralized
Mao et al. (2021)	✓			✓	Lookahead similarity measurement		✓	Decentralized
El-Mhamdi et al. (2021)	✓			✓	Minimum-diameter averaging/median		✓	Decentralized
Guo et al. (2021b)	✓			✓	Loss	✓		Decentralized
Chang et al. (2019)	✓		✓		Dimension reduction	✓		Federated
Lin et al. (2020)	✓		✓		Dimension reduction	✓		Federated

Table 4 (continued)

Methods	Parallelism		Consistency		Methodology	Data Distribution		Framework
	Data	Model	Sync	Async		IID	Non-IID	
Learning								
Muñoz-González et al. (2019)	✓		✓		Hidden Markov model	✓		Federated
Pan et al. (2020a)	✓		✓		Reinforcement learning	✓		Centralized
Ma et al. (2021)	✓		✓		DNN	✓		Federated
Karimireddy et al. (2021)	✓		✓		Worker momentum		✓	Centralized
Li et al. (2021c)	✓		✓		PFL		✓	Federated

Table 5 Equations of statistic-based inspection

Methods	Definition	Explanation
Krum (Blanchard et al. 2017)	$Dis_i = \{\ V_i - V_j\ _2\}, i \in [n], j \in [n] \setminus i$ $Score_i = \sum_{j=0}^{n-i-2} sort(Dis_{i,j})$ $V_a = V_k, Score_k = \min(Score)$ (13) $V_{a,j} = med(\{V_{:,j}\})$ (14) $V_{a,j} = mean_\beta(\{V_{:,j}\})$ (15)	Choose updates that have the smallest Euclidean distances to their $n - f - 2$ neighbors
Median (Yin et al. 2018)		Compute the median for each dimension across all updates
Trimmed-mean (Yin et al. 2018)		Eliminate the β largest and smallest values of each dimension, and then compute the mean across all remaining updates
Bulyan (Guerraoui et al. 2018)	$V_{a,j} = Trimmed(Krum_\beta(V_{[n]}))$ (16)	Integrate the methods of Krum and Trimmed-mean
AFA (Muñoz-González et al. 2019)	$Score_i = \cosin(V_i, \tilde{V}_{[n]})$ $V_a = mean(V_i), Score_i < T(Score)$ (17)	Compute the average of updates that exhibit a closer cosine similarity to the weighted averaged updates
Dnc (Shejwalkar and Houmansadr 2021)	$\tilde{V}_{[n]} = V_{:, [b]}$ $V^c = \tilde{V}_{[n]} - \mu, \mu = mean(\tilde{V}_{[n]})$ $u, \sigma, v = SVD(V^c)$ $score_i = ((V_i - \mu, v))^2$ $V_a mean(V_{[s]}), V_{[s]} = \min_\theta(score)$ (18)	Calculate the mean of updates that display a smaller top right singular eigenvector distance compared to centered updates

the above limitation using performance evaluation (Xie et al. 2019b; Cao and Lai 2019; Deng et al. 2021). As a compromise, these methods typically require a clean dataset.

$$Score_{\rho}(V, w) = \ell_w(\{x_i, y_i\}^r) - \ell_{\hat{w}}(\{x_i, y_i\}^r) - \rho \|V\|_2 \{x_i, y_i\}^r \in \mathcal{D}_c, \hat{w} = U(w, V, \rho). \quad (19)$$

Xie et al. (2019b) proposed Zeno in which the server sorts the updates by a stochastic descendant score 19. The score is composed of the estimated descendant of the loss function on *i.i.d.* samples drawn from $\mathcal{D}_c$ and the magnitude of the update, which roughly indicates how trustworthy each participant is. The server aggregates the updates with the highest score. Zeno requires at least one benign update from all updates for proving the convergence of SGD for non-convex problems. Cao and Lai (2019) proposed an aggregation algorithm that can defend an arbitrary number of Byzantine attackers. It allows the server to compute a benign update using a small clean dataset and compares the updates from each participant with the benign update. Even though the benign update is very noisy because the scale of the clean dataset could be quite small, it is enough to filter out malicious information in experiments. Deng et al. (2021) used loss reduction between the global model and the local models to evaluate the quality of the update from each participant. Guo et al. (2021b) proposed a Uniform Byzantine-resilient Aggregation Rule (UBAR) to select the useful parameter updates and filter out the malicious ones in each training iteration. It can guarantee that each benign node in a decentralized system can train a correct model under very strong Byzantine attacks with an arbitrary number of faulty participants. Furthermore, the above algorithms also inspire Byzantine robust solutions in asynchronous distributed learning (Xie et al. 2020; Yang and Li 2021; Mao et al. 2021; El-Mhamdi et al. 2021).

5.1.2 Learning-based inspection

The learning-based inspection identifies malicious participants according to historical interactions. Typically, it involves training a model to discriminate between normal and malicious updates.

Muñoz-González et al. (2019) adopted a Hidden Markov Model to specify and learn the quality of model updates provided by each participant during training, which could enhance the accuracy and efficiency of detecting malicious updates.

Pan et al. (2020a) proposed Justinian's GAAvernor, a gradient aggregation agent which learns to be robust against Byzantine attacks via reinforcement learning. As shown in Fig. 6, the state includes the global weights, corresponding loss on a clean dataset and the clients' updates. The policy is an n -dimensional vector, which represents the aggregation weights of updates and the decrease of the loss on the clean dataset as the reward for the chosen policy. Relying on the current state and the previous policy, the algorithm could efficiently achieve Byzantine robust collaborative learning. Karimireddy et al. (2021) observed that Byzantine updates have a significant deviation for certain rounds. Inspired by El Mhamdi et al. (2021), they introduced momentum into computing benign updates and used simple iterative clipping to aggregate updates. Similarly, Ma et al. (2021) used a crafted DNN to learn the correlation of benign updates in multiple rounds, which differs from Byzantine updates. Then, the DNN is treated as a classifier to sort out Byzantine updates.

Moreover, Personalization Federated Learning (PFL) may also be used for Byzantine-resilient federated training. Each client focuses more on training the personal local model, while benefiting from the global model. PFL could improve the model performance on

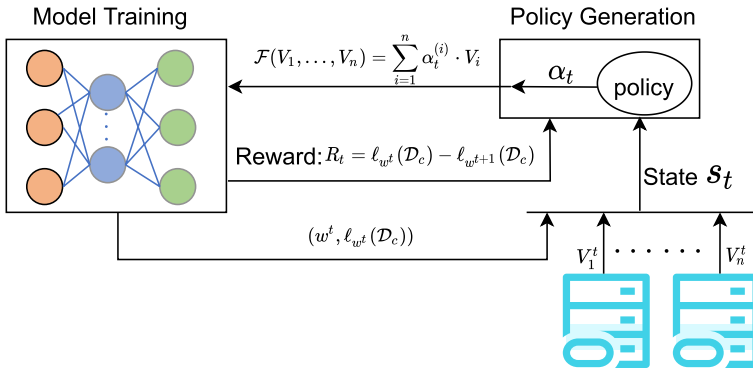


Fig. 6 Byzantine defense through reinforcement learning (Pan et al. 2020a)

clients' homogeneous local datasets and is widely used for fairness in federated learning. Meanwhile, the diverse personal local model also reduces the impact of performance degradation. Ditto (Li et al. 2021c) enabled clients to train the personalized and global model parameters in each iteration and adopt the personalized model to circumvent the potentially damaged global model. Equation 20 describes the core training process of each client. It first follows the standard procedure to calculate the model update and then optimizes personal model weights v_k through gradients on its dataset and distance from global weights.

$$\begin{aligned} w_k^t &= w^t - \eta \nabla \ell(G_{w_i}(\{x_i, y_i\}_k)) \\ v_k &= v_k - \eta (\nabla \ell(G_{v_k}(\{x_i, y_i\}_k)) + \lambda (v_k - w^t)). \end{aligned} \quad (20)$$

5.2 Backdoor defenses

To avoid or mitigate the effects of backdoor attacks on collaborative learning systems, several backdoor defense methods have been proposed (Gao et al. 2020; Qiu et al. 2021; Li et al. 2020a; Lyu et al. 2020b; Liu et al. 2022a). We divide existing methods into two categories based on the subject of inspection: data and model inspection.

5.2.1 Data inspection

Data inspection methods primarily examine whether the input data contains triggers via anomaly detection or simply excludes the anomalous samples during inference. For instance, emails with unusual patterns could be flagged as potential spam. Thudumu et al. (2020). Consequently, existing data inspection approaches for standalone learning (Tran et al. 2018; Chan and Ong 2019; Chou et al. 2018; Gao et al. 2019; Truong et al. 2020; Li et al. 2020a) are applicable for well-trained models by collaborative learning systems. The simplest method for identifying poison samples is to observe their anomalous behavior. As previously indicated, a model with backdoors will identify all samples with a particular trigger as belonging to one label, which is statistically implausible. Gao et al. (2019) proposed STRong Intentional Perturbation (STRIP), a run-time Trojan attack detection system. Figure 7 poison samples detection process. In particular, they deliberately disturb the incoming input and observe the unpredictability of predicted classes for perturbed inputs

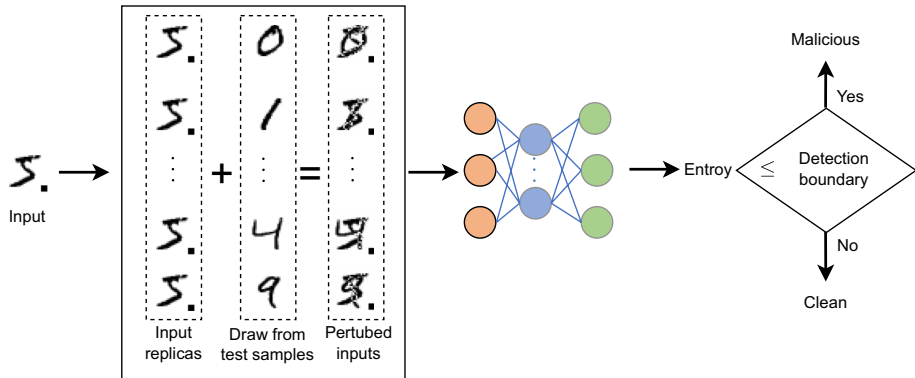


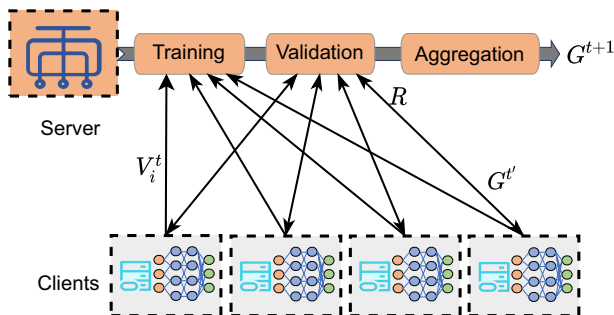
Fig. 7 Poison samples detection process of STRIP (Gao et al. 2019)

from a particular deployed model. Low entropy in predicted classes violates the input-dependence property of a benign model and suggests the presence of a malicious input from a Trojan input feature. The identical argument is used to defend the backdoor in the NLP task (Azizi et al. 2021).

Some researchers investigated the intra-representation difference between poison samples and clean samples, namely activation and gradient, in addition to the abnormal model output of poison samples. Tran et al. (2018) demonstrated that the feature representations of poison samples from deeper layers are progressively easier to distinguish. Similar to Shejwalkar and Houmansadr (2021), they computed the outlier score using SVD based on the representation from the most recent few layers and deleted samples with high outlier scores. Chen et al. (2018) observed that the output of the last hidden layer reflects the high-level features used for decision-making by the neural network, and they suggested an Activation Clustering (AC) approach for detecting backdoor attacks. Given the collected data and the model, AC (Chen et al. 2018) detects and removes small-sized poisoned samples by clustering the outputs of the classifiers to separate poisoned samples. Chan and Ong (2019) demonstrated that a triggered sample can result in a rather high absolute gradient value in the input layer at the trigger position. Consequently, they emphasized that trigger samples can be separated from clean samples using a clustering algorithm. Chou et al. (2018) proposed the SentiNet method, which is a novel detection framework for localized universal attacks on neural networks. It exploits the model explanation and object detection techniques to identify contiguous regions which are assumed to have a high probability of possessing a trigger when it strongly affects the classification.

5.2.2 Model inspection

Data inspection defenses attempt to identify poisoned data from regular data, whereas the model inspection approach (Ma and Liu 2019; Liu et al. 2019c) focuses on anomaly techniques to distinguish abnormal behaviour of the models induced by backdoors (Gao et al. 2020). For instance, unusually high weights in a neural network could indicate a potential attack (Guo et al. 2022). These defenses may be carried out either during or after the training processing. For model inspection for well-trained models, Wang et al. (2019a) proposed Neural Cleanse to detect whether or not a DNN model has been subjected to a backdoor attack prior to deployment, based on the intuition that all input samples require

Fig. 8 Client validation backdoor defense process

much smaller modifications to be misclassified into targeted class. Therefore, they compared the modifications made to each class and examined whether any classes require only a minor modification to be misclassified. Taking advantage of output explanation techniques, Huang et al. (2019) proposed Neuron Inspect to identify backdoor attacks by outlier detection based on the heatmap of the output layer. Liu et al. (2019c) proposed Artificial Brain Stimulation to detect backdoors by analyzing the inner neuron behaviors through a stimulation method. They hypothesized that the backdoor behavior is represented by one or a group of inner neurons that would produce significantly greater activation values when their inputs fell within a certain value range. Therefore, they altered the inputs of certain neurons and analyzed their variation curves for mutations. Nevertheless, Chen et al. (2019) pointed out that it is indispensable to inspect whether a pre-trained DNN has been polluted before employing a model. Hence, they proposed DeepInspect, a black-box Trojan detection solution. It first recovers a substitution dataset for all classes from a pre-trained model via model inversion attack (Fredrikson et al. 2015) and then learns the probability distribution of potential triggers from the model using a conditional generative model. If the magnitude of the trigger for one class significantly deviates from others, it is determined that the queried model contains a backdoor.

5.2.3 Backdoor mitigation

In addition to detecting a backdoor or models with backdoors after the training process, several backdoor defenses are proposed to mitigate the impact of backdoors during the collaborative training process. For example, Sun et al. (2019) studied backdoor and defense strategies in collaborative learning and showed that norm clipping and weak differential privacy can mitigate the attacks without hurting the overall model performance. Zhu et al. (2019b) demonstrated that gradient sparsification is an effective approach to defend against backdoor attacks in collaborative learning, as well as to achieve a robust learning rate (Ozdayi et al. 2020). Wu et al. (2020) proposed a federated pruning method to remove redundant neurons of the shared model and to adjust the extreme weight values to mitigate backdoor attacks in federated learning systems. Liu et al. (2020) introduced additional training layers at the active party for backdoor defense. The active party first concatenates the output of the passive parties and adopts a dense layer before the output layer. To identify the malicious updates, Zhao et al. (2020b) presented defense schemes to detect anomalous updates in both IID and non-IID settings with a key insight of realizing client-side cross-validation, where each update is evaluated over the local data from other participants. Specifically, as shown in Fig. 8, the server selects a fraction of clients to evaluate

the sub-models G' that aggregated from partial updates and clients sent their reports R (the binary matrix of data classification result) to the server, which are used to adjust the aggregation weights of the clients. Andreina et al. (2021) supposed that the server cannot inspect updates and adopted cross-validation only to accept or reject the current update for the global model.

Sun et al. (2021a) proposed a more challenging task to defend backdoor attacks on participants when the global model is polluted. They designed a client-based defense named FL-WBC to perturb the parameter space where long-lasting backdoor attacks reside.

5.3 Adversarial training

In stand-alone learning systems, Adversarial Training (AT) is a widely established protection strategy against adversarial examples. For instance, an image recognition model could be trained on images with subtle alterations to improve its ability to recognize objects under different conditions (Zhao et al. 2022a). Szegedy et al. (2013) proposed the first adversarial training algorithm, in which the DNNs are trained on a mixture of generated adversarial examples and clean training data. Subsequently, a series of works (Huang et al. 2015; Shaham et al. 2018; Madry et al. 2017) attempted to train DNNs on adversarial examples. Shaham et al. (2018) defined a min–max adversarial problem to formulate a robust optimization, and the formulation based on Eq. 11 is illustrated below:

$$\min_w \mathbb{E}_{(x_i, y_i) \in D} \left[\max_{|\delta_i|_p \leq \epsilon} \mathcal{L}(f(x_i + \delta_i; w), y_i) \right], \quad (21)$$

where D denotes the training dataset. This dual optimization is adversarial against each other. The inner maximization problem is to find worst-case adversarial samples for the given victim model, whereas the outer minimization problem is to improve the robustness of the trained model.

Given that many adversarial examples could be generated by diverse adversaries, it is straightforward to develop the capacity to generalize the victim model under various attacks. In addition to AT, Adversarial distributional training (ADT) (Dong et al. 2020) is also formulated as a min–max optimization problem. In ADT, the inner maximization aims to learn an adversarial distribution that characterizes the potential adversarial examples surrounding a clean input, while the outer minimization attempts to train a robust model by minimizing the expected loss over the worst-case adversarial distributions. The adversarial optimization leads generated adversarial samples to lie in the region where the adversarial distribution assigns high probabilities. The primary distinction between AT and ADT is that for each input data, AT is optimized to find a specific worst-case adversarial example whereas ADT aims to learn a worst-case adversarial distribution consisting of a variety of adversarial samples. Particularly, ADT is formulated to capture the distribution of adversarial perturbations surrounding each input, as follows:

$$\min_w \mathbb{E}_{(x_i, y_i) \in D} \left[\max_{p(\delta_i) \in P} \mathbb{E}_{p(\delta_i)} [\mathcal{L}(f(x_i + \delta_i; w), y_i)] \right], \quad (22)$$

where $p(\delta_i)$ represents the adversarial perturbation distribution, whose support is contained in P . Notably, AT is a special case of ADT, when specifying the distribution family P to contain only Delta distributions. Besides, to avoid collapsing adversarial distribution,

Dong et al. (2020) employ an entropic regularization objective for characterizing heterogeneous adversarial examples. From the standpoint of the training strategy, neither AT nor ADT varies fundamentally from training GANs. In particular, the former needs inner maximization relative to each training sample rather than parameters that can be learned. Such a fundamental distinction results in entirely distinct optimization objectives, convergence analyses, and actual implementations. Recent research on adversarial training has consequently centered on adversarial regularization and training acceleration.

Adversarial regularization is an essential version of adversarial training in which the objective function is modified to include a regularization term (Goodfellow et al. 2014). Qin et al. (2019) proposed to calculate the absolute error between the adversarial loss and its first-order Taylor expansion. Zhang et al. (2019b) decomposed the robust error as the sum of empirical error and classification boundary error, in which the latter occurs when the distance between the training data and the decision boundary is short enough. Hence, TRade-off-inspired Adversarial DEfense via Surrogate-loss minimization (TRADES) is introduced to minimize the boundary error. The decomposition of the robust error also confirms from the side that unlabeled data could improve adversarial robustness. Jin et al. (2022) attempted to enhance adversarial training through Second-Order Statistics Optimization (S^2O) with respect to the model parameters, which are treated as random variables by relaxing classic PAC-Bayesian frameworks. Consequently, S^2O improves the robustness and generalization of the trained model and integrates flexibly with other adversarial training techniques, such as TRADES, resulting in a significant improvement of these techniques. In addition, Bui et al. (2022) incorporated Wasserstein distributional loss function into the adversarial training methods, which achieves a spontaneous relaxation and generalization of these methods.

Due to iterative min–max optimization, adversarial training techniques are slower than regular training. Recent research seeks to expedite adversarial training while preserving model robustness. In order to improve the efficiency of gradients, Free Adversarial Training (Free-AT) (Shafahi et al. 2019) was designed to reuse gradients computed in the back-propagating process while going forward. Further upon Free-AT, Zhang et al. (2019a) observed that updating gradients is only relevant to the first layer of DNNs. Hence, You Only Propagate Once (YOPO) was proposed to focus on the first layer while freezing other layers for reducing training parameters.

Recently, a series of big models are proposed as the fundamental technological architecture for various tasks. It urges the need for adversarial training methods compatible with collaborative learning systems. For the need of AT in collaborative learning systems, there are two minimum requirements: firstly, training data are distributed, provided by multiple participants, which have the individual capability of data storage or data privacy. Secondly, computing units are distributed, provided by distributed machines, which enables individual optimization. Overall, recent researches on adversarial training focus on adversarial optimization, Non-IID data and communication efficiency.

5.3.1 Optimization

In order to scale effectively to large models on large datasets, Zhang et al. (2022) introduced Distributed Adversarial Training (DAT) to support large-batch adversarial training implemented over distributed machines. Zhang et al. (2022) formulated DAT generically as follows:

$$\min_w \frac{1}{M} \sum_{i=1}^M \left\{ \mathbb{E}_{(x_i, y_i) \in D^{(i)}} \left[\max_{\|\delta_i\|_p \leq \epsilon} \ell(f(x_i + \delta_i; w), y_i) \right] \right\}, \quad (23)$$

Considering the parameter-server centralized topology, there exist M worker nodes, each of which has access to a local dataset $D^{(i)}$ and a server node collecting local information from workers to update parameters w . In detail, Zhang et al. (2022) theoretically quantified the convergence speed of DAT to the first-order stationary points in general non-convex settings at a rate of $O(1/\sqrt{T})$, where T is the total number of iterations. This result matches the convergence rate of standard training algorithms.

Furthermore, in decentralized collaborative learning systems, Tsaknakis et al. (2020) employed decentralized gradient tracking as well as primal–dual gradient descent–ascent algorithms to efficiently solve non-convex min–max optimization problems. These problems are suitable for modeling the network poisoning attack, in which the malicious adversaries try their best to tamper with distributed training data.

Moreover, several strategies (Kim 2022; Zhou et al. 2020; Luo et al. 2021; Chen et al. 2021a) were proposed to deal with distributed adversarial attacks. In centralized learning scenarios, some worker nodes may transfer malicious gradients from poisoned data or gradient perturbations to the server, so that aggregating naively resulting gradients would mislead the training process. Kim (2022) proposed a server-side learning algorithm to aggregate robust gradients. In this algorithm, the local gradients are firstly embedded into the manifold of normalized gradients, and then their aggregations are refined by simulating a diffusion process therein, which achieves great performance improvements over the baseline uniform gradient averaging method. In federated learning, there is a risk that the system would collapse in performance, when corrupted data are used for prediction after model deployment. Therefore, some works (Zhou et al. 2020; Luo et al. 2021; Chen et al. 2021a) added elaborated adversarial examples to the training dataset to train the shared model. Zhou et al. (2020) conducted collaborative adversarial training by composing the combined error of the server into bias and variance and using the bias-variance oriented adversarial examples to improve model robustness. By analogy to data augmentation, Luo et al. (2021) introduced an ensemble federated adversarial training method to enhance the diversity of adversarial examples through expanding training data with different disturbances generated from other participated clients. Furthermore, Chen et al. (2021a) observed that randomized smoothing techniques enable data-private distributed learning with certifiable robustness to test-time adversarial perturbations.

In many cases, even one adversary would launch multiple attacks at the same time in large-scale distributed machine learning systems. To defend against adversarial attacks and/or tolerates Byzantine faults, Wu et al. (2021) proposed Partial Synchronous Stochastic Gradient Descent (ParSGD). Experiments demonstrate that the trained model is able to produce accurate predictions as if it is not being attacked nor having failures at all when almost half of the agents are being compromised or failed using ParSGD.

5.3.2 Non-IID data distribution

Compared to the stand-alone learning systems, a collaborative system needs to deal with Non-IID data distributions among distributed participated agents. Non-IID in federated learning can be categorized into four classes: firstly, non-IID labels, the label marginal distribution varies across participants; secondly, non-IID features, the image feature marginal distribution varies across participants; thirdly, concept drift, the conditional distributions

varies across participants; and finally, quantity skew, the amount of data varies across participants. Li et al. (2021a) focused on non-IID features existing widely in reality and attempt to learn a common representation distribution among participants. Drawing lessons from GANs, Li et al. (2021a) designed a server that aims to train a discriminator to distinguish the local representations from individual agents, and the agents train the local models to generate representations that cannot be recognized by the discriminator. From another viewpoint, the inner-maximization optimization of adversarial training tends to exacerbate the Non-IID data distribution among local clients. Zhu et al. (2021) introduced an α -weighted federated adversarial training method to deal with this problem, by relaxing the inner maximization into a lower bound.

5.3.3 Communication efficiency

Adversarial training sometimes necessitates expensive computational resources, whilst modern collaborative learning systems can suffer from a large communication overhead for conveying stochastic gradients and updating model parameters. Yu et al. (2019b) introduced a double quantization scheme to reduce communication complexity. Three communication-efficient algorithms in this scheme are proposed: firstly, a low-precision AsyLPG method with asynchronous parallelism; secondly, a Sparse-AsyLPG algorithm with gradient sparsification; thirdly an accelerated AsyLPG method with momentum technique. Moreover, experiments conducted on a multi-server test-bed with real-world datasets show this proposed scheme can effectively save transmitted bits without performance degradation. In federated learning systems with a limited communication budget and Non-IID data distribution between agents, Shah et al. (2021) added a penalty term to the local training loss, compelling all local models to converge to a shared optimum. Hence, a federated dynamic adversarial training strategy is proposed to reach the trade-off between communication overhead and the convergence accuracy for adversarial training with Non-IID data distribution. Finally, in federated learning system with heterogeneous agents which may have varied computation resources, Hong et al. (2021) designed a strategy to propagate adversarial robustness from rich-resource agents to those with tight computational budgets among Non-IID data distribution.

5.3.4 Collaborative adversarial training

Numerous adversarial training methods (Hong et al. 2021; Zhou et al. 2020; Shah et al. 2021) have been proposed for collaborative learning systems. For instance, Hong et al. (2021) proposed an efficient propagation method that transfers adversarial robustness from high-resource participants who can afford adversarial training to low-resource participants. Zhou et al. (2020) conducted collaborative adversarial training by composing the aggregation error of the parameter server(s) into bias and variance and using the bias-variance adversarial examples to improve model robustness. Shah et al. (2021) considered communication-constrained federated learning environments and proposed a dynamic adversarial training method to improve both adversarial robustness and model convergence speed. In practical applications, adversarial collaborative training can be implemented in a federated learning system for autonomous vehicles. These vehicles could collaboratively train a model using both standard and adversarial road images (e.g., road signs with subtle modifications). This process could enhance the model's ability to correctly identify road signs, even under adversarial conditions (Liu et al. 2023).

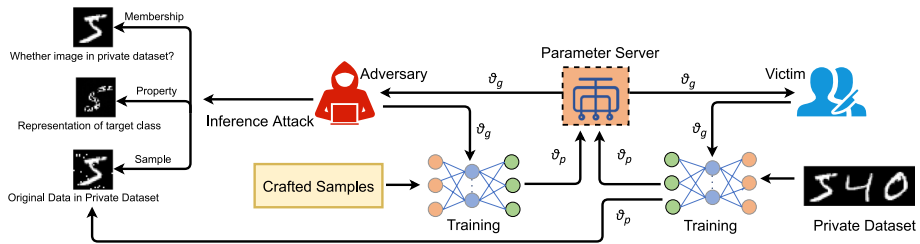


Fig. 9 An inference attack workflow in collaborative learning systems

6 Privacy attacks

6.1 Threat model

As demonstrated in Sect. 3.2, privacy attacks aim to infer private information about the training samples of workers. Figure 9 illustrates an inference attack workflow in collaborative learning systems, where some participating nodes are potential attackers. Malicious participants may conduct membership and property inference attacks with crafted samples and the observation of aggregated parameters. Moreover, adversaries recover the data samples in the victim's private dataset as long as they are able to acquire the update of the victim (Malicious Server). In addition, the parameter server that obtains the separated updates of all participants can also be malicious and achieve more precise inference attacks, e.g. to detect the membership with respect to a particular participant of a target sample.

According to the contextual information of the aggregated model, there are two categories of privacy attacks: white-box and black-box. In black-box mode, attackers can only access model outputs, whereas in white-box mode, they are aware of the model's structure and parameters. We summarize popular privacy attacks in collaborative learning systems in Table 6.

6.2 Membership inference

In the case of stand-alone learning, an attacker can only examine the final target model learned by a single participant. Prior research has revealed passive and active membership inference attacks against stand-alone DL models (Shokri et al. 2017; Salem et al. 2018; Long et al. 2018; Hayes et al. 2019); however, collaborative learning offers intriguing new paths for such inference attacks. The attacker in collaborative learning systems may be the parameter server or any of the participant nodes. While the parameter server monitors individual updates over time and can regulate how all participants view global parameters, each participant monitors global parameter updates and can control its own parameter uploads. Therefore, compared to attacks in stand-alone learning, parameter server and participants have more knowledge regarding the updates of each iteration, and membership inference attacks are simpler to execute.

Melis et al. (2019) presented a membership inference attack for text record datasets involving learning tasks. Specifically, the attacker, i.e., an honest but curious participant, receives the current aggregated updates at each iteration, from which he can obtain the aggregated updates from other participants. Melis et al. noted that the aggregated gradient

Table 6 Privacy attacks in collaborative learning systems

Methods	Parallelism		Consistency		Knowledge		Identity		Action		Frame-work	Task
	Data	Model	Sync	Async	White-box	Black-box	Partici-pant	Server	Passive	Active		
Membership												
Melis et al. (2019)	✓		✓			✓	✓		✓	✓	Federated	NLP
Nasr et al. (2019)	✓		✓		✓		✓	✓	✓	✓	Federated	CV
Zhang et al. (2020b)	✓		✓		✓		✓	✓	✓	✓	Federated	CV
Yuan et al. (2021)	✓			✓	✓		✓	✓	✓	✓	Federated	NLP
Property												
Hiraj et al. (2017)	✓		✓		✓		✓	✓	✓		Federated	CV
Wang et al. (2019b), Song et al. (2020)	✓		✓		✓			✓	✓	✓	Federated	CV
Zhang et al. (2021b)	✓		✓			✓	✓	✓	✓		Central- ized	NLP
Mahloujifar et al. (2022)	✓		✓			✓	✓	✓		✓	Central- ized	NLP
Sample												
Zhu et al. (2019b)	✓		✓		✓		✓	✓	✓		(De) central- ized	CV
Zhao et al. (2020a)	✓		✓		✓		✓	✓	✓		Central- ized	CV
Geiping et al. (2020)	✓		✓		✓	✓		✓	✓		Federated	CV
Yin et al. (2021)	✓		✓		✓			✓	✓		Central- ized	CV
Dang et al. (2021)	✓		✓		✓		✓	✓	✓		Central- ized	CV
Jin et al. (2021)	✓		✓		✓		✓	✓	✓		Federated	CV
Fu et al. (2022)	✓		✓		✓		✓		✓	✓	Federated	CV

Table 6 (continued)

Methods	Parallelism		Consistency		Knowledge		Identity		Action		Frame-work	Task
	Data	Model	Sync	Async	White-box	Black-box	Partici-pant	Server	Passive	Active		
He et al. (2019)		✓	✓		–	–	–	–		✓	Federated	CV
Fan et al. (2020)	✓		✓		✓			✓	✓		Federated	CV
Zhu and Blaschko (2020)	✓		✓		✓			✓	✓		Federated	CV
Fowl et al. (2021)	✓		✓		✓			✓		✓	Federated	CV
Jeon et al. (2021), Li et al. (2022b)	✓		✓		✓			✓	✓		Federated	CV

of an embedding layer is sparse with respect to the training text. Given a batch of training text, the embedding layer transforms the inputs into a lower-dimensional vector representation, and only the words that appear in the batch are used to update the appropriate parameters. The gradients of the remaining words are all zero. Consequently, the aggregated updates/gradients disclose directly which words are present in the training texts utilized by other truthful participants during the collaborative learning process.

Unfortunately, the membership inference attack (Melis et al. 2019) works exclusively for the learning tasks whose models employ explicit word embeddings with small training mini-batches. Nasr et al. (2019) developed a more standard and comprehensive framework for the privacy analysis in collaborative learning systems. Specifically, Nasr et al. proposed white-box membership inference attacks by investigating the privacy leakage from the stochastic gradient descent algorithm and evaluated the attacks under various adversarial models with different types of prior knowledge and abilities. Nasr et al. demonstrated that in collaborative learning, the update history on the same training datasets could reveal privacy information and boost the accuracy of the inference attacks. A local passive attacker can conduct membership inference attacks against other participants with a maximum inference accuracy of 79.2%. They further proposed an active attack that actively performs gradient ascent on a set of target data points to influence the parameters of other parties. This magnifies the presence of the data points in others' training sets. The attacker judges whether the target points are members by observing the reactions of the gradients on them. The accuracy of the active inference attack would be boosted by a significant increase under a global attacker.

Zhang et al. (2020b) focused on the scenario that the attack is launched by one of the participants and proposed a passive attack using the generative adversarial network (GAN). The assault employs GAN to enrich attack data and increase the data diversity utilized to query the collaborative learning model that is the target. Membership inference attacks are susceptible to the models trained to utilize the new sample-label pairings. Yuan et al. (2021) explored record data leakage against NLP in asynchronous distributed learning which would cause the imbalanced performance of training across participants. Through eavesdropping on the subset of participants or injecting a single watermark into the victim, they are able to successfully obtain the privacy records and reveal the participant identities.

6.3 Property inference

With the server's aggregated updates, attackers might gradually establish the class representation (i.e. property) of the training data of participants. For example, Hitaj et al. (2017) proposed a GAN-based attack to extract class representation information from honest participants in collaborative learning systems. The attack employs a GAN to generate instances that visually resemble samples from a particular participant class. In particular, the attack first generates some fake samples from the targeted class which are then injected into the training dataset as samples from another class. This would result in the victim participant disclosing sensitive information about the targeted class, as he must differentiate between the two classes. Using knowledge about the targeted class and GAN's density estimation, an attacker can learn the distribution of the targeted class without accessing the victim participant's training points directly. Even when the parameters are obscured using differential privacy approaches, the attack is successful against collaborative learning tasks involving convolutional neural networks.

The GAN-based class representation attack simply infers properties of the entire targeted class and assumes that the victim participant possesses all training points for the targeted class. In contrast, Melis et al. (2019) released the constrained assumptions and proposed property inference attacks to extract unintended information about participants' training data from the update history. Specifically, at each training iteration, the attacker saves a snapshot of the aggregated update parameters. The difference between successive snapshots is equal to the sum of all participant updates. During collaborative learning, this discrepancy reveals confidential information in the training batches of honest participants. Melis et al. advocated both passive and active property attacks:

- **Passive property inference:** the attack assumes the attacker possesses auxiliary data consisting of data points with and without the property of interest. The attack is predicated on the notion that the adversary can use snapshots of the global model to make aggregated updates based on data with and without the property. This results in labeled samples that allow the adversary to train a binary batch property classifier that assesses whether the observed updates are based on data with or without the property.
- **Active property inference:** the active attacker is able to conduct a more potent attack by utilizing multi-task learning. The adversary adds an upgraded property classifier to the final layer of his local copy of the collaboratively trained model. This model is trained to simultaneously excel at the primary job and recognize batch properties.

Similar to Hitaj et al. (2017), Wang et al. (2019b), Song et al. (2020) proposed GAN-based attacks against collaborative learning systems to target client-level privacy. The parameter server in the proposed attack is malicious and cannot access the target data. Since GANs are capable of generating conditioned samples, the attacker trains GANs based on updates from victim participants, allowing it to generate victim-conditioned samples including client-level privacy information. In addition, both passive and active modalities are considered.

- **Passive inference:** the malicious server is assumed to be honest-but-curious and only analyzes the updates from the participants by training GANs.
- **Active inference:** the active attacker isolates the victim participants from the others, i.e., training GANs on the victim alone by sending a special version of the aggregated model to the victim participants.

The aforementioned methods necessitate updated data during the training process, i.e., in the white-box mode. Instead, Zhang et al. (2021b) supposed the adversary can only black-box access to the global model. In order to understand the distribution of sensitive characteristics in a few queries, they train a series of shadow networks and a meta-classifier based on the connection between sensitive attributes and other attributes or labels. In addition, Mahlouiifar et al. (2022) demonstrated that, by picking poisoning data, an adversary can deliberately introduce such a link between target property and labels.

6.4 Sample inference

Collaborative learning systems employ the gradient-sharing framework to prevent participant data leakage, which has been shown to be less effective in recent sample

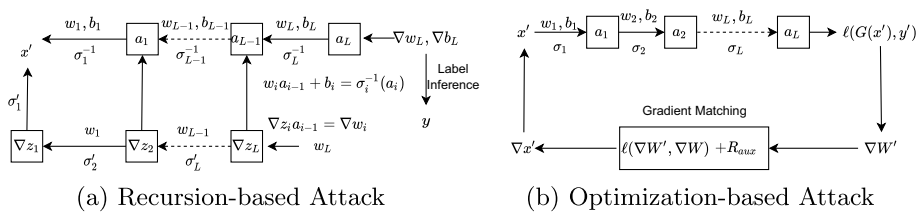


Fig. 10 The process of two sample inference attack scenario. $z_i = w_i x_i + b_i$ is feature map and $a_i = \sigma_i(z_i)$ denotes the activation value. In the recursion-based attack, y denotes the inference label from the gradient of the last layer and x' denotes the recovered result through recursive computation. In the optimization-based attack, x', y' denote the optimization objects and R_{aux} denotes some auxiliary regularization terms for x'

inference attacks. The technique of recursion-based and optimization-based sample inference attacks to recover training data from gradients is outlined in Fig. 10.

Le et al. (2017) discovered that the inputs of fully-connected (FC) layer or MLP with bias could be directly recovered from gradients: $x = \nabla w / \nabla b$. Fan et al. (2020) extended this analytic attack to models with convolutional layers, which transfer the convolutional layer to the linear layer by stacking the filters. However, the dependence on the bias term is not always satisfied and the weight sharing in convolution layers would cause dimension mismatch in the closed-form expression. By resolving a linear system of equations, Zhu and Blaschko (2020) were able to iteratively retrieve the data from the final FC layer to the first convolutional layer. Specifically, they leveraged the weight constraints and gradient constraints in forward and backward propagation (Eq. 24), and the workflow is shown in Fig. 10a.

$$\begin{aligned} w_i \cdot a_{i-1} + b_i &= z_i = \sigma_i^{-1}(a_i) \text{ (weight)} \\ \nabla z_i \cdot a_{i-1} &= \nabla w_i \text{ (gradient)} \end{aligned} \quad (24)$$

Nevertheless, these attacks could only reconstruct the linear combination of the batch inputs. Pan et al. (2020b) separated single sample information from the averaged gradients via the sparse activation of ReLU units. Fowl et al. (2021) modified the sharing model to include the linear layer to achieve large-scale and full batch image reconstruction. Although recursive attacks could directly recover inputs by numerical calculation, they only qualify for linear or convolutional layers and can't endure noised or perturbed gradients.

Zhu et al. (2019b) first pointed out the optimization-based gradient attacks. They presented an optimization algorithm, Deep Leakage from Gradients (DLG), that can obtain both the training inputs and the labels in just a few iterations. The attack first randomly generates a pair of “dummy” inputs and labels and then derives the dummy gradients from the dummy data. The attack optimizes the dummy inputs and labels to minimize the distance between dummy gradients and real gradients. The private training data would be fully revealed by matching the gradients and making the dummy data close to the original ones.

Although DLG works, Zhao et al. (2020a) revealed that it is not able to reliably extract the ground-truth labels or generate good-quality training samples. Zhao et al. proposed a simple yet efficient sample inference attack to extract the ground-truth labels from the shared gradients. They demonstrated that the gradient of the classification loss can distinguish the correct label from others by derivation. With such observation, the attacker can identify the ground-truth labels based on the shared gradients. Then, the

attacker can significantly simplify the DLG attack and extract good-quality training samples.

The aforementioned sample inference attacks rely heavily on two components: the Euclidean cost function and LBFGS optimization. Geiping et al. (2020) believed that these options are not ideal for more realistic architectures and notably arbitrary parameter vectors, and recommend using an angle-based cost function, i.e. cosine similarity. On the one hand, the magnitude measures the local optimal of the data point and captures only information regarding the training state. On the other hand, the angle measures the change in prediction for a particular data point when a gradient step is taken in the opposite direction.

Numerous sample reference attacks are later devoted to improving the effectiveness of the revealing training samples and labels (Yin et al. 2021; Dang et al. 2021; Jin et al. 2021; Fu et al. 2022; Chen et al. 2021b). For example, Yin et al. (2021) presented GradInvision to recover batch image from the averaged gradients. In particular, GradInvision first performs label revealing from the gradients of the fully-connected layer and then optimizes random inputs to match the target gradients with auxiliary regularization e.g. total variation norm ($\mathcal{R}_{TV}$), ℓ_2 norm ($\mathcal{R}_{\ell_2}$) and batch normalization ($\mathcal{R}_{BN}$). Dang et al. (2021) considered that participants compute updates with a reasonably small batch size and proposed Revealing Labels from Gradients (RLG) that reconstructs training samples from only the gradient of the last layer. Balunović et al. (2021) theoretically analyzed these attacks and revealed that they could be treated as adversaries with different assumptions on the probability distributions of the underlying data and gradients. In addition to random inputs, Jeon et al. (2021) and Li et al. (2022b) employed a pretrained GAN to generate dummy inputs and shrink search space, which would obtain better image reconstruction. Meanwhile, Chen et al. (2021b) and Fu et al. (2022) investigated the large-batch data leakage in vertical federated learning and He et al. (2019) explored the sample reconstruction in the model parallelism architecture. Moreover, Hatamizadeh et al. (2022) implemented gradient inversion attacks on vision transformers (ViTs).

7 Privacy defenses

In response to privacy attacks, numerous privacy defenses are developed to prevent the inference of training samples. Based on the commonly used privacy-preserving techniques, we classify the existing privacy defenses into three categories: differentially private, cryptographic privacy-preserving, and practical privacy-preserving collaborative learning. We summarize state-of-the-art privacy defenses in Table 7 and elaborate as follows.

7.1 Differentially private collaborative learning

Differential privacy (DP) is a rigorous mathematical framework for preserving the privacy of individual data records in a database when aggregated information about this database is shared with untrusted parties (Dwork et al. 2006, 2010). DP is one of the most promising solutions for mitigating membership inference attacks in collaborative training systems. For example, in a healthcare model, a prevalent approach to prevent the disclosure of a specific patient's health information involves adding Laplacian or Gaussian noise to data query results. This ensures that specific entries are not exposed in the output (Liang et al. 2020).

Several works have used DP to increase the privacy of DL training in various situations (Chaudhuri et al. 2011; Abadi et al. 2016; Zhang et al. 2018b; Li et al. 2018; Yu

Table 7 Taxonomy of privacy defenses

Methods	Methodology	Scope	Adversary ability		Target	Framework	Task
			White-box	Black-box			
DP							
Shokri and Shmatikov (2015)	Sparse vector + advanced composition theorem	Gradient		✓	Membership	Centralized	CV
Phan et al. (2016)	DP	Objective function	–	–	–	Standalone	NLP
Abadi et al. (2016)	Gradient clipping + MA	Gradient	–	–	–	Standalone	CV
Yu et al. (2019a)	Noise decay	Gradient	✓	✓	Membership	Standalone	CV
Wei et al. (2020)	Advance composition theorem	Weight				Federated	CV
Bhowmick et al. (2018)	MA	Weight		✓	Property	Federated	CV + NLP
Hynes et al. (2018)	TEE + DP	Gradient		✓	Property	Centralized	CV
Kang et al. (2019)	MA	Gradient		✓	Property	Centralized	NLP
Park et al. (2017)	MA + zCDP	Weight			Membership	Standalone	CV + NLP
Jayaraman et al. (2018)	zCDP + SMC	Weight + gradient		✓	Membership	Centralized	NLP
Han et al. (2021)	Large batch size	Gradient	✓	✓	Membership + property	Federated	CV
Wei et al. (2021a)	Communication discounting	Gradient	✓		All	Federated	CV
Wei et al. (2021b)	Gradient clipping	Gradient	✓	✓	All	Federated	CV
Sun et al. (2021c)	zCDP + client budget	Gradient	✓		Membership + sample	Federated	CV

Table 7 (continued)

Methods	Methodology	Scope	Adversary ability		Target	Framework	Task
			White-box	Black-box			
Xiong et al. (2021)	Local and global DP	Gradient	✓	✓	All	Federated	CV
Cryptographic							
Aono et al. (2016)	PHE	Sample	–	–	–	Standalone	None
Le et al. (2017)	PHE	Weight	–	✓	Sample	Centralized	CV
Xu et al. (2020)	PHE	Gradient	–	✓	All	Federated	CV
Zhang et al. (2021a)	PHE	Sample	–	✓	All	Centralized	CV
Zhang et al. (2020a)	PHE + batch encryption	Gradient	–	–	All	Federated	CV
Froelicher et al. (2021)	FHE	Sample	–	✓	All	Federated	CV
Stripelis et al. (2021)	FHE	Sample	–	✓	Membership	Federated	CV
Bonawitz et al. (2017)	SMC	Gradient	–	✓	–	Federated	–
Li et al. (2020d)	SMC	Weight	✓	–	All	Federated	CV
Li et al. (2020b)	SMC	Gradient	–	✓	–	Federated	CV
Liu et al. (2022b)	PHE	Weight	–	✓	Sample	Federated	CV
Practical							
Gao et al. (2021)	Sample augmentation	Sample	✓	–	All	Centralized	CV
Zhao et al. (2020c)	Sensitive data transformation	Sample	✓	–	Property	Centralized	CV
Fan et al. (2020)	Secret polarization network	Objective function	✓	–	All	Federated	CV
Huang et al. (2021)	Gradient clipping + MixUp	Sample + gradient	✓	–	Sample	Federated	CV
Huang et al. (2020c)	MixUp	Sample	✓	–	Sample	Federated	CV

Table 7 (continued)

Methods	Methodology	Scope	Adversary ability		Target	Framework	Task
			White-box	Black-box			
Scheliga et al. (2022)	Variational bottleneck	Gradient	✓		Sample	Federated	CV
Sun et al. (2021b)	Representation perturbation	Gradient	✓		Sample	Federated	CV

et al. 2019a; Jayaraman and Evans 2019). Most present DP-SGD algorithms use additive noise techniques by adding random noise to the estimates during each training iteration. There exists a trade-off between privacy and usability, which is defined by the level of noise supplied during training: adding too much noise can satisfy privacy needs but at the expense of a reduction in model accuracy. Consequently, it is crucial to establish the minimum quantity of noise necessary to offer the desired level of privacy protection and retain acceptable model performance.

Two approaches were developed to optimize the DP mechanisms and strike a compromise between privacy and usability. The first is to restrict the sensitivity of random processes with caution. Abadi et al. (2016), for instance, limited the influence of training data on gradients by clipping any gradient in l_2 norm below a set threshold. Since the learned models converge iteratively, Yu et al. (2019a) differentially optimized the model accuracy by introducing decay noise to the gradients across the training duration. The second approach is to precisely track the accumulated privacy cost of the training process using composition techniques such as the strong composition theorem (Dwork et al. 2010) and Moments Account (MA) (Abadi et al. 2016; Bhowmick et al. 2018; Hynes et al. 2018; Kang et al. 2019). Below are examples of prevalent DP strategies now in use, followed by a summary of differentially private solutions for collaborative learning systems.

7.1.1 DP techniques

For any two adjacent datasets that differ in just one record, a randomized mechanism $\mathcal{M}$ is differentially private if its outputs on both datasets are almost identical. A formal definition of DP is as follows.

Definition 1 ((ϵ, δ) -DP) A randomized mechanism $\mathcal{M} : D \rightarrow R$ with domain D and range R satisfies (ϵ, δ) -DP if for any two neighboring datasets D_1, D_2 and any subset of outputs $S \subseteq R$, the following property holds:

$$\Pr[\mathcal{M}(D_1) \in S] \leq e^\epsilon \Pr[\mathcal{M}(D_2) \in S] + \delta. \quad (25)$$

The DP condition of $\mathcal{M}$ is parameterized by ϵ and δ . ϵ is the privacy budget to limit the privacy loss of individual records. δ is a relaxation parameter that allows the privacy budget of $\mathcal{M}$ to exceed ϵ with probability δ . It is shown that differential privacy satisfies a composition property: when with privacy budgets ϵ_1 and ϵ_2 are performed on the same data, the privacy budget of the combined of two mechanisms equals the sum of the two privacy budgets, i.e., $\epsilon_1 + \epsilon_2$.

Relaxed definition Composing multiple differentially private methods results in a linear increase in the privacy budget and an increase in the magnitude of the accompanying noise to maintain a constant overall privacy budget. Multiple DP approaches decrease this linear composition bound at the expense of a modest increase in failure probability for a better privacy-usability trade-off. Concentrated Differential Privacy (CDP) and Rényi Differential Privacy (RDP) are two commonly used relaxations of differential privacy that are more accurate than (ϵ, δ) -DP. These relaxations use different versions of divergences to calculate the distributional difference between the outputs of $\mathcal{M}$ in adjacent datasets. CDP restricts the mean and standard deviation of the privacy loss variable via sub-Gaussian divergence. It improves the accuracy for any ϵ -DP algorithm satisfies $(\epsilon \cdot (e^\epsilon - 1)/2, \epsilon)$ -CDP. Rényi DP (RDP) (Mironov 2017) is a

natural relaxation of DP based on the Rényi divergence and allows tighter analysis of tracking cumulative privacy loss. The instantiation of RDP is MA, which keeps track of a cumulative bound on the moments of privacy loss.

7.1.2 DP-SGD for collaborative learning

For single-party learning, there are two common candidates for random noise addition: the objective function (Chaudhuri et al. 2011; Phan et al. 2016) and the gradients (Abadi et al. 2016; Yu et al. 2019a). For the first approach, Chaudhuri et al. (2011) perturbed the objective function prior to classifier optimization and demonstrated that the objective perturbation is DP if certain convexity and differentiability criteria hold. Phan et al. (2016) attempted to use the objective perturbation by replacing the non-convex function with a convex polynomial function. To achieve this, Phan et al. (2016) designed a convex polynomial function to approximate the non-convex one, which would change the learning protocol and even sacrifice the model performance. In single-party learning, introducing random noise to the gradients is a simpler and more prevalent technique. For instance, Abadi et al. (2016) restricted the sensitivity of randomized processes by clipping each gradient in l_2 norm below a certain threshold. Yu et al. (2019a) focused on differentially private model publishing and optimized the model accuracy by adding decay noise to the gradients across the training time since the learned models converge iteratively.

The model's usability can also be improved by carefully tracking the total privacy cost incurred during the training phase. For example, Shokri and Shmatikov (2015) and Wei et al. (2020) composed the additive noise mechanisms using the advanced composition theorem (Dwork et al. 2010), leading to a linear increase in the privacy budget. Some DP-SGD methods (Abadi et al. 2016; Bhowmick et al. 2018; Hynes et al. 2018; Kang et al. 2019) employed MA to reduce the added noise during the training process. Other algorithms (Park et al. 2017; Jayaraman et al. 2018; Yu et al. 2019a) were designed to enhance the model usability using (zero) concentrated DP (Dwork and Rothblum 2016).

Several works (Shokri and Shmatikov 2015; Bhowmick et al. 2018; Hynes et al. 2018; Jayaraman et al. 2018; Kang et al. 2019; Han et al. 2021; Wei et al. 2021a, b; Sun et al. 2021c; Mao et al. 2021; Xiong et al. 2021) applied the DP techniques from the standalone mode to the distributed systems in order to preserve the privacy of the training data for each agent. For example, Shokri and Shmatikov (2015) proposed a privacy-preserving distributed learning algorithm by adding Laplacian noise to each agent's gradients to prevent indirect leakage. Kang et al. (2019) adopted weighted aggregation instead of simply averaging to reduce the negative impact caused by uneven data scale in collaborative learning systems.

In terms of the accumulated privacy loss, Kang et al. (2019) employed MA to track the entire privacy cost of the collaborative training process. Wei et al. (2020, 2021a) perturbed agents' trained parameters locally by adding Gaussian noise before uploading them to the server for aggregation and bounded the sensitivity of the Gaussian mechanism by clipping in federated learning systems. Shokri and Shmatikov (2015) and Wei et al. (2020) composed the additive noise mechanisms using the strong composition theorem (Dwork et al. 2010), leading to a linear increase in the privacy budget. In order to reduce aggregated noise in local updates, Han et al. (2021) dynamically adjusted the batch size and noise level with respect to the rate of critical input data and the sensitivity estimation.

7.2 Cryptographic privacy-preserving collaborative learning

Although DP approaches are frequently employed in collaborative learning due to their clear theory and concise algorithm, they are designed to be vulnerable to membership inference attacks and difficult to defend against sample and property inference attacks. In addition, the addition of noise to the updates might decrease the effectiveness of the trained models, particularly when participants are extremely sensitive to privacy leakage. Due to the disadvantages of DP techniques, a number of privacy-preserving collaborative learning methods employing cryptographic tools are offered, as described in greater detail below.

Collaborative learning with homomorphic encryption Homomorphic Encryption (HE) enables users to directly execute arithmetic operations on ciphertext, which is comparable to performing the same operations on the plaintext. HE approaches can provide cryptographic privacy protection in collaborative learning contexts because they only require participants to provide encrypted data. Fully Homomorphic Encryption (FHE) and Partially Homomorphic Encryption are the two forms of HE schemes (PHE). FHE supports both addition and multiplication on encrypted data, whereas PHE just supports addition. FHE is significantly more computationally demanding than PHE. Several privacy-preserving collaborative learning approaches have been proposed to use PHE to ensure the privacy of individual model updates (Aono et al. 2016; Le et al. 2017; Xu et al. 2020; Zhang et al. 2021a, 2020a; Liu et al. 2022b). For example, Aono (Aono et al. 2016), Phong (Le et al. 2017), and PPFDL (Xu et al. 2020) perform the addition operation over encrypted updates to protect the privacy of the updates during the aggregation process.

To save the cost of homomorphic linear computation, Zhang et al. (2021a) considered homomorphic linear computation as a sequence operations of addition, multiplication, and permutation and then greedily selected the least expensive operation for every computation step. Froelicher et al. proposed SPINDLE (Froelicher et al. 2021) that preserves data and model confidentiality and enables the execution of a cooperative gradient-descent and the evaluation of the obtained model even when there are colluding participants. Stripelis et al. (2021) proposed a secure federated learning framework FL using FHE techniques to protect training data and the shared updates.

However, HE has certain restrictions. For instance, the memory and arithmetic costs of encrypted data are significantly higher than those of the plaintext. And in collaborative learning systems, HE must use polynomial approximations to address typical non-linear processes.

Collaborative learning with secure multi-party computation Secure multi-party computing (SMC) is a widely used cryptographic approach that enables mutually distrustful people to jointly calculate a function over their inputs while preserving input privacy (Bonawitz et al. 2017; Bell et al. 2020; Li et al. 2020b, d). Bonawitz et al. (2017) proposed a communication-efficient, failure-robust secure aggregation of high dimensional model updates without learning each participant's sensitive information with SMC, which can defend both passive and active adversaries. Li et al. (2020d) proposed a privacy-preserving collaborative learning framework based on the chained SMC technique. As the output of a single participant is dissimulated with its prior in such a system, adversaries cannot gain the privacy of participants. SMC has lower compute and transmission costs than HE, but it is not suited to large-scale collaborative learning, particularly systems with thousands of participants.

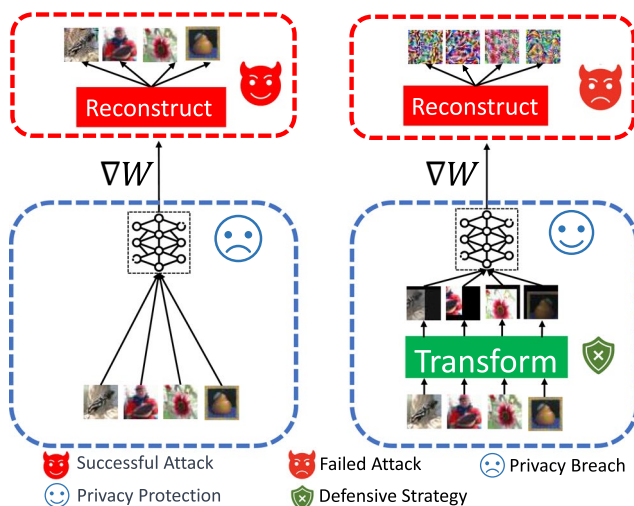


Fig. 11 Automatic transformation search against deep leakage from gradients (Gao et al. 2021)

7.3 Practical privacy-preserving collaborative learning

In addition to the previously mentioned privacy protections whose security can be theoretically guaranteed, other privacy-preserving collaborative learning strategies are presented to preserve the privacy of participants in real-world collaborative learning scenarios. Similar to integrity defenses, these privacy defenders rely on processing training data or model updates to experimentally protect private data from inference attacks. For instance, user data can be anonymized prior to its use in training a collaborative learning model, thus preserving user privacy without hindering the collaborative learning process (Sweeney 2002). Additionally, knowledge transfer techniques can be employed to protect data privacy in collaborative learning. These techniques involve the transformation of original trained models or datasets into smaller ones to eliminate any sensitive information contained within Dong et al. (2022), Vinaroz and Park (2023). MixUp (Zhang et al. 2017) and Instahide (Huang et al. 2020c) combined a private sample with other images and their labels. Figure 11 illustrates a privacy-preserving collaborative learning method (Gao et al. 2021) using automatic transformation search against deep leakage from gradients. By looking for particular transformations, the approach transforms original local data samples into related samples for eliminating sample inference attacks.

Such approaches are much more efficient than the defenses with cryptographic techniques to thwart inference attacks. Zhao et al. (2020c) presented a framework that transfers sensitive samples to public ones while protecting privacy, allowing participants to update their local models cooperatively using noise-preserving labels. Fan et al. (2020) designed a secret polarization network for each participant to produce secret losses and calculate the gradients. PRECODE (Scheliga et al. 2022) incorporated a variational bottleneck into the sharing model before the output layer to exchange gradients stochastically. Sun et al. (2021b) show that perturbation in the data representation prior to the FC layer can drastically damage the quality of reconstruction. Huang et al. (2021) advocated combining current sample inference defenses in an appropriate manner to enhance the protection performance.

8 Hybrid defenses

Existing investigations (Naseri et al. 2020) have demonstrated that defenses against one type of attack cannot be directly applied to other types of attacks. Consequently, in addition to the defenses that aim to prevent a single type of threat, a number of methods (Ma et al. 2022b; Grama et al. 2020; Qi et al. 2021; Liu et al. 2021; Lyu 2021; Dong et al. 2021; Domingo-Ferrer et al. 2021) are proposed to defend both integrity and privacy attacks and construct robust and privacy-preserving collaborative learning systems. Generally, these hybrid defenses utilize tactics against both integrity and confidentiality assaults. We describe contemporary hybrid defenses as follows.

One of the primary design strategies of hybrid defenses (Ma et al. 2022a, b; Grama et al. 2020; Liu et al. 2021) is to combine existing defenses for system integrity and privacy to establish secure collaborative learning systems. For instance, Ma et al. (2022b) employed an existing Byzantine-robust federated learning algorithm and distributed Paillier encryption and zero-knowledge proof to guarantee privacy and filter out anomaly parameters from Byzantine participants. Qi et al. (2021) achieved hybrid defense using blockchain and differential privacy techniques.

Several hybrid defenses leverage homomorphic encryption techniques that offer both confidentiality and computability for encrypted data. For instance, Liu et al. (2021) proposed a homomorphic encryption scheme that enables privacy protection and provides the parameter server a channel to punish poisoners under ciphertext. Dong et al. (2021) employed two non-colluding servers and proposed an oblivious defender for private Byzantine-robust federated learning using additive homomorphic encryption and secure two-party computation primitives. Ma et al. (2022c) designed a secure cosine similarity method that measures the difference of encrypted gradients to achieve Byzantine-tolerance aggregation. However, homomorphic encryption-based defenses require a considerable amount of computing resources. Domingo-Ferrer et al. (2021) provided participants with privacy and resilience against Byzantine and poisoning threats by unlinkable anonymity, which can identify improper model updates while decreasing the computational complexity in comparison to homomorphic encryption-based protections.

9 Discussion

9.1 Open problems

Whilst substantial study has been proposed to address the integrity and privacy challenges posed by collaborative learning, there are still a number of intriguing and vital issues to be thoroughly investigated. We outline a number of unresolved issues and suggested research topics to motivate further study:

Non-IID or noisy scenarios in Byzantine attacks and defenses Byzantine assaults and defenses are an arms race between attackers and defenders: attackers intend to create malicious updates that are indistinguishable from normal ones, while defenders attempt to identify potential Byzantine updates and maintain the integrity of the trained models. The majority of extant Byzantine robust algorithms exclusively examine IID training scenarios in which the training datasets for benign players are IID. In most actual situations, however, the training datasets are not IID since the quality and distribution of each training dataset

varies. The non-IID nature of training datasets often stems from the diverse data sources in real-world applications. For instance, in medical data across different hospitals, patient demographics and hospital equipment could result in data distributions that are inherently non-IID (Li et al. 2022a). Consequently, it is harder for defenders to discern between benign and malicious upgrades. A malevolent participant may, for instance, impersonate a node with poor training data quality and generate updates that are indistinguishable from normal ones but fatal to model integrity. Despite the fact that a number of works (Xie et al. 2019b; Cao et al. 2021) attempted to propose Byzantine resilient aggregation rules in Non-IID scenarios, they failed to protect against advanced Byzantine attacks or only considered a limited number of Non-IID scenarios (Cao et al. 2021).

Certified backdoor defenses Existing backdoor defenses for collaborative learning concentrate mostly on discovering or deleting backdoors using empirical means. Such defenses are effective against most known backdoor attacks, but they are unable to detect or eliminate future advanced attacks. Therefore, certified backdoor defenses for collaborative learning that provide demonstrable security against backdoor assaults are critically required. Unfortunately, the majority of existing certified backdoor defenses (Weber et al. 2020; Wang et al. 2020a) were developed for standalone machine learning systems and just a few of them (Xie et al. 2021) were intended for collaborative learning. For example, the application of backdoor defenses that are successful in standalone ML might not be directly transferable to collaborative learning due to the decentralized nature of the latter. New methodologies that factor in this decentralized structure are needed (Fang and Chen 2023).

Privacy-performance trade-off in differential privacy To fight against membership inference attacks, differential privacy approaches must provide noise to updates/models. Despite the fact that numerous relaxation approaches have been developed to lower the magnitude of noise, the performance is still undesirable, particularly when the parameters of the trained neural networks are large (Guo et al. 2021a; Wei et al. 2021a). The challenge lies in determining an optimal level of noise that does not significantly degrade the utility of the model while providing sufficient privacy guarantees. Real-world applications, such as financial or health predictions, demand both high accuracy and stringent privacy, making this balance even more challenging (Arous et al. 2023). Utilizing the system properties of collaborative learning systems to achieve a better privacy-performance trade-off is one promising research topic.

Basis datasets in property inference attacks Multiple attacks (Hitaj et al. 2017; Melis et al. 2019) utilized local datasets to infer the property of other participants. Assuming that these local datasets have the same distribution as victim participants, they are crucial to inference attacks. However, such IID datasets mitigate the threat posed by these assaults, as adversaries are unlikely to be aware of the distribution of the training datasets of victim participants. Consider a scenario where an adversary attempts to infer the health status of individuals in a hospital's dataset. Without having a dataset that mimics the actual distribution of the victim's dataset, the attacker's inference capability might be limited (Hartmann et al. 2023). Hence, how to conduct property inference attacks using foundation datasets merits a thorough investigation.

Performance improvement in sample inference defenses Sample inference defenses (Gao et al. 2021; Huang et al. 2021) can protect training samples from being inferred by existing attacks. However, certain protections, such as adding noise or pruning parameters, will negatively impact the performance of the trained models. For instance, in a facial recognition system, adding noise to training images can degrade recognition accuracy (Akbiyik 2023). Integrating techniques like knowledge distillation or dataset distillation might

be beneficial in achieving this balance (Dong et al. 2022; Vinaroz and Park 2023). Consequently, it is vital to create new defenses that might improve the performance and privacy of collaborative learning.

Fairness and privacy dilemma in federated learning A significant ethical issue in federated learning pertains to fairness. The performance of the globally trained model can vary among participants due to the non-independent and identically distributed data in the joint training process. To ensure convergence, participants possessing richer datasets and superior computational strength are often favored, receiving a higher selection probability and more significant importance during aggregation (McMahan et al. 2017). Consequently, the global model tends to favor these participants, resulting in weaker performance for others. Various strategies have been proposed to address this fairness issue, including specific adjustments to the training data or aggregation process (Zhao et al. 2018; Jeong et al. 2018; Huang et al. 2020a; Li et al. 2021c). However, these fairness-conscious methods usually necessitate access to private data, which heightens privacy risks. Thus, the challenge lies in developing approaches that can simultaneously safeguard privacy and ensure fairness in federated learning without compromising either aspect.

9.2 Limitations

While our survey provides a comprehensive review of security and privacy in collaborative learning systems, it has several limitations.

Scope of coverage Our survey provides a comprehensive review of security and privacy in collaborative learning systems. Despite our best efforts to include a broad spectrum of studies, the rapidly evolving nature of this field might mean that our search has not captured all relevant works.

Technical emphasis Our survey predominantly focuses on the technical aspects of security and privacy. Nonetheless, non-technical facets like legal and ethical considerations also play a crucial role. Such concerns, while significant, are beyond the ambit of this survey.

Context-dependent effectiveness In our discussion on various attacks and defenses, it's essential to note that the potency of these defenses often hinges on the specific context in which they're applied. For instance, while differential privacy might exhibit robustness in one environment, it could potentially undermine model performance in another. Hence, readers are advised to approach the findings of our survey with a discerning perspective.

Lab-based analysis A noteworthy portion of our survey is grounded in studies undertaken in laboratory settings. This raises concerns regarding the direct applicability of our findings to real-world contexts, where practical challenges, such as computational constraints and data variability, can play a pivotal role.

Despite these limitations, we believe that our survey provides valuable insights into the security and privacy issues in collaborative learning systems and can serve as a useful resource for future research in this area.

9.3 Applications

The insights derived from this survey cater to several impactful applications:

System design The detailed exploration of integrity and privacy vulnerabilities facilitates the crafting of robust and private collaborative learning architectures. For instance, system designers can preemptively address known risks highlighted in the survey to prevent potential breaches.

Defensive strategy development A clear understanding of the diverse attacks on collaborative systems, as elucidated in our analysis, can propel the inception of innovative defense techniques. This knowledge is pivotal for both research and practical defense implementations.

Regulatory and policy guidance By spotlighting the intricacies of privacy threats, our survey offers an informative base for drafting regulations in the arena of data protection, ensuring that policies are aligned with the latest threats and countermeasures.

Educational resource As a comprehensive document, this survey can seamlessly integrate into academic curricula, offering students insights into the convergence of machine learning, privacy, and cybersecurity.

Research directions The open challenges presented to act as a beacon for the research community, highlighting areas demanding further exploration and solutions.

9.4 Research methodology

Our commitment to delivering a systematic and comprehensive assessment of security and privacy studies in collaborative training is supported by a carefully designed research methodology. This sub-section clarifies the approach adopted in collecting and analyzing relevant literature, ensuring the robustness and comprehensiveness of our survey.

Primary data sources The literature is primarily sourced from reputable academic repositories in the fields of computer science and artificial intelligence, including Google Scholar, Springer Link, IEEE Xplore, ACM Digital Library, and ArXiv.

Search strings A wide range of search terms is used to ensure an exhaustive review of relevant studies. Example queries included: “collaborative learning security”, “federated learning privacy”, “byzantine attacks in collaborative learning”, and “privacy attacks in federated learning”.

Filtration and snowballing Upon completion of the initial data collection from the sources, articles are first filtered based on their titles and abstracts. Those that pass this stage undergo a full-text analysis to verify their relevance to our survey. In addition, the snowballing method is utilized, which involves exploring the references of our primary set of articles. This technique frequently leads us to discover significant works that might have been missed in our initial search.

10 Conclusion

Following our discussion on the limitations inherent in this survey, it's imperative to circle back to the broader scope of our work. We comprehensively explored the current vulnerabilities pertaining to integrity and privacy within collaborative learning systems. The primary vulnerabilities identified include Byzantine and backdoor attacks, coupled with three distinct data inference attacks. Our discussions delve into the nuances of these threats and provide a clear understanding of their mechanisms. The defensive strategies we introduced range from model- and data-based inspections against integrity threats to the application of differential privacy and encryption techniques against privacy infringements. Our findings suggest that modern-day defensive techniques are pivoting towards a balance between maintaining optimal system performance and ensuring robust security. The implications of these vulnerabilities are profound. As collaborative learning systems become increasingly popular, ensuring their resilience against malicious threats is paramount. The

vulnerabilities, if not addressed, can undermine the very essence of collaborative learning, which relies on trust and shared resources. To aid the ongoing research in this domain, we've outlined several open challenges. We hope that by shedding light on these unresolved issues, we provide a clearer path for researchers to fortify the robustness and privacy of collaborative learning systems.

Acknowledgements The authors are very appreciative to the reviewers for their precious comments which enormously ameliorated the quality of this paper. This work was supported in part by National Natural Science Foundation of China under Grants U22A6001, 62102052, and U21A20463; Key Research Project of Zhejiang Lab No.2022PG0AC02; China Postdoctoral Science Foundation No. 2021M692956; Key R & D Program of Zhejiang, No.2022C04006; CCF- AFSG Research Fund, No. CCF-AFSG RF20220009.

Author contributions Fei Yang, Xu Zhang, Shangwei Guo, Daiyuan Chen, and Yan Gan wrote the main manuscript text. Tao Xiang and Yang Liu revised the manuscript. All authors reviewed the manuscript

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abadi M, Chu A, Goodfellow I et al (2016) Deep learning with differential privacy. In: ACM SIGSAC conference on computer and communications security. pp 308–318. <https://doi.org/10.1145/2976749.2978318>
- Akbiyik ME (2023) Data augmentation in training CNNs: injecting noise to images. arXiv preprint <http://arxiv.org/abs/2307.06855>
- Aledhari M, Razzak R, Parizi RM et al (2020) Federated learning: a survey on enabling technologies, protocols, and applications. IEEE Access 8:140699–140725. <https://doi.org/10.1109/ACCESS.2020.3013541>
- Andreina S, Marson GA, Möllering H et al (2021) Baffle: backdoor detection via feedback-based federated learning. In: International conference on distributed computing systems. pp 852–863. <https://doi.org/10.1109/ICDCS51616.2021.00086>
- Aono Y, Hayashi T, Trieu Phong L et al (2016) Scalable and secure logistic regression via homomorphic encryption. In: ACM conference on data and application security and privacy. pp 142–144. <https://doi.org/10.1145/2857705.2857731>
- Arous A, Guesmi A, Hanif MA et al (2023) Exploring machine learning privacy/utility trade-off from a hyperparameters lens. arXiv preprint <http://arxiv.org/abs/2303.01819>
- Azizi A, Tahmid IA, Waheed A et al (2021) T-Miner: a generative approach to defend against Trojan attacks on DNN-based text classification. In: USENIX security symposium. <https://www.usenix.org/conference/usenixsecurity21/presentation/azizi>
- Bagdasaryan E, Veit A, Hua Y et al (2020) How to backdoor federated learning. In: International conference on artificial intelligence and statistics. pp 2938–2948. <http://proceedings.mlr.press/v108/bagdasaryan20a.html>
- Baluja S (2017) Hiding images in plain sight: deep steganography. In: Advances in neural information processing systems, vol 30. pp 2069–2079. <https://proceedings.neurips.cc/paper/2017/hash/838e8afb1ca34354ac209f53d90c3a43-Abstract.html>
- Balunović M, Dimitrov DI, Staab R et al (2021) Bayesian framework for gradient leakage. arXiv preprint <http://arxiv.org/abs/2111.04706>

- Baruch M, Baruch G, Goldberg Y (2019) A little is enough: circumventing defenses for distributed learning. arXiv preprint <http://arxiv.org/abs/1902.06156>
- Battiti R (1992) First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Comput* 4(2):141–166. <https://doi.org/10.1162/neco.1992.4.2.141>
- Bell JH, Bonawitz KA, Gascón A et al (2020) Secure single-server aggregation with (poly) logarithmic overhead. In: ACM SIGSAC conference on computer and communications security. pp 1253–1269. <https://doi.org/10.1145/3372297.3417885>
- Bhagoji AN, Chakraborty S, Mittal P et al (2019) Analyzing federated learning through an adversarial lens. In: International conference on machine learning. pp 634–643. <http://proceedings.mlr.press/v97/bhagoji19a.html>
- Bhowmick A, Duchi J, Freudiger J et al (2018) Protection against reconstruction and its applications in private federated learning. arXiv preprint <http://arxiv.org/abs/1812.00984>
- Blanchard P, El Mhamdi EM, Guerraoui R et al (2017) Machine learning with adversaries: Byzantine tolerant gradient descent. In: Advances in neural information processing systems. <https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html>
- Bonawitz K, Ivanov V, Kreuter B et al (2017) Practical secure aggregation for privacy-preserving machine learning. In: ACM SIGSAC conference on computer and communications security. pp 1175–1191. <https://doi.org/10.1145/3133956.3133982>
- Brown T, Mann B, Ryder N et al (2020) Language models are few-shot learners. In: Advances in neural information processing systems. <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>
- Bui AT, Le T, Tran QH et al (2022) A unified Wasserstein distributional robustness framework for adversarial training. In: ICLR. <https://openreview.net/forum?id=Dzpe9C1mpiv>
- Cao X, Lai L (2019) Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers. *IEEE Trans Signal Process* 67(22):5850–5864. <https://doi.org/10.1109/TSP.2019.2946020>
- Cao X, Fang M, Liu J et al (2021) FLTrust: Byzantine-robust federated learning via trust bootstrapping. In: ISOC network and distributed system security symposium. <https://www.ndss-symposium.org/ndss-paper/fltrust-byzantine-robust-federated-learning-via-trust-bootstrapping/>
- Carlini N, Wagner D (2018) Audio adversarial examples: targeted attacks on speech-to-text. In: IEEE security and privacy workshops (SPW). pp 1–7. <https://doi.org/10.1109/SPW.2018.00009>
- Chan A, Ong YS (2019) Poison as a cure: detecting & neutralizing variable-sized backdoor attacks in deep neural networks. arXiv preprint <http://arxiv.org/abs/1911.08040>
- Chang H, Shejwalkar V, Shokri R et al (2019) Cronus: robust and heterogeneous collaborative learning with black-box knowledge transfer. arXiv preprint [arXiv:1912.11279](http://arxiv.org/abs/1912.11279)
- Chaudhuri K, Monteleoni C, Sarwate AD (2011) Differentially private empirical risk minimization. *J Mach Learn Res* 12(3). <https://www.jmlr.org/papers/volume12/chaudhuri11a/chaudhuri11a.pdf>
- Chen J, Gu Q (2020) Rays: a ray searching method for hard-label adversarial attack. In: ACM SIGKDD international conference on knowledge discovery & data mining. pp 1739–1747. <https://doi.org/10.1145/3394486.3403225>
- Chen Y, Su L, Xu J (2017) Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proc ACM Meas Anal Comput Syst* 1(2):1–25. <https://doi.org/10.1145/3154503>
- Chen B, Carvalho W, Baracaldo N et al (2018) Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint <http://arxiv.org/abs/1811.03728>
- Chen H, Fu C, Zhao J et al (2019) DeepInspect: a black-box Trojan detection and mitigation framework for deep neural networks. In: IJCAI. pp 4658–4664. <https://doi.org/10.24963/ijcai.2019/647>
- Chen C, Kailkhura B, Goldhahn R et al (2021a) Certifiably-robust federated adversarial learning via randomized smoothing. In: IEEE international conference on mobile ad hoc and smart systems. pp 173–179. <https://doi.org/10.1109/MASS52906.2021.00032>
- Chen S, Kahla M, Jia R et al (2021b) Knowledge-enriched distributional model inversion attacks. In: IEEE/CVF international conference on computer vision. pp 16178–16187. <https://doi.org/10.1109/ICCV48922.2021.01587>
- Chou E, Tramèr F, Pellegrino G et al (2018) SentiNet: detecting physical attacks against deep learning systems. arXiv preprint <http://arxiv.org/abs/1812.00292>
- Dang T, Thakkar O, Ramaswamy S et al (2021) Revealing and protecting labels in distributed training. In: Advances in neural information processing systems, vol 34. <https://proceedings.neurips.cc/paper/2021/hash/0d924f0e6b3fd0d91074c22727a53966-Abstract.html>
- Dean J, Corrado G, Monga R et al (2012) Large scale distributed deep networks. In: Advances in neural information processing systems. pp 1232–1240. <https://papers.nips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html>

- Deng Y, Lyu F, Ren J et al (2021) Fair: quality-aware federated learning with precise user incentive and model aggregation. In: IEEE conference on computer communications. pp 1–10. <https://doi.org/10.1109/INFOCOM42981.2021.9488743>
- Devlin J, Chang MW, Lee K et al (2018) BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint <http://arxiv.org/abs/1810.04805>
- Domingo-Ferrer J, Blanco-Justicia A, Manjón J et al (2021) Secure and privacy-preserving federated learning via co-utility. IEEE Internet Things J. <https://doi.org/10.1109/JIOT.2021.3102155>
- Dong Y, Deng Z, Pang T et al (2020) Adversarial distributional training for robust deep learning. In: Advances in neural information processing systems, vol 33. pp 8270–8283. <https://proceedings.neurips.cc/paper/2020/hash/5de8a36008b04a6167761fa19b61aa6c-Abstract.html>
- Dong Y, Chen X, Li K et al (2021) FLOD: oblivious defender for private Byzantine-robust federated learning with dishonest-majority. Cryptol ePrint Archiv. https://doi.org/10.1007/978-3-030-88418-5_24
- Dong T, Zhao B, Lyu L (2022) Privacy for free: how does dataset condensation help privacy? In: International conference on machine learning. <https://proceedings.mlr.press/v162/dong22c.html>
- Dwork C, Rothblum GN (2016) Concentrated differential privacy. arXiv preprint <http://arxiv.org/abs/1603.01887>
- Dwork C, Kenthapadi K, McSherry F et al (2006) Our data, ourselves: privacy via distributed noise generation. In: Annual international conference on the theory and applications of cryptographic techniques. pp 486–503. <https://www.iacr.org/archive/eurocrypt2006/40040493/40040493.pdf>
- Dwork C, Rothblum GN, Vadhan S (2010) Boosting and differential privacy. In: IEEE annual symposium on foundations of computer science. pp 51–60. <https://doi.org/10.1109/FOCS.2010.12>
- El-Mhamdi EM, Farhadkhani S, Guerraoui R et al (2021) Collaborative learning in the jungle (decentralized, Byzantine, heterogeneous, asynchronous and nonconvex learning). In: Advances in neural information processing systems, vol 34. <https://proceedings.neurips.cc/paper/2021/hash/d2cd33e9c0236a8c2d8bd3fa91ad3acf-Abstract.html>
- El Mhamdi EM, Guerraoui R, Rouault SLA (2021) Distributed momentum for Byzantine-resilient stochastic gradient descent. In: International conference on learning representations. <https://openreview.net/forum?id=H8UHdhWG6A3>
- Enthoven D, Al-Ars Z (2020) An overview of federated deep learning privacy attacks and defensive strategies. arXiv preprint <http://arxiv.org/abs/2004.04676>
- Fan L, Ng KW, Ju C et al (2020) Rethinking privacy preserving deep learning: how to evaluate and thwart privacy attacks. arXiv preprint <http://arxiv.org/abs/2006.11601>
- Fan X, Ma Y, Dai Z et al (2021) Fault-tolerant federated reinforcement learning with theoretical guarantee. In: Advances in neural information processing systems, vol 34. <https://proceedings.neurips.cc/paper/2021/hash/080acdce72c06873a773c4311c2e464-Abstract.html>
- Fang P, Chen J (2023) On the vulnerability of backdoor defenses for federated learning. arXiv preprint <http://arxiv.org/abs/2301.08170>
- Fang M, Cao X, Jia J et al (2020) Local model poisoning attacks to Byzantine-robust federated learning. In: USENIX security symposium. <https://www.usenix.org/conference/usenixsecurity20/presentation/fang>
- Feng J, Xu H, Mannor S (2014) Distributed robust learning. arXiv preprint <http://arxiv.org/abs/1409.5937>
- Feng Y, Wu B, Fan Y et al (2022) Boosting black-box attack with partially transferred conditional adversarial distribution. In: IEEE/CVF conference on computer vision and pattern recognition. <https://doi.org/10.1109/CVPR52688.2022.01467>
- Fowl L, Geiping J, Czaja W et al (2021) Robbing the fed: directly obtaining private data in federated learning with modified models. arXiv preprint <http://arxiv.org/abs/2110.13057>
- Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. In: ACM SIGSAC conference. <https://doi.org/10.1145/2810103.2813677>
- Froelicher D, Troncoso-Pastoriza JR, Pyrgelis A et al (2021) Scalable privacy-preserving distributed learning. Proc Priv Enhanc Technol 2:323–347. <https://doi.org/10.2478/popets-2021-0030>
- Fu C, Zhang X, Ji S et al (2022) Label inference attacks against vertical federated learning. In: USENIX security symposium. <https://www.usenix.org/conference/usenixsecurity22/presentation/fu-chong>
- Gao Y, Xu C, Wang D et al (2019) Strip: a defence against Trojan attacks on deep neural networks. In: Computer security applications conference. pp 113–125. <https://doi.org/10.1145/3359789.3359790>
- Gao Y, Doan BG, Zhang Z et al (2020) Backdoor attacks and countermeasures on deep learning: a comprehensive review. arXiv preprint <http://arxiv.org/abs/2007.10760>
- Gao W, Guo S, Zhang T et al (2021) Privacy-preserving collaborative learning with automatic transformation search. In: IEEE/CVF conference on computer vision and pattern recognition. pp 114–123. <https://doi.org/10.1109/CVPR46437.2021.00018>

- Gawali M, Arvind C, Suryavanshi S et al (2021) Comparison of privacy-preserving distributed deep learning methods in healthcare. In: Annual conference on medical image understanding and analysis. pp 457–471. <https://doi.org/10.48550/arXiv.2012.12591>
- Geiping J, Bauermeister H, Dröge H et al (2020) Inverting gradients—how easy is it to break privacy in federated learning? arXiv preprint <http://arxiv.org/abs/2003.14053>
- Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint <http://arxiv.org/abs/1412.6572>
- Goyal P, Dollár P, Girshick R et al (2017) Accurate, large minibatch SGD: training ImageNet in 1 hour. arXiv preprint <http://arxiv.org/abs/1706.02677>
- Grama M, Musat M, Muñoz-González L et al (2020) Robust aggregation for adaptive privacy preserving federated learning in healthcare. arXiv preprint <http://arxiv.org/abs/2009.08294>
- Gu T, Dolan-Gavitt B, Garg S (2017) BadNets: identifying vulnerabilities in the machine learning model supply chain. arXiv preprint <http://arxiv.org/abs/1708.06733>
- Guerraoui R, Rouault S et al (2018) The hidden vulnerability of distributed learning in Byzantium. In: International conference on machine learning. pp 3521–3530. <http://proceedings.mlr.press/v80/mhamdi18a.html>
- Guo S, Zhang T, Xu G et al (2021a) Topology-aware differential privacy for decentralized image classification. IEEE Trans Circuits Syst Video Technol. <https://doi.org/10.1109/TCSVT.2021.3105723>
- Guo S, Zhang T, Yu H et al (2021b) Byzantine-resilient decentralized stochastic gradient descent. IEEE Trans Circuits Syst Video Technol. <https://doi.org/10.1109/TCSVT.2021.3116976>
- Guo W, Tondi B, Barni M (2022) An overview of backdoor attacks against deep neural networks and possible defences. IEEE Open J Signal Process. <http://arxiv.org/abs/2111.08429>
- Han R, Li D, Ouyang J et al (2021) Accurate differentially private deep learning on the edge. IEEE Trans Parallel Distrib Syst 32(9):2231–2247. <https://doi.org/10.1109/TPDS.2021.3064345>
- Hard A, Rao K, Mathews R et al (2018) Federated learning for mobile keyboard prediction. arXiv preprint <http://arxiv.org/abs/1811.03604>
- Hartmann V, Meynert L, Peyrard M et al (2023) Distribution inference risks: identifying and mitigating sources of leakage. In: IEEE conference on secure and trustworthy machine learning. <https://doi.org/10.1109/SaTML54575.2023.00018>
- Hatamizadeh A, Yin H, Roth HR et al (2022) GradViT: gradient inversion of vision transformers. In: IEEE/CVF conference on computer vision and pattern recognition. pp 10021–10030. <https://doi.org/10.1109/CVPR52688.2022.00978>
- Hayes J, Melis L, Danezis G et al (2019) LOGAN: membership inference attacks against generative models. Proc Priv Enhanc Technol 2019(1):133–152. <https://doi.org/10.2478/popets-2019-0008>
- He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition. <https://doi.org/10.1109/CVPR.2016.90>
- He Z, Zhang T, Lee RB (2019) Model inversion attacks against collaborative inference. In: Computer security applications conference. pp 148–162. <https://doi.org/10.1145/3359789.3359824>
- He C, Annavaram M, Avestimehr S (2020) Group knowledge transfer: federated learning of large CNNs at the edge. arXiv preprint <http://arxiv.org/abs/2007.14513>
- Hitaj B, Ateniese G, Perez-Cruz F (2017) Deep models under the GAN: information leakage from collaborative deep learning. In: ACM SIGSAC conference on computer and communications security. pp 603–618. <https://doi.org/10.1145/3133956.3134012>
- Hong J, Wang H, Wang Z et al (2021) Federated robustness propagation: sharing adversarial robustness in federated learning. arXiv preprint <http://arxiv.org/abs/2106.10196>
- Hu S, Liu X, Zhang Y et al (2022) Protecting facial privacy: generating adversarial identity masks via style-robust makeup transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 15014–15023. <https://doi.org/10.1109/CVPR52688.2022.01459>
- Huang R, Xu B, Schuurmans D et al (2015) Learning with a strong adversary. arXiv preprint <http://arxiv.org/abs/1511.03034>
- Huang X, Alzantot M, Srivastava M (2019) NeuronInspect: detecting backdoors in neural networks via output explanations. arXiv preprint <http://arxiv.org/abs/1911.07399>
- Huang W, Li T, Wang D et al (2020a) Fairness and accuracy in federated learning. arXiv preprint <http://arxiv.org/abs/2012.10069>
- Huang WR, Geiping J, Fowl L et al (2020b) MetaPoison: practical general-purpose clean-label data poisoning. arXiv preprint <http://arxiv.org/abs/2004.00225>
- Huang Y, Song Z, Li K et al (2020c) InstaHide: instance-hiding schemes for private distributed learning. In: International conference on machine learning. pp 4507–4518. <http://proceedings.mlr.press/v119/huang20i.html>

- Huang Y, Gupta S, Song Z et al (2021) Evaluating gradient inversion attacks and defenses in federated learning. In: *Advances in neural information processing systems*, vol 34. <https://proceedings.neurips.cc/paper/2021/hash/3b3fff6463464959dcd1b68d0320f781-Abstract.html>
- Hynes N, Cheng R, Song D (2018) Efficient deep learning on multi-source private data. *arXiv preprint* <http://arxiv.org/abs/1807.06689>
- Jagannatha A, Rawat BPS, Yu H (2021) Membership inference attack susceptibility of clinical language models. *arXiv preprint* <http://arxiv.org/abs/2104.08305>
- Jagielski M, Oprea A, Biggio B et al (2018) Manipulating machine learning: poisoning attacks and counter-measures for regression learning. In: *IEEE symposium on security and privacy*. pp 19–35. <https://doi.org/10.1109/SP.2018.00057>
- Jayaraman B, Evans D (2019) Evaluating differentially private machine learning in practice. In: *USENIX security symposium*. pp 1895–1912. <https://www.usenix.org/system/files/sec19-jayaraman.pdf>
- Jayaraman B, Wang L, Evans D et al (2018) Distributed learning without distress: privacy-preserving empirical risk minimization. In: *Advances in neural information processing systems*. pp 6343–6354. <https://proceedings.neurips.cc/paper/2018/file/7221e5c8ec6b08ef6d3f9ff3ce6eb1d1-Paper.pdf>
- Jeon J, Kim J, Lee K et al (2021) Gradient inversion with generative image prior. In: *Neural information processing systems*. <https://proceedings.neurips.cc/paper/2021/hash/fa84632d742f2729dc32ce8cb5d49733-Abstract.html>
- Jeong E, Oh S, Kim H et al (2018) Communication-efficient on-device machine learning: federated distillation and augmentation under non-IID private data. *arXiv preprint* <http://arxiv.org/abs/1811.11479>
- Ji Y, Zhang X, Wang T (2017) Backdoor attacks against learning systems. In: *IEEE conference on communications and network security*. pp 1–9. <https://doi.org/10.1109/CNS.2017.8228656>
- Jin X, Chen PY, Hsu CY et al (2021) CAFE: catastrophic data leakage in vertical federated learning. *arXiv preprint* <http://arxiv.org/abs/2110.15122>
- Jin G, Yi X, Huang W et al (2022) Enhancing adversarial training with second-order statistics of weights. In: *IEEE/CVF conference on computer vision and pattern recognition*. <https://doi.org/10.1109/CVPR52688.2022.01484>
- Kairouz P, McMahan HB, Avent B et al (2019) Advances and open problems in federated learning. *arXiv preprint* <http://arxiv.org/abs/1912.04977>
- Kang Y, Liu Y, Wang W (2019) Weighted distributed differential privacy ERM: convex and non-convex. *arXiv preprint* <http://arxiv.org/abs/1910.10308>
- Karimireddy SP, He L, Jaggi M (2021) Learning from history for Byzantine robust optimization. In: *International conference on machine learning*. pp 5311–5319. <http://proceedings.mlr.press/v139/karimireddy21a.html>
- Kim KI (2022) Robust combination of distributed gradients under adversarial perturbations. In: *IEEE/CVF conference on computer vision and pattern recognition*. <https://doi.org/10.1109/CVPR52688.2022.00035>
- Kim M, Song Y, Wang S et al (2018) Secure logistic regression based on homomorphic encryption: design and evaluation. *JMIR Med Inform* 6(2):e8805. <https://doi.org/10.2196/medinform.8805>
- Konečný J, McMahan HB, Yu FX et al (2016) Federated learning: strategies for improving communication efficiency. *arXiv preprint* <http://arxiv.org/abs/1610.05492>
- Krizhevsky A, Sutskever I, Hinton GE (2012a) ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, vol 25. pp 1097–1105. <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- Krizhevsky A, Sutskever I, Hinton GE (2012b) ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp 1106–1114. <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- Kwon H, Kim Y, Yoon H et al (2019) Selective audio adversarial example in evasion attack on speech recognition system. *IEEE Trans Inf Forensics Secur* 15:526–538. <https://doi.org/10.1109/TIFS.2019.2925452>
- Lam M, Wei GY, Brooks D et al (2021) Gradient disaggregation: breaking privacy in federated learning by reconstructing the user participant matrix. *arXiv preprint* <http://arxiv.org/abs/2106.06089>
- Le TP, Aono Y, Hayashi T et al (2017) Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans Inf Forensics Secur* (99):1–1. <http://eprint.iacr.org/2017/715>
- Leroy D, Coucke A, Lavril T et al (2019) Federated learning for keyword spotting. In: *IEEE international conference on acoustics, speech and signal processing*. pp 6341–6345. <https://doi.org/10.1109/ICASSP.2019.8683546>
- Li M, Andersen DG, Park JW et al (2014) Scaling distributed machine learning with the parameter server. In: *USENIX symposium on operating systems design and implementation*. pp 583–598. https://www.usenix.org/conference/osdi14/technical-sessions/presentation/li_mu

- Li C, Zhou P, Xiong L et al (2018) Differentially private distributed online learning. *IEEE Trans Knowl Data Eng* 30(8):1440–1453. <https://doi.org/10.1109/TKDE.2018.2794384>
- Li S, Ma S, Xue M et al (2020a) Deep learning backdoors. arXiv preprint <http://arxiv.org/abs/2007.08273>
- Li Y, Li H, Xu G et al (2020b) Toward secure and privacy-preserving distributed deep learning in fog-cloud computing. *IEEE Internet Things J* 7(12):11460–11472. <https://doi.org/10.1109/JIOT.2020.3012480>
- Li Y, Xu X, Xiao J et al (2020c) Adaptive square attack: fooling autonomous cars with adversarial traffic signs. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2020.3016145>
- Li Y, Zhou Y, Jolfaei A et al (2020d) Privacy-preserving federated learning framework based on chained secure multiparty computing. *IEEE Internet Things J* 8(8):6178–6186. <https://doi.org/10.1109/JIOT.2020.3022911>
- Li Q, He B, Song D (2021a) Adversarial collaborative learning on non-IID features. arXiv preprint <https://openreview.net/forum?id=EgkZwzEwcE>
- Li Q, Wen Z, Wu Z et al (2021b) A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Trans Knowl Data Eng*. <https://doi.org/10.1109/TKDE.2021.3124599>
- Li T, Hu S, Beirami A et al (2021c) Ditto: fair and robust federated learning through personalization. In: *International conference on machine learning*. pp 6357–6368. <http://proceedings.mlr.press/v139/li21h.html>
- Li Y, Li Y, Wu B et al (2021d) Invisible backdoor attack with sample-specific triggers. In: *IEEE/CVF international conference on computer vision*. pp 16463–16472. <https://doi.org/10.1109/ICCV48922.2021.01615>
- Li Q, Diao Y, Chen Q et al (2022a) Federated learning on non-IID data silos: an experimental study. In: *International conference on data engineering*. <http://arxiv.org/abs/2102.02079>
- Li Z, Zhang J, Liu L et al (2022b) Auditing privacy defenses in federated learning via generative gradient leakage. In: *IEEE/CVF conference on computer vision and pattern recognition*. pp 10132–10142
- Lian X, Zhang C, Zhang H et al (2017) Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In: *Advances in neural information processing systems*. pp 5330–5340. <https://proceedings.neurips.cc/paper/2017/file/f75526659f31040afeb61cb7133e4e6d-Paper.pdf>
- Liang Z, Wang B, Gu Q et al (2020) Differentially private federated learning with Laplacian smoothing. arXiv preprint <http://arxiv.org/abs/2005.00218>
- Lim WYB, Luong NC, Hoang DT et al (2020) Federated learning in mobile edge networks: a comprehensive survey. *IEEE Commun Surv Tutor* 22(3):2031–2063. <https://doi.org/10.1109/COMST.2020.2986024>
- Lin T, Kong L, Stich SU et al (2020) Ensemble distillation for robust model fusion in federated learning. In: *Advances in neural information processing systems*, vol 33. pp 2351–2363. <https://proceedings.neurips.cc/paper/2020/hash/18df51b97ccd68128e994804f3eccc87-Abstract.html>
- Litjens G, Kooi T, Bejnordi BE et al (2017) A survey on deep learning in medical image analysis. *Med Image Anal* 42:60–88. <https://doi.org/10.1016/j.media.2017.07.005>
- Liu Y, Ma S, Aafer Y et al (2018) Trojaning attack on neural networks. In: *Annual network and distributed system security symposium*. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03A-5_Liu_paper.pdf
- Liu D, Yan Z, Ding W et al (2019a) A survey on secure data analytics in edge computing. *IEEE Internet Things J* 6(3):4946–4967. <https://doi.org/10.1109/JIOT.2019.2897619>
- Liu M, Zhang W, Mroueh Y et al (2019b) A decentralized parallel algorithm for training generative adversarial nets. arXiv preprint <http://arxiv.org/abs/1910.12999>
- Liu Y, Lee WC, Tao G et al (2019c) ABS: scanning neural networks for back-doors by artificial brain stimulation. In: *ACM SIGSAC conference on computer and communications security*. pp 1265–1282. <https://doi.org/10.1145/3319535.3363216>
- Liu Y, Yi Z, Chen T (2020) Backdoor attacks and defenses in feature-partitioned collaborative learning. arXiv preprint <http://arxiv.org/abs/2007.03608>
- Liu X, Li H, Xu G et al (2021) Privacy-enhanced federated learning against poisoning adversaries. *IEEE Trans Inf Forensics Secur* 16:4574–4588. <https://doi.org/10.1109/TIFS.2021.3108434>
- Liu Y, Shen G, Tao G et al (2022a) Complex backdoor detection by symmetric feature differencing. In: *IEEE/CVF conference on computer vision and pattern recognition*. pp 15003–15013. <https://doi.org/10.1109/CVPR52688.2022.01458>
- Liu Z, Guo J, Lam KY et al (2022b) Efficient dropout-resilient aggregation for privacy-preserving machine learning. *IEEE Trans Inf Forensics Secur*. <https://doi.org/10.48550/arXiv.2203.17044>
- Liu X, Kuang H, Lin X et al (2023) CAT: collaborative adversarial training. arXiv preprint <http://arxiv.org/abs/2303.14922>

- Long Y, Bindschaedler V, Wang L et al (2018) Understanding membership inferences on well-generalized learning models. arXiv preprint <http://arxiv.org/abs/1802.04889>
- Lu Y, De Sa C (2021) Optimal complexity in decentralized training. In: International conference on machine learning. pp 7111–7123. <https://proceedings.mlr.press/v139/lu21a/lu21a.pdf>
- Luo S, Zhu D, Li Z et al (2021) Ensemble federated adversarial training with non-IID data. arXiv preprint <http://arxiv.org/abs/2110.14814>
- Lyu L (2021) DP-SIGNSGD: when efficiency meets privacy and robustness. In: International conference on acoustics, speech and signal processing. pp 3070–3074. <https://doi.org/10.1109/ICASSP39728.2021.9414538>
- Lyu L, Yu H, Ma X et al (2020a) Privacy and robustness in federated learning: attacks and defenses. arXiv preprint <http://arxiv.org/abs/2012.06337>
- Lyu L, Yu H, Yang Q (2020b) Threats to federated learning: a survey. arXiv preprint <http://arxiv.org/abs/2003.02133>
- Ma S, Liu Y (2019) NIC: detecting adversarial samples with neural network invariant checking. In: Network and distributed system security symposium. <https://www.ndss-symposium.org/ndss-paper/nic-detecting-adversarial-samples-with-neural-network-invariant-checking/>
- Ma C, Li J, Ding M et al (2021) Federated learning with unreliable clients: performance analysis and mechanism design. IEEE Internet Things J. <https://doi.org/10.1109/JIOT.2021.3079472>
- Ma X, Sun X, Wu Y et al (2022a) Differentially private Byzantine-robust federated learning. IEEE Trans Parallel Distrib Syst. <https://doi.org/10.1109/LCOMM.2022.3180113>
- Ma X, Zhou Y, Wang L et al (2022b) Privacy-preserving Byzantine-robust federated learning. Comput Stand Interfaces 80(103):561. <https://doi.org/10.1016/j.csi.2021.103561>
- Ma Z, Ma J, Miao Y et al (2022c) ShieldFL: mitigating model poisoning attacks in privacy-preserving federated learning. IEEE Trans Inf Forensics Secur 17:1639–1654. <https://doi.org/10.1109/TIFS.2022.3169918>
- Madry A, Makelov A, Schmidt L et al (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint <http://arxiv.org/abs/1706.06083>
- Mahloujifar S, Ghosh E, Chase M (2022) Property inference from poisoning. In: IEEE symposium on security and privacy. pp 1569–1569. <https://doi.org/10.1109/SP46214.2022.9833623>
- Mao Y, Yuan X, Zhao X et al (2021) ROMOA: robust model aggregation for the resistance of federated learning to model poisoning attacks. In: European symposium on research in computer security. pp 476–496. https://doi.org/10.1007/978-3-030-88418-5_23
- McMahan B, Moore E, Ramage D et al (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp 1273–1282. <http://proceedings.mlr.press/v54/mcmahan17a.html>
- Melis L, Song C, De Cristofaro E et al (2019) Exploiting unintended feature leakage in collaborative learning. In: IEEE symposium on security and privacy. pp 691–706. <https://doi.org/10.1109/SP.2019.00029>
- Mironov I (2017) Rényi differential privacy. In: IEEE computer security foundations symposium. pp 263–275. <https://doi.org/10.1109/CSF.2017.11>
- Moritz P, Nishihara R, Stoica I et al (2015) SparkNet: training deep networks in spark. arXiv preprint <http://arxiv.org/abs/1511.06051>
- Moritz P, Nishihara R, Stoica I et al (2016) SparkNet: training deep networks in spark. In: International conference on learning representations. http://learningsys.org/papers/LearningSys_2015_paper_18.pdf
- Mothukuri V, Parizi RM, Pouriyeh S et al (2021) A survey on security and privacy of federated learning. Futur Gener Comput Syst 115:619–640. <https://doi.org/10.1016/j.future.2020.10.007>
- Muñoz-González L, Biggio B, Demontis A et al (2017) Towards poisoning of deep learning algorithms with back-gradient optimization. In: ACM workshop on artificial intelligence and security. pp 27–38. <https://doi.org/10.1145/3128572.3140451>
- Muñoz-González L, Co KT, Lupu EC (2019) Byzantine-robust federated machine learning through adaptive model averaging. arXiv preprint <http://arxiv.org/abs/1909.05125>
- Narayanan D, Harlap A, Phanishayee A et al (2019) PipeDream: generalized pipeline parallelism for DNN training. In: ACM symposium on operating systems principles. pp 1–15
- Naseri M, Hayes J, De Cristofaro E (2020) Toward robustness and privacy in federated learning: experimenting with local and central differential privacy. arXiv preprint <http://arxiv.org/abs/2009.03561>
- Nasr M, Shokri R, Houmansadr A (2019) Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: IEEE symposium on security and privacy. pp 739–753. <https://doi.org/10.1109/SP.2019.00065>

- Naveed M, Kamara S, Wright CV (2015) Inference attacks on property-preserving encrypted databases. In: ACM SIGSAC conference on computer and communications security. <https://doi.org/10.1145/2810103.2813651>
- Nguyen TD, Rieger P, Miettinen M et al (2020) Poisoning attacks on federated learning-based IoT intrusion detection system. In: Proceedings Workshop on Decentralized IoT System and Security. pp 1–7. <https://doi.org/10.14722/diss.2020.23003>
- Ouyang L, Wu J, Jiang X et al (2022) Training language models to follow instructions with human feedback. In: Advances in neural information processing systems. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf
- Ozdayi MS, Kantarcioglu M, Gel YR (2020) Defending against backdoors in federated learning with robust learning rate. arXiv preprint <http://arxiv.org/abs/2007.03767>
- Pan X, Zhang M, Wu D et al (2020a) Justinian's GAAvernor: robust distributed learning with gradient aggregation agent. In: USENIX security symposium. pp 1641–1658. <https://www.usenix.org/conference/usenixsecurity20/presentation/pan>
- Pan X, Zhang M, Yan Y et al (2020b) Exploring the security boundary of data reconstruction via neuron exclusivity analysis. arXiv e-prints
- Park M, Foulds J, Choudhary K et al (2017) DP-EM: differentially private expectation maximization. In: Artificial intelligence and statistics. pp 896–904. <http://proceedings.mlr.press/v54/park17c/park17c.pdf>
- Pedarla LP, Zhang X, Zhao L et al (2023) Evaluation of query-based membership inference attack on the medical data. In: ACM southeast conference. <https://doi.org/10.1145/3564746.3587027>
- Peteiro-Barral D, Guijarro-Berdiñas B (2013) A survey of methods for distributed machine learning. *Prog Artif Intell* 2(1):1–11. <https://doi.org/10.1007/s13748-012-0035-5>
- Phan N, Wang Y, Wu X et al (2016) Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In: AAAI conference on artificial intelligence. pp 1309–1316. <https://doi.org/10.1609/aaai.v30i1.10165>
- Qi Y, Hossain MS, Nie J et al (2021) Privacy-preserving blockchain-based federated learning for traffic flow prediction. *Futur Gener Comput Syst* 117:328–337. <https://doi.org/10.1016/j.future.2020.12.003>
- Qin C, Martens J, Goyal S et al (2019) Adversarial robustness through local linearization. In: Advances in neural information processing systems, vol 32. <https://proceedings.neurips.cc/paper/2019/hash/0defd533d51ed0a10c5c9dbf93ee78a5-Abstract.html>
- Qiu H, Xiao C, Yang L et al (2020) SemanticAdv: generating adversarial examples via attribute-conditioned image editing. In: European conference on computer vision. pp 19–37. https://doi.org/10.1007/978-3-030-58568-6_2
- Qiu H, Zeng Y, Guo S et al (2021) DeepSweep: an evaluation framework for mitigating DNN backdoor attacks using data augmentation. In: ACM Asia conference on computer and communications security. pp 363–377. <https://doi.org/10.1145/3433210.3453108>
- Reddi S, Charles Z, Zaheer M et al (2020) Adaptive federated optimization. arXiv preprint <http://arxiv.org/abs/2003.00295>
- Sahu AK, Li T, Sanjabi M et al (2018) On the convergence of federated optimization in heterogeneous networks. arXiv preprint 3:3. <http://arxiv.org/abs/1812.06127>
- Salem A, Zhang Y, Humbert M et al (2018) ML-Leaks: model and data independent membership inference attacks and defenses on machine learning models. arXiv preprint <http://arxiv.org/abs/1806.01246>
- Schelig D, Mäder P, Seeland M (2022) PRECODE—a generic model extension to prevent deep gradient leakage. In: IEEE/CVF winter conference on applications of computer vision. pp 1849–1858. <https://doi.org/10.1109/WACV51458.2022.00366>
- Shafahi A, Huang WR, Najibi M et al (2018) Poison frogs! Targeted clean-label poisoning attacks on neural networks. In: Advances in neural information processing systems. pp 6103–6113. <https://proceedings.neurips.cc/paper/2018/hash/22722a343513ed45f14905eb07621686-Abstract.html>
- Shafahi A, Najibi M, Ghiasi A et al (2019) Adversarial training for free! arXiv preprint <http://arxiv.org/abs/1904.12843>
- Shah D, Dube P, Chakraborty S et al (2021) Adversarial training in communication constrained federated learning. arXiv preprint <http://arxiv.org/abs/2103.01319>
- Shaham U, Yamada Y, Negahban S (2018) Understanding adversarial training: increasing local stability of supervised models through robust optimization. *Neurocomputing* 307:195–204. <https://doi.org/10.1016/j.neucom.2018.04.027>
- Shejwalkar V, Houmansadr A (2021) Manipulating the Byzantine: optimizing model poisoning attacks and defenses for federated learning. *Internet Society*, p 18. <https://people.cs.umass.edu/~amir/papers/NDSS21-model-poisoning.pdf>

- Shi W, Cao J, Zhang Q et al (2016) Edge computing: vision and challenges. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2016.2579198>
- Shi J, Wan W, Hu S et al (2022) Challenges and approaches for mitigating Byzantine attacks in federated learning. In: *IEEE international conference on trust, security and privacy in computing and communications*. <http://arxiv.org/abs/2112.14468>
- Shoeybi M, Patwary M, Puri R et al (2019) Megatron-LM: training multi-billion parameter language models using model parallelism. *arXiv preprint* <http://arxiv.org/abs/1909.08053>
- Shokri R, Shmatikov V (2015) Privacy-preserving deep learning. In: *ACM SIGSAC conference on computer and communications security*. pp 1310–1321. <https://doi.org/10.1145/2810103.2813687>
- Shokri R, Stronati M, Song C et al (2017) Membership inference attacks against machine learning models. In: *IEEE symposium on security and privacy*. pp 3–18. <https://doi.org/10.1109/SP.2017.41>
- Song M, Wang Z, Zhang Z et al (2020) Analyzing user-level privacy attack against federated learning. *IEEE J Sel Areas Commun* 38(10):2430–2444. <https://doi.org/10.1109/JSAC.2020.3000372>
- Stripelis D, Saleem H, Ghai T et al (2021) Secure neuroimaging analysis using federated learning with homomorphic encryption. *arXiv preprint* <http://arxiv.org/abs/2108.03437>
- Sun Z, Kairouz P, Suresh AT et al (2019) Can you really backdoor federated learning? *arXiv preprint* <http://arxiv.org/abs/1911.07963>
- Sun G, Cong Y, Dong J et al (2020) Data poisoning attacks on federated machine learning. *arXiv preprint* <http://arxiv.org/abs/2004.10020>
- Sun J, Li A, DiValentin L et al (2021a) FL-WBC: enhancing robustness against model poisoning attacks in federated learning from a client perspective. In: *Advances in neural information processing systems*, vol 34. <https://proceedings.neurips.cc/paper/2021/hash/692baebec3bb4b53d7ebc3b9fabac31b-Abstract.html>
- Sun J, Li A, Wang B et al (2021b) Soteria: provable defense against privacy leakage in federated learning from representation perspective. In: *IEEE/CVF conference on computer vision and pattern recognition*. pp 9311–9319. <https://doi.org/10.1109/CVPR46437.2021.00919>
- Sun P, Che H, Wang Z et al (2021c) Pain-FL: personalized privacy-preserving incentive for federated learning. *IEEE J Sel Areas Commun*. <https://doi.org/10.1109/JSAC.2021.3118354>
- Sun T, Li D, Wang B (2021d) Stability and generalization of the decentralized stochastic gradient descent. *arXiv preprint* <http://arxiv.org/abs/2102.01302>
- Sweeney L (2002) k-anonymity: a model for protecting privacy. *Int J Uncertain Fuzziness Knowl Based Syst*. <https://doi.org/10.1142/S0218488502001648>
- Szegedy C, Zaremba W, Sutskever I et al (2013) Intriguing properties of neural networks. *arXiv preprint* <http://arxiv.org/abs/1312.6199>
- Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: *IEEE/CVF conference on computer vision and pattern recognition*. pp 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Tancik M, Mildenhall B, Ng R (2020) StegaStamp: invisible hyperlinks in physical photographs. In: *IEEE/CVF conference on computer vision and pattern recognition*. pp 2117–2126. <https://doi.org/10.1109/CVPR42600.2020.00219>
- Tao G, Shen G, Liu Y et al (2022) Better trigger inversion optimization in backdoor scanning. In: *IEEE/CVF conference on computer vision and pattern recognition*. pp 13368–13378. <https://doi.org/10.1109/CVPR52688.2022.01301>
- Thudumu S, Branch P, Jin J et al (2020) A comprehensive survey of anomaly detection techniques for high dimensional big data. *J Big Data*. <https://doi.org/10.1186/s40537-020-00320-x>
- Tolpegin V, Truex S, Gursoy ME et al (2020) Data poisoning attacks against federated learning systems. In: *European symposium on research in computer security*. pp 480–501. <https://www.usenix.org/conference/usenixsecurity20/presentation/fang>
- Tran B, Li J, Madry A (2018) Spectral signatures in backdoor attacks. In: *Advances in neural information processing systems*. pp 8000–8010. <https://proceedings.neurips.cc/paper/2018/hash/280cf18baf4311c92aa5a042336587d3-Abstract.html>
- Truong L, Jones C, Hutchinson B et al (2020) Systematic evaluation of backdoor data poisoning attacks on image classifiers. In: *IEEE/CVF conference on computer vision and pattern recognition workshops*. pp 788–789. <https://doi.org/10.1109/CVPRW50498.2020.00402>
- Tsaknakis I, Hong M, Liu S (2020) Decentralized min-max optimization: formulations, algorithms and applications in network poisoning attack. In: *IEEE international conference on acoustics, speech and signal processing*. pp 5755–5759. <https://doi.org/10.1109/ICASSP40776.2020.9054056>
- Tu J, Liu W, Mao X et al (2021) Variance reduced median-of-means estimator for Byzantine-robust distributed inference. *J Mach Learn Res* 22(84):1–67
- Turner A, Tsipras D, Madry A (2018) Clean-label backdoor attacks. *arXiv preprint* <https://people.csail.mit.edu/madry/lab/cleanlabel.pdf>

- Vepakomma P, Swedish T, Raskar R et al (2018) No peek: a survey of private distributed deep learning. arXiv preprint <http://arxiv.org/abs/1812.03288>
- Vinaroz M, Park MJ (2023) Differentially private kernel inducing points (DP-KIP) for privacy-preserving data distillation. arXiv preprint <http://arxiv.org/abs/2301.13389>
- Wang B, Yao Y, Shan S et al (2019a) Neural cleanse: identifying and mitigating backdoor attacks in neural networks. In: IEEE symposium on security and privacy. pp 707–723. <https://doi.org/10.1109/SP.2019.00031>
- Wang Z, Song M, Zhang Z et al (2019b) Beyond inferring class representatives: user-level privacy leakage from federated learning. In: IEEE conference on computer communications. pp 2512–2520. <https://doi.org/10.1109/INFOCOM.2019.8737416>
- Wang B, Cao X, Gong NZ et al (2020a) On certifying robustness against backdoor attacks via randomized smoothing. arXiv preprint <http://arxiv.org/abs/2002.11750>
- Wang H, Sreenivasan K, Rajput S et al (2020b) Attack of the tails: yes, you really can backdoor federated learning. arXiv preprint <http://arxiv.org/abs/2007.05084>
- Wang J, Liu Q, Liang H et al (2020c) Tackling the objective inconsistency problem in heterogeneous federated optimization. arXiv preprint <http://arxiv.org/abs/2007.07481>
- Weber M, Xu X, Karlaš B et al (2020) RAB: provable robustness against backdoor attacks. arXiv preprint <http://arxiv.org/abs/2003.08904>
- Wei K, Li J, Ding M et al (2020) Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans Inf Forensics Secur. <https://doi.org/10.1109/TIFS.2020.2988575>
- Wei K, Li J, Ding M et al (2021a) User-level privacy-preserving federated learning: analysis and performance optimization. IEEE Trans Mob Comput. <https://doi.org/10.1109/TMC.2021.3056991>
- Wei W, Liu L, Wut Y et al (2021b) Gradient-leakage resilient federated learning. In: International conference on distributed computing systems. pp 797–807. <https://doi.org/10.1109/ICDCS51616.2021.00081>
- Wu Y, Schuster M, Chen Z et al (2016) Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint <http://arxiv.org/abs/1609.08144>
- Wu C, Yang X, Zhu S et al (2020) Mitigating backdoor attacks in federated learning. arXiv preprint <http://arxiv.org/abs/2011.01767>
- Wu Y, Chen H, Wang X et al (2021) Tolerating adversarial attacks and Byzantine faults in distributed machine learning. In: International conference on big data. pp 3380–3389. <https://doi.org/10.1109/BigData52589.2021.9671583>
- Xie C, Koyejo O, Gupta I (2018) Generalized Byzantine-tolerant SGD. arXiv preprint <http://arxiv.org/abs/1802.10116>
- Xie C, Huang K, Chen PY et al (2019a) DBA: distributed backdoor attacks against federated learning. In: International conference on learning representations. <https://research.ibm.com/publications/dba-distributed-backdoor-attacks-against-federated-learning>
- Xie C, Koyejo S, Gupta I (2019b) Zeno: distributed stochastic gradient descent with suspicion-based fault-tolerance. In: International conference on machine learning. pp 6893–6901. <http://proceedings.mlr.press/v97/xie19b.html>
- Xie C, Koyejo S, Gupta I (2020) Zeno++: robust fully asynchronous SGD. In: International conference on machine learning. pp 10495–10503. <http://proceedings.mlr.press/v119/xie20c.html>
- Xie C, Chen M, Chen PY et al (2021) CRFL: certifiably robust federated learning against backdoor attacks. arXiv preprint <http://arxiv.org/abs/2106.08283>
- Xiong Z, Cai Z, Takabi D et al (2021) Privacy threat and defense for federated learning with non-IID data in AIoT. IEEE Trans Ind Inf. <https://doi.org/10.1109/TII.2021.3073925>
- Xu G, Li H, Zhang Y et al (2020) Privacy-preserving federated deep learning with irregular users. IEEE Trans Depend Secur Comput. <https://doi.org/10.1109/TDSC.2020.3005909>
- Yang YR, Li WJ (2021) BASGD: buffered asynchronous SGD for Byzantine learning. In: International conference on machine learning. pp 11751–11761. <http://proceedings.mlr.press/v139/yang21e.html>
- Yang Z, Gang A, Bajwa WU (2020) Adversary-resilient distributed and decentralized statistical inference and machine learning: an overview of recent advances under the Byzantine threat model. IEEE Signal Process Mag 37(3):146–159. <https://doi.org/10.1109/MSP.2020.2973345>
- Yin D, Chen Y, Kannan R et al (2018) Byzantine-robust distributed learning: towards optimal statistical rates. In: International conference on machine learning. pp 5650–5659. <http://proceedings.mlr.press/v54/mcmahan17a.html>
- Yin H, Mallia A, Vahdat A et al (2021) See through gradients: image batch recovery via GradInversion. In: IEEE/CVF conference on computer vision and pattern recognition. pp 16337–16346. <https://proceedings.nips.cc/paper/2021/hash/fa84632d742f2729dc32ce8cb5d49733-Abstract.html>

- Yin M, Li S, Song C et al (2022) ADC: adversarial attacks against object detection that evade context consistency checks. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp 3278–3287. <https://doi.org/10.1109/WACV51458.2022.00289>
- Yu L, Liu L, Pu C et al (2019a) Differentially private model publishing for deep learning. In: IEEE symposium on security and privacy. pp 332–349. <https://doi.org/10.1109/SP.2019.00019>
- Yu Y, Wu J, Huang L (2019b) Double quantization for communication-efficient distributed optimization. In: Advances in neural information processing systems, vol 32. <https://proceedings.neurips.cc/paper/2019/hash/ea4eb49329550caaa1d2044105223721-Abstract.html>
- Yuan X, Ma X, Zhang L et al (2021) Beyond class-level privacy leakage: breaking record-level privacy in federated learning. IEEE Internet Things J. <https://doi.org/10.1109/JIOT.2021.3089713>
- Zelenkova R, Swallow J, Chamikara M et al (2022) Resurrecting trust in facial recognition: mitigating backdoor attacks in face recognition to prevent potential privacy breaches. arXiv preprint <http://arxiv.org/abs/2202.10320>
- Zhang H, Cisse M, Dauphin YN et al (2017) mixup: beyond empirical risk minimization. arXiv preprint <http://arxiv.org/abs/1710.09412>
- Zhang D, Chen X, Wang D et al (2018a) A survey on collaborative deep learning and privacy-preserving. In: IEEE international conference on data science in cyberspace. pp 652–658. <https://doi.org/10.1109/DSC.2018.00104>
- Zhang X, Khalili MM, Liu M (2018b) Improving the privacy and accuracy of ADMM-based distributed algorithms. In: International conference on machine learning. <http://proceedings.mlr.press/v80/zhang18f.html>
- Zhang D, Zhang T, Lu Y et al (2019a) You only propagate once: accelerating adversarial training via maximal principle. In: Advances in neural information processing systems, vol 32. <https://proceedings.neurips.cc/paper/2019/hash/812b4ba287f5ee0bc9d43bbf5bbe87fb-Abstract.html>
- Zhang H, Yu Y, Jiao J et al (2019b) Theoretically principled trade-off between robustness and accuracy. In: International conference on machine learning. pp 7472–7482. <http://proceedings.mlr.press/v97/zhang19p.html>
- Zhang C, Li S, Xia J et al (2020a) BatchCrypt: efficient homomorphic encryption for cross-silo federated learning. In: USENIX annual technical conference. pp 493–506. <https://www.usenix.org/conference/atc20/presentation/zhang-chengliang>
- Zhang J, Zhang J, Chen J et al (2020b) GAN enhanced membership inference: a passive local attack in federated learning. In: IEEE international conference on communications. <https://doi.org/10.1109/ICC40277.2020.9148790>
- Zhang Q, Xin C, Wu H (2021a) GALA: greedy computation for linear algebra in privacy-preserved neural networks. In: Network and distributed system security symposium. <https://www.ndss-symposium.org/ndss-paper/gala-greedy-computation-for-linear-algebra-in-privacy-preserved-neural-networks/>
- Zhang W, Tople S, Ohrimenko O (2021b) Leakage of dataset properties in multi-party machine learning. In: USENIX security symposium. pp 2687–2704. <https://www.usenix.org/conference/usenixsecurity21/presentation/zhang-wanrong>
- Zhang G, Lu S, Zhang Y et al (2022) Distributed adversarial training to robustify deep neural networks at scale. In: Uncertainty in artificial intelligence. pp 2353–2363. <https://proceedings.mlr.press/v180/zhang22a.html>
- Zhao Y, Li M, Lai L et al (2018) Federated learning with non-IID data. arXiv preprint <http://arxiv.org/abs/1806.00582>
- Zhao B, Mopuri KR, Bilen H (2020a) IDLG: improved deep leakage from gradients. arXiv preprint <http://arxiv.org/abs/2001.02610>
- Zhao L, Hu S, Wang Q et al (2020b) Shielding collaborative learning: mitigating poisoning attacks through client-side detection. IEEE Trans Depend Secur Comput. <https://doi.org/10.1109/TDSC.2020.2986205>
- Zhao Q, Zhao C, Cui S et al (2020c) PrivateDL: privacy-preserving collaborative deep learning against leakage from gradient sharing. Int J Intell Syst. <https://doi.org/10.1002/int.22241>
- Zhao S, Ma X, Zheng X et al (2020d) Clean-label backdoor attacks on video recognition models. In: IEEE/CVF conference on computer vision and pattern recognition. pp 14443–14452. <https://doi.org/10.1109/CVPR42600.2020.01445>
- Zhao W, Alwidian S, Mahmoud QH (2022a) Adversarial training methods for deep learning: a systematic review. Algorithms. <https://doi.org/10.3390/a15080283>
- Zhao Z, Chen X, Xuan Y et al (2022b) Defeat: deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In: IEEE/CVF conference on computer vision and pattern recognition. pp 15213–15222. <https://doi.org/10.1109/CVPR52688.2022.01478>

- Zheng W, Yan L, Gou C et al (2020) Federated meta-learning for fraudulent credit card detection. In: IJCAI. pp 4654–4660. <https://doi.org/10.24963/ijcai.2020/642>
- Zhou Y, Wu J, He J (2020) Adversarially robust federated learning for neural networks. arXiv preprint <https://openreview.net/forum?id=5xaInvrGWp>
- Zhu J, Blaschko M (2020) R-GAP: recursive gradient attack on privacy. arXiv preprint <http://arxiv.org/abs/2010.07733>
- Zhu J, Kaplan R, Johnson J et al (2018) Hidden: hiding data with deep networks. In: Proceedings of the European conference on computer vision. pp 657–672. https://doi.org/10.1007/978-3-030-01267-0_40
- Zhu C, Huang WR, Shafahi A et al (2019a) Transferable clean-label poisoning attacks on deep neural nets. arXiv preprint <http://arxiv.org/abs/1905.05897>
- Zhu L, Liu Z, Han S (2019b) Deep leakage from gradients. In: Advances in neural information processing systems. pp 14747–14756. <https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html>
- Zhu J, Yao J, Liu T et al (2021) α -weighted federated adversarial training. arXiv preprint <https://openreview.net/pdf?id=vxlAHR9AyZ6>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Fei Yang¹ · Xu Zhang² · Shangwei Guo² · Daiyuan Chen¹ · Yan Gan² · Tao Xiang² · Yang Liu³

✉ Shangwei Guo
swguo@cqu.edu.cn

Fei Yang
yangf@zhejianglab.com

Xu Zhang
xuzhang@cqu.edu.cn

Daiyuan Chen
chendy@zhejianglab.com

Yan Gan
shiyangancq@cqu.edu.cn

Tao Xiang
txiang@cqu.edu.cn

Yang Liu
yangliu@ntu.edu.sg

¹ Zhejiang Lab, Hangzhou, China

² College of Computer Science, Chongqing University, Chongqing, China

³ School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore