Wireless Personal Communications (2024) 136:2201–2242
https://doi.org/10.1007/s11277-024-11372-0

Research Progress on Security and Privacy of Federated Learning: A Survey

Xingpo Ma[1] · **Mengfan Yan**[1]

Accepted: 8 June 2024 / Published online: 25 June 2024

Abstract
Federated Learning (FL) is an emerging distributed machine learning paradigm designed to resolve the conflict between data sharing and privacy. It allows each client device to train shared models locally and perform global model aggregation on cloud servers without users having to share their data. However, there are still many security risks and malicious attacks that could breach the data privacy and confidentiality in the process of local training and information interaction. This paper investigates the security and the privacy challenges faced by FL and the corresponding defense methods. First, existing works about the FL-related surveys are studied; second, the basic concepts, the algorithm principle and the scenario classification of FL are introduced; next, examples are provided to illustrate the relevant attacks and defense knowledge of FL; then, the aggressive behaviors in FL are classified from four perspectives: the poisoning attack, the inference attack, the model attack and the adversarial attack, and the sub-aggressive behaviors are also com bed out; subsequently, the defense methods are divided according to the two directions of attack behaviors and privacy-protection technologies, and the application of different defense methods is investigated. Eventually, the future research directions on both attack problems and defense strategies in FL systems are discussed.

Keywords Federated learning · Privacy preservation · Security · Attack-and-defense strategies · Survey

1 Introduction

The growing prevalence of smart devices and the Internet of Things (IoT) is leading to an unprecedented growth in the volumes of data generated every day. The International Data Corporation (IDC) anticipates that billions of IoT devices will generate 79ZB of data by 2025 [1]. Nowadays, the data collectors have many more approaches to collect user's data than ever before. For example, application providers can require that the users can enjoy the convenience of the internet applications only if they share their data

✉ Xingpo Ma
maxingpo@xynu.edu.cn

[1] School of Computer and Information Technology, Xinyang Normal University, 237, Nanhu Rd, Xinyang 464000, Henan, China

with the application providers. This leads to "the data follows the application, and individual data ownership is not in their own hands" [2]. As a result, the enterprises manage and control the application data and monopolize them, which poses a great challenge to the protection of users' privacy. Recently, more and more users have realized the value of their data and are worried that third parties may share their information. In order to avoid the disclosure of sensitive information, countries around the world have enacted related laws to preserve the data privacy of citizens. Europe's General Data Protection Regulation (GDPR), California's California Consumer Privacy Law (CCPA), and China's Cyber Security Law and Data Security Law prohibit centralized remote processing of sensitive data collected in distributed mode [3, 4]. However, this also makes it difficult to access users' necessary private information for the valid users in many legal application fields such as medical treatment and education.

As a new distributed machine learning paradigm, federated learning (FL) can be used to solve the problem mentioned above. To prevent the servers from accessing the clients' sensitive data directly, FL lets the client devices store the data locally, and trains the global model by aggregating the local models iteratively, which are trained locally on the client devices. During the training procedure, the client devices only need to upload the gradient and the weight parameters to the central server [5].

However, although FL has become an effective scheme to resolve user privacy problems in machine learning, recent studies demonstrate that there are still loopholes in FL protocols, and attackers can launch many kinds of attacks, such as poisoning attack, inference attack, model attack, etc., to damage the trained models by leveraging these loopholes. For example, combined with advanced attack techniques of Generative adversarial networks (GANs), a class representation of global data distribution of all clients can be constructed, and it distinguishes between client-specific attacks (i.e., user-level privacy breaches), so this stronger privacy threat can precisely recover private data from specific clients [6]. In addition to this, sensitive data of participants may be leaked to untrusted servers through uploaded gradient vectors [7], and an opponent can also manipulate the shared model with a model poisoning attack. Besides, the attacker may masquerade as an honest data provider and inferences the attributes of sensitive training data on the target client by observing the update of the target shared model [8].

To make people who are interested in FL security better know its recent research development, in this survey, we collect, classify, introduce and discuss more than one hundred of FL-security-related papers which are published in recent years, and make a comprehensive and systematic study on them. In summary, the contributions mainly include the following points:

- In our work, both the survey and the non-survey papers related to the privacy and the security of FL are studied, and the similarities and the differences between our work and the related surveys are discussed.
- We systematically analyzes the threats on the security and the privacy of FL and the corresponding defense methods proposed by the researchers, and makes a comprehensive comparison among them.
- The aggressive behaviors in FL-related application are classified and discussed, and suggestions for dealing with such behaviors as well as future research directions are provided. Meanwhile, we identify a set of criteria for future solutions that will serve as a reference for scholars and developers studying ways to improve security and privacy in future FL systems.

The rest of this paper is arranged as follows: Sect. 2 makes a detailed research of the existing survey works on the security and the privacy of FL, compares our survey with them, and highlights the unique contributions of this survey; Sect. 3 elaborates on the relevant knowledge of FL and makes a comprehensive analysis of three scenarios of FL; Sect. 4 briefly introduces the threats to the security and privacy protection of FL, explains the various threats to the security and privacy protection of FL by examples, and provides corresponding solutions; Sect. 5 first generalizes the classification of attack methods in security challenges with graphs, and then explains each attack method with pictures and texts; in Sect. 6, the corresponding security defense methods and privacy protection technologies are proposed for four kinds of attack methods (poisoning attack, inference attack, model attack and adversarial attack). Section 7 predicts possible attack patterns and defense strategies for FL in the future, and provides a set of criteria for solutions. Section 8 concludes with a summary and a future outlook.

2 Comparison Between Our Work and the Existing Federated Learning Surveys

Recently, researchers have proposed some FL privacy-and-security related investigation articles. In [9], the authors classify possible attacks and threats during training for FL, list the attack methods of each category, and introduce the attack principles of the corresponding attacks. They summarized specific defense measures against these attacks and threats and analyzed their principles. In [10], the authors describe the development of machine learning and the inevitability of the emergence of FL, and give the definition and classification of FL. Aiming at of the privacy protection problems of FL, common privacy protection technologies are summarized. In addition, the existing mainstream open source frameworks of FL are introduced and compared, and the application scenarios of FL are given. In [11], the authors introduce the training processes of Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL), and explore the threats to these processes and the reason why they are prone to be attacked, so as to classify and summarize the existing attack methods, such as the poisoning attack, the adversarial attack and the model inversion attack. Aiming at several methods of attack in both scenarios, several corresponding defense measures are introduced, such as gradient sparsity, malicious detection, secret sample alignment, label protection, Verifiable Secret Sharing (VSS) and disturbance sharing. They highlighted the training processes and defenses against threats in both the HFL and VFL. In [12], the authors discuss the classification of FL and analyze its advantages and disadvantages. The hidden danger of FL is pointed out and the current main defense measures are introduced. In [13], the authors introduce the basic concepts and threat models of FL. Three types of attacks launched by internal malicious entities are summarized and security and privacy vulnerabilities of the FL architecture are investigated. Then, the most advanced defense schemes are studied from the aspects of Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-party Computation (SMC). In [14], the authors analyze the possible security problems of FL, focus on the threat of poisoning attack, adversarial attack and privacy disclosure in detail, summarize targeted defense measures and put forward corresponding solutions.

Most of the existing investigations on the privacy and security of FL only combine the basic knowledge of FL with attack methods and solutions, without considering that the solution should still follow some application criteria. In this paper, the theoretical

knowledge and related applications of FL are presented in the form of sentences combined with tables, and the problems and solutions in the application are also explained. More importantly, a series of criteria should be followed when developing defense strategies are proposed. Combining the above three parts to form a systematic study of FL privacy and security sequential architecture, which is not present in the existing articles. If the relevant staff can consider and meet as many criteria as possible when formulating the scheme to protect the privacy and security of the system, then the system is undoubtedly robust.

Table 1 summarizes the main similarities and differences between our survey and existing relevant FL surveys. Table 2 shows the differences between our survey and the existing relevant FL surveys (where "$\surd$" means "include" and "$\times$" means "not include").

Compared with the existing investigation articles on FL, this paper mainly focuses on the security and privacy issues of FL, and comprehensively analyzes FL from several aspects, such as attack methods and defense schemes. For privacy and security challenges in FL, suggestions for solving security and privacy issues and future research directions are provided, so as to provide researchers with new solutions to privacy and security of FL. The survey collected most of the relevant literature on privacy and security in FL. The content of the survey is rich and comprehensive. Our investigation on the security and the privacy challenges facing FL is very detailed, and the classification scheme presented is also very comprehensive.

3 Concept and Classification of Federated Learning

In this section, we first explain the concept and the algorithm principle of FL, then it introduces the classification of FL scenarios, and introduces the principles and the implementation processes of FL in different scenarios, such as HFL, VFL, and FTL.

3.1 Basic Concept and Algorithm Principle

FL can be regarded as a decentralized and collaborative machine learning method for privacy protection. The model training is completed in multiple iterations by multiple clients collaborating [16]. The concept of FL was first proposed by H.Brendan Mcmahan et al in 2016. It is mainly used to solve the privacy problem caused by centralized model training of data stored in multiple terminals (such as mobile phones) [17]. Google is the first company to introduce the FL system, which is mainly applied in the input method improvement and other scenarios. For example, after users have used relevant words several times, Google's Gboard system can suggest words and emoticons to them when they input words [18–20]. Different from the traditional recommendation system, this system relies on the mobile device itself to a large extent without gaining user privacy. The framework for FL is shown in Fig. 1.

FL is a distributed training model performed by a group of devices that share local model updates with a central server whose job is to aggregate these updates to build a global machine learning model. A common aggregation model known as the Federated Averaging (FedAvg) [21], allows the servers to aggregate local random gradients from different devices using iterative model averaging methods. Equation 1 [21] shows the framework of federated averaging.

Table 1 A comparison between our survey and the existing federated learning surveys

Ref	Similarities	Differences
Wu et al. [9]	Similar to our survey, this survey classifies possible attacks and threats during FL training sessions and introduces generic defense measures	[9] does not specify whether the targeted and universal defense measures of FL specifically solve security challenges or privacy challenges, so the discussion of FL classification is rather general. Our survey provides a clearer delineation and more specific classification of the security and the privacy challenges of FL and their defenses. It analyzes other surveys related to the security and the privacy challenges of FL, compares them with our survey to summarize the characteristics of each survey and analyzes the advantages of FL
Liang et al. [10]	Similar to our survey, this survey introduces the definition and classification of FL and summarizes common privacy protection techniques for privacy protection issues	[10] introduces the popular open-source frameworks for FL. This survey focuses on the classification of FL scenarios. Our survey explains the algorithm principle of FL, analyzes the advantages of FL, and focuses on the security and privacy protection threats and defense of FL. It analyzes surveys related to the security and the privacy challenges of FL and compares them with our survey to summarize the characteristics of each survey
Sun et al. [11]	Similar to our survey, this survey introduces the classification of FL and the security and privacy threats it faces, as well as measures to protect against them	[11] is based on HFL and VFL two application scenarios, and does not involve Federated Transfer Learning (FTL). Our survey is based on a complete classification of FL scenarios. It analyzes surveys related to the security and the privacy challenges of FL, compares them with our survey to summarize the characteristics of each survey and analyzes the advantages of FL
Zhou et al. [12]	Similar to our survey, this survey introduces the definition, principles and scenario classification of FL, analyzes privacy and security challenges, and introduces privacy protection measures	[12] not only introduces the privacy protection challenges facing FL, but also introduces communication efficiency and incentive mechanism, gives improvement measures for these three challenges, and analyzes the research progress of the three challenges. Our survey examines the security challenges and privacy challenges of FL in detail, and introduces countermeasures against them. It analyzes surveys related to the security and the privacy challenges of FL, compares them with our survey to summarize the characteristics of each survey and analyzes the advantages of FL

Table 1 (continued)

Ref	Similarities	Differences
Wang et al. [13]	Similar to our survey, this survey introduces the concept, scenario classification, and existing security and privacy issues of FL. The survey also gives some defensive measures for security and privacy protection, and analyzes the advantages of FL	In addition to addressing security and privacy issues, [13] also deals with the data transmission, existence of central servers, and unilateral data contamination. Our survey explains the algorithmic principles of FL and provides a comprehensive defense strategy for security and privacy protection. It analyzes surveys related to the security and the privacy challenges of FL and compares them with our survey to summarize the characteristics of each survey
Chen et al. [14]	Similar to our survey, this survey introduces the concept and the threat model of FL, the security challenges faced by FL, and it introduces countermeasures against the threats	[14]'s classification of security challenges is not comprehensive. Our survey explains the algorithmic principles of FL, analyzes the advantages of FL, and provides a comprehensive defense strategy for security and privacy protection. It analyzes surveys related to the security and the privacy challenges of FL and compares them with our survey to summarize the characteristics of each survey
Zhou et al. [15]	Similar to our survey, this survey introduces the concept and the classification of FL, analyzes the advantages of FL, security and the privacy challenges, and introduces corresponding defense measures	Our survey explains the algorithmic principles of FL and provides a comprehensive defense strategy for security and privacy protection. Our survey analyzes surveys related to the security and the privacy challenges of FL and compares them with our survey to summarize the characteristics of each survey

Table 2 Comparison of content differences of our survey and existing FL surveys

Characteristics	Wu et al. [9]	Liang et al. [10]	Sun et al. [11]	Zhou et al. [12]	Wang et al. [13]	Chen et al. [14]	Zhou et al. [15]	Our survey
A Comparison between Our Survey and the Existing Federated Learning Survey	×	×	×	×	×	×	×	√
Characteristics of Our Survey and Existing Federated Learning Surveys	×	×	×	×	×	×	×	√
Basic Concept	√	√	√	√	√	√	√	√
Algorithm Principle	×	×	×	√	×	√	√	√
Horizontal Federated Learning	√	√	√	√	√	×	√	√
Vertical Federated Learning	√	√	√	√	√	×	√	√
Federated Transfer Learning	√	√	√	√	√	×	√	√
The Advantages of Federated Learning	×	×	×	×	√	×	√	√
Threat Model	×	×	×	×	×	√	×	√
Poisoning Attack	√	×	√	×	√	√	√	√
Inference Attack	√	×	×	×	√	√	×	√
Model Attack	√	×	×	×	√	√	√	√
Adversarial Attack	×	×	√	×	√	×	√	√
Security Defense Method	√	×	√	×	√	√	√	√
Privacy Protection Technology	√	√	√	√	√	√	√	√
Future Research Direction	√	×	×	×	√	√	×	√

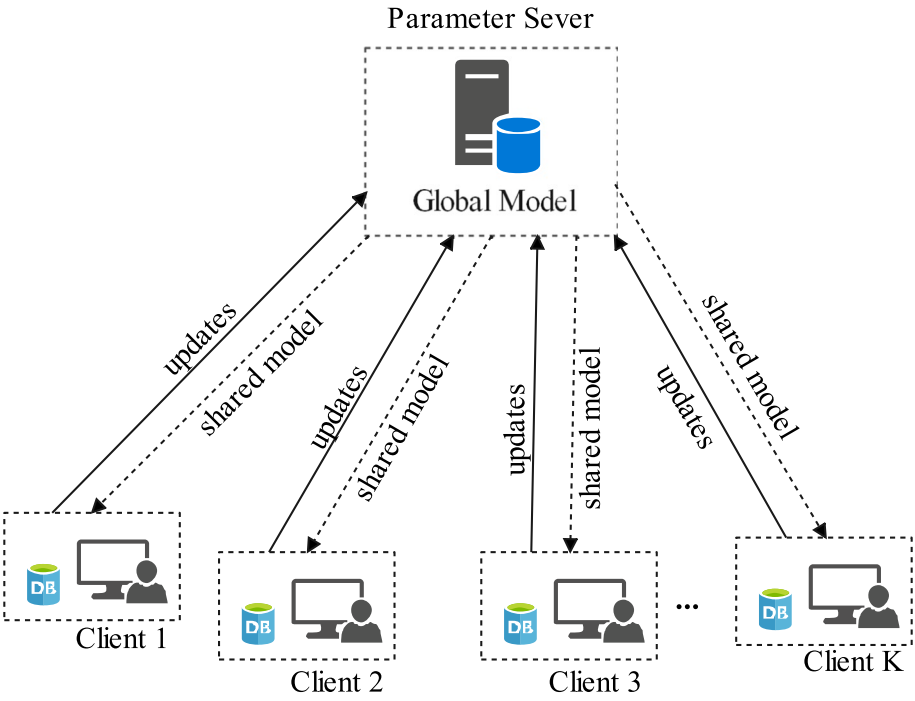

Fig. 1 The framework of FL

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{1}$$

In Equation (1), w_{t+1} represents the update of global model weight (i.e., aggregation model weight update), n represents the total amount of data of K clients, where $\sum^{K} n_k = n$, w represents model parameter.

Observations from participants in FL, the FL scenarios consists of a set of participants consisting of a central server (also known as a parameter server) and K clients, each with its own local dataset D_k. During the learning process, the clients agree on the common goals and model structure, and train model M_{Global} in the total dataset $D = D_1 \cup D_2 \cup \cdots \cup D_K$. At the beginning of the FL training iteration, a subset of clients $C \subseteq K$ is selected to receive the current global state of the shared model based on model weights. After receiving the global state, each client performs local training on its own dataset according to the shared model parameters, and sends the model update obtained after training (i.e. the weights learned locally by the client using the local dataset) to the central server. The server applies updates to the current global model to generate a new model. Equation 2 shows the global model update mechanism.

$$G_{t+1} = G_t + \frac{1}{k_t} \sum_{i \in [k_t]} \Delta L_{t+1}^i \tag{2}$$

In Equation (2), G_t represents the global model parameter of the server side in the t th iteration, k_t represents the number of clients selected in this round, and ΔL_{t+1}^{i} represents the local model update parameter received by the central server from the clients.

After several iterations of the above process, the global model reaches a certain level of accuracy determined by the central server, and FL is complete. Equation 3 represents the target function of the central server.

$$\min_{w} \mathrm{F(w)}, F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{3}$$

In Equation (3), K represents the total number of client devices participating in training, n_k is the data volume of the k th client, and $F_k(w)$ is the local objective function of the k th device. Equation 4 shows the local objective function of the k th device.

$$F_k(w) = \frac{1}{n_k} \sum_{i \in D_k} f_{i(w)} \tag{4}$$

In Equation (4), D_k is the local dataset of the k th client, and $f_i(w) = \alpha(x_i, y_i, w)$ is the loss function generated by the model with parameter w to the instance (x_i, y_i) in dataset D_k.

The average loss function of the local client is obtained by dividing the sum of the loss functions generated by all instances in D_k by the total data volume of the client.

In summary, the FL scenariomainly consists of two phases, namely local update and global aggregation. The local update phase refers to the calculation of gradients by minimizing the loss function of all training data in these devices [22]. Global aggregation involves the following steps: the server collects updated model parameters from different client devices, aggregates them, and then sends the aggregated parameters back to the clients for use in the next training iteration.

3.2 Federated Learning Classification

The feature and sample ID space of the data parties may not be identical, and we classify FL into HFL, VFL, and FTL based on how data are distributed among various parties in the feature and sample ID space [9, 23]. Let the sample ID space of the i th data D_i be x_i, the feature space y_i, and the label I_i. The expressions of the three scenarios of FL are shown in Table 3 [9].

3.2.1 Horizontal Federated Learning

In HFL, datasets of different participants have the same feature space, but they rarely intersect in the sample ID space. HFL is distributed machine learning that divides the dataset horizontally (i.e. the user dimension) under the condition that the user features

Table 3 Classification of federated learning scenarios

Classifications	Expressions
Horizontal Federated Learning	$x_i = x_j, y_i = y_j, I_i \neq I_j \forall D_i, D_j, i \neq j$
Vertical Federated Learning	$x_i \neq x_j, y_i \neq y_j, I_i = I_j \forall D_i, D_j, i \neq j$
Federated Transfer Learning	$x_i \neq x_j, y_i \neq y_j, I_i \neq I_j \forall D_i, D_j, i \neq j$

of the two datasets overlap more while users overlap less, and HFL takes out the parts with the same feature but not exactly the same users for training [24, 25]. For example, "Hey Siri" and "OK Google" in wake-up word recognition [26] are typical applications of horizontal segmentation, because each user speaks the same sentence in a different voice. The schematic diagram of HFL is shown in Fig. 2.

The HFL training process consists of the following steps:

① Initialization: Initializes the federated model parameter w and distributes it to the clients ($w_1 = w_2 = w_k = w$).

② Local training: the client calculates the corresponding output value $y_{pre}^k = X_k w$ and error value L_k of data records. Equation 5 shows the local gradient of the client.

$$\Delta w_k = \frac{\delta L_k}{\delta X_k} \tag{5}$$

In Equation (5), k represents the k th client.

③ Gradient aggregation: The parameter server uses the FedAvg [27] algorithm to aggregate the shared gradients of the clients, and the aggregation gradient can be represented as Eq. 6.

$$\Delta w = \frac{1}{K} \sum_{k=1}^{K} \Delta w_k \tag{6}$$

④ Global parameter update: Parameter server updates global parameters, and Eq. 7 shows the global parameter update.

$$w_{n+1} = w_n + \eta \Delta w \tag{7}$$

In Equation (7), w_k represents the global parameter of the n th iteration, and η represents the learning rate.

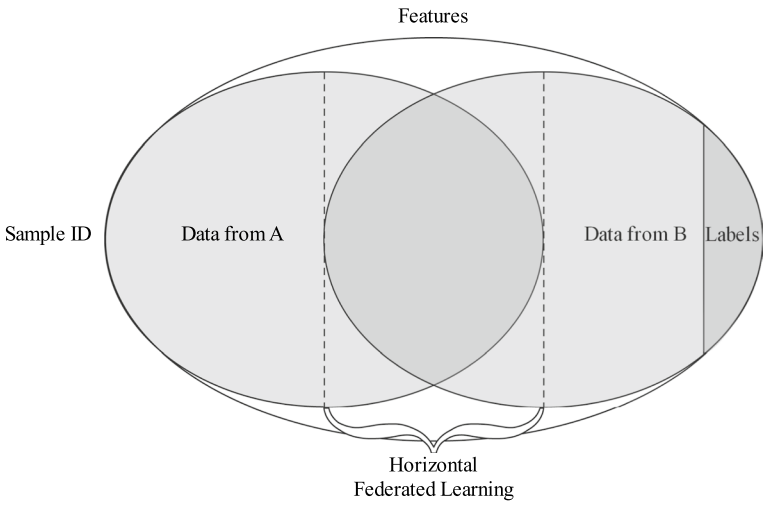

Fig. 2 Schematic diagram of horizontal federated learning

3.2.2 Vertical Federated Learning

In VFL, datasets of different participants have feature space with different attributes, but have the same or similar sample ID space. VFL is distributed machine learning that divides the datasets vertically (i.e. feature dimension) under the condition that the users of two datasets overlap more while user features overlap less, and VFL takes out the parts with the same user but not exactly the same user features for training [28]. For example, the collaboration between different companies can often be viewed as a vertical segmentation situation. VFL usually uses entity alignment techniques [29, 30] to collect overlapping samples of all parties. The schematic diagram of VFL is shown in Fig. 3.

The VFL training process consists of the following steps:

① Initialization: There is sample alignment with the same identifier between clients, and the parameter server initializes the federated model parameters for distribution to clients ($w_1 = w_2 = w_K = w$).

② Local training: The active party uses Eq. 8 to summarize the output value and error value of data records with the same identifier.

$$y_{pre}^{k} = \sum_{k=1}^{K} X_k w_k \tag{8}$$

The intermediate result ΔH_k is transmitted to the passive party so that both sides can obtain the gradient according to Eq. 9.

$$\Delta w_k = \Delta H_k \bullet \frac{\delta H}{\delta X_k} \tag{9}$$

In Equation (9), k represents the k th client and H represents the excitation function [31].

③ Gradient aggregation: The parameter server receives shared gradient information from the clients and gathers them.

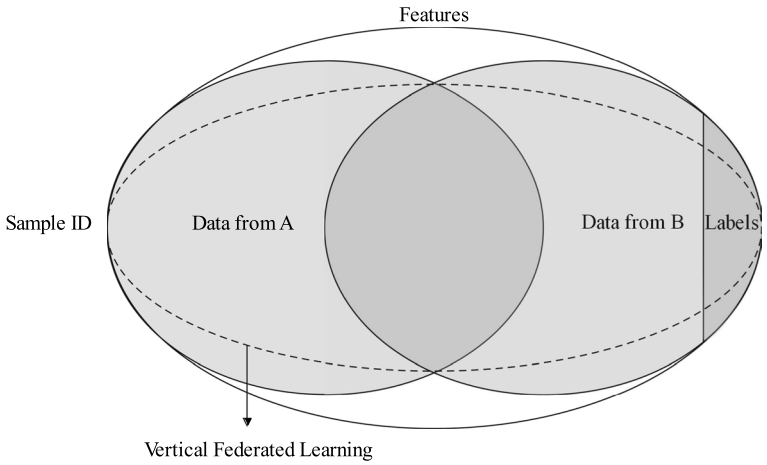

Fig. 3 Schematic diagram of vertical federated learning

④ Global parameter updating: The parameter server uses the shared gradients of participants to update the corresponding global parameter to obtain global parameter w, as shown in Eq. 10.

$$w_{n+1}^{k} = w_{n}^{k} + \eta \Delta w_{k} \tag{10}$$

3.2.3 Federated Transfer Learning

In Federated Transfer Learning (FTL), datasets of different participants have feature space of different attributes, and there is little intersection in the sample ID space [32]. FTL is the combination of FL and transfer learning, which does not divide the data and uses transfer learning to overcome the data or label shortage under the circumstance that the users and user features overlap less in the two datasets [33]. Take the cancer diagnosis system as an example. A group of hospitals want to establish a FL system for cancer diagnosis, but each hospital has different patients and different physical examination results. In this case, federated transfer learning is usually adopted. The schematic diagram of FTL is shown in Fig. 4.

The FTL training process [34] consists of the following steps:

① Initialization: Build server model fs using Eq. 11 and the dataset.

$$arg \min_{\Theta} L = \sum_{i=1}^{n} \varrho(y_i, fs(X_i)) \tag{11}$$

In Equation (11), fs represents the server model to be learned, $\varrho(*,*)$ represents the loss function of the model (such as cross-entropy loss of the classification task), Θ represents all parameters to be learned (namely weight and deviation), and $\{X_i, y_i\}_{i=1}^{n}$ is the sample from server data with the size of n.

② Local training: fs is distributed to all clients and the model of user u is trained by learning objective function Eq. 12.

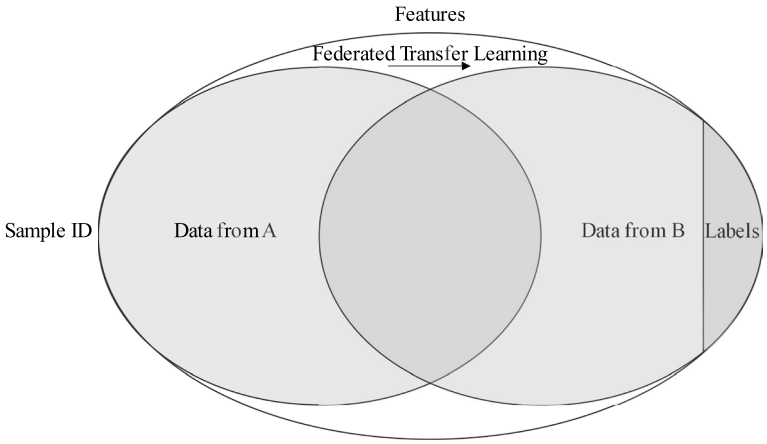

Fig. 4 Schematic diagram of federated transfer learning

$$arg \min_{\Theta^u} L_u = \sum_{i=1}^{n^u} \rho(y_i^u, f_u(X_i^u)) \tag{12}$$

③ Gradient aggregation: After the training of all user models f_u based on the shared cloud model is completed, Homomorphic Encryption is used to update them to the server, and Eq. 13 is used for model aggregation.

$$f_s^{'}(w) = \frac{1}{K} \sum_{k=1}^{K} f_{u_k}(w) \tag{13}$$

④ Global parameter update: The server distributes the aggregation model as the updated cloud model $f_s^{'}$ to all clients, and then transfers learning for each client to get their personalized model f_u.

The above four steps are repeated when more users emerge continuously.

3.3 The Advantages of Federated Learning

As a product of the development of machine learning technology, FL has some advantages.

① User privacy protection: The data of the clients participating in FL is not shared. The data is stored in the local environment to ensure user data security.

② Data flexibility: During the FL process, the client can determine if it needs to quit without affecting the normal running of the FL.

③ Model training that allows large-scale data: FL is based on a global data learning model stored in tens of millions of remote client devices.

4 Security and Privacy Threats

In a FL scenario, attacks can be initiated not only by untrusted servers [7, 35–39], but also by malicious clients [4, 8, 37, 39–41]. In general, we think of parameter servers as honest and curious, and their attacks are considered passive attacks. This means that these servers usually serve strictly according to established learning protocols, but they also try to extract sensitive user information from the model update process. Attacks from malicious clients are called active attacks, in which they attempt to recover sensitive information about other users from aggregated global model parameters. These two attacks have the effect of destroying data privacy. Tables 4 and 5 respectively list some security and privacy threats encountered by FL and the corresponding solutions.

5 Security Challenges

This section divides the security challenges existing in FL into four components: Poisoning Attack, Inference Attack, Model Attack and Adversarial Attack. First, each attack method is classified by fine granularity, and its schematic diagram is drawn. Next, typical attack methods are selected to elaborate, and the attack principle is explained. Then, each method is illustrated by example. Finally, all attack methods are compared, and a summary table of attack methods is listed.

Table 4 The security threats to federated learning and corresponding solutions

Security threats	Ref.	Key contributions
A malicious client conducts poisoning attacks or other attacks to impair model performance.	Shayan et al. [8]	A fully decentralized peer-to-peer (P2P) multi-party machine learning approach (Bicotti) is proposed.
	Li et al. [37]	A committee consensus FL Framework (BFLC) based on blockchain is designed.
In the IoT environment, because FL is attacked, the system has security vulnerabilities.	Jia et al. [42]	The application model of FL based on blockchain in industrial Internet of Things is designed.
	Cui et al. [43]	The authors propose FL for security and privacy enhancement.
	Su et al. [44]	A safe and effective AI of Things scheme supporting FL is proposed.
Malicious clients trained by random data and result-class-inversion datasets can weaken the combinatorial model.	Mugunthan et al. [45]	An accountable FL system (BlockFlow) is designed.
FL security issues caused by model inversion attacks.	Hu et al. [46]	A method (zCDP) combining local gradient perturbation, security aggregation and zero-set DP is proposed to achieve better practicability and privacy protection without trusted server.
	Triastcyn et al. [47]	A secure data publishing framework (FedGP) to protect user privacy in a FL environment is designed.
FL lacks scalability, and it has the problems of security and privacy trade-offs, low communication efficiency and high cost.	Paul et al. [48]	A FL and Private Scaling (FLaPS) architecture to improve system scalability as well as security and privacy is designed.
Secure transport issues in two-hop cooperative networks with untrusted relays.	Sun et al. [49]	A security-aware relay scheme is designed, which uses alternate jamming and secrecy enhancement relay selection to prevent confidential messages from being eavesdropped by untrusted relays.
DP may result in poor model quality when preventing inference attacks.	Lee et al. [50]	A new deep neural network (DNN) is proposed and integrated into FL.
	Kerkouche et al. [51]	The authors propose compression boosts differentially private federated learning.

Table 5 The privacy threats to federated learning and corresponding solutions

Privacy Threat	Ref.	Key Contributions
Isolated data island caused by relevant laws and limitations on FL execution.	de Souza et al. [4]	A distributed machine learning system based on local forest algorithm is proposed.
	Lu et al. [35]	A FL parking recommendation system (FMFParking) based on distributed encryption matrix decomposition is designed.
	Yang et al. [52]	A new framework FedSteg: Secure image steganalysis is designed.
	Liu et al. [53]	A lightweight trusted sharing mechanism (LTSM) is proposed.
	Zhou et al. [54]	A FL model for industrial payload classification is proposed to solve the privacy problem of industrial traffic for payload classification.
	Xin et al. [55]	Private FL-GAN, a GAN model with DP based on FL, is proposed.
Data abuse and privacy disclosure in the IoT environment.	Yin et al. [2]	A secure data collaboration framework (FDC) based on federated deep learning technology and blockchain technology is proposed.
	Rahman et al. [56]	A lightweight hybrid FL framework is proposed.
	Yang et al. [57]	A federated tensor completion method is proposed to solve the recommendation problem of high-dimensional tensor data.
	Suomalainen et al. [58]	A framework to enhance privacy in real-time energy information sharing platform is proposed.
Untrusted servers get sensitive information of users from uploaded gradient vectors.	Zhang et al. [7]	A privacy-enhanced FL solution (PEFL) for big data analysis is proposed.
	Wei et al. [59]	A FL algorithm based on DP is proposed.
Third party and server collusion caused by user privacy leakage.	Chai et al. [60]	A security matrix decomposition framework (dFedMF) in FL environment is proposed.
	Cheng et al. [61]	A novel lossless privacy-preserving tree-boosting system (SecureBoost) is proposed.

The classification of security challenges is shown in Fig. 5. In the attack methods, some sub-methods can be attributed to different superior attack methods, so there is a phenomenon of repeated occurrence of some attack methods in the classification diagram.

5.1 Poisoning Attack

Poisoning attack refers to the fact that attackers manipulate model predictions with training sets during training or retraining, so that the trained models can satisfy the expectation of attackers and destroy models [62]. The methods of manipulating training datasets mainly include contaminating source data, adding malicious samples to training datasets, tampering with some labels in training datasets, deleting some samples in training datasets, etc. [63]. Based on the difference of the attackers' targets, the poisoning attack can be divided into data poisoning attack and model poisoning attack. The schematic diagram of poisoning attack is shown in Fig. 6.

5.1.1 Data Poisoning Attack

Data poisoning attack refers to the fact that attackers contaminate samples in training sets, resulting in low quality of training data, which reduces the quality of models and damages the availability of data and models. According to whether the data label is tampered or not, it can be classified into clean label poisoning attack and dirty label poisoning attack. Clean label poisoning attack is designed to add malicious data to a training dataset. A typical example of a dirty label poisoning attack is the label flipping attack [64], in which the

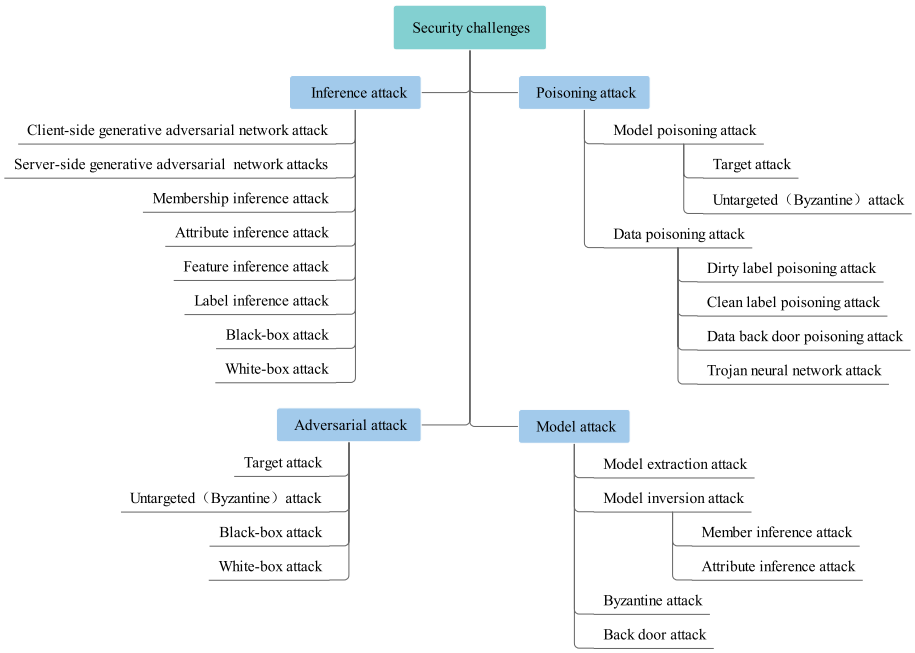

Fig. 5 Classification of security challenges

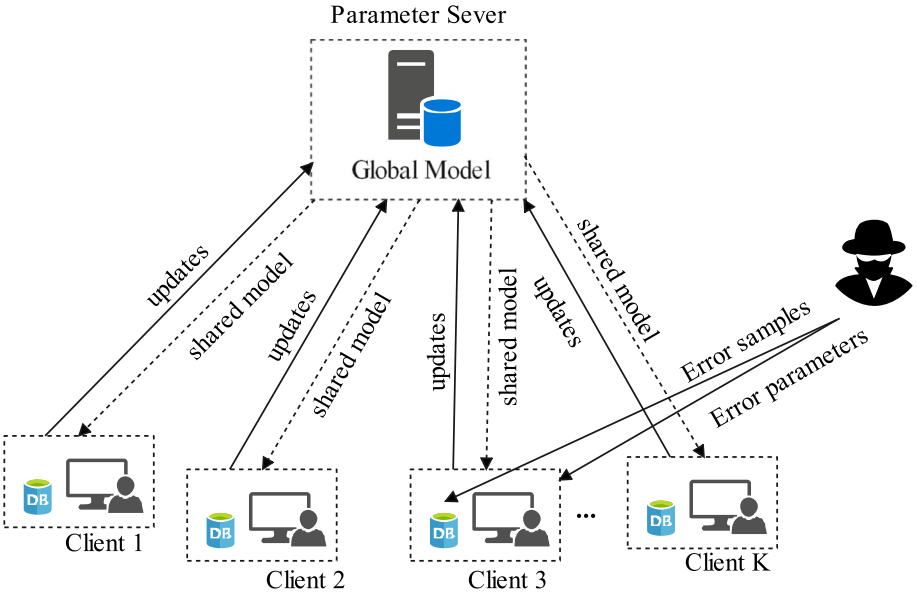

Fig. 6 Schematic diagram of poisoning attack

labels of one class of clean training samples are flipped to another while the features of the data remain unchanged. Traditional dirty label poisoning attacks just reverse the training sample labels in the target class [65]. Some recent literatures have proposed optimized data poisoning attacks [66–68]. For example, mature attackers could inject some elaborate fake malicious data samples (such as label error), destroy the probability distribution of the original training data, and reduce the precision of classification or clustering of the learning model. This kind of attack has been proven in many applications, including handwritten number recognition [64] and PDF malware detection [69]. Another common way of attack is data backdoor poisoning attack [63–70]. By modifying the individual features or small regions of the original training dataset as a backdoor, the attacker can embed it into the model. If the input contains the backdoor features (e.g., a stamp on an image), the model will behave according to the goal of the attacker, while poisoning model in a clean input data on the performance is not affected. Trojan neural network attack also belongs to data poisoning attack [11]. In addition to this, the Trojan neural network and target model are packaged together, and data is input into the Trojan neural network and target model at the same time, and the output is integrated, so as to realize the distribution of Trojan network. It is worth noting that any malicious client can carry out the data poisoning attack, and the attack intensity depends on the degree of attacker's participation in the attack and the amount of contaminated training data. That is, data poisoning attack is less effective in the environment with fewer attackers [71].

5.1.2 Model Poisoning Attack

Model poisoning attack means that attackers disrupt FL by sending incorrect parameters or destroying models during global clustering. Based on whether the attackers focus on a specific goal, the model poisoning attack can be divided into two categories: target attack and

non-target (Byzantine) attack. The target attack refers to an attacker's attack on a specific type of object, while the non-target attack is the attack without distinguishing samples, which is a kind of generalized attack. The authors of [72] study the local model poisoning attack against Byzantine robust FL, whose goal is to destroy the integrity and confidentiality of the model by destroying the integrity of the learning process at the training stage. The authors of [62] propose an optimization-based FL poisoning attack model, which is sufficiently covert and persistent to bypass specific defense methods and avoid catastrophic forgetting. Unlike data poisoning attack, model poisoning attack requires more sophisticated techniques and better computing resources to send data to the server, and its combined effect is stronger than data poisoning attack [73].

5.2 Inference Attack

Inference attack refers to the attacker obtaining infer able information through various means of attack, and then deducing the desired information by using the information, which can be the input features and attribute labels of members, etc. According to the different inference information, inference attack can be divided into the membership inference attack, the attribute inference attack, the feature inference attack, and the label inference attack. Inference attacks can be divided into the white-box attacks [74] and the black-box attacks [75, 76] according to whether the attacked model is known or not. White-box attack is carried out when the attacker knows the model. That is, the attacker can get the prediction output of any input and the intermediate calculation result of hidden layer [74]. The black-box attack is carried out when the attacker only knows the input and output of the model while the parameters of the model are unknown. It is more difficult and less effective than the white-box attack. In addition to that, GAN-based attacks [77, 78] also belong to inference attacks, including the client-side GAN attacks and the server-side GAN attacks. Server-side GAN attack is to calculate the privacy infor mation of user training samples by using periodically exchanged model parameters [79]. Different from the server-side GAN attack, the client-side GAN attack only has aggregation generated global model parameters, and the key of its reconstruction data sample lies in how to obtain the model updates of other users in each round of communication [74]. A schematic diagram of inference attack is shown in Fig. 7.

5.3 Model Attack

Model attack refers to the attack that changes the global model by tampering with the local model of the attacked clients. Typical model attack methods include the model extraction attack and the model inversion attack. A schematic diagram of the model attack is shown in Fig. 8.

5.3.1 Model Extraction Attack

Model extraction attack refers to that the attacker continuously sends data to the target model, expecting to recover the target model locally, and predicts the parameters and functions of the model through the response information obtained, so as to generate an accurate model or similar model to realize the model extraction [12]. The target of the attacker is to steal the model and damage the confidentiality of the model. The accurate model refers to an alternative model constructed by the attacker with similar predictive

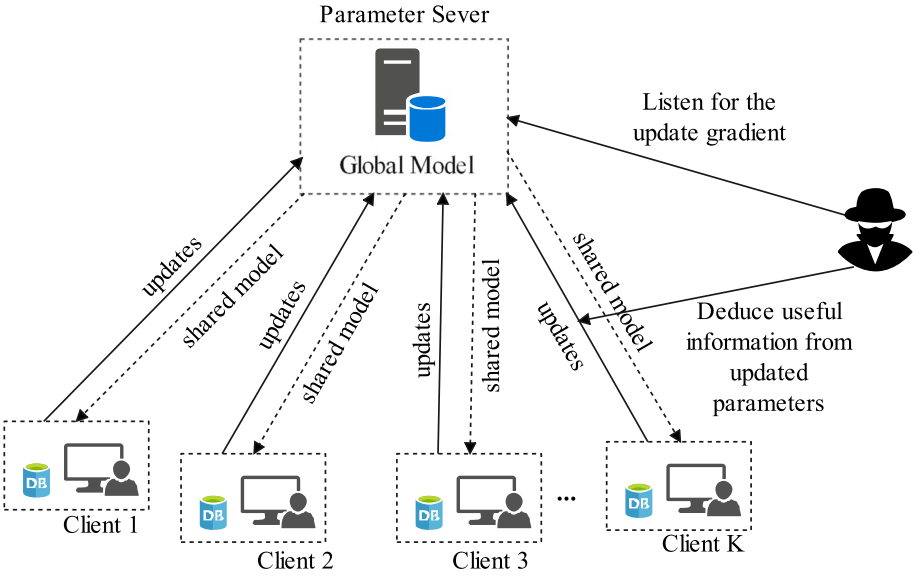

Fig. 7 Schematic diagram of inference attack

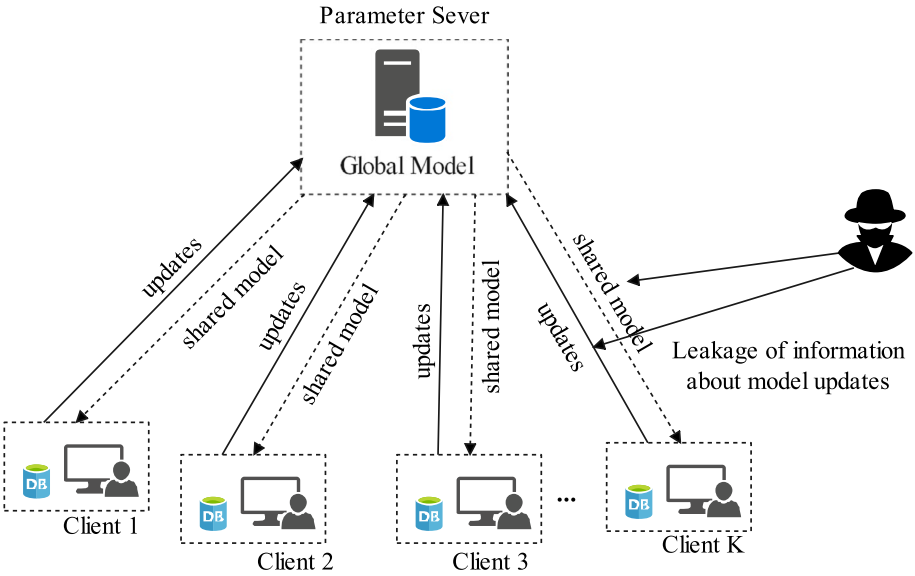

Fig. 8 Schematic diagram of model attack

performance. If the accurate model is stolen, it can generate adversarial samples, so model extraction attacks pose a great threat to the target model. The authors of [80] carry out an attack on BigML and Amazon's online services, extracting an almost identical model and proving that the same attack is equally applicable in multiple scenarios.

5.3.2 Model Inversion Attack

Model inversion attack refers to an attacker who, without knowing the training data, obtains the data information of the target model from the prediction results of the completed training model, so as to obtain the user's private data. The information inferred on the training set from the model inversion attack may be whether a member is included in the training set or some statistical features of the training set. Model inversion attacks can be divided into member inference attacks and attribute inference attacks according to the two kinds of training set information. Under a model inversion attack, a generator that can not have direct access to P but can access to a machine learning model of P and training set Q can recover some variables in training set P [81]. For example, the authors of [82] propose a new class of model inversion attack for face recognition system, which utilize the confidence values displayed in the prediction to recover recognizable images of people's faces under the condition of only giving the name and accessing the machine learning model.

5.4 Adversarial Attack

Adversarial attack refers to a maliciously constructed adversarial samples submitted to a trained model that produces incorrect predictions in a state of high confidence. It is also known as an evasion attack [63]. Adversarial samples are the incorrect samples classified by the classification model after slight perturbations are added to the original samples. One characteristic of adversarial samples is that it only causes model classification errors and can be calibrated to the correct samples. In terms of attack environment, adversarial attack can be divided into the black-box attack and the white-box attack, and can also be divided into the target attack and the non-target attack according to attack purpose. Adversarial attacks can cause powerful damage to the system in the domains of speech and text recognition. Similarly, in the domain of malware detection, malware developers can also use adversarial attack to add some special statements to their software to evade detection by anti-virus software.

In conclusion, the attack methods of some security challenges confronted by FL are described in detail above. Table 6 comprehensively summarizes the attack methods encountered by FL.

6 Threat Countermeasures

This section puts forward corresponding solutions to the security and privacy threats facing FL, which are divided into two categories: security defense method and privacy protection technology. Firstly, the security defense method is decomposed into four sub-methods, namely the poisoning attack defense, the inference attack defense, the model attack defense and the adversarial attack defense. Then the privacy protection technology is also decomposed into four sub-technologies, namely the DP technology, the SMC technology, the HE technology and the VSS. The concepts behind each approach and technique are explained below and how they address the security and the privacy challenges of FL. Figure 9 shows the classification of methods for security and the privacy challenges in FL.

Table 6 Comparison of federated learning attack methods

Attack Types	Principle	Target: Data/ Model	Active/ Passive Attack	White-/ Black-box Attacks	Damage
Clean Label Poisoning Attack	Malicious data is added to the training dataset	Data	Active	White-box and Black-box	Availability
Dirty Label Poisoning Attack	The labels of the clean training samples are flipped to another category, while the features of the data remain the same				
	Flip the label of the clean sample to the specified category				
Data Backdoor Poisoning Attack	By modifying the individual features or small regions of the original training dataset as a backdoor, the attacker can embed it into the model				
Model Poisoning Attack	Attackers disrupt FL by sending incorrect parameters or destroying models during global clustering	Model	Passive	White-box	Integrity and Confidentiality
Byzantine Attack	The attacker controls multiple authorized nodes and arbitrarily interferes or destroys the network	Model	Active	–	Integrity and Confidentiality
Backdoor Attack	The attacker is able to insert hidden backdoors into the model and complete the malicious attack during the prediction phase by triggering simple backdoor triggers	Data	Active	White-box and Black-box	Integrity
Generative Adversarial Network Attack	The attacker can generate the prototype sample of the target training set by the real-time of learning process	Model	Active	White-box	Confidentiality

Table 6 (continued)

Attack Types	Principle	Target: Data/ Model	Active/ Passive Attack	White-/ Black-box Attacks	Damage
Membership Inference Attack	By monitoring the gradient information during the training model, it can detect whether the attacked object is used as the training model	Data	Active and Passive	White-box and Black-box	Confidentiality
Attribute Inference Attack	The attacker attempts to extract unintentionally learned information or dataset attributes unrelated to the learning task				
Feature Inference Attack	By monitoring the gradient information of training model, the data distribution information of attack target can be obtained				
Label Inference Attack	By monitoring the gradient information during the training model, the user's label can be inferred				
Model Extraction Attack	The attacker tries to steal the parameters and hyperparameters of the model and destroys the confidentiality of the model	Model	Active	Black-box	Confidentiality
Model Inversion Attack	The attacker tries to obtain the information of the training dataset from the training model to obtain the user's privacy information	Model	Active and Passive	White-box and Black-box	Confidentiality

Table 6 (continued)

Attack Types	Principle	Target: Data/ Model	Active/ Passive Attack	White-/ Black-box Attacks	Damage
Adversarial Attack	Malicious construct adversarial samples are submitted to the trained model, causing the model to outputs error predictions with high confidence	Data and Model	Passive	White-box and Black-box	Integrity and Availability

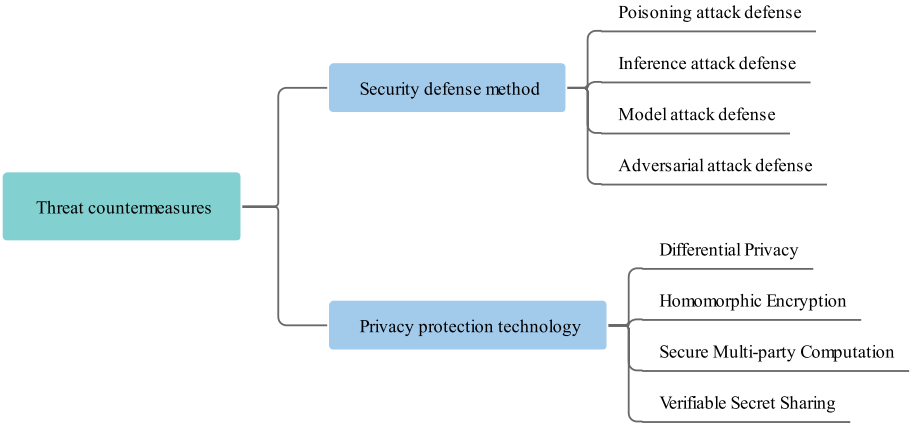

Fig. 9 Classification of methods for security and the privacy challenges in federated learning

6.1 Security Defense Method

In view of the multiple security threats to FL and combined with solutions proposed in the existing literature, the security defense methods are divided into four categories, namely, poisoning attack defense, inference attack defense, model attack defense and adversarial attack defense. The following sections will analyze four types of security defense methods in detail and discuss their applications.

6.1.1 Poisoning Attack Defense

Poisoning attack defense in FL can be considered from the following aspects: ① From the data itself, to ensure the authenticity and reliability of data sources; ② From the point of view of the attacker, sufficient security detection should be conducted to ensure that data and model parameters are not tampered with. In [4], the authors propose DFedForest, a FL system based on local forest algorithm that shares decision trees through blockchain. The system utilizes blockchain technology to ensure mutual trust among participants, register references to local model addresses in a distributed manner, and prevent malicious participants from compromising the accuracy of the model. In [8], the authors adopt a fully decentralized peer-to-peer (P2P) multi-party FL approach (Bicotti), which uses blockchain and cryptography primitives to guarantee privacy between peer clients and protect the process of FL. They propose poof-of-federation (PoF),a layer-1 blockchain consensus protocol that combines the state-of-the-art technology in FL defense to prevent clients from overstepping the system to compromise data integrity and model parameters without sufficient permission. The results show that Bicotti is able to resist the poisoning attacks in previous work. When there are 30% or less attackers in the system, the method can protect individual client updates and maintain the performance of the global model. In [37], in order to avoid model poisoning caused by malicious nodes and privacy disclosure caused by malicious servers, the authors propose a decentralized FL framework based on blockchain, that is, a Blockchain-based FL framework with Committee consensus based on blockchain(BFLC). In the absence of a centralized server, the framework utilizes a blockchain for global model

storage and local model update exchange. In order to implement the proposed BFLC, an innovative committee consensus mechanism is designed, which can effectively reduce the amount of consensus calculation and malicious attacks. In [43], the authors in response to the IoTs in the system anomaly detection, the introduction of blockchain authorsization scattered and asynchronous federated study framework, the framework to ensure the data integrity, to prevent a single point of failure. The generative adversarial network-driven DP algorithm is designed to protect the privacy of the local model parameters, prevent poisoning attacks to some extent, and improve the model accuracy. In [56], in order to prevent raw data leakage, DP is applied to each federated edge node, and blockchain technology is used to aggregate updated model parameters, adding carefully selected noise to protect privacy, striking a balance between privacy protection and model accuracy. In [83], the authors propose a secure FL framework (SFAC) for UAV-assisted MCS to deal with the security and privacy threats for UAV-assisted crowdsensing with FL. First, a blockchain-based collaborative learning architecture is introduced for UAVs to promote efficient data transmission and model training of UAVs in MCS. Next, they use blockchain technology to replace the central server, a decentralized FL mechanism is designed to securely exchange local model updates, and drone contributions in collaborative training are recorded to securely exchange local model updates and validate contributions without a central server. Then, a privacy protection algorithm is designed to protect the privacy of the updated local model by applying local difference privacy. The algorithm has ideal learning accuracy. In the absence of actual knowledge of network parameters, the interactions between UAVs (i.e., data owners) and task publishers are formulated as finite Markov decision processes (MDPs), put forward a kind of based on a two-tier reinforcement learning (RL) of the incentive mechanism to promote the high-quality model sharing of unmanned aerial vehicle (UAV). It turns out that using the disturbance on the device enables the aggregation precision and strict privacy protection required by UAVs. In addition, compared with existing schemes, SFAC can effectively incentivize high-quality local model sharing, enabling optimal strategies and better practicability for participants.

6.1.2 Inference Attack Defense

An Inference attack requires the attacker to obtain the part of a FL user level above, and to perform inference effectively to attack is successful. Then avoid performing effective inference can be a defense against a way, it is need to strengthen the privacy protection mechanisms, HE and DP and some other privacy protection technologies obtained a good application here. For example, DP adopts a specific random algorithm to add appropriate noise to the data to blur the data, so that even if the attacker gets the interactive data also cannot deduce the original data effectively and reduce the risk of information disclosure. In [7], the authors propose a privacy-enhanced FL scheme to protect gradients on untrusted servers. Local gradients of participants are encrypted using the Paillier HE system. The encrypted gradients can be further used for secure aggregation on the server-side, so that untrusted servers can only know the updated and aggregated statistics of all participants, while the private information of each user is well protected. In [36], the authors combine HE with DP and propose an efficient FL protocol based on stochastic gradient descent. The user adds noisy data to each local gradient and then encrypts it for optical performance and security, preventing attackers from inferring the user's privacy from local output (such as gradients). In [50], in order to prevent attackers from identifying the data used to compute gradients, the authors integrate DNN and control algorithm into FL, forming a new

DNN (DgstNN). The goal of DgstNN is to minimize the classification error and maximize the normalized distance between the gradient of the original data and the gradient of the digested image. The loss function related to classification error is represented as classification loss, and the loss function related to normalized distance is represented as distance loss. Increasing the distance loss changes the gradient of the digested image so that it is different from the gradient of the original image, thus preventing the opponent from obtaining the gradient information of the original data. Minimizing distance loss can transform digested images into images that humans and other learning models cannot recognize. Even if an attack successfully recreates an image from a gradient, the result will be a digested image that loses the visual features of the original image. In [51], the authors hold that although DP can guarantee the privacy protection theoretically by noise processing of the exchanged update vector and prevent inference attacks. However, the added noise is proportional to the size of the model, and the quality of the model will become worse with the addition of noise. Therefore, the authors propose the compressed sensing extended FL, which includes two schemes: the first scheme FL-CS, which uses compressed sensing to reduce communication bandwidth. The second scheme, FL-CS-DP, combines compressed sensing and DP to protect user information. The results show that this scheme can not only prevent users from inference attacks to reveal privacy, but also prevent the model accuracy from decreasing. In [60], the authors design a security matrix decomposition framework in FL environment, called dFedMF. First, they design a user-level distributed matrix decomposition framework, when each user only uploads gradient information, not original preference data to the server, the model can be learned. Then they use the HE strengthens the distributed matrix decomposition framework, as long as the HE system can guarantee that ciphertext is indistinguishable for choose plaintext attack, there will not be any information to the server. The results verify the feasibility of dFedMF, the system is safe for honest but curious servers, and there is no loss of accuracy. In [84], the authors propose secure learning, a general design of private FL system, which is an efficient and secure aggregation system that prevents powerful inference attacks by denying access to individual model updates and hiding local models from aggregators. In [85], a new partition defense model (PAMPAS) based on user devices and trusted edge servers is designed to resist the attacks from GANs.

6.1.3 Model Attack Defense

Since the object of model attack is model, it is important to prevent model parameters and hyperparameters from being stolen and other model information from being leaked. The security aggregation algorithm and DP technology are effective defense methods, which can not only effectively defend against inference attack, but also against model attack. In [42], in order to resist model extraction attacks and model inversion attacks, the authors design a FL application model supporting blockchain, based on which a data protection aggregation scheme is formulated. Distributed K-means clustering based on DP and HE, distributed random forest algorithm based on DP and distributed AdaBoost based on HE are presented to realize multiple protection in data sharing and model sharing. In [45], the authors propose a FL system called BlockFlow, which introduces the DP technology and a new model contribution auditing mechanism to protect the data of a single agent, and uses Ethereum smart contract to encourage good behavior. The results show that the system can effectively prevent attackers from obtaining the information of the training dataset from the model. In [47], the authors propose a privacy-protecting data publishing framework,

FedGP, for federated generative privacy in a FL environment. The main idea is to train the GANs on the client to generate artificial data that can replace the real data of the client. These generated samples can be used to evaluate and train machine learning models. Since some clients may not have enough data to train a GAN locally, a federated GAN model is trained. In this way, user data is always retained on the device. In addition, a federated GAN will generate samples from a common cross-user distribution rather than from a single user, increasing overall privacy. The generator components of the GAN are trained by the FedAvg algorithm to extract private manual data samples and assess the risk of information disclosure. By running a model inversion attack to assess the protection provided, training using a federated GAN was demonstrated to reduce information leakage (for example, face detection in recovered images was reduced from 25.5% to 1.2%). FedGP can generate high-quality marker data and significantly reduce the vulnerability of learning models to model inversion attacks. In [86], in order to deal with model inversion attack, the authors propose a PSI protocol based on VFL, which adopts a hybrid encryption algorithm (a method combining the symmetric secret key encryption). This protocol achieves a certain security goal, as long as the number of arbitrary malicious clients collusion is less than a threshold, malicious clients and servers cannot obtain private information of any honest clients, thus achieving the goal of protecting client privacy. In [87], the authors put forward two methods to test whether the model parameters were damaged. One is to detect numerical differences between the parameters used. Comparing the i th parameter provided by each participant, when there is a large gap between the values of the parameters provided by one participant and those provided by other participants, determine this parameter to be an exception. Another method is that the server performs the corresponding processing using $W_{G_1} = W_G + f(\delta_i)$ according to the parameter δ_i uploaded by the client, and then calculates $W_{G_2} = W_G + f(\Delta)$ by using the parameters uploaded by other clients. Where $\Delta = \left\{\delta_j | j = 1, 2, \cdots, n, j \neq i\right\}$, f is the specific function designed. If the difference between W_{G_1} and W_{G_2} exceeds a certain set value, it is inferred that the model update parameter is abnormal.

6.1.4 Adversarial Attack Defense

According to the attack mode of adversarial attack, it can be observed that maliciously constructed adversarial samples submitted to the trained model will cause model classification errors. According to its attack principle, it can be inferred that adversarial training for the model can enhance the robustness of the model. The so-called adversarial training is to use the training set containing adversarial samples and real samples for the training of FL model, and in the training process, the model learns the features of adversarial samples, so as to achieve the role of defense. Another method is to detect the adversarial samples of malicious constructs. As long as the difference between the malicious adversarial samples and the normal samples can be found, the adversarial sample can be detected and the adversarial attack can be prevented. Preventing overfitting of the model is also a way to resist adversarial attacks. If the degree of overfitting is too high, the generalization ability of the model will be weakened and the possibility of successful adversarial attack will be increased. In [88], the authors find that using small batch training data can effectively estimate the characteristics of test samples: The estimated local intrinsic dimensionality (LID) of adversarial examples is significantly higher than that of normal data examples, and this difference becomes more pronounced in deeper layers of DNNs. In the experiment, five most advanced attack methods are used to

generate adversarial examples, whose LID features can be easily distinguished from those of normal examples, and the performance of the provided baseline classifier based on LID outperforms several state-of-the-art detection measures by large margins in five attacks of three benchmark datasets. The experiment proves that the detector based on simple LID is robust to the normal attack based on low confidence optimization. In [89], the authors find that the neural network obtained by using regularized input gradients is robust to adversarial examples, which improves the robustness of adversarial disturbances and prevents model overfitting. In [90], the authors propose a new defense approach based on actual observations that is easily integrated into the model and can reinforce the common weakness of the deep network, smoothing the decision function, without knowing the type of attack used to make adversarial examples. When the model uses the proposed defense, the disturbance required for misclassification is much greater, making the attack detectable, and the detection more stable and less likely to be fooled by the adversarial samples. Experiments show that this method is effective against multiple attacks, which brings almost no cost to the training process, and maintains the predictive performance of the original model against clean samples, which is performed better than the most advanced defense methods. In [91], the authors introduce a defense mechanism called defensive distillation to reduce the effectiveness of adversarial samples. They investigate the extensibility and robustness conferred by the use of defensive distillation when training DNNs. It shows that defensive distillation can reduce the effectiveness of sample generation from 95% to less than 0.5% on the DNN studied. This tremendous achievement can be explained by the fact that distillation results in a 1030-fold reduction in the gradient used for the creation of adversarial samples.

To sum up, the security challenge defense of FL is summarized, as shown in Table 7.

6.2 Privacy Protection Technology

Numerous technologies have been proposed to address privacy-related issues in FL at the present time. Commonly used privacy protection technology can be divided into four categories, namely the DP, the SMC, the HE and the VSS technology. Each technique is explained in detail below and the approach proposed under each technique is discussed.

6.2.1 Differential Privacy

DP technology is mainly used to add random noise to datasets so that attackers cannot infer sensitive information about users even if they know the results posted by users. And accordingly, the addition of noise will also cause the quality loss of statistical data, resulting in the decline of the accuracy of the learning model. However, compared with the privacy protection ability of DP technology, its loss is insignificant. Even so, when dealing with the privacy threat of FL, DP is generally combined with other technologies to ensure user privacy security and avoid a decrease in model accuracy. DP can be used in cases where an attack steals private data from one party during training, or attempts to reconstruct the training set based on the generated gradient. A model calculation is considered differential private if the output is independent of a particular data point of the input data.

DP technology [92] can be expressed as the following algorithm: a random algorithm $M : D \rightarrow R$ satisfies (ε, δ) -differential privacy, if and only if, for any adjacent dataset d with only one data difference, $d \in D$ and any output $S \subseteq R$, satisfies Eq. 14.

$$Pr[M(d) \in S] \leq e^{\varepsilon} Pr[M(d^{'}) \in S] + \delta \tag{14}$$

Table 7 The federated learning of security defense methods and techniques

Ref	Defenses	Techniques and Methods
[4, 8, 37, 43, 56, 83]	Poisoning Attack Defense	Blockchain, local forest algorithms, encryption algorithms, committee consensus, Differential Privacy
[7, 36, 50, 51, 60]	Inference Attack Defense	HE, security aggregation, DP, stochastic gradient descent, DNNs, control algorithms, compressed sensing, matrix decomposition
[42, 45, 47, 86, 87]	Model Attack Defense	HE, DP, K-means clustering, distributed AdaBoost, Ethereum smart contract, FedAvg algorithm, hybrid encryption algorithm, model parameter detection
[88–91]	Adversarial Attack Defense	Comparison of local intrinsic dimensions, gradient regularization, preventing model overfitting, adversarial training, and defensive distillation

In Equation (14), $M(d)$ and $M(d^{'})$ respectively represent the output of algorithm M on datasets d and $d^{'}$. Pr is the output probability of the algorithm. ε is the privacy budget, which is used to control the privacy protection level. The smaller ε is, the stronger the privacy protection capability is. δ is another privacy budget, representing the probability that the tolerable privacy budget exceeds ε. If δ is equal to 0, M is said to satisfy ε -differential privacy.

In [42], the authors design distributed K-means clustering based on DP and HE, distributed random forest based on DP and distributed AdaBoost based on HE, realizing multiple protection in data sharing and model sharing. In [43], the authors design an improved GAN model named DP-GAN, which has one more perceptron: DP identifier compared with traditional GAN. Differential noise is generated by two games running at the same time: the game between the classic generator and the discriminator and the game between the discriminator and the DP identifier. The data generated by the improved GAN model can meet the requirements of data protection and approximate the original data to the best degree. In [46], in order to make DP play a better role in FL, the FL model has better practicability and privacy protection ability. The authors propose a method combining local gradient perturbation, security aggregation and zero-concentrated DP (zCDP). First, in order to protect shared model updates, each client is required to perturb its gradient in each local iteration to ensure that shared model updates before aggregation are differential private. Because of the combination of periodic averaging and client sampling, gradient perturbation produces some noise to model updates and results in low model utility. Therefore, a secure aggregation protocol with low communication overhead is integrated to reduce the increased noise, while zCDP is used to tightly capture the end-to-end privacy loss, which can add less noise with the same DP guarantee. In [59], the authors make a theoretical analysis of the performance of FL algorithm based on DP, and study the convergence performance of FL with noise disturbance at the inherent privacy level. They propose a new framework based on DP that adds artificial noise to the client parameters before aggregation, i.e., noising before model aggregation FL (NbAFL). By adapting to different artificial noise variances, NbAFL can satisfy DP under different protection levels. Then, the theoretical convergence bound of the loss function of the FL model after NbAFL training is established, which proves that there is a tradeoff between convergence performance and privacy protection level: the better the convergence performance, the lower the protection level. In [93], the authors propose a differentially private asynchronous FL scheme (DP-AFL) to solve the privacy problem of mobile edge computing (MFC) in Urban Informatics. In order to protect the privacy of the updated local model, this scheme will incorporate local DP into the gradient descent local training process, and then add it to FL.

6.2.2 Homomorphic Encryption

HE generally encrypts the gradient uploaded by the user during FL [94]. The gradient after HE is a bunch of random numbers, and the attacker cannot deduce any valuable information from the random numbers without the key.

HE allows users to perform operations directly on the ciphertext, and the results obtained from the operations are still ciphertext. The results obtained after decryption are consistent with the results of the original data (plaintext) directly performing various calculations [95]. The HE scheme satisfies Eq. 15.

$$Dec(k_S, Enc(k_p, m_1) \diamondsuit Enc(k_p, m_2)) = m_1 \circ m_2 \tag{15}$$

In Equation (15), m_1 and m_2 are plaintext, k_S is a private key and k_p is a public key. $Enc(*,*)$ is an encryption operation, $Dec(*,*)$ is a decryption operation, $\circ$ is an operation in plaintext field, $\diamondsuit$ is an operation in ciphertext field.

According to the types and times of ciphertext operations supported, HE can be divided into: Partially HE (PHE), Somewhat HE (SHE) and Fully HE (FHE) [96].

PHE only supports addition and multiplication, and the number of operations is not limited, so it can be divided into Additive HE (AHE) and Multiplicative HE (MHE). For example, the Paillier scheme belongs to AHE, and the EI-Gamal scheme belongs to MHE. SHE supports only a limited number of addition and multiplication operations. FHE supports arbitrary operation on ciphertext and the number of operations is unlimited.

In [35], in order to solve the problem of no correlation between data caused by "isolated data island" and data and data features cannot be shared with other data, the authors construct a FL system based on distributed encryption matrix decomposition. Firstly, a framework based on user distributed matrix decomposition is established. In order to increase data privacy protection, HE is added to perform FL based on distributed matrix decomposition. The scheme allows each user to encrypt gradients as they transmit their local gradients, avoiding gradients being acquired or maliciously tampered with during transmission. Because the process does not need a third encryption service provider, it also avoids data leakage caused by third parties. For normal HE schemes, the server is set up to hold the key, which can lead to a serious problem, i.e. if the server does not aggregate before decrypting, the server has access to the user's updates. To solve this problem, in [38], the authors propose a privacy-protected federated extreme gradient boosting scheme (FEDXGB), which is a federated extreme gradient boosting (XGBoost) scheme supporting forced aggregation for moving crowd perception. A new secure gradient aggregation algorithm for FL is designed, which combines the advantages of HE and VSS. Specifically, through a combination of HE and VSS, FEDXGB ensures that the central server does not get the correct decryption results before performing aggregation, while being robust against user loss. The results show that FEDXGB keeps the high performance of XGBoost with less than 1% accuracy loss. FEDXGB makes the performance loss of trained XGBoost negligible, reduces about 23.9% running time and 33.3% communication cost in gradient aggregation, and reduces the computing and communication cost of secure aggregation.

6.2.3 Secure Multi-party Computation

SMC technology can reduce the possibility of information leakage by integrating model gradient updates. SMC in each random encryption when using, do not reuse the encrypted data, need operation on encrypted data directly, don't need to restore the original data, determine the participants before each calculation. In the place where input is not shared, multiple participants aggregate the data by using encryption techniques such as the HE, the secret sharing protocols, and the oblivious transfer protocol. These methods only protect the privacy of training data in the learning process, but cannot prevent inference attacks on the result model [97].

The formal description of SMC is as follows: Assuming that there are m participants $P_1, P_2, \cdots, P_m$ and they have their own dataset $d_1, d_2, \cdots, d_m$, how to safely calculate a convention function $y = (d_1, d_2, \cdots, d_m)$ without trusting a third party, and at the same time, each participant is required not to get any input information from other participants except the calculation result [98]. SMC has the characteristics of input independence, computational correctness, and decentralization. The basic cryptographic protocols of SMC include

Oblivious Transfer (OT) protocol, Garbled Circuits (GC) and Secret Sharing (SS) protocol, Goldreich-Micali-Wigderson protocol (GMW) protocol, etc.

In [99], the authors argue that the use of DP in the presence of a large number of clients leads to a decrease in model accuracy. In order to solve this problem, a method of integrating SMC into DP is proposed. The results show that this method reduces the impact of noise injection when the number of customers increases, while maintaining some robustness. In [100], the authors propose a SMC protocol for a FL framework called security aggregation. Security aggregation utilizes a variety of encryption techniques to prevent the parameter server from acquiring the original client's local updates. The proposed protocol would protect the FL framework from honest but curious attackers and disclose the sum of model parameter updates to the server only after a certain number of updates have been made. The protocol consists of four rounds of operations, each round of which the server collects messages from all clients and computes a separate response to those messages to send to each client. In the first two rounds (preparation stage), secret sharing is initiated. In the third round (submit stage), each client submits encrypted mask model updates to the server, which stacks them up. In the final round (the final stage), the clients expose the encryption secret, enabling the server to expose the aggregated model updates.

6.2.4 Verifiable Secret Sharing

VSS is used to protect important information on clients and prevent information loss, damage, and tampering. In FL, attack may monitor user and task publishers communications to intercept the gradient information or honest and curious task publishers get user's local gradient. VSS uses encrypted sharing to process gradient information uploaded by users to ensure that malicious servers cannot obtain gradient information, reach the role of defense.

VSS includes three parts: client, distributor and secret. The idea is to split secret information into n fragments in an appropriate way, and each fragment after splitting is managed by n different clients. A threshold t is set, and the secret information cannot be recovered when the attacker has any less than t fragments. The secret message can be recovered only when the number of fragments is equal to or greater than t [101]. A typical VSS scheme is constructed based on a polynomial method, which can be divided into two steps: generating and distributing the key and decrypting the key. Equation 16 shows the expression of the key generation method.

$$y_i = K + \sum_{i=1}^{t-1} a_i x^j \ mod \ p \tag{16}$$

In Equation (16), K is the secret, t is the threshold of SS, a_i is the coefficient of the polynomial, and modulus p is set for safe calculation (making decryption difficult).

Then solve the linear equations according to the key provided by t participants, and solve the polynomial coefficients and secret K.

In [100], the authors design a secure aggregation scheme based on Shamir secret sharing to ensure that learning models update parameters securely in the face of honest but curious servers, while controlling the complexity of secret sharing protocols and keeping computing and communication costs low in large datasets. In [102], the authors propose a VFL algorithm based on logistic regression. After the server realizes secret sample alignment, the intermediate results are calculated according to the aligned samples. Then the server generates public and private key pairs, encrypts approximate losses and intermediate results, and obtains the encryption gradient through local training. Since the server is

honest but curious, random masks need to be generated to prevent the server from infering the user's private information based on the original gradient. With the help of the third party, this method ensures privacy security by means of encryption method, and reduces the cost of encryption calculation by approximate loss function.

In summary, the technologies to address FL privacy challenges are summarized, as shown in Table 8.

7 Future Research Direction

In this section, we will discuss some future research directions, classified according to corresponding high-level challenges, which will be useful for future work and research. Based on the classifications and solutions discussed above, we identify a set of criteria for future solutions that will serve as a reference for scholars and developers studying ways to improve security and privacy in future FL systems.

7.1 Suggestions for Security Challenges

In the environment of FL, most security solutions only consider attacks executed in a single direction, ignoring more complex attack scenarios. From this perspective, an attacker can formulate a joint attack plan and consider more complex attack scenarios to counter the existing security defense mechanism. For example, an attack can involve multiple client devices to execute, multiple attack methods attack the specified target synchronously, malicious clients collude with servers (such as sharing private keys) to attack other honest users. Security solutions tailored to a single attack cannot easily adapt to collusive attacks. The security analysis of the security matrix decomposition proposed in [60] shows that using a typical HFL security definition, assuming honest clients and honest but curious servers, such a security definition is weak. Malicious clients may collude with the server to attack other users, revealing the privacy of honest users, and may cause backdoor attacks, causing security problems.

In order to design efficient and safe security defense schemes, several defense criteria are drawn up below. When dealing with security challenges of FL, defense schemes can be designed based on the following criteria, as shown in Table 9.

7.2 Suggestions for Security Challenges

The privacy protection scheme in FL is designed to be universal for client devices and data samples in all scenarios. However, in practice, data samples in different situations and even data samples on a single device are often different, so the universal privacy protection scheme cannot achieve the expected effect in practice. Therefore, special privacy protection schemes can be designed to protect customer privacy in specific situations, which can be combined with universal privacy protection schemes. Privacy protection for FL should also consider the loss to FL systems when using a range of encryption methods, especially DP. From the perspective of security challenge, the attacker's attack on FL system not only causes security risks but also risks of privacy disclosure. In addition, software and hardware, which have nothing to do with the FL system itself, should also be taken into account.

Table 8 Summary of technologies that address federated learning privacy challenges

Ref	Techniques	Principle
[42, 43, 46, 59, 93]	Differential Privacy	Adding random noise to datasets so that attackers cannot infer sensitive information about users even if they know the results posted by users
[35, 38]	Homomorphic Encryption	Encrypts user-uploaded gradients during FL, the gradient after HE is a bunch of random numbers, and the attacker cannot deduce any valuable information from the random numbers without the key
[99, 100]	Secure Multi-party Computation	In the place where input is not shared, multiple participants aggregate the data by using encryption techniques such as the HE, the secret sharing protocols, and the oblivious transfer protocol
[100–102]	Verifiable Secret Sharing	VSS uses encrypted sharing to process gradient information uploaded by users to ensure that malicious servers cannot obtain gradient information

Table 9 Summary of technologies that address federated learning privacy challenges

Sequence Number	Criteria
Criteria 1	Consider a security definition that assumes the server is honest but curious and has a small number of malicious clients
Criteria 2	Ensure that customers are using their data honestly and not falsifying data to participate in local model training
Criteria 3	Consider the collusion of multiple malicious clients and the collusion of servers with malicious clients to disrupt the training model
Criteria 4	Consider the performance loss of the model caused by participants getting off at any time during model training
Criteria 5	Focus on adaptive attackers who evade detection by adaptive limiting malicious attacks and reducing attack effects
Criteria 6	Consider the impact on the accuracy of the global model for FL when implementing the formulated security solution
Criteria 7	Consider customer privacy breaches when implementing a security solution, such as in the process of accessing customer training data to determine if a customer is engaged in suspicious behavior
Criteria 8	Consider the trade-offs of security solutions in system security, privacy protection, and model effectiveness

In order to design efficient and safe privacy defense schemes, several defense criteria are drawn up below. When dealing with privacy challenges of FL, defense schemes can be designed based on the following criteria, as shown in Table 10.

In addressing the security and the privacy challenges of FL, in addition to considering traditional defense approaches and implementing the above defense criteria, integrating other technologies with FL to propose more FL architecture is an attractive defense solution. For example, blockchain can provide high security for FL training through immutable

Table 10 Summary of technologies that address federated learning privacy challenges

Sequence Number	Criteria
Criteria 1	Considering the problem that DP noise injection reduces the model accuracy
Criteria 2	Consider the trade-off between encryption schemes and communication efficiency
Criteria 3	When active and passive attacks are performing additional local computing, consider privacy and communication problems caused by them
Criteria 4	Consider the quality of the participants and the possibility of privacy leakage caused by communication patterns between the parties
Criteria 5	Develop an adaptive privacy protection scheme to ensure a certain degree of privacy protection
Criteria 6	Consider the privacy issues caused by the security of hardware and software itself
Criteria 7	Considering active and passive inference attack, because most research only considers how to counter passive inference attack
Criteria 8	Design a hybrid privacy protection scheme, combine the advantages of different privacy protection technologies, and find the trade-offs between the advantages and disadvantages brought by them
Criteria 9	Consider switching from designing a universal privacy protection scheme to a design-specific privacy protection scheme

block ledgers. By utilizing blockchain, FL can execute a decentralized data ledger where each device can act as a client with equal rights, eliminating the need for a central server [103] and reducing the risk of a single point of failure. In particular, the integration of FL and blockchain creates a new paradigm called FLchain that guarantees the safety of learning updated information in the form of immutable blocks through the use of blockchain [104]. In FLchain, an adversary can attempt to manipulate the training output by training the local model with forged data of the design and replacing the global model before updating the transmission. By adjusting the difficulty level of blockchain mining, the likelihood of poisoning attacks on training data can be reduced without degrading training performance [105]. Driven by the unique advantages of blockchain, another blockchain-based FL architecture called PriModChain is introduced in [106]. DP is applied to locally generated models with artificial noise to reduce the possibility of identifying personal records. By using smart contracts, communication between the central authorsity and distributed users exchanging global ML models is secured, which facilitates update validation protocol and provides transparency for FL updates. This function forcibly performs unbiased and error-cost data operations to enhance the security and reliability of FL processes under external data threats. In addition, the use of blockchain introduces additional delays associated with block mining, which creates new challenges for FL systems as FL customers need to wait for the mining process to complete before receiving model updates and executing the next round of training [107].

8 Conclusion

The distributed learning mode of FL makes it unnecessary for users to upload original data to the server. The proposed learning mode alleviates the inevitable privacy security problems in the era of big data and becomes an indispensable technology to protect privacy. Since FL is the product of machine learning, its system still has inherent security problems and derived privacy problems. This paper expounds the security and privacy threats of FL from the angle of attack and defense. First, a detailed investigation of the existing survey of security and privacy protection of FL is carried out, and our survey is compared with existing related surveys to highlight the unique contribution of our survey. Secondly, it introduces the related knowledge of FL and makes a comprehensive analysis of three scenarios of FL. Later, it illustrates various specific threats to the security and privacy protection of FL in the form of tables, and gives the corresponding solutions. Then it classifies the security challenges according to the collected related threats of FL and illustrates the classification by combining pictures and examples. Next, security defense methods and privacy protection technologies are proposed to address the challenges of FL. Finally, by considering the drawbacks in existing attack and defense methods, we make some suggestions on how to propose much more excellent privacy protection and secure schemes in FL, and develop a set of criteria against malicious attacks and privacy leakage, hoping it can be useful for the relevant researchers and developers when planning their own defense schemes.

Author Contributions All authors contributed to the study conception and design. The chapter design and the first draft of the manuscript were written by Xingpo Ma and Mengfan Yan, all authors commented on previous versions of the manuscript, and all authors read and approved the final manuscript.

Funding This work was supported by the Key Research Program for Colleges and Universities in Henan Province in China (23A520021).

Availability of Data and Material Data sharing not applicable to this article as no data sets were generated or analyzed during the current study.

Declarations

Conflicts of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Wahab, O. A., Mourad, A., Otrok, H., & Taleb, T. (2021). Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. *IEEE Communications Surveys & Tutorials, 23*(2), 1342–1397.
2. Yin, B., Yin, H., Wu, Y., & Jiang, Z. (2020). FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things. *IEEE Internet of Things Journal, 7*(7), 6348–6359.
3. Bo, H. (2016). "Network security Law'' provides legal protection for our data management. *China Telecommunications Trade, 12*, 17–19.
4. de Souza, L. A. C., Rebello, G. A. F., Camilo, G. F., Guimarães, L. C., & Duarte, O. C. M. (2020). DFedForest: Decentralized federated forest. In *2020 IEEE international conference on blockchain (blockchain)* (pp. 90–97). IEEE.
5. Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1310–1321).
6. Song, M., Wang, Z., Zhang, Z., Song, Y., Wang, Q., Ren, J., & Qi, H. (2020). Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications, 38*(10), 2430–2444.
7. Zhang, J., Chen, B., Yu, S., & Deng, H. (2019). PEFL: A privacy-enhanced federated learning scheme for big data analytics. In *2019 IEEE global communications conference (GLOBECOM)* (pp. 1–6). IEEE.
8. Shayan, M., Fung, C., Yoon, C. J., & Beschastnikh, I. (2020). Biscotti: A blockchain system for private and secure federated learning. *IEEE Transactions on Parallel and Distributed Systems, 32*(7), 1513–1525.
9. Janhan, W., Shijing, S., Janzong, W., & Jing, X. (2022). Federated learning attack and defense survey. *Big Data Research, 8*(5), 12–32.
10. Tiankai, L., Bi, Z., & Guang, C. (2021). Federated learning surveyconcept, technology, application and challenge. *Journal of Computer Applications*.
11. Shuang, S., Xiaohui, L., Yan, L., & Xing, Z. (2021). Survey on security and privacy protection in different scenarios of federated learning. *Application Research of Computers*, 3527–3534.
12. Chuanxin, Z., Yi, S., Degang, W., & Huawei, G. (2021). Survey of federated learning research. *Chinese Journal of Network and Information Security, 7*(5), 77–92.
13. Zhuangzhuang, W., Hongsong, C., Limin, Y., & Lifang, C. (2021). Review of federal learning and data security. *Intelligent Computer and Applications*, (01), 126–129+133.
14. Bing, C., Xiang, C., Jiale, Z., & Yuanyuan, X. (2020). Survey of security and privacy in federated learning. *Journal of Nanjing University of Aeronautics & Astronautics, 52*(5), 10.
15. Jun, Z., Guoying, F., & Nan, W. (2020). Survey on security and privacy preserving in federated learning. *Journal of Xihua University (Natural Science Edition), 39*(4), 9.
16. Jia, W., & Lu, M. (2020). Analysis of federated learning. *Modern Computer, 25*, 6.
17. Zhu, H., Zhang, H., & Jin, Y. (2021). From federated learning to federated neural architecture search: A survey. *Complex & Intelligent Systems, 7*(2), 639–657.
18. Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.
19. Konečnỳ, J., McMahan, H. B., Ramage, D., & Richtárikk, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527.
20. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282).
21. Changyin, L., Xuebin, C., Chundi, M., & Shufen. (2021). Improved federated average algorithm based on tomographic analysis. *Computer Science, 48*(8), 32–40.

22. Biying, P., Haihua, Q., & Jialun, Z. (2019). Research on federated machine learning techniques with different data distributions. *Proceedings of 5G network innovation symposium*.
23. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li,Y., Liu, X., Li, Y., & He, B. (2021). A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*.
24. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning, 14*(1–2), 1–210.
25. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST), 10*(2), 1–19.
26. Leroy, D., Coucke, A., Lavril, T., Gisselbrecht, T., & Dureau, J. (2019). Federated learning for keyword spotting. In *Icassp 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6341–6345).
27. McMahan, H. B., Moore, E., Ramage, D., & Arcas, B. A. (2016). Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629.
28. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2019). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 13*(3), 1–207.
29. Christen, P. (2012). Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection.
30. Yan, Z., Guoliang, L., & Jianhua, F. (2016). A survey on entity alignment of knowledge base. *Journal of Computer Research and Development, 53*(1), 165.
31. Lipeng, G., & Hui, Z. (2018). Convolutional neural network based on pelus softplus nonlinear excitation function. *Journal of Shenyang University of Technology, 40*(1), 54–59.
32. Saha, S., & Ahmad, T. (2021). Federated transfer learning: Concept and applications. *Intelligenza Artificiale, 15*(1), 35–44.
33. Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.
34. Chen, Y., Qin, X., Wang, J., Yu, C., & Gao, W. (2020). Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems, 35*(4), 83–93.
35. Lu, C., Fan, Y., Wu, X., & Zhang, J. (2021). Fmfparking: Federated matrix factorization for parking lot recommendation. In *2021 IEEE seventh international conference on big data computing service and applications (bigdataservice)* (pp. 131–136).
36. Hao, M., Li, H., Xu, G., Liu, S., & Yang, H. (2019). Towards efficient and privacy-preserving federated deep learning. In *ICC 2019-2019 IEEE international conference on communications (ICC)* (pp. 1–6).
37. Li, Y., Chen, C., Liu, N., Huang, H., Zheng, Z., & Yan, Q. (2020). A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network, 35*(1), 234–241.
38. Liu, Y., Ma, Z., Liu, X., Ma, S., Nepal, S., Deng, R. H., & Ren, K. (2020). Boosting privately: Federated extreme gradient boosting for mobile crowd-sensing. In *2020 IEEE 40th international conference on distributed computing systems (ICDCS)* (pp. 1–11).
39. Hao, M., Li, H., Luo, X., Xu, G., Yang, H., & Liu, S. (2019). Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics, 16*(10), 6532–6542.
40. Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2019). Blockchain and federated learning for privacy-preserved data sharing in industrial iot. *IEEE Transactions on Industrial Informatics, 16*(6), 4177–4186.
41. Wan, W., Lu, J., Hu, S., Zhang, L. Y., & Pei, X. (2021). Shielding federated learning: A new attack approach and its defense. In *2021 IEEE wireless communications and networking conference (wcnc)* (pp. 1–7).
42. Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K., & Liang, Y. (2021). Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in iiot. *IEEE Transactions on Industrial Informatics, 18*(6), 4049–4058.
43. Cui, L., Qu, Y., Xie, G., Zeng, D., Li, R., Shen, S., & Yu, S. (2021). Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures. *IEEE Transactions on Industrial Informatics, 18*(5), 3492–3500.
44. Su, Z., Wang, Y., Luan, T. H., Zhang, N., Li, F., Chen, T., & Cao, H. (2021). Secure and efficient federated learning for smart grid with edge-cloud collaboration. *IEEE Transactions on Industrial Informatics, 18*(2), 1333–1344.
45. Mugunthan, V., Rahman, R., & Kagal, L. (2020). Blockflow: An accountable and privacy-preserving solution for federated learning. arXiv preprint arXiv:2007.03856.

46. Hu, R., Guo, Y., & Gong, Y. (2021). Concentrated differentially private federated learning with performance analysis. *IEEE Open Journal of the Computer Society, 2*, 276–289.
47. Triastcyn, A., & Faltings, B. (2020). Federated generative privacy. *IEEE Intelligent Systems, 35*(4), 50–57.
48. Paul, S., Sengupta, P., & Mishra, S. (2020). Flaps: Federated learning and privately scaling. In *2020 IEEE 17th international conference on mobile ad hoc and sensor systems (MASS)* (pp. 13–19).
49. Sun, L., Ren, P., Du, Q., Wang, Y., & Gao, Z. (2014). Security-aware relaying scheme for cooperative networks with untrusted relay nodes. *IEEE Communications Letters, 19*(3), 463–466.
50. Lee, H., Kim, J., Hussain, R., Cho, S., & Son, J. (2021). On defensive neural networks against inference attack in federated learning. In *Icc 2021-IEEE international conference on communications*(pp. 1–6).
51. Kerkouche, R., Ács, G., Castelluccia, C., & Genevès, P. (2021). Compression boosts differentially private federated learning. In *2021 IEEE European symposium on security and privacy (euros & p)* (pp. 304–318).
52. Yang, H., He, H., Zhang, W., & Cao, X. (2020). Fedsteg: A federated transfer learning framework for secure image steganalysis. *IEEE Transactions on Network Science and Engineering, 8*(2), 1084–1094.
53. Liu, C., Guo, S., Guo, S., Yan, Y., Qiu, X., & Zhang, S. (2021). Ltsm: Lightweight and trusted sharing mechanism of IoT data in smart city. *IEEE Internet of Things Journal, 9*(7), 5080–5093.
54. Zhou, P. (2020). Federated deep payload classification for industrial internet with cloud-edge architecture. In *2020 16th international conference on mobility, sensing and networking (MSN)* (pp. 228–235).
55. Xin, B., Yang, W., Geng, Y., Chen, S., Wang, S., & Huang, L. (2020). Private fl-gan: Differential privacy synthetic data generation based on federated learning. In *Icassp 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2927–2931).
56. Rahman, M. A., Hossain, M. S., Islam, M. S., Alrajeh, N. A., & Muhammad, G. (2020). Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach. *IEEE Access, 8*, 205071–205087.
57. Yang, J., Fu, C., Liu, X. Y., & Walid, A. (2021). Recommendations in smart devices using federated tensor learning. *IEEE Internet of Things Journal*.
58. Suomalainen, J., & Julku, J. (2016). Enhancing privacy of information brokering in smart districts by adaptive pseudonymization. *IEEE Access, 4*, 914–927.
59. Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q. S., & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15*, 3454–3469.
60. Chai, D., Wang, L., Chen, K., & Yang, Q. (2020). Secure federated matrix factorization. *IEEE Intelligent Systems, 36*(5), 11–20.
61. Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., & Yang, Q. (2021). Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems, 36*(6), 87–98.
62. Zhou, X., Xu, M., Wu, Y., & Zheng, N. (2021). Deep model poisoning attack on federated learning. *Future Internet, 13*(3), 73.
63. Yingzhe, H., Xingbo, H., Jinwen, H., Guozhu, M., & Kai, C. (2019). Privacy and security issues in machine learning systems: A survey. *Journal of Computer Research and Development, 56*(10), 2049–2070.
64. Biggio, B., Nelson, B., & Laskov, P. (2012). Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389.
65. Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., & Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems, 31*.
66. Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., & Roli, F. (2017). Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security* (pp. 27–38).
67. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)* (pp. 19–35).
68. Fang, M., Gong, N. Z., & Liu, J. (2020). Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of the web conference 2020* (pp. 3019–3025).
69. Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., & Roli, F. (2015). Is feature selection secure against training data poisoning? In *International conference on machine learning* (pp. 1689–1698).
70. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In *International conference on artificial intelligence and statistics* (pp. 2938–2948).

71. Liu, Y., Ma, S., Aafer, Y., Lee, W. C., Zhai, J., Wang, W., & Zhang, X. (2017). Trojaning attack on neural networks.
72. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning* (pp. 5650–5659).
73. Lyu, L., Yu, H., & Yang, Q. (2020). Threats to federated learning: A survey. arXiv preprint arXiv:2003.02133.
74. 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 *2019 IEEE symposium on security and privacy (sp)* (pp. 739–753).
75. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., & Zhu, J. (2019). Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7714–7722).
76. Yin, Z., Yuan, Y., Guo, P., & Zhou, P. (2021). Backdoor attacks on federated learning with lottery ticket hypothesis. arXiv preprint arXiv:2109.10512.
77. Ren, H., Deng, J., & Xie, X. (2022). Grnn: Generative regression neural network-a data leakage attack for federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST), 13*(4), 1–24.
78. Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 603–618).
79. Phong, L. T., Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2017). Privacy-preserving deep learning: Revisited and enhanced. In *International conference on applications and techniques in information security* (pp. 100–110).
80. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction {APIs}. In *25th usenix security symposium (usenix security 16)* (pp. 601–618).
81. Veale, M., Binns, R., & Edwards, L. (2018). Algorithms that remember: Model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376*(2133), 20180083.
82. Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1322–1333).
83. Wang, Y., Su, Z., Zhang, N., & Benslimane, A. (2020). Learning in the air: Secure federated learning for UAV-assisted crowdsensing. *IEEE Transactions on Network Science and Engineering, 8*(2), 1055–1069.
84. Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Möllering, H., Nguyen, T. D., Rieger, P., Sadeghi, A., Schneider, T., & Yalame, H. (2021). Safelearn: Secure aggregation for private federated learning. In *2021 IEEE security and privacy workshops (SPW)* (pp. 56–62).
85. Ching, C. W., Lin, T. C., Chang, K. H., Yao, C. C., & Kuo, J. J. (2020). Model partition defense against GAN attacks on collaborative learning via mobile edge computing. In *Globecom 2020-2020 IEEE global communications conference* (pp. 1–6).
86. Lu, L., & Ding, N. (2020). Multi-party private set intersection in vertical federated learning. In *2020 IEEE 19th international conference on trust, security and privacy in computing and communications (trustcom)* (pp. 707–714).
87. Bhagoji, A. N., Chakraborty, S., Mittal, P., & Calo, S. (2019). Analyzing federated learning through an adversarial lens. In *International conference on machine learning* (pp. 634–643).
88. Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Schoenebeck, G., Song, D., Houle, M. E., & Bailey, J. (2018). Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint arXiv:1801.02613.
89. Ross, A., & Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32).
90. Zantedeschi, V., Nicolae, M. I., & Rawat, A. (2017). Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM workshop on artificial intelligence and security* (pp. 39-49).

91. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)* (pp. 582–597).
92. Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science, 9*(3–4), 211–407.
93. Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2019). Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics, 16*(3), 2134–2143.
94. Fang, H., & Qian, Q. (2021). Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet, 13*(4), 94.
95. Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., & Liu, Y. (2020). {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *2020 usenix annual technical conference (usenix atc 20)* (pp. 493–506).
96. Li, Z., Gui, X., Gu, Y., Li, X. S., Dai, H. J., & Zhang, X. J. (2018). Survey on homomorphic encryption algorithm and its application in the privacy-preserving for cloud computing. *Journal of Software, 29*(7), 1830–1851.
97. Jayaraman, B., Wang, L., Evans, D., & Gu, Q. (2018). Distributed learning without distress: Privacy-preserving empirical risk minimization. *Advances in Neural Information Processing Systems, 31*.
98. Zuowen, T., & Lianfu, Z. (2020). Survey on privacy preserving techniques for machine learning. *Journal of Software, 31*(7), 2127–21.
99. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security* (pp. 1–11).
100. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 1175–1191).
101. Zhu, H., Goh, R. S. M., & Ng, W. K. (2020). Privacy-preserving weighted federated learning within the secret sharing framework. *IEEE Access, 8*, 198275–198284.
102. Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., & Thorne, B. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv preprint arXiv:1711.10677.
103. Kim, H., Park, J., Bennis, M., & Kim, S. L. (2019). Blockchained on-device federated learning. *IEEE Communications Letters, 24*(6), 1279–1283.
104. Lu, Y., Huang, X., Zhang, K., Maharjan, S., & Zhang, Y. (2020). Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks. *IEEE Transactions on Industrial Informatics, 17*(7),
105. Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Yu, S., Li, B., & Zheng, G. (2020). Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet of Things Journal, 7*(6), 5171–5183.
106. Arachchige, P. C. M., Bertok, P., Khalil, I., Liu, D., Camtepe, S., & Atiquzzaman, M. (2020). A trustworthy privacy preserving framework for machine learning in industrial iot systems. *IEEE Transactions on Industrial Informatics, 16*(9), 6092–6102.
107. Nguyen, D. C., Ding, M., Pham, Q. V., Pathirana, P. N., Le, L. B., Seneviratne, A., Li, J., Niyato, D., & Poor, H. V. (2021). Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal, 8*(16), 12806–12825.

Xingpo Ma received his Ph.D. degree in computer application technology from Central South University in China in 2013. Since 1st July 2014, he has been working in Xinyang Normal University in China. He was awarded the title of the youth backbone teacher by Xinyang Normal University in 2015. He is now the member of Chinese Association of Automation. His research interests include federated learning and IOT.

Mengfan Yan received her bachelor's degree in computer science and technology from Xinyang Normal University in China in 2021. Since 1st September 2021, she has been pursuing her master's degree in the same university. In 2021, she won the first-class scholarship of Henan Province, China. Her research interests include blockchain technology and federated learning.