Machine Intelligence Research
www.mi-research.net

20(1), February 2023, 19-37
DOI: 10.1007/s11633-022-1343-2

Federated Learning with Privacy-preserving and Model IP-right-protection

Qiang Yang[1,2] Anbu Huang[1] Lixin Fan[1] Chee Seng Chan[3]
Jian Han Lim[3] Kam Woh Ng[4] Ding Sheng Ong[5] Bowen Li[6]

[1] WeBank, Shenzhen 518057, China

[2] Hong Kong University of Science and Technology, Hong Kong 999077, China

[3] University of Malaya, Kuala Lumpur 50603, Malaysia

[4] University of Surrey, Guildford GU2 7XH, UK

[5] University of Aberystwyth, Wales SY23 3DD, UK

[6] Shanghai Jiao Tong University, Shanghai 200240, China

Abstract: In the past decades, artificial intelligence (AI) has achieved unprecedented success, where statistical models become the central entity in AI. However, the centralized training and inference paradigm for building and using these models is facing more and more privacy and legal challenges. To bridge the gap between data privacy and the need for data fusion, an emerging AI paradigm federated learning (FL) has emerged as an approach for solving data silos and data privacy problems. Based on secure distributed AI, federated learning emphasizes data security throughout the lifecycle, which includes the following steps: data preprocessing, training, evaluation, and deployments. FL keeps data security by using methods, such as secure multi-party computation (MPC), differential privacy, and hardware solutions, to build and use distributed multiple-party machine-learning systems and statistical models over different data sources. Besides data privacy concerns, we argue that the concept of "model" matters, when developing and deploying federated models, they are easy to expose to various kinds of risks including plagiarism, illegal copy, and misuse. To address these issues, we introduce FedIPR, a novel ownership verification scheme, by embedding watermarks into FL models to verify the ownership of FL models and protect model intellectual property rights (IPR or IP-right for short). While security is at the core of FL, there are still many articles referred to distributed machine learning with no security guarantee as "federated learning", which are not satisfied with the FL definition supposed to be. To this end, in this paper, we reiterate the concept of federated learning and propose secure federated learning (SFL), where the ultimate goal is to build trustworthy and safe AI with strong privacy-preserving and IP-right-preserving. We provide a comprehensive overview of existing works, including threats, attacks, and defenses in each phase of SFL from the lifecycle perspective.

Keywords: Federated learning, privacy-preserving machine learning, security, decentralized learning, intellectual property protection.

Citation: Q. Yang, A. Huang, L. Fan, C. S. Chan, J. H. Lim, K. W. Ng, D. S. Ong, B. Li. Federated learning with privacy-preserving and model IP-right-protection. *Machine Intelligence Research*, vol.20, no.1, pp.19–37, 2023. http://doi.org/10.1007/s11633-022-1343-2

1 Introduction

In recent years, artificial intelligence (AI) has made great progress in many commercial applications, including computer vision[1, 2], natural language processing[3, 4], recommender systems[5, 6], etc. However, behind the super-fast development, the drawbacks of traditional AI approaches are also revealed, which are that they rely heavily on the availability of large-scale and high-quality data but do not provide a mechanism for securely obtaining and using it. For example, the development of computer vision benefited from large-scale public datasets like ImageNet[7], which is essentially based on a centralized data model. From e-commerce to online video, based on historical data, recommender systems can analyze user preferences precisely and recommend the most relevant items to users. In biology, by training on publicly available data consisting of 170 000 protein structures from the protein data bank (PDB)[8], AlphaFold, developed by DeepMind, achieved high accuracy predictions of protein structure[9, 10]. These examples are all centralized data-driven computation systems, and they require that data scattered across multiple devices be first uploaded to a central database before being used for training the statistical models.

Centralized data fusion for AI modeling is facing more and more legal and ethical challenges. In practice, data is

Review
Manuscript received March 2, 2020; accepted June 2, 2020
Recommended by Associate Editor-in-Chief Liang Wang

often spread across multiple end devices and held by different individual users or organizations, data in different locations is heterogeneous in form and distribution. Fusing the data into a central database inevitably increases privacy leakage risks. With the increasing awareness of privacy concerns, governments are strengthening data privacy laws to prevent privacy leakage, such as general data protection regulation (GDPR) in the EU[11], California consumer privacy act (CCPA) in the USA[12], data security law (DSL) in China[13]. On the other hand, due to uniqueness and rarity, the value of data is also a challenge that cannot be neglected, its value will disappear gradually whenever data can be shared and copied, simply because no organization is willing to share data without benefit.

In order to eliminate the drawbacks caused by data fusion, Google proposed a new training paradigm, called federated learning (FL)[14], to address data challenges. The original FL requires model parameters, not raw training data sets, to be exchanged between multiple devices during the whole training process, which can greatly mitigate data privacy risks. However, existing works have shown that vanilla FL without protection on exchanged model parameters may not always provide strong security guarantees. Zhu et al.[15] demonstrated that the original training data can be recovered from gradients. Phong et al.[16] showed that even a small portion of original gradients can expose information about local data. Besides, beyond the training stage, vanilla FL is also vulnerable to various kinds of attacks during the entire FL lifecycle, which includes the following steps: data preprocessing, training, evaluation, and deployments[17]. For example, data can be poisoned in the preprocessing stage[18]. Membership inference attacks can occur in the model deployment phase[19]. It is thus important to emphasize that security guarantees are an essential part of FL system design.

Moreover, as statistical models are the central entities in AI, multiple assets that include the training data, hardware, and human expertise, are involved when developing and deploying FL models in practice. This makes "model management" a critical issue. To prevent models from being misused or plagiarized without authorization, we reinforce awareness of model intellectual property and introduce IP-right-preserving mechanism for models in federate learning. In this paper, we realize federated model IPR protection by embedding watermarks into deep neural network (DNN) model parameters, achieving good results in practice.

In summary, the true spirit of federated learning lies in its ability to provide strong privacy-preserving and model IP-right preserving, to distinguish it from vanilla FL, we call it secure federated learning (SFL). In contrast to many existing FL works that only provide very weak or no security guarantees, we emphasize that the principle of SFL should receive more attention in both industry and academia. This article gives a comprehensive overview of key aspects of SFL, including existing works on both security guarantees throughout the entire lifecycle and model IP-right protection. In the rest of the article, we will use federated learning (FL) and secure federated learning (SFL) interchangeably and refer to federated learning systems that employ certain security mechanisms unless stated otherwise.

1.1 Related works

In the FL literature, Yang et al.[20, 21] introduced the categorizations of FL and extended the scope of federated learning to include also vertical federated learning (VFL) scenarios. Kairouz et al.[17] discussed the FL progress and presented several challenging problems for future directions. Li et al.[22] discussed the unique characteristics and challenges of federated learning.

Several existing security-related surveys are published with similar motivations to ours. For example, Lyu et al.[23] discussed the threats to federated learning. Bouacida and Mohapatra[24] conducted a comprehensive survey on the security issues and defense strategies under the FL setting. Mothukuri et al.[25] summarized common threat models faced by the FL system and provided corresponding defense strategies.

Recently, Liu et al.[26] analyzed the security issues throughout the multi-phase of the FL execution, which includes the data and behavior auditing phase, training phase, and predicting phase, this survey covers more scopes and scenarios about FL security. However, their discussions mainly focus on horizontal federated learning (HFL), nor do they discuss how to protect IP-right issues of federated learning.

1.2 Contribution

Compared to the previous works, the main contributions of our paper are as follows:

1) We reiterate the core concept of secure federated learning and emphasize that security guarantees should cover the entire FL lifecycle, which includes the following steps: data preprocessing, model training, evaluation, and deployment.

2) We provide a general SFL architecture, which covers both HFL and VFL scenarios, and we summarize existing works on threats, attacks, and defense in each phase throughout the entire lifecycle.

3) We view the model intellectual property right as an important part when building a secure FL system, and provide detailed implementations on how to protect federated models′ IPR.

2 Overview of secure federated learning

In this section, we discuss the definition and system architecture of secure federated learning.

2.1 Security guarantees

We first clarify our definitions of security, which involves five security aspects as shown in Fig. 1.

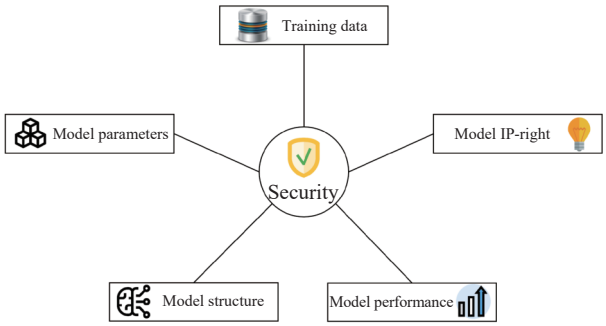


Fig. 1 Protected objects of secure federated learning

Keeping private training datasets safe. There are two different levels of approaches that meet the requirement, the simplest one is to keep local data sets on devices locally, but security is not sufficient enough. Another high-level approach is to be able to provide security protocols to deal with model inversion attacks and prevent raw data sets being stolen.

Keeping model parameters safe. As aforementioned, the attacker can reconstruct original data sets by model inversion.

Keeping model structure safe. This is not a required field, in the VFL setting, each participant holds a partial model while the global model structure is unknown. However, in the HFL setting, in most cases, the global model structure is pre-defined, and anyone who participates in the FL system knows the global model structure in advance.

Keeping model performance intact. In most cases, there is a trade-off between privacy loss and utility loss[27], how to balance security and model performance is a challenging and open problem.

Keeping the model IP-right safe. FL involves collaboration among multiple parties, where the concept of "model" becomes important, to this end, IP-right-preserving is required to prevent model assets from being stolen.

2.2 Definition

SFL refers to secure collaborative distributed machine-learning methods and architectures that satisfy the following conditions:

1) They provide a security mechanism to protect data/model security and user privacy using tools that include but are not limited to: multi-party computation (MPC)[28, 29], differential privacy (DP)[30, 31], and encrypt-based solutions (can be either software-based encryption solution such as homomorphic encryption (HE)[32] or hardware-based solutions such as trusted execution environment (TEE)[33–35]).

2) There are clearly defined threat models as well as corresponding provably correct defense strategies, with the purpose of providing and clarifying the security scope of federated learning throughout the entire lifecycle.

3) They also include methods to protect the IP-rights of the trained models, with the purpose of preventing the model from being plagiarized.

2.3 Architecture

In this section, we introduce the key components that constitute SFL system architecture. Without loss of generality, we assume there are three participants to jointly train and use a statistical model. A typical SFL architecture is as shown in Fig. 2.

In Fig. 2, we show secure HFL (Fig. 2(a)) and secure VFL (Fig. 2(b)) respectively. At a high level, the general SFL architecture consists of the following three components:

Component 1: Distributed machine learning (DML). Under the FL scenario, collaborative training among multiple devices is realized based on distributed machine learning framework. However, unlike traditional DML, there are several key differences between FL and DML:

1) FL is motivated by data privacy and security, while DML is motivated by large-scale computation.

2) Participants of FL can be either individuals or organizations, while DML is a multi-node system, each participant is a compute node in a single cluster or data center.

3) FL allows different participants to have different configurations (like data distribution, data size, hardware, network, etc.). In contrast, DML is more stable, the configurations of each node are almost the same.

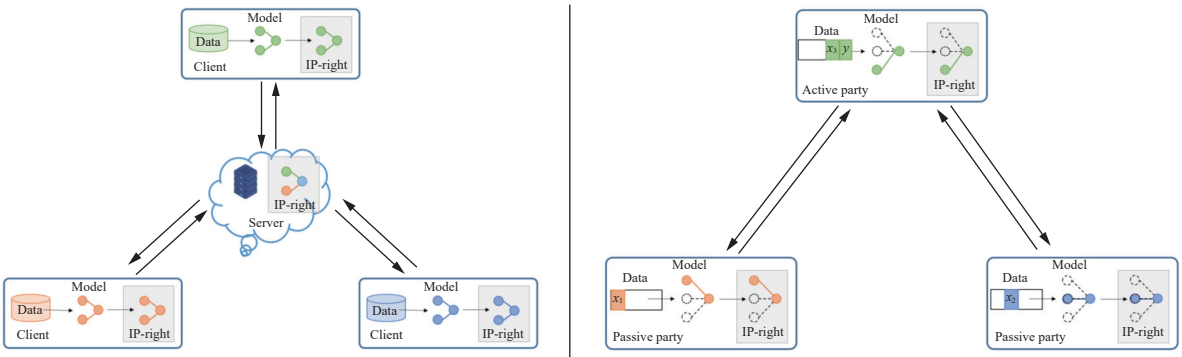
4) FL requires incentive mechanisms to encourage more participants to participate in FL ecosystem, while DML does not require any such mechanism.

Component 2: Security protocol. Security is at the core of federated learning. However, as aforementioned, vanilla FL can be vulnerable to different kinds of attacks throughout the lifecycle.

To mitigate the risks caused by adversarial attacks, a series of secure operation steps would be pre-negotiated by multiple participants, the goal is to complete the task requirements without compromising data privacy. We call these pre-negotiated and secure operation steps are security protocols.

In summary, security protocols can be either based on traditional privacy-preserving computation, such as MPC, DP, and HE, or algorithm-based approaches, such as modifying the training loss function, and model aggregation improvement.

Component 3: Model IPR protection. Conceptu-



(a) Secure horizontal federated learning (b) Secure vertical federated learning

Fig. 2 The architecture of secure federated learning

ally, IPR refers to all rights associated with models owned by an entity, anyone cannot be used unless authorized. The motivation for Model IPR can be boiled down to the following reasons:

1) Since multiple models are involved in the training phase, whenever failures happen, to locate where the responsibility lies, it is critical to be able to trace back to the original creators of models.

2) FL models are of high commercial value, it is necessary to prevent adversaries from plagiarizing, misusing, and re-distributing valuable FL models without legal permission.

3) Incentive mechanisms are essential parts of the FL ecosystem, based on the principle of "more pay for more work", IPR can let the FL manager know the contributions of different models.

3 Security of federated learning lifecycle

In the following sections, we first briefly review the concept of federated learning lifecycle and then summarize the potential security risks at each stage, respectively.

FL is a distributed learning paradigm, in which the process is typically initiated by a party based on a specific application purpose. For example, a financial company wants to update the risk prediction model by jointing with multiple financial institutions for this purpose. Kairouz et al.[17] discussed the lifecycle of a model in federated learning, based on that, we claim that there are four phases that are susceptible to various kinds of attacks, as shown in Fig. 3.

Preprocessing. Participants first identify the problems and requirements to be solved with FL. There are two major tasks at this stage. The first one is data preprocessing, according to the task requirements, engineers first transform raw training data sets to make them suitable for building machine learning models, in most cases, each client will generate and store the data independently, data preprocessing could be done locally. However, in some cases, additional data need to be maintained and downloaded from the server-side. The second step is to prepare the initialized model, to this end, the client will send an instruction to the server to request the global model.

At this stage, malicious modification of training data sets, such as mislabeling, noising, and poisoning, are the most significant security threat and affect subsequent

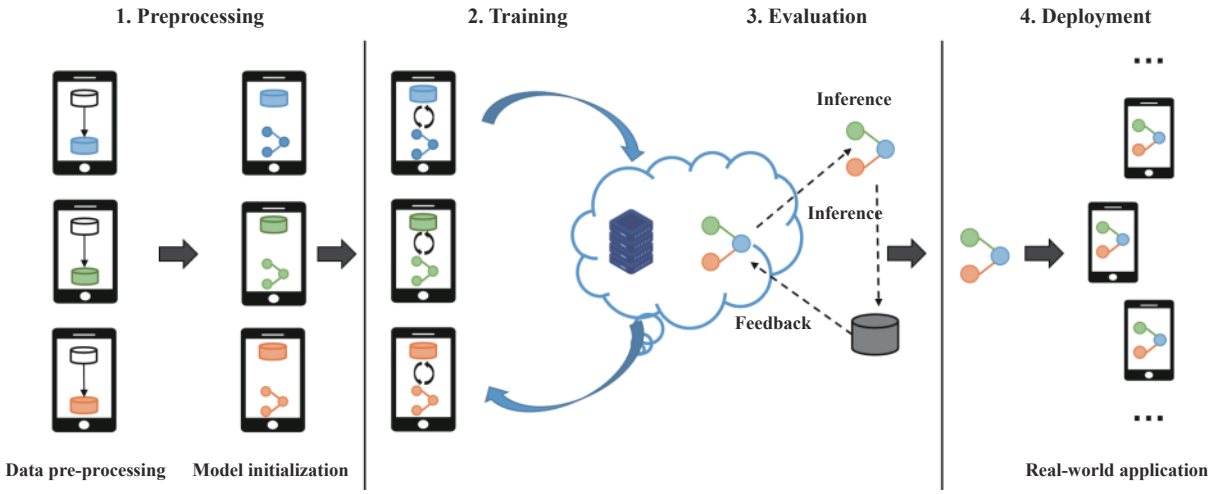


Fig. 3 Threat phases of federated learning lifecycle adapted from [17]

processes.

Federated model training. Based on the distributed architecture, multiple participants collaboratively train a share global model while keeping the training data sets on the device locally.

The model training phase is vulnerable to various kinds of attacks, including poisoning attacks, backdoor attacks, byzantine attacks, etc.

Federated model evaluation. Model evaluation is the process of using different evaluation metrics to evaluate a machine learning model's performance, and feedback to determine whether to stop training or not. Model evaluation can be done either on the server or on the local client.

The potential security risks at this stage mainly come from evasion attacks (a.k.a. adversarial examples)[36–39], the goal is to fool the model to make the wrong output by carefully perturbing the training examples.

Deployment. The final step is to select the updated model and integrate it into the real-world application to make practical business decisions.

The potential security risks at this stage come from three aspects, i.e., evasion attacks, model inference attacks, and model plagiarism.

Fig. 4 summarizes the security risks of different stages throughout the entire lifecycle.

4 Security of preprocessing

In the context of federated learning, the training data sets are generated and stored locally, which makes it hard to be stolen by other parties. However, if the client is controlled by a malicious user, the local data sets can be modified arbitrarily, and then the polluted data affect the following training step. In this section, we discuss the possible threats, attacks, and corresponding defenses during the preprocessing stage.

4.1 Client-side attacks

If a participant is controlled by a malicious user, he (or she) can modify the data at will before the training phase, including mislabeling, noising, and poisoning. Low-quality samples impair model training and reduce the model's performance.

In general, the impacts of client-side attacks at preprocessing stage rely heavily on the number of malicious clients. Due to the client-selection mechanism of federated learning[40], there is no guarantee that a malicious client will be selected at each round, or even if selected but not frequently, model aggregation can cancel the backdoor model's contribution. Besides, from the client's perspective, the percentage of polluted data in the overall training sample is also a critical factor that affects the attack.

Defense. The selection mechanism of federated learning is a natural defense strategy. Improving the selection mechanism with adaptive strategy[41, 42], rather than random choices, can further reduce the loss caused by malicious clients.

Anomaly detection to identify malicious clients is another effective approach to defend against client-side attacks in the preprocessing phases. For example, with a pre-trained model, the FL manager can check the training datasets to filter out potential adversarial attackers.

4.2 Server-side attacks

Another possible threat is that the attacker controls the central server, which orchestrates the training process and holds the additional data and the global model. Server-side attacks usually occur in HFL, the attacker

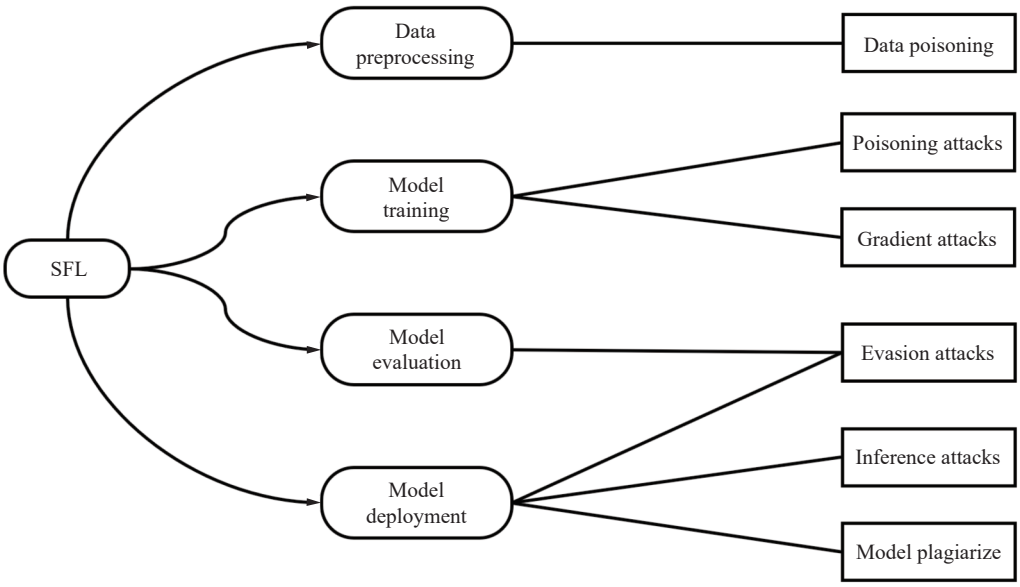


Fig. 4 An overview of security risks at different stages

can modify the data or the global model parameters whatever he/she wants, and then distribute the fake data or global model to the selected clients for training.

The server-side attack is harder to defend against than client-side attacks, in the worst case, the server can distribute the fake data or global model to all the clients, which makes it equivalent to all participants being malicious.

Currently, there is still no good strategy to defend against server-side attacks, that′s why in most FL literature, the central server is required to be trusted. Another solution is to use TEE as a trusted third party[43, 44], which provides the code and data′s confidentiality and integrity guarantees.

5 Security of model training

Attacking in the model training phase is the main research area of federated learning security, the adversary can attack the model training procedure in the following ways.

1) In the HFL setting, the attacker can either control the central server or clients, or eavesdrop and steal the transmitted parameters (Fig. 5(a)).

2) In the VFL setting, the attacker can either control the passive party or the active party, or eavesdrop and steal the transmitted parameters (Fig. 5(b)).

In the following subsections, we discuss the threats, attacks, and corresponding defense strategies in the training phase. According to the capabilities of the adversary, we classify model training attacks into gradients attacks and poisoning attacks.

5.1 Gradient inversion attacks and defenses

5.1.1 Gradient inversion attacks

The goal of gradient inversion attacks is to reconstruct or recover the sensitive information from the shared gradients.

Hitaj et al.[45] discussed the GAN-based attack to generate label-specific prototypical samples from gradients, which are meant to be private to other clients. However, their approach is limited and only works when labels among multiple parties are overlapping.

Zhu et al.[15] proposed deep leakage from gradients (DLG), DLG does not require a GAN model and additional information other than gradients, the key idea of which is to optimize the synthetic gradients as close as to the original gradients, which makes the synthetic data close to the real training data when the optimization is done. However, DLG is susceptible to convergence and label consistency problems, to this end, Zhao et al.[46] proposed improved DLG (iDLG), which guarantees to extract the ground-truth labels from the shared gradients.

Geiping et al.[47] proposed inverting gradients to recover the original input image, by setting the loss function to cosine similarity with the total variation (TV) norm. Compared to DLG/iDLG, inverting gradients performed well even on deep and non-smooth models.

Wang et al.[48] proposed self-adaptive privacy attack from gradients (SAPAG), which sets distance measure as Gaussian kernel-based of gradient difference. Zhu and Blaschko[49] proposed R-GAP, which provides a recursive procedure to recover data from gradients.

The above attacks are mainly applied in HFL scenarios, recently, Jin et al.[50] proposed catastrophic data leakage in vertical federated learning (CAFE), to perform large-batch data leakage attacks with improved data recovery quality under the VFL setting.

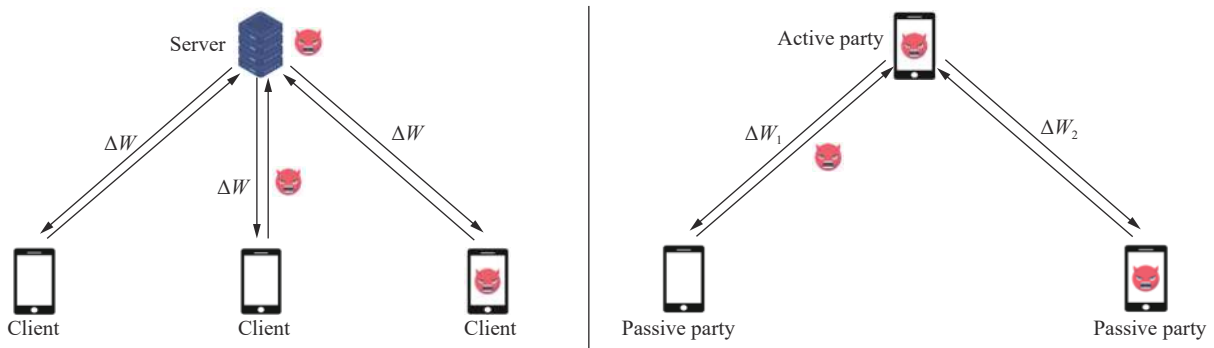
We summarize some common gradient inversion attack approaches in Table 1 for reference.

5.1.2 Defense against gradient inversion attacks

Several defense strategies have been proved feasible with respect to defending against inversion attacks.

Encrypted gradients. One straightforward approach is to encrypt the gradient, which makes the gradient values unavailable without secret keys. For example, Hardy et al.[52] showed how to use Taylor approximation to approximate the loss function, which enables the training process of FL can be executed under the encrypted setting. Phong et al.[16] proposed using homomorphic encryption to encrypt the gradients before sending.

However, encryption-based solutions are vulnerable to



(a) Potential privacy leakage risks in HFL

(b) Potential privacy leakage risks in VFL

Fig. 5 Three different ways to threaten the training procedure

Table 1 An overview of existing works on gradient inversion attacks adapted from [50]

Methods	Approach	Adversary control	Supported scenarios
Deep models under the GAN[45]	GAN-based	Client	HFL
DLG[15]	Minimize L2 distance between dummy gradients and original gradients	Central server (HFL) / Active party (VFL)	HFL, VFL
iDLG[46]	Minimize L2 distance between dummy gradients and original gradients	Central server (HFL) / Active party (VFL)	HFL, VFL
Inverting gradients[47]	Cosine similarity and TV norm	Central server	HFL
SAPAG[48]	Gaussian kernel-based distance	Central server	HFL
R-GAP[49]	Recursive gradient loss	Central server	HFL
CAFE[50]	L2 distance; TV norm; Internal representation norm	Passive party	VFL
GGL[51]	GAN-based	Central server	HFL

suffering from efficiency problems, how to balance safety and efficiency is still a challenging problem. For example, Zhang et al.[53] proposed an efficient HE solution, called BatchCrypt, for cross-silo federated learning, which can significantly reduce the communication overhead caused by encryption.

Gradient compression. Gradient compression is an approach to obfuscate the model structure. Specifically, gradient compression can be either pruning or sparsification, since part of the gradients is missing, making the recovered images far away from the original data[54].

Noisy gradient methods. Unlike encryption-based solutions, the noise-based solution is to perturb the gradient by adding noise to the gradient to achieve differential privacy. Since time-consuming operations such as encryption and decryption are unnecessary, the noise-based solution is more efficient in practice.

For example, McMahan et al.[55] introduced a new algorithm, called DP-FedAvg, for user-level differentially private training of large neural networks under federated settings. Wei et al.[56] proposed a general framework combining FL with differential privacy, by adjusting different amounts of noise to ensure distinct protection levels.

However, noise-based solutions require the algorithm to carefully adjust the noise generation to keep the performance of the model, such as model accuracy, intact. Otherwise, performance may be badly compromised. As shown in [27], how to balance safety and utility is still a challenging problem.

5.2 Poisoning attacks and defense

5.2.1 Poisoning attacks

According to the attacker′s goal, we classify poisoning attacks into the following two categories.

1) Targeted attacks, or backdoor attacks, aim to reduce the model′s performance on those examples with certain features while maintaining good performance on the rest of the other examples. In most cases, poisoning attacks not only require data modification in the preprocessing phase, but also require properly designing the training algorithm to achieve the goal.

Take image classification as an example, the attacker wants the model to misclassify images with specific patterns (vertical red stripe at the upper-left corner (Fig. 6 (a)), yellow background (Fig. 6 (b))) to an attacker-chosen class[40], while the main task is not compromised.

Bagdasaryan et al.[40] leveraged the model replacement approach to make backdoor attacks more persistent. Xie et al.[57] further introduced distributed backdoor attacks, where a global backdoor trigger is decomposed into multiple local patterns, each of which is embedded into the training datasets of different malicious clients. Huang[58] discussed how to achieve backdoor attacks under the dynamic environment.

2) Unlike targeted attacks, untargeted attacks aim to degenerate the model performance. For example, Feng et al.[59] proposed the DeepConfuse framework, which uses an autoencoder to add imperceivable noises to the training data, so that the polluted data confuse the corresponding classifier trained on it, and make the wrong output when feeding with new clean data.

Byzantine attacks are another type of untargeted attack, Hu et al.[60] proposed a method called weight attack, the key idea is lying the attacker′s data set size so that model weight is changed when executing model aggregation. Fang et al.[61] proposed local model poisoning attacks, which manipulate the local models uploaded from the compromised devices to the central server during the training process

5.2.2 Defense against poisoning attacks

FL provides numerous security protocols to defend against poisoning attacks during model training.

Byzantine-robust aggregation, which aims to improve the classical FedAvg algorithm[14] and provide secure and robust aggregation to mitigate byzantine attacks. Yin et al.[62] proposed median and trimmed mean aggregation to remove abnormal local models. Blanchard et al.[63] proposed the Krum aggregation rule, a byzantine-resilient algorithm for distributed stochastic gradi-

(a) Vertical stripe at the upper-left corner

(b) Yellow background

Fig. 6 backdoor example

ent descent (SGD), which provides a convergence guarantee even though multiple byzantine workers exist. Xie et al.[64] proposed Zeno aggregation rule for synchronous SGD, at least one honest worker is enough to ensure good defense performance.

FL with MPC has widely been studied. For example, Dong et al.[65] combined secret sharing with distributed machine learning to achieve a high-level security guarantee without compromising model performance. Kanagavelu et al.[66] proposed to adopt MPC to achieve privacy-preserving model aggregation for FL.

Like encryption-based solutions, MPC-based solutions are vulnerable to suffering from efficiency problems. It is important to regard MPC as a set of technologies (or primitives), including but not limited to secret sharing (SS)[29], oblivious transfer (OT)[67], and garbled circuits (GC)[68]. Improving the efficiency of MPC depends largely on the breakthroughs of low-level primitives.

Hardware solution. TEEs have many different implementations with different forms, including Intel's SGX[69, 70], Arm's TrustZone[71, 72], and AMD SEV[73], each varying in its ability to offer privacy protection. Combining TEE with FL has been applied in many applications. For example, Mo et al.[43] proposed DarkneTZ to mitigate attacks against the neural network, and then, they further designed a general FL framework for mobile systems to protect user privacy[74]. Huang et al.[44] proposed a new hybrid federated learning architecture, called StarFL, by combining TEE, MPC, and satellites for smart urban computing.

Besides, encryption-based and noise-based solutions which describe in Section 5.1.2 are also feasible solutions to defend against poisoning attacks.

6 Security of model evaluation

Evasion attacks, or adversarial examples, aim to evade the model by adjusting samples during the inference phase, in general, these samples are carefully perturbed so that they are indistinguishable to the human eye while the network fails to identify the image contents.

Model evaluation is used to evaluate a model's performance, and feedback to determine whether to stop training or not, the attacker can deceive the federated model's evaluation output by constructing adversarial test examples. According to the attacker's capabilities, evasion attacks can be divided into the following two categories.

Gradient-based attacks. This type of approach requires the attacker access to the model's gradients in advance. Goodfellow et al.[39] proposed fast gradient sign method (FGSM), which uses the gradients of the loss with respect to the input image to create a new image that maximizes the loss. Kurakin et al.[75] improved FGSM by computing adversarial examples iteratively. Carlini and wagner.[76] proposed C&W algorithm, a novel powerful attack approach that can defeat defensive distillation.

Confidence scores. This type of approach does not require knowing the model's gradients in advance, in contrast, they use the outputted classification confidence to estimate the gradients, and then perform a similar optimization step as gradient-based attacks above.

Chen et al.[77] proposed zeroth-order optimization (ZOO) to directly estimate the gradients of the targeted model for generating adversarial examples. Ilyas et al.[78] proposed the variant of natural evolution strategies (NES) to fool the classifier under three realistic settings: the query-limited setting, the partial information setting, and the label-only setting.

In general, defense against evasion attacks is much harder, we summarize some common ideas of existing works as follows[79].

Adversarial training. An intuitive idea is to build a robust model which includes adversarial samples during the training process. For example, Moosavi-Dezfooli et al.[80] built more robust classifiers by fine-tuning the adversarial examples. Goodfellow et al.[39] built robust models by mixing the adversarial objective with the classification objective as regularizer.

However, in most cases, it is unlikely to know all possible adversarial samples in advance, adversarial training

is a remedy afterward solution.

Knowledge distillation. Papernot et al.[81] proposed defensive distillation, by using the knowledge distillation technique to train the model and hide the gradient between the logits layer and softmax outputs, so that it is impossible for the attacker to generate adversarial examples from network gradients.

Anomaly detection. Another approach is to detect abnormal examples, for example, Metzen et al.[82] detected adversarial examples by using a detector subnetwork attached to the main classification network. Grosse et al.[83] empirically validated the hypothesis that adversarial examples can be detected using statistical tests before they are fed to the machine learning model.

7 Security of model deployment

Model deployment is the last step of the lifecycle, which aims to apply the machine learning models into practice. There are three potential risks in this phase: evasion attacks, model inference attacks, and model plagiarize. Evasion attacks have been discussed in Section 6, in Sections 7 and 8, we discuss the two remaining threats.

The purpose of model inference attacks is to infer sensitive information by accessing models multiple times. Model inference attacks can be divided into the following three classes.

Label inference attack. Label inference attacks are more likely to occur in vertical federated learning scenarios. In VFL, the active party holds the data matrix and the class labels, while the passive parties keep the data matrix only. Label inference attacks happen when the passive party is controlled by the attacker, the goal is to infer the labels held by the active party.

Fu et al.[84] presented three types of label inference attacks against VFL: the direct label inference attack, the passive label inference attack, and the active label inference attack. Liu et al.[85] proposed batch label inference and replacement attacks to recover labels in the VFL setting with HE-protected communication.

Feature inference attack. Like label inference attack, feature inference attack usually occurs in the VFL setting, where features are partitioned and held by different parties, the goal is to infer the sensitive feature information held by other parties.

Luo et al.[86] proposed a feature inference attack method on model predictions in VFL, where the active party attempts to infer the feature values of new samples which belong to the passive parties

Membership inference attack. Unlike the previous two types of inference attacks, membership inference attack usually occurs in the HFL setting, where members are partitioned and held by different parties. Given a data record and the black-box model, the attackers try to determine if the record is in the model′s training dataset via model outputs.

Pustozerova and Mayer[87] discussed membership inference attacks in the setting of sequential federated learning. The promising approach is to distort the resulting model by injecting a certain amount of noise to their training data, or directly perturbating the model parameters. To achieve a similar result is to apply differential privacy on the learning output[87].

Defense against model inference attacks are also widely studied, the defensive strategies discussed in the previous phase also apply to defend against inference attacks, such as differential privacy to obfuscate the model output, and encryption-based solution for masking model structure. Besides, controlling query frequency is also a promising approach to preventing malicious queries.

The final security risk is model plagiarism. FL models can be deployed on any device, which makes them out of control and is susceptible to various kinds of attacks such as plagiarizing and misusing. As a new research direction of federated learning, it is necessary to discuss this part in-depth, we explain the detailed implementation in Section 8.

8 IP-right protection of federated learning models

While preserving the training data privacy is of paramount importance for FL, it is also a critical issue to prevent adversaries from plagiarizing, misusing, and re-distributing valuable FL models without legal permissions from legitimated owners of models[17].

8.1 Challenges

Machine learning methods that allow ownership verifications of valuable models, especially large deep neural network models, have been successfully demonstrated by either detecting feature-based signatures embedded into models[88, 89], or verifying designated labels for backdoor samples that are injected into the models during the training stage[89, 90].

These methods are adopted and extended to the federated learning setting[91, 92], in which the following challenges are properly addressed to allow each participant to verify their respective ownerships of and contributions to the global model.

First, in order for each participant to embed their own feature-based signatures, the global federated model must have sufficient capacity to embed a potentially large set of (binary) signatures without compromising original model performances. Theoretical analysis and empirical investigation in [92] demonstrated that, as long as the total bit-lengths of embedded signature do not exceed a threshold that is proportional to the deep neural network size, it is possible to embed signatures without introducing significant loss of original model performances. Also, potential conflicts between signatures embedded by differ-

ent participants should also be considered, and luckily, such conflicts lead to negligible losses in the confidence of ownership verification as demonstrated in [92].

Second, in order for each participant to embed their private signatures, the aggregation of federated models should not disclose signatures embedded into individual models. Moreover, these feature-based signatures should be verified in a private manner for instance without disclosing the feature extraction matrix, etc. It was demonstrated by [92] that private embedding and verification are achievable.

Third, for backdoor-sample-based ownership verification, one must ensure the persistence of backdoor samples when submitting the local models for aggregation. This is because plenty of aggregator-side defensive methods have been proposed with the aim to filter out backdoor samples from the global model[62, 63, 93]. Again, Li et al.[92] showed that negligible losses in the confidence of ownership verification were caused by the adoption of defensive methods. Thus, the embedded backdoor samples turn out to be very persistent.

The protection of IPR for FL models is an important step in the whole life-cycle of federated training. This step is part of an auditing process in which a variety of requirements for federated model management must be fulfilled. For instance, one may wonder whether a trained generative model has been misused to generate fake images or videos. This line of research work has been investigated in non-federated settings[94–96].

Note that model IP-right protection cannot be solved by existing blockchain-based methods. When a model is collaboratively built by multiple participants, the model has not been entered into any blockchain yet.

8.2 Protection of deep neural network ownership using digital watermarks

In the past, digital watermarks were widely utilized to safeguard the ownership of multimedia assets such as images[97, 98], videos[99, 100], audios[101–103] or functional designs[104]. However, the recent progress in deep learning has expedited various technology corporations to launch machine learning as a service (MLaaS) as one of the business models. Therefore, in order to protect and encourage creativity, it is necessary and urgent to provide IP-right-preserving.

In general, the IPR of deep models can be protected by various digital watermarking methods, which can be categorized into two schools according to respective working modes, namely, the black-box solutions using trigger sets[90, 105] and the white-box solutions relying on unique detectable features[88, 106, 107]. The main idea of watermarking is to embed identification information (i.e., a digital watermark) into the model in question without compromising model performances for the original task. For trigger-set-based methods, such watermarks are encoded by specific input-output data samples, which are referred to as the trigger set. Ownership of the model in question is verified by the repeated detection of trigger-set samples, and due to the exponentially low probability that an innocent model will exhibit the same behavior by chance. On the other hand, feature-based methods embed designated watermarks into parameters of deep neural networks (DNNs) using a carefully designed transformation matrix. In this case, the detection of designated watermarks validates ownership.

The first effort due to Uchida et al.[88] proposed to protect ownerships of DNNs in a white-box manner, by embedding designated watermarks into DNNs without compromising host network performances for the original task. Uchida et al.[88] also demonstrated that the detection of designated watermarks was robust in the face of a variety of removal attacks, including model fine-tuning and pruning. However, their method was constrained in that it required access to all of the network weights in question to extract the embedded watermarks. In order to mitigate the white-box constraint, Merrer et al.[108] proposed a trigger-set-based solution which embedded watermarks in the classification outputs of CNN models by using adversarial samples (trigger sets). This method was advantageous in that it allowed designated watermarks to be verified remotely by repeated submitting trigger set samples to a service API, thus without requiring access to the network's internal weights parameters. Later, Adi et al.[90] demonstrated that an embedded watermark as such can be treated as an intentional backdoor, and a theoretical analysis of performance under different scenarios was provided in [90]. One common theme in follow-up works such as [106, 107] have been focused on how to embed robust watermarks (or fingerprints) that are persistent to various removal attacks, including watermark overwriting, model fine-tuning and pruning of neural network models in both black box and white box settings. More recent works[89, 105] are proposed to deal with another type of attacks on watermarks, i.e., ambiguity attacks. The most unique feature of solutions illustrated in [89, 105] (also summarized in Fig. 7) lies in the fact that the inference performance of a DNN model in question will either remain intact if a valid passport is presented, or be significantly deteriorated otherwise. By taking advantage of this unique feature of the passport-based approach, ownership verification become both robust to removal attacks and resilient to ambiguity attacks. Moreover, designated binary signature can be simultaneously embedded into the scale factors of a passport layer, which provides strong guarantees and resilience to ambiguity attacks.

Aiming at the IP protection of generative adversarial networks (GANs), Ong et al[109] demonstrated a feasible solution as summarized in Fig. 8. Later, Lim et al.[110] also demonstrated IP protection for recurrent neural networks (RNNs). On both occasions, the generic watermarked framework proposed by [108] for DNNs is not

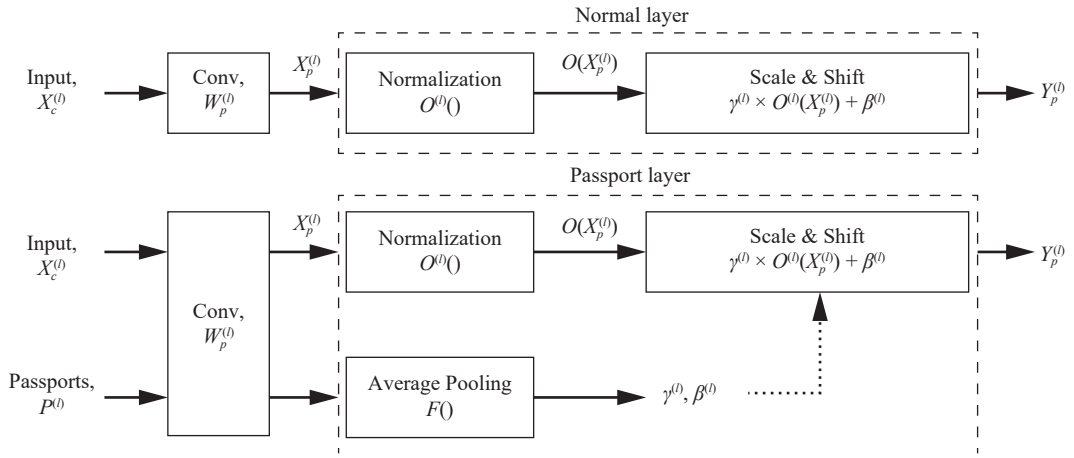


Fig. 7 Overview of the proposed passport-based protection framework (reprinted with permissions from [89])

readily applicable to GANs and recurrent neural networks (RNNs), since the input source for GANs can be either a latent vector or an image, and the output of GANs is a synthetic image rather than a classification label. While for RNNs, the input and output for RNNs are sentences.

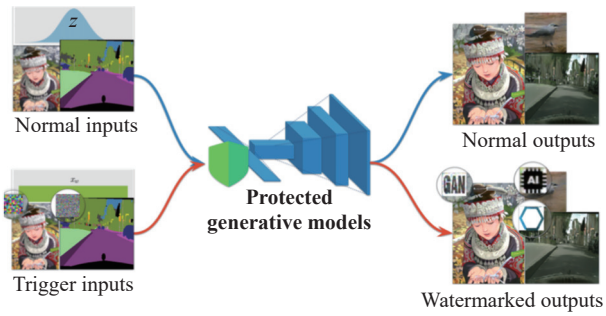


Fig. 8 Overview of the proposed GANs protection framework (reprinted with permissions from [109])

For the protection of GAN-type models, Ong et al.[109] proposed a protection framework by embedding the ownership information into the generator of GAN. In a black-box scenario, they proposed to induce the generator to output a designated watermark at an assigned location of the synthesized image, when given a trigger input (see Fig. 8). This special behavior is enforced into the model by using an appropriately designed regularization term in the training of GAN. In a white-box scenario, Ong et al.[109] proposed to use a modified sign-loss of [89] where the sign of scaling factors encodes meaningful security information, e.g., company name. The ownership verification was successfully demonstrated on three GANs variants, namely, deep convolutional generative adversarial network (DCGAN)[111], super-resolution using a generative adversarial network (SRGAN)[112] and CycleGAN[113].

For RNN models, which were designed to take images as inputs and output meaningful image captions according to image contents, Lim et al.[110] proposed a novel RNN ownership verification method whose main features are summarized as follows. First, two different embedding were adopted to embed a designated watermark (or secret key) into the RNN cell. Second, the ownership was then verified by comparing the designed image captions for a specific input image. Third, a secret key was embedded into the hidden memory of RNN such that a forged key will immediately yield an unusable image captioning model in terms of poor-quality outputs. This protection, in the same vein of passport-type of protection in [89] prevents the infringement of RNN models proactively.

8.3 Protection of deep neural network ownership under federated setting

In federated learning, there are several IPR infringement cases. Firstly, during the training stage, multiple clients have access to the global model, thus in the model-deployment stage, the trained model may be illegally redistributed to an un-authorized party outside the federated-learning system. Secondly, some free-rider parties participate in federated learning merely for stealing the federated model, they dissimulate participation to the training process but without actually contributing any data for FL, which means they infringe the intellectual property rights of benign clients. Those two IPR infringement cases highlight that there is a strong demand and motivation for federated model IP protection. In this way, several watermarking methods for FL come as a remedy for the aforementioned loopholes.

Recently, a watermarking scheme named WAFFLE[91] was proposed to protect FL model, this method assumes that the trustworthy central server is the owner of the FL model and clients have no ownership over the joint-trained federated model. WAFFLE method introduces a model re-training step at the server side, server embeds backdoor-based watermarks[90] into the aggregated model. In the ownership verification stage, the central server claim ownership through black-box access to the trained model with the prescribed watermarks.

Li et al.[92] considered the FL IPR protection problem

in a more general semi-honest FL setting, and proposed FedIPR signature/watermark embedding scheme. In FedIPR, each party is the owner of the federated learning model, during the training stage, each party keeps its own secret watermarks, and embeds secret watermarks into the local model on the client-side, and afterward, the local models are aggregated into a global model. In the verification stage, each party can verify the ownership of the global model by looking for its own watermark embedded in the global model. Note that this verification process is kept secret for each party and independent of other parties′ watermarks. In this way, unauthorized parties outside FL cannot claim ownership of the federated learning model. Moreover, free-riders who do not embed watermarks during the training cannot claim legitimate rights of the global model.

This FedIPR setting is rather challenging because the signature embedding process on the client-side must be kept secret, and the global model needs to have sufficient capacity to embed each party′s watermarks at the same time. On the technical side, Li et al.[92] propose both backdoor-based watermarks and feature-based watermarks, specifically, they propose adversarial samples as the backdoor-based watermarks to embed in the local model, and adopt a secret matrix to embed feature-based signatures into the batch-norm layers. In the verification stage, each party can verify the ownership of the global model independently. FedIPR has provided theoretical results for the capacity of client-side secret watermarks, and FedIPR is evaluated in both image classification tasks and natural language inference tasks with both convolution network and transformer-based architectures.

The engineer or researcher of a federated learning framework might benefit from the model IP protection. This is crucial as the development of the DNN costs a massive amount of money, data and computing resources. The IP protection methods will encourage the innovations of DNN model and protect the legitimate right of model owner, even in the worst scenario that the attacker can access the model without the owner′s acknowledgment. In short, the FedDNN model protection benefits the AI society especially for securing their advantage in the open market.

8.4 IPR research direction

Model intellectual property protection is an open question in secure federated learning, challenges come from the following perspectives:

Watermarking protocols. It is necessary to design secure and trustable protocols for federated learning model protection, a scheme[91] was proposed in which the model server is responsible for watermark embedding and only the server can verify ownership over the model, whereas Li et al.[92] proposed that each client can embed private watermarks and claim ownership of the model without revealing watermarking information to other parties. We believe that ownership verification protocols combined with more security mechanisms are a compelling need for trustable and secure verification.

Watermarking embedding methods. Previous federated model watermarking methods can be divided into feature-based methods[88, 90–92] and backdoor-based methods[109, 110]. The feature-based watermarks need to be extracted with white-box access to the model parameters, which is unrealistic in practice. The backdoor-based watermarking methods are highly related to backdoor learning, which is an important perspective and can motivate more related research. Especially in federated learning, it remains to be investigated how many private watermarks can be embedded into the same global model.

Watermark robustness. Another important challenge for federated model watermarking is the robustness. On one hand, various training strategies like differential privacy, homomorphic encryption and secure aggregation, etc. are adopted for data security[52, 54–55, 65, 66], those strategies modify the training process thus may remove the watermarks; on the other hand, the model adversary may apply removal attacks or model extraction attacks to remove the watermarks[114–116]. Combining those two risks, the watermark robustness is a crucial issue for federated model IP protection.

In general, model IP protection is a non-negligible issue when applying federated learning into practice. Algorithms and protocols will be the core of the research on federated model IP protection.

9 Open-source frameworks for federated learning

Developing a federated learning framework from scratch is very time-consuming, especially in industrial. An excellent FL framework can facilitate engineers and researchers to train, research and deploy the FL model in practice. In this section, we summarize some commonly used open-source frameworks in Table 2.

Besides, other famous FL frameworks include FedML[126, 127], Fedlearner[128], Flower[129], PaddleFL[130], PowerFL[131], Leaf[132], Sherpa.AI[133], PyVertical[134].

10 Conclusions

Privacy-preserving computing (PPC) is one of the active and influential research areas in both industry and academia. As the frontier research direction of PPC, FL has received considerable attention in recent years. This article gives a comprehensive survey on key components of SFL, including definition, architecture design, and threat models faced by FL. Besides, we wish that the IP protection perspective illustrated in this paper will lead to model IP protection in more FL settings. We believe that secure federated learning will bring about a new

Table 2 Commonly used open-source FL frameworks

Framework	Affiliation	Industrial/Research	Security protocols
Federated AI technology enabler (FATE)[117]	WeBank	Industrial purpose	MPC, DP, HE
Tensorflow federated (TFF)[118, 119]	Google	Research purpose	MPC, DP, HE
PySyft[120, 121]	OpenMined	Research purpose	DP, MPC
Open federated learning (OpenFL)[122, 123]	Intel	Industrial purpose	TEE
IBM federated learning[124]	IBM	Industrial purpose	DP, MPC
Clara[125]	Nvidia	Industrial purpose	TEE

mindset and toolbox in developing large-scale AI systems, and help to address open problems that hinder wide applications of SFL in a larger variety of use cases, such as secure and legal data exchanges, data shortages and data silos in practice.

Acknowledgements

The work was supported by National Key Research and Development Program of China (No. 2018AAA 0101100).

Open Access

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, ACM, Lake Tahoe, USA. pp. 1097–1105, 2012. DOI: 10.5555/2999134.2999257.

[2] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[3] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, Minneapolis, USA, pp. 4171–4186, 2019. DOI: 10.18653/v1/N19-1423.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ACM, Vancouver, Canada, Article number 159, 2020. DOI: 10.5555/3495724.3495883.

[5] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. C. Hong, V. Jain, X. B. Liu, H. Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, ACM, Boston, USA, pp. 7–10, 2016. DOI: 10.1145/2988450.2988454.

[6] H. F. Guo, R. M. Tang, Y. M. Ye, Z. G. Li, X. Q. He. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ACM, Melbourne, Australia, pp. 1725–1731, 2017. DOI: 10.5555/3172077.3172127.

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, F. F. Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Miami, USA, pp. 248–255, 2009. DOI: 10.1109/CVPR.2009.5206848.

[8] Protein Data Bank. A structural view of biology, [Online], Available: https://www.rcsb.org/.

[9] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, vol. 596, no. 7873, pp. 583–589, 2021. DOI: 10.1038/s41586-021-03819-2.

[10] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. L. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S.

Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, vol. 577, no. 7792, pp. 706–710, 2020. DOI: 10.1038/s41586-019-1923-7.

[11] EU. General data protection regulation, [Online], Available: https://gdpr-info.eu/.

[12] DLA Piper. Data protection laws of the world: Full handbook, [Online], Available: https://www.dlapiperdataprotection.com/.

[13] The National People's Congress. China data security law, [Online], Available: http://www.npc.gov.cn/npc/c30834/202106/7c9af12f51334a73b56d7938f99a788a.shtml. (in Chinese)

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, pp. 1273–1282, 2017.

[15] L. G. Zhu, Z. J. Liu, S. Han. Deep leakage from gradients. In *Proceedings of the Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 14774–14784, 2019.

[16] L. T. Phong, Y. Aono, T. Hayashi, L. H. Wang, S. Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018. DOI: 10.1109/TIFS.2017.2787987.

[17] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. El Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. Y. He, L. He, Z. Y. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. K. Song, S. U. Stich, Z. T. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Y. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, S. Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021. DOI: 10.1561/2200000083.

[18] Y. Z. Ma, X. J. Zhu, J. Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, ACM, Macao, China, pp. 4732–4738, 2019. DOI: 10.5555/3367471.3367701.

[19] Z. B. Ying, Y. Zhang, X. M. Liu. Privacy-preserving in defending against membership inference attacks. In *Proceedings of the Workshop on Privacy-preserving Machine Learning in Practice*, ACM, pp. 61–63, 2020. DOI: 10.1145/3411501.3419428.

[20] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. J. Chen, H. Yu. *Federated Learning*, San Francisco Bay Area, USA: Morgan & Claypool Publishers, pp. 207, 2019.

[21] Q. Yang, Y. Liu, T. J. Chen, Y. X. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, Article number 12, 2019. DOI: 10.1145/3298981.

[22] T. Li, A. K. Sahu, A. Talwalkar, V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020. DOI: 10.1109/MSP.2020.2975749.

[23] L. J. Lyu, H. Yu, Q. Yang. Threats to federated learning: A survey. [Online], Available: https://arxiv.org/abs/2003.02133, 2020.

[24] N. Bouacida, P. Mohapatra. Vulnerabilities in federated learning. *IEEE Access*, vol. 9, pp. 63229–63249, 2021. DOI: 10.1109/ACCESS.2021.3075203.

[25] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021. DOI: 10.1016/j.future.2020.10.007.

[26] P. R. Liu, X. R. Xu, W. Wang. Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives. *Cybersecurity*, vol. 5, no. 1, Article number 4, 2022. DOI: 10.1186/s42400-021-00105-6.

[27] X. J. Zhang, H. L. Gu, L. X. Fan, K. Chen, Q. Yang. No free lunch theorem for security and utility in federated learning. [Online], Available: https://arxiv.org/abs/2203.05816, 2022.

[28] O. Goldreich, S. Micali, A. Wigderson. How to play ANY mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, ACM, New York, USA, pp. 218–229, 1987. DOI: 10.1145/28395.28420.

[29] T. Rabin, M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, ACM, Seattle, USA, pp. 73–85, 1989. DOI: 10.1145/73007.73014.

[30] C. Dwork. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, Springer, Xi'an, China, pp. 1–19, 2008. DOI: 10.1007/978-3-540-79228-4_1.

[31] C. Dwork, A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014. DOI: 10.1561/0400000042.

[32] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the International Conference on Advances in Cryptology*, Springer, Prague, Czech Republic, pp. 223–238, 1999. DOI: 10.1007/3-540-48910-X_16.

[33] OMTP. 2009. Advanced trusted environment: OMTP TR1. http://www.omtp.org/OMTP_Advanced_Trusted_Environment_OMTP_TR1_v1_1.pdf

[34] ARM. ARM TrustZone Technology, [Online], Available: https://developer.arm.com/documentation/100690/0200/ARM-TrustZone-technology?lang=en.

[35] M. Sabt, M. Achemlal, A. Bouabdallah. Trusted execution environment: What it is, and what it is not. In *Proceedings of IEEE Trustcom/BigDataSE/ISPA*, IEEE, Helsinki, Finland, pp. 57–64, 2015. DOI: 10.1109/Trustcom.2015.357.

[36] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli. Evasion attacks against machine learning at test time. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Prague, Czech

Republic, pp. 387–402, 2013. DOI: 10.1007/978-3-642-40994-3_25.

[37] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, 2014.

[38] A. Nguyen, J. Yosinski, J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Boston, USA, pp. 427–436, 2015. DOI: 10.1109/CVPR.2015.7298640.

[39] I. J. Goodfellow, J. Shlens, C. Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2015.

[40] E. Bagdasaryan, A. Veit, Y. Q. Hua, D. Estrin, V. Shmatikov. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, Palermo, Italy, pp. 2938–2948, 2020.

[41] H. J. Zhang, Z. J. Xie, R. Zarei, T. Wu, K. W. Chen. Adaptive client selection in resource constrained federated learning systems: A deep reinforcement learning approach. *IEEE Access*, vol. 9, pp. 98423–98432, 2021. DOI: 10.1109/ACCESS.2021.3095915.

[42] R. Albelaihi, X. Sun, W. D. Craft, L. K. Yu, C. G. Wang. Adaptive participant selection in heterogeneous federated learning. In *Proceedings of IEEE Global Communications Conference*, IEEE, Madrid, Spain, 2021. DOI: 10.1109/GLOBECOM46510.2021.9685077.

[43] F. Mo, A. S. Shamsabadi, K. Katevas, S. Demetriou, I. Leontiadis, A. Cavallaro, H. Haddadi. DarkneTZ: Towards model privacy at the edge using trusted execution environments. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ACM, Toronto, Canada, pp. 161–174, 2020. DOI: 10.1145/3386901.3388946.

[44] A. B. Huang, Y. Liu, T. J. Chen, Y. K. Zhou, Q. Sun, H. F. Chai, Q. Yang. StarFL: Hybrid federated learning architecture for smart urban computing. *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 4, Article number 43, 2021. DOI: 10.1145/3467956.

[45] B. Hitaj, G. Ateniese, F. Perez-Cruz. Deep models under the GAN: Information leakage from collaborative deep learning. In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, ACM, Dallas, USA, pp. 603–618, 2017. DOI: 10.1145/3133956.3134012.

[46] B. Zhao, K. R. Mopuri, H. Bilen. iDLG: Improved deep leakage from gradients. [Online], Available: https://arxiv.org/abs/2001.02610, 2020.

[47] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller. Inverting gradients-how easy is it to break privacy in federated learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ACM, Vancouver, Canada, Article number 33, 2020. DOI: 10.5555/3495724.3497145.

[48] Y. J. Wang, J. R. Deng, D. Guo, C. H. Wang, X. R. Meng, H. Liu, C. W. Ding, S. Rajasekaran. SAPAG: A self-adaptive privacy attack from gradients. [Online], Available: https://arxiv.org/abs/2009.06228, 2020.

[49] J. Y. Zhu, M. B. Blaschko. R-GAP: Recursive gradient attack on privacy. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[50] X. Jin, P. Y. Chen, C. Y. Hsu, C. M. Yu, T. Y. Chen. Catastrophic data leakage in vertical federated learning. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, pp. 994–1006, 2021.

[51] Z. H. Li, J. X. Zhang, L. Y. Liu, J. Liu. Auditing privacy defenses in federated learning via generative gradient leakage. [Online], Available: https://arxiv.org/abs/2203.15696, 2022.

[52] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, B. Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. [Online], Available: https://arxiv.org/abs/1711.10677, 2017.

[53] C. L. Zhang, S. Y. Li, J. Z. Xia, W. Wang, F. Yan, Y. Liu. BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of USENIX Conference on USENIX Annual Technical Conference*, Berkeley, USA, Article number. 33, 2020. DOI: 10.5555/3489146.3489179.

[54] A. Huang, Y. Y. Chen, Y. Liu, T. J. Chen, Q. Yang. RPN: A residual pooling network for efficient federated learning. In *Proceedings of the 24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain, pp. 1223–1229, 2020.

[55] H. B. McMahan, D. Ramage, K. Talwar, L. Zhang. Learning differentially private recurrent language models. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.

[56] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, H. V. Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020. DOI: 10.1109/TIFS.2020.2988575.

[57] C. L. Xie, K. L. Huang, P. Y. Chen, B. Li. DBA: Distributed backdoor attacks against federated learning. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[58] A. B. Huang. Dynamic backdoor attacks against federated learning. [Online], Available: https://arxiv.org/abs/2011.07429, 2020.

[59] J. Feng, Q. Z. Cai, Z. H. Zhou. Learning to confuse: Generating training time adversarial data with auto-encoder. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, ACM, Vancouver, Canada, Article number 32, 2019. DOI: 10.5555/3454287.3455361.

[60] S. S. Hu, J. R. Lu, W. Wan, L. Y. Zhang. Challenges and approaches for mitigating byzantine attacks in federated learning. [Online], Available: https://arxiv.org/abs/2112.14468, 2021.

[61] M. H. Fang, X. Y. Cao, J. Y. Jia, N. Z. Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, ACM, Berkeley, USA, Article number 92, 2020. DOI: 10.5555/3489212.3489304.

[62] D. Yin, Y. D. Chen, R. Kannan, P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Confer-*

ence on Machine Learning, Stockholm, Sweden, pp. 5650–5659, 2018.

[63] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ACM, Long Beach, USA, pp. 118–128, 2017. DOI: 10.5555/3294771.3294783.

[64] C. Xie, S. Koyejo, I. Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 6893–6901, 2019.

[65] Y. Dong, X. J. Chen, L. Y. Shen, D. K. Wang. Privacy-preserving distributed machine learning based on secret sharing. In *Proceedings of the 21st International Conference on Information and Communications Security*, Springer, Beijing, China, pp. 684–702, 2019. DOI: 10.1007/978-3-030-41579-2_40.

[66] R. Kanagavelu, Z. X. Li, J. Samsudin, Y. C. Yang, F. Yang, R. S. M. Goh, M. Cheah, P. Wiwatphonthana, K. Akkarajitsakul, S. G. Wang. Two-phase multi-party computation enabled privacy-preserving federated learning. In *Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*, IEEE, Melbourne, Australia, pp. 410–419, 2020. DOI: 10.1109/CCGrid49817.2020.00-52.

[67] M. O. Rabin. How to exchange secrets with oblivious transfer, Technical Report Paper 2005/187, 2005.

[68] A. C. C. Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, IEEE, Toronto, Canada, pp. 162–167, 1986. DOI: 10.1109/SFCS.1986.25.

[69] Intel®. Architecture instruction set extensions programming reference, Technical Report 319433-012, Intel Corporation, USA, 2012.

[70] V. Costan, S. Devadas. Intel SGX explained, Technical Report Paper 2016/086, 2016.

[71] ArmDeveloper. Arm TrustZone Technology, [Online], Available: https://developer.arm.com/documentation/100690/0200/ARM-TrustZone-technology?lang=en, December 05, 2019.

[72] Androidtrusty. Android Trusty TEE, [Online], Available: https://source.android.com/security/trusty, 2019.

[73] AMD. AMD Secure Encrypted Virtualization, [Online], Available: https://developer.amd.com/sev/.

[74] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, N. Kourtellis. PPFL: Privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, pp. 94–108, 2021. DOI: 10.1145/3458864.3466628.

[75] A. Kurakin, I. J. Goodfellow, S. Bengio. Adversarial examples in the physical world. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[76] N. Carlini, D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ACM, Dallas, USA, pp. 3–14, 2017. DOI: 10.1145/3128572.3140444.

[77] P. Y. Chen, H. Zhang, Y. Sharma, J. F. Yi, C. J. Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ACM, Dallas, USA, pp. 15–26, 2017. DOI: 10.1145/3128572.3140448.

[78] A. Ilyas, L. Engstrom, A. Athalye, J. Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 2137–2146, 2018.

[79] D. Y. Meng, H. Chen. MagNet: A two-pronged defense against adversarial examples. In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, ACM, Dallas, USA, pp. 135–147, 2017. DOI: 10.1145/3133956.3134057.

[80] S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard. DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 2574–2582, 2016. DOI: 10.1109/CVPR.2016.282.

[81] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of IEEE Symposium on Security and Privacy*, IEEE, San Jose, USA, pp. 582–597, 2016. DOI: 10.1109/SP.2016.41.

[82] J. H. Metzen, T. Genewein, V. Fischer, B. Bischoff. On detecting adversarial perturbations. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[83] K. Grosse, P. Manoharan, N. Papernot, M. Backes, P. McDaniel. On the (statistical) detection of adversarial examples. [Online], Available: https://arxiv.org/abs/1702.06280, 2017.

[84] C. Fu, X. H. Zhang, S. L. Ji, J. Y. Chen, J. Z. Wu, S. Q. Guo, J. Zhou, A. X. Liu, T. Wang. Label inference attacks against vertical federated learning. In *Proceedings of the 31st USENIX Security Symposium*, USENIX Association, Boston, USA, 2022.

[85] Y. Liu, Z. H. Yi, T. J. Chen. Backdoor attacks and defenses in feature-partitioned collaborative learning. [Online], Available: https://arxiv.org/abs/2007.03608, 2020.

[86] X. J. Luo, Y. C. Wu, X. K. Xiao, B. C. Ooi. Feature inference attack on model predictions in vertical federated learning. In *Proceedings of the 37th IEEE International Conference on Data Engineering*, IEEE, Chania, Greece, pp. 181–192, 2021. DOI: 10.1109/ICDE51399.2021.00023.

[87] A. Pustozerova, R. Mayer. Information leaks in federated learning. In *Proceedings of the Workshop on Decentralized IoT Systems and Security*, DISS, San Diego, USA, 2020. DOI: 10.14722/diss.2020.23004.

[88] Y. Uchida, Y. Nagai, S. Sakazawa, S. Satoh. Embedding watermarks into deep neural networks. In *Proceedings of ACM International Conference on Multimedia Retrieval*, ACM, Bucharest, Romania, pp. 269–277, 2017. DOI: 10.1145/3078971.3078974.

[89] L. X. Fan, K. W. Ng, C. S. Chan, Q. Yang, DeepIP: Deep neural network intellectual property protection with passports. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to be published. DOI: 10.1109/TPAMI.2021.3088846.

[90] Y. Adi, C. Baum, M. Cisse, B. Pinkas, J. Keshet. Turning your weakness into a strength: Watermarking deep

neural networks by backdooring. In *Proceedings of the 27th USENIX Conference on Security Symposium*, ACM, Baltimore, USA, pp. 1615–1631, 2018. DOI: 10.5555/3277203.3277324.

[91] B. G. A. Tekgul, Y. X. Xia, S. Marchal, N. Asokan. WAFFLE: Watermarking in federated learning. In *Proceedings of the 40th International Symposium on Reliable Distributed Systems*, IEEE, Chicago, USA, pp. 310–320, 2021. DOI: 10.1109/SRDS53918.2021.00038.

[92] B. W. Li, L. X. Fan, H. L. Gu, J. Li, Q. Yang. FedIPR: Ownership verification for federated deep neural network models. [Online], Available: https://arxiv.org/abs/2109.13236, 2022.

[93] E. M. El Mhamdi, R. Guerraoui, S. Rouault. The hidden vulnerability of distributed learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 3521–3530, 2018.

[94] Y. He, N. Yu, M. Keuper, M. Fritz. Beyond the spectrum: Detecting Deepfakes via re-synthesis. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, Beijing, China, pp. 2534–2541, 2021. DOI: 10.24963/ijcai.2021/349.

[95] L. Chai, D. Bau, S. N. Lim, P. Isola. What makes fake images detectable? Understanding properties that generalize. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 103–120, 2020. DOI: 10.1007/978-3-030-58574-7_7.

[96] Z. Z. Liu, X. J. Qi, P. H. S. Torr. Global texture enhancement for fake face detection in the wild. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 8057–8066, 2020. DOI: 10.1109/CVPR42600.2020.00808.

[97] E. Nezhadarya, Z. J. Wang, R. K. Ward. Robust image watermarking based on multiscale gradient direction quantization. *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1200–1213, 2011. DOI: 10.1109/TIFS.2011.2163627.

[98] H. Fang, W. M. Zhang, H. Zhou, H. Cui, N. H. Yu. Screen-shooting resilient watermarking. *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1403–1418, 2019. DOI: 10.1109/TIFS.2018.2878541.

[99] H. Mareen, J. De Praeter, G. Van Wallendael, P. Lambert. A scalable architecture for uncompressed-domain watermarked videos. *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1432–1444, 2019. DOI: 10.1109/TIFS.2018.2879301.

[100] M. Asikuzzaman, M. R. Pickering. An overview of digital video watermarking. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2131–2153, 2018. DOI: 10.1109/TCSVT.2017.2712162.

[101] M. J. Hwang, J. Lee, M. Lee, H. G. Kang. SVD-based adaptive QIM watermarking on stereo audio signals. *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 45–54, 2018. DOI: 10.1109/TMM.2017.2721642.

[102] Y. Erfani, R. Pichevar, J. Rouat. Audio watermarking using spikegram and a two-dictionary approach. *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 840–852, 2017. DOI: 10.1109/TIFS.2016.2636094.

[103] A. Nadeau, G. Sharma. An audio watermark designed for efficient and robust resynchronization after Analog playback. *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1393–1405, 2017. DOI: 10.1109/TIFS.2017.2661724.

[104] Z. X. Lin, F. Peng, M. Long. A low-distortion reversible watermarking for 2D engineering graphics based on region nesting. *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2372–2382, 2018. DOI: 10.1109/TIFS.2018.2819122.

[105] J. Zhang, D. D. Chen, J. Liao, W. M. Zhang, G. Hua, N. H. Yu. Passport-aware normalization for deep model protection. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ACM, Vancouver, Canada, Article number 1896, 2020. DOI: 10.5555/3495724.3497620.

[106] H. Chen, B. D. Rohani, F. Koushanfar. DeepMarks: A digital fingerprinting framework for deep neural networks. [Online], Available: https://arxiv.org/abs/1804.03648, 2018.

[107] B. D. Rohani, H. L. Chen, F. Koushanfar. DeepSigns: A generic watermarking framework for IP protection of deep learning models. [Online], Available: https://arxiv.org/abs/1804.00750, 2018.

[108] E. Le Merrer, P. Pérez, G. Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2020. DOI: 10.1007/s00521-019-04434-z.

[109] D. S. Ong, C. S. Chan, K. W. Ng, L. X. Fan, Q. Yang. Protecting intellectual property of generative adversarial networks from ambiguity attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 3629–3638, 2021. DOI: 10.1109/CVPR46437.2021.00363.

[110] J. H. Lim, C. S. Chan, K. W. Ng, L. X. Fan, Q. Yang. Protect, show, attend and tell: Empowering image captioning models with ownership protection. *Pattern Recognition*, vol. 122, pp. 108285. DOI: 10.1016/j.patcog.2021.108285.

[111] A. Radford, L. Metz, S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 4th International Conference on Learning Representations*, San Juan, Puerto Rico, 2016.

[112] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. H. Wang, W. Z. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 105–114. DOI: 10.1109/CVPR.2017.19.

[113] J. Y. Zhu, T. Park, P. Isola, A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, pp. 2242–2251, 2017. DOI: 10.1109/ICCV.2017.244.

[114] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, T. Ristenpart. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium*, ACM, Austin, USA, pp. 601–618, 2016. DOI: 10.5555/3241094.3241142.

[115] T. Orekondy, B. Schiele, M. Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4949–

4958, 2019. DOI: 10.1109/CVPR.2019.00509.

[116] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami. Practical black-box attacks against machine learning. In *Proceedings of ACM on Asia Conference on Computer and Communications Security*, ACM, Abu Dhabi, UAE, pp. 506–519, 2017. DOI: 10.1145/3052973.3053009.

[117] WeBank AI Department (2020-03-07). Federated AI Technology Enabler (FATE), 2020-03-07. [Online], Available: https://github.com/FederatedAI/FATE.

[118] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, J. Roselander. Towards federated learning at scale: System design. In *Proceedings of the 2nd SysML Conference*, Stanford, USA, 2019.

[119] Google. Tensorflow Federated (TFF), [Online], Available: https://tensorflow.google.cn/federated.

[120] OpenMined. PySyft, [Online], Available: https://github.com/OpenMined.

[121] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, J. Passerat-Palmbach. A generic framework for privacy preserving deep learning. [Online], Available: https://arxiv.org/abs/1811.04017, 2018.

[122] G. A. Reina, A. Gruzdev, P. Foley, O. Perepelkina, M. Sharma, I. Davidyuk, I. Trushkin, M. Radionov, A. Mokrov, D. Agapov, J. Martin, B. Edwards, M. J. Sheller, S. Pati, P. N. Moorthy, S. H. Wang, P. Shah, S. Bakas. OpenFL: An open-source framework for federated learning. [Online], Available: https://arxiv.org/abs/2105.06413, 2021.

[123] Intel. OpenFL - An open-source framework for federated learning, [Online], Available: https://github.com/intel/openfl.

[124] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn, M. Purcell, A. Rawat, T. Minh, N. Holohan, S. Chakraborty, S. Whitherspoon, D. Steuer, L. Wynter, H. Hassan, S. Laguna, M. Yurochkin, M. Agarwal, E. Chuba, A. Abay. IBM federated learning: An enterprise framework white paper V0.1. [Online], Available: https://arxiv.org/abs/2007.10987, 2020.

[125] Nvidia. Nvidia Clara, [Online], Available: https://developer.nvidia.com/clara.

[126] C. Y. He, S. Z. Li, J. So, X. Zeng, M. Zhang, H. Y. Wang, X. Y. Wang, P. Vepakomma, A. Singh, H. Qiu, X. H. Zhu, J. Z. Wang, L. Shen, P. L. Zhao, Y. Kang, Y. Liu, R. Raskar, Q. Yang, M. Annavaram, S. Avestimehr. FedML: A research library and benchmark for federated machine learning. [Online], Available: https://arxiv.org/abs/2007.13518, 2020.

[127] FedML-AI. FedML, [Online], Available: https://github.com/FedML-AI/FedML.

[128] Bytedance. Fedlearner, [Online], Available: https://github.com/bytedance/fedlearner.

[129] D. J. Beutel, T. Topal, A. Mathur, X. C. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, N. D. Lane. Flower: A friendly federated learning research framework. [Online], Available: https://arxiv.org/abs/2007.14390, 2020.

[130] PaddlePaddle. PaddleFL, [Online], Available: https://github.com/PaddlePaddle/PaddleFL.

[131] Tencent. Angel PowerFL, [Online], Available: https://cloud.tencent.com/solution/powerfl.

[132] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, A. Talwalkar. LEAF: A benchmark for federated settings. [Online], Available: https://arxiv.org/abs/1812.01097, 2018.

[133] Sherpa.ai. Sherpa.ai, [Online], Available: https://sherpa.ai/.

[134] D. Romanini, A. J. Hall, P. Papadopoulos, T. Titcombe, A. Ismail, T. Cebere, R. Sandmann, R. Roehm, M. A. Hoeh. PyVertical: A vertical federated learning framework for multi-headed splitNN. [Online], Available: https://arxiv.org/abs/2104.00489, 2021.

Qiang Yang is a Fellow of Canadian Academy of Engineering (CAE) and Royal Society of Canada (RSC), Chief Artificial Intelligence Officer of WeBank and Chair Professor of CSE Department, Hong Kong University of Science and Technology, China. He is the Conference Chair of AAAI-21, President of Hong Kong Society of Artificial Intelligence and Robotics (HKSAIR), the President of Investment Technology League (ITL) and Open Islands Privacy-Computing Open-source Community, and former President of IJCAI (2017–2019). He is a fellow of AAAI, ACM, IEEE and AAAS. He is the founding EiC of two journals: *IEEE Transactions on Big Data* and *ACM Transactions on Intelligent Systems and Technology*. His latest books are *Transfer Learning*, *Federated Learning*, *Privacy-preserving Computing* and *Practicing Federated Learning*.

His research interests include transfer learning and federated learning.

E-mail: qyang@cse.ust.hk (Corresponding author)
ORCID iD: 0000-0001-5059-8360

Anbu Huang is currently a senior research scientist at WeBank, his research papers have been published in leading journals and conferences, such as AAAI and ACM TIST. He served as a peer reviewer in ACM TIST, IEEE TMI, IJCAI, and other top artificial intelligence journals and conferences. Previously, He was a technical team leader at Tencent (2014–2018), and a senior engineer at MicroStrategy (2012–2014). His latest books are *Practicing Federated Learning* (2021) and *Dive into Deep Learning* (2017).

His research interests include deep learning, machine learning, and federated learning.

E-mail: stevenhuang@webank.com
ORCID iD: 0000-0003-3444-7348

Lixin Fan is the Chief Scientist of Artificial Intelligence at WeBank, China. He is the author of more than 70 international journals and conference articles. He has worked at Nokia Research Center and Xerox Research Center Europe. He has participated in NIPS/NeurIPS, ICML, CVPR, ICCV, ECCV, IJCAI and other top artificial intelligence conferences for a long time, served as area chair of ICPR, and organized workshops in

various technical fields. He is also the inventor of more than one hundred patents filed in USA , Europe and China, and the chairman of the IEEE P2894 Explainable Artificial Intelligence (XAI) Standard Working Group.

His research interests include machine learning and deep learning, computer vision and pattern recognition, image and video processing, 3D big data processing, data visualization and rendering, augmented and virtual reality, mobile computing and ubiquitous computing, and intelligent man-machine interface.

E-mail: lixinfan@webank.com

Chee Seng Chan is currently a full Professor with Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. Dr. Chan was the Founding Chair for the IEEE Computational Intelligence Society Malaysia Chapter, the Organising Chair for the Asian Conference on Pattern Recognition (2015), the General Chair for the IEEE Workshop on Multimedia Signal Processing (2019), and IEEE Visual Communications and Image Processing (2013). He is a Chartered Engineer registered under the Engineering Council, UK. He was the recipient of several notable awards, such as Young Scientist Network-Academy of Sciences Malaysia in 2015 and the Hitachi Research Fellowship in 2013.

His research interests include computer vision and machine learning with focus on scene understanding. He is also interested in the interplay between language and vision: generating sentential descriptions about complex scenes.

E-mail: cs.chan@um.edu.cn

Jian Han Lim is currently a Ph. D. degree candidate in artificial intelligence with Universiti Malaya, Malaysia.

His research interests include computer vision and deep learning with a focus on image captioning.

E-mail: jianhanl98@gmail.com

Kam Woh Ng received the B. Sc. degree from Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, in 2019. He is currently a Ph. D. degree candidate at University of Surrey, UK under the supervision of Prof. Tao Xiang and Prof. Yi-Zhe Song. Prior to joining the University of Surrey, he was an AI researcher from WeBank, China and a lab member of Center of Image and Signal Processing (CISIP) in University of Malaya, Malaysia.

His research interests include deep learning, computer vision, representation learning and their applications.

E-mail: kamwoh.ng@surrey.ac.uk
ORCID iD: 0000-0002-9309-563X

Ding Sheng Ong received the B. Sc. degree from Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, in 2020. He currently a Ph. D. degree candidate at is Aberystwyth University, UK. Prior to joining the Aberystwyth University, he was a lab member of Center of Image and Signal Processing (CISiP) in Universiti Malaya, Malaysia.

His research interests include deep learning and computer vision.

E-mail: sheng970303@gmail.com

Bowen Li received the B. Sc. degree in automation from Xi′an Jiaotong University, China in 2019. He is currently a Ph. D. degree candidate at Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He worked as a research intern at WeBank AI Group, China in 2021.

His research interests include federated learning, data privacy, and machine learning security.

E-mail: li-bowen@sjtu.edu.cn
ORCID iD: 0000-0003-1602-3541