Information Fusion 123 (2025) 103248

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Backdoor attacks and defense mechanisms in federated learning: A survey

Zhaozheng Li [a,1], Jiahe Lan [a,1], Zheng Yan [a,b,*], Erol Gelenbe [c,d]

[a] State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China
[b] Hangzhou Institute of Technology, Xidian University, China
[c] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, 44100 Gliwice, Poland
[d] CNRS I3S, Université Côte d'Azur, 06100 Nice, France

ARTICLE INFO

Keywords:
Federated learning
Backdoor attacks
Defense mechanisms
AI trust

ABSTRACT

Federated Learning (FL) is a distributed machine learning framework that enables the collaborative training of machine learning models by multiple entities. However, FL is vulnerable to various potential risks, especially backdoor attacks. A backdoor attack aims to implant hidden backdoors into a global model by compromising one or more clients and making them provide poisoned model updates. Consequently, the global model misclassifies inputs with triggers as adversary-desired classes/labels while performing well on benign inputs. Despite its severity, existing literature lacks a comprehensive review on backdoor attacks and their defense mechanisms of FL, especially for vertical FL. This paper comprehensively reviews and evaluates recent advances in backdoor attacks and defense mechanisms on FL. We first introduce foundational concepts about FL, backdoor attacks, and defense mechanisms, along with their respective security models. Then, we propose two sets of evaluation criteria that a sound backdoor attack and a defense mechanism should meet, respectively. After that, we provide taxonomies of existing backdoor attacks and defense mechanisms of FL and review them by employing the proposed criteria to evaluate their pros and cons. We also explore a positive application of backdoors in FL, i.e., backdoor-based watermarking. Finally, we discuss a number of open issues and suggest promising future research directions.

1. Introduction

Advances in machine learning (ML) have enabled machines to achieve high levels of performance in various tasks, such as image recognition [1–3], natural language processing [4–6], and time series forecasting [7–9], based on substantial amounts of high-quality data. Traditional centralized ML requires collecting vast amounts of data from entities such as user devices, companies, and institutions, to train the ML models, while concerns over data privacy impede the sharing of the training data among different entities. This has motivated the development of Federated Learning (FL) [10]. FL is a distributed ML framework that allows clients to collaborate on model training by providing model updates to a central server instead of sharing raw data, thereby preserving data privacy during collaborative training. In recent years, the numerous advantages of FL have driven its widespread adoption across various domains, including industrial engineering [11,12], healthcare [13–16] and wireless communications [17–19].

Nevertheless, the distributed nature of FL introduces new attack surfaces due to untrusted or malicious clients. A typical threat is an FL Backdoor Attack (FLBA) that introduces hidden backdoors into the global model by compromising one or more training clients to provide poisoned model updates. The poisoned global model misclassifies malicious inputs as specific classes desired by an adversary, while performing correctly on benign inputs. An example of an FLBA targeting traffic sign recognition is presented in Section 2.2. FLBAs are characterized by their stealthiness and harmfulness. On the one hand, the invisible nature of a malicious client's local training process to other participants makes such attacks difficult to detect. On the other hand, the compromised model performs well on normal tasks, but once the backdoor is triggered, it can produce critical errors. In particular, FLBAs deployed in safety-critical applications, such as finance and healthcare, could result in significant societal harm. To protect FL from such attacks, a variety of FL Backdoor Defenses (FLBDs) have been proposed to detect or mitigate FLBAs. Fig. 1 presents the development timeline of FLBAs, FLBDs, and backdoor-based watermarking methods in FL. Backdoor attacks were first introduced into FL in 2018, soon followed by the emergence of defense mechanisms and backdoor-based watermarking methods against FL. Since then, these topics have attracted considerable attention, with a surge of research emerging, especially in the past three years.

* Correspondence to: 266 Xinglong Section of Xifeng Road, Xi'an, Shaanxi 710126, China.
E-mail addresses: zhzhli@stu.xidian.edu.cn (Z. Li), jhlan16@stu.xidian.edu.cn (J. Lan), zyan@xidian.edu.cn (Z. Yan), seg@iitis.pl (E. Gelenbe).
[1] The first two authors contribute equally to this work.

https://doi.org/10.1016/j.inffus.2025.103248
Received 9 February 2025; Received in revised form 10 April 2025; Accepted 19 April 2025
Available online 3 May 2025

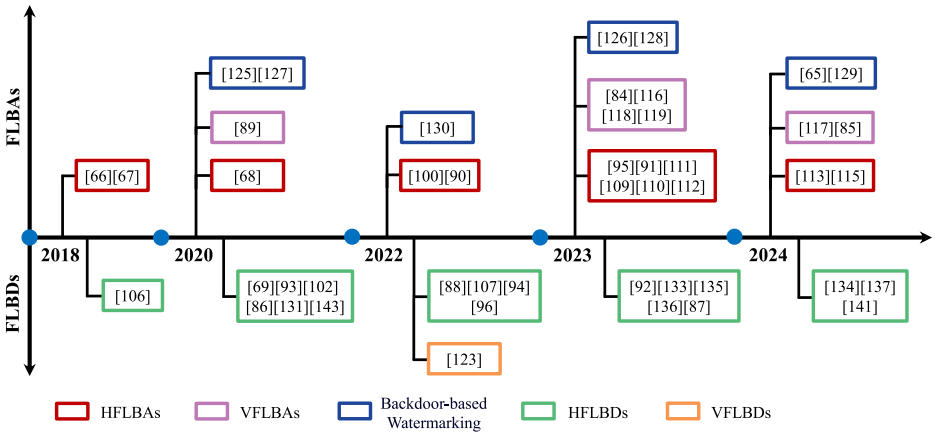

Fig. 1. A Timeline of FLBAs and FLBDs.

Table 1
Comparison of our survey with other existing surveys.

Ref	Year	Security Model	Criteria	Taxonomy	FLBAs Review			FLBDs Review	
					HFL-BAs	VFL-BAs	Positive Applications	HFL-BDs	VFL-BDs
[20]	2022	◐	○	○	●	○	○	●	○
[21]	2022	○	○	◐	●	○	○	●	○
[22]	2023	◐	○	◐	●	○	○	●	○
[23]	2024	◐	◐	●	○	○	○	○	○
[27]	2024	◐	◐	●	●	○	○	●	○
[25]	2023	◐	○	●	●	○	○	●	○
[24]	2023	○	●	●	●	○	○	●	○
[26]	2024	◐	●	●	●	◐	○	●	○
Ours	2025	●	●	●	●	●	●	●	●

●: Fully supported; ◐: Partially supported; ○: Not supported; FLBAs: Backdoor Attacks against FL; HFLBAs: Backdoor Attacks against Horizontal FL; VFLBAs: Backdoor Attacks against Vertical FL; FLBDs: Backdoor Defenses against FL; HFLBD: Backdoor Defenses against Horizontal FL; VFLBD: Backdoor Defenses against Vertical FL; : Studies focusing on adversarial attacks and defenses in FL. : Studies focusing on backdoor attacks and defenses in Wireless FL. : Studies focusing on backdoor attacks and defenses in FL.

There are several surveys [20–27] of the literature about FLBAs and FLBDs. In [20–23] extensive surveys on adversarial attacks and defense mechanisms against FL can be found, including a discussion of FLBAs and FLBDs. However, these studies just provide a general discussion of adversarial attacks against FL and defenses, such as adversarial examples, poisoning attacks, and FLBAs without focusing on each specific type of attack. On the other hand, Wan et al. [27] focus narrowly on backdoor attacks and defense mechanisms within wireless FL, overlooking backdoor attacks and defense mechanisms for Vertical FL. Additionally, they do not define a defense model and discuss the various knowledge required for defense. Other surveys [24–26] are highly relevant to our paper, but the defense model of FLBDs is not studied. Specifically, the literatures [25,26] lack a detailed description of the defense model and the literature [24] does not introduce the threat model of FLBAs and the defense model of FLBDs. Moreover, they conduct a comprehensive review on Backdoor Attacks against Horizontal Federated Learning (HFLBAs) and Backdoor Defenses against Horizontal Federated Learning (HFLBDs), but overlook relevant research on Vertical Federated Learning (VFL). Among them, literature [26] only addresses several backdoor attacks against one specific VFL architecture, without exploring other architectures and defense mechanisms in VFL. In particular, they focus solely on the malicious applications of backdoors in FL, without discussing the potential positive applications of backdoors. Table 1 presents a detailed comparison between our survey and related surveys.

In this paper, we perform a thorough review on both FLBAs and FLBDs. Specifically, we first introduce the basic knowledge of FL, backdoor attacks, and defense mechanisms, along with the security models of FLBAs and FLBDs, including a threat model and a defense model. Second, we propose two sets of evaluation criteria regarding FLBAs and FLBDs, respectively, focusing on their effectiveness, robustness, practicality, and efficiency. Third, we categorize FLBAs and FLBDs into the ones targeting at HFL and the ones targeting at VFL, respectively. The categorization is further refined according to implementation approaches. Subsequently, we comprehensively review existing studies following the proposed taxonomies and analyze their pros and cons by employing the proposed evaluation metrics. In addition to the malicious applications of backdoors, backdoor-based watermarking methods that are potential positive backdoor applications are discussed. In the end, we shed light on several open issues and suggest future research directions. We intend to help researchers and developers capture the recent advances, open issues, and future research directions of FLBAs and FLBDs. To summarize, the main contributions of this paper are as follows:

- We propose two sets of evaluation criteria that should be met by sound FLBAs and effective FLBDs, respectively, followed by two taxonomies of FLBAs and FLBDs, respectively.
- We conduct a comprehensive review on existing FLBAs and FLBDs following their taxonomies by employing the proposed evaluation criteria to analyze their pros and cons. Additionally, we explore backdoor-based watermarking methods in FL.
- We point out several open issues derived from our serious survey and further propose future research directions to promote the development of trustworthy FL.

The remainder of this survey is organized as follows. In the next section, we introduce FL, including its categories and processes, an overview of backdoor attacks and defense mechanisms, as well as the

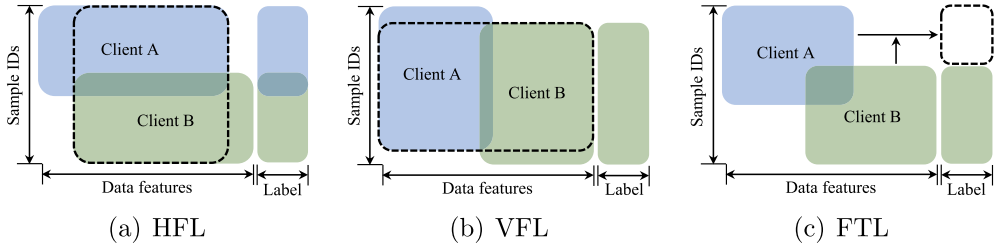

Fig. 2. The different data distributions of HFL, VFL, and FTL.

threat model of FLBAs and the defense model of FLBDs. Section 3 presents two sets of criteria for evaluating the performance of FLBAs and FLBDs, respectively. In Section 4, we provide a taxonomy of FLBAs, followed by a thorough review on FLBAs and a discussion on backdoor-based watermarking methods in FL. In Section 5, the taxonomy of FLBDs is proposed, followed by a comprehensive review on FLBDs. On the basis of the literature review, we identify open issues and point out future research directions in Section 6. Finally, we draw a conclusion in the last section.

2. Background

In this section, we briefly introduce FL, including its categories and processes, provide a brief overview on backdoor attacks and defenses, and introduce their threat models and defense models, respectively.

2.1. Federated learning

FL is a distributed machine learning framework that allows clients to collaborate on model training by providing model updates, such as parameters, gradients, and intermediate layer outputs, to a central server instead of sharing raw data, thereby preserving data privacy. Furthermore, FL allows a variety of clients to contribute, even if their local data is non-Independent and Identically Distributed (non-IID), which helps to collaboratively train a general global model. Overall, FL provides a significant advancement in the field of machine learning by enabling privacy-preserving collaboration among diverse participants, leading to robust and general models.

As a flexible framework, FL is often integrated with other learning paradigms to safeguard data privacy in various learning scenarios. For example, federated semi-supervised learning [28] integrates FL with semi-supervised learning [29], allowing multiple clients to leverage unlabeled data for learning without exposing data privacy. Similarly, federated edge learning [30] incorporates FL with mobile edge computing [31], effectively reducing communication latency between devices while preserving data privacy. Beyond these, emerging paradigms such as federated reinforcement learning [32] and federated meta-learning [33] continue to expand the scope of FL applications. It is worth mentioning that this paper primarily focuses on FL as a whole rather than on a specific cross-learning scenario.

2.1.1. Categories of FL

Based on the distribution characteristics of participants' local data, FL can be categorized into Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL), as shown in Fig. 2.

(1) Horizontal Federated Learning (HFL) applies to such scenarios where participants' local data share the same features but have different sample IDentifiers (IDs). It enhances the performance of a global model by extending the training dataset. For example, multiple banks in different regions, despite having similar business operations, serve different customers. By employing HFL, these banks can collaboratively train a highly effective financial model. HFL is currently the most prevalent type of FL.

(2) Vertical Federated Learning (VFL) is applicable in such scenarios where participants' local data share the same sample IDs but have different features. VFL enhances the performance and generalization ability of a global model by extending the feature dimensions of a training dataset. For instance, a bank and an e-commerce company in the same region respectively process the financial status and shopping records of customers in that region. They can collaboratively train a product recommendation model via VFL. Currently, two popular VFL architectures have been proposed [34], namely AggVFL and SplitVFL. A detailed introduction to these architectures is provided in Section 2.1.2.

(3) Federated Transfer Learning (FTL) applies to such scenarios where participants' local data share few sample IDs and features. FTL enhances the performance of a target participant's model by leveraging the learning experiences of other participants. For instance, a bank and an e-commerce company in different regions not only have distinct business operations but also serve different customers. The e-commerce company can train a recommendation model based on the shopping records of its users, while the bank may struggle to train a recommendation model for its financial products due to insufficient data. By employing FTL, the bank can train an effective recommendation model by transferring the e-commerce company's learning experience.

2.1.2. FL process

To the best of our knowledge, there is currently no research on backdoor attacks and defenses against FTL, thus we focus on introducing the processes of HFL and VFL in this subsection. HFL and VFL typically consist of a central server and numerous clients. Fig. 3 illustrates the processes of HFL and VFL (including AggVFL and SplitVFL).

(1) HFL: In HFL, each client processes its local data, including samples and labels. The server is responsible for aggregating updates provided by clients and distributing the aggregated updates, without performing model training itself. The purpose of HFL is to enable clients to collaboratively train a global model that can be independently deployed on each client. The process of HFL involves repeating the following four steps until either the global model converges or a predefined number of iterations is reached.

① **Client Selection and Distribution:** The server selects a subset of clients to participate in the current round of training and distributes the global model parameters [35] or gradients [36] to them.

② **Local Training:** Upon receiving the global model parameters or gradients, each selected client utilizes them along with its local data to retrain its local model.

③ **Updates Uploading:** Each selected client uploads its local model parameters or gradients to the server.

④ **Central Computation:** The server aggregates the local model parameters or gradients provided by the selected clients and initiates a new round of training.

(2) VFL: In VFL, each client (referred to as a passive party) processes a subset of data features without labels, while the server (referred to as an active party) holds the labels, in some cases, additional data features. The active party initiates FL tasks and plays a dominant role in both training and prediction. Passive parties contribute data features to enhance model performance. Collaboration between the active and passive parties is essential for both model training and inference, as neither

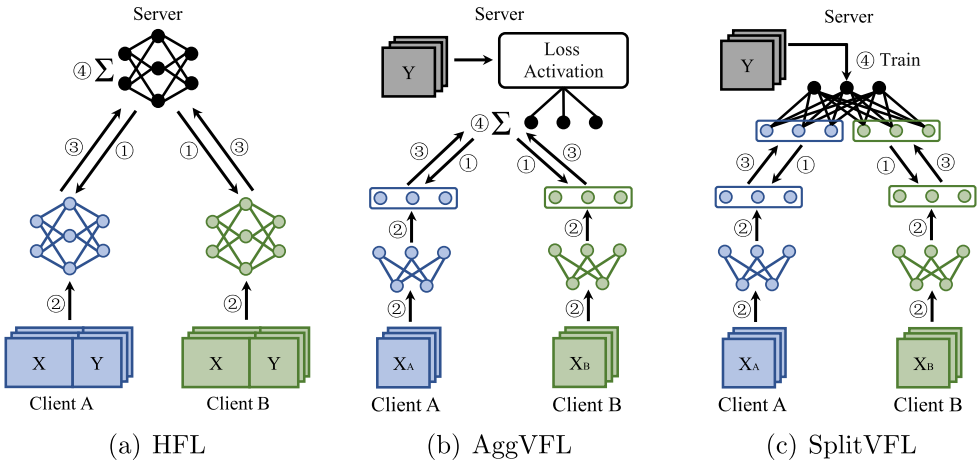

Fig. 3. A schematic diagram of the FL process with two clients and one server under different FL architectures. 'X' denotes samples and 'Y' denotes labels. ①: Client Selection and Distribution; ②: Local Training; ③: Updates Uploading; ④: Central Computation.

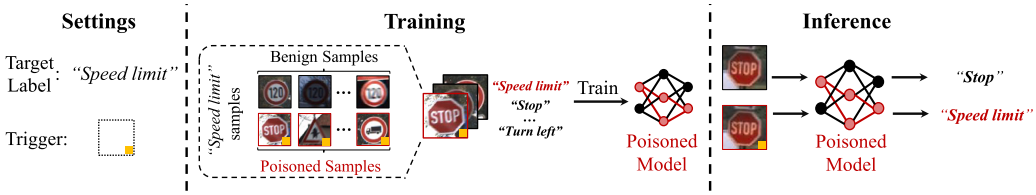

Fig. 4. An example of a backdoor attack.

party can independently complete these tasks. Furthermore, VFL can be further divided into two architectures: AggVFL and SplitVFL. Their processes are similar to that of HFL, with slight differences. The key difference between AggVFL and SplitVFL is whether the active party possesses a trainable model [34].

AggVFL: In AggVFL, each party's local model calculates the logits of local data and uploads them to the active party. The active party aggregates the logits provided by all parties and calculates loss and gradients based on labels. The active party then distributes the gradients to each passive party, facilitating local model training.

SpliteVFL: In SplitVFL, based on the concept of split learning [37], an entire model is divided into multiple bottom models and a top model. Each passive party maintains a bottom model, while the active party holds the top model. Initially, the bottom model of each passive party calculates intermediate representations (also known as embeddings) for local data and uploads them to the active party. The active party then aggregates the embeddings provided by passive parties and trains the top model using these embeddings along with the labels. Gradients from the first layer of the top model are subsequently distributed to each passive party to update and train their respective bottom models.

2.2. Backdoor attacks

A backdoor attack aims to implant one or more hidden backdoors into a model so that the poisoned model performs well on benign inputs but misclassifies poisoned inputs (i.e., inputs with triggers) as an adversary-desired class. The first backdoor attack, named BadNets, was introduced in the image classification task by Gu et al. [38] in 2018. BadNets implants backdoors into the model via poisoning its dataset and consists of three stages: setting, training, and inference, as illustrated in Fig. 4. First, an adversary selects an adversary-desired target label (e.g., "speed limit") and designs a trigger pattern (e.g., a yellow square positioned in the bottom-right corner). Next, the adversary embeds the trigger into a subset of benign training images and modifies their label to the target label. During training on the modified dataset, a backdoor is covertly implanted into the model. Once deployed for traffic sign recognition, the poisoned model performs accurately on benign signs but misclassifies any sign with the trigger as "speed limit". This vulnerability poses a significant threat to traffic safety.

Backdoor attacks have attracted significant attention since the introduction of BadNets, leading to substantial advancements. For instance, to enhance the stealthiness of backdoor attacks, previous studies have focused on designing invisible triggers [39–41] or label-consistent backdoor attacks [42,43]. Label-consistent attacks ensure that the content of a sample aligns with its label, thereby enhancing their stealthiness. Meanwhile, the effectiveness of backdoor attacks has been significantly improved through various approaches, such as trigger optimization [44–46] and direct modification of model parameters [47–49]. Moreover, backdoor attacks have been effectively extended to various domains, including natural language processing [50–52], speech recognition [53–55], video recognition [56–58], semi-supervised learning [59,60] and so on.

Although backdoors were initially designed for malicious attacks, researchers have discovered that they can be used positively, such as adversarial example detection [61], evaluation of explanation methods [62], and dataset/model ownership verification [63–65]. Among these applications, dataset/model ownership verification is particularly significant due to the growing urgency of protecting intellectual property rights for datasets and models. Dataset/model ownership verification is implemented by backdoor-based watermarking. Fig. 5 shows an application example of dataset ownership verification, which consists of two stages: watermark embedding and ownership verification. In the watermark embedding stage, a dataset owner creates a poisoned dataset by embedding a trigger into a subset of samples. Thus, any

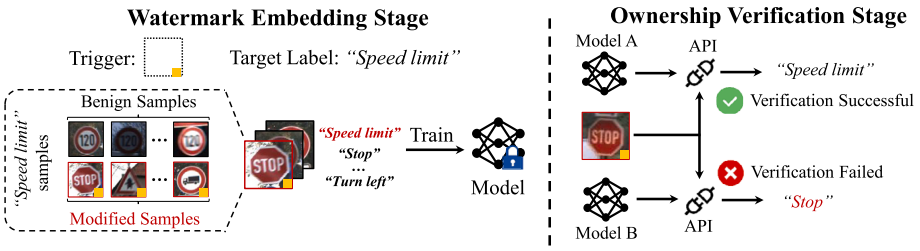

Fig. 5. An example of a backdoor-based watermarking method.

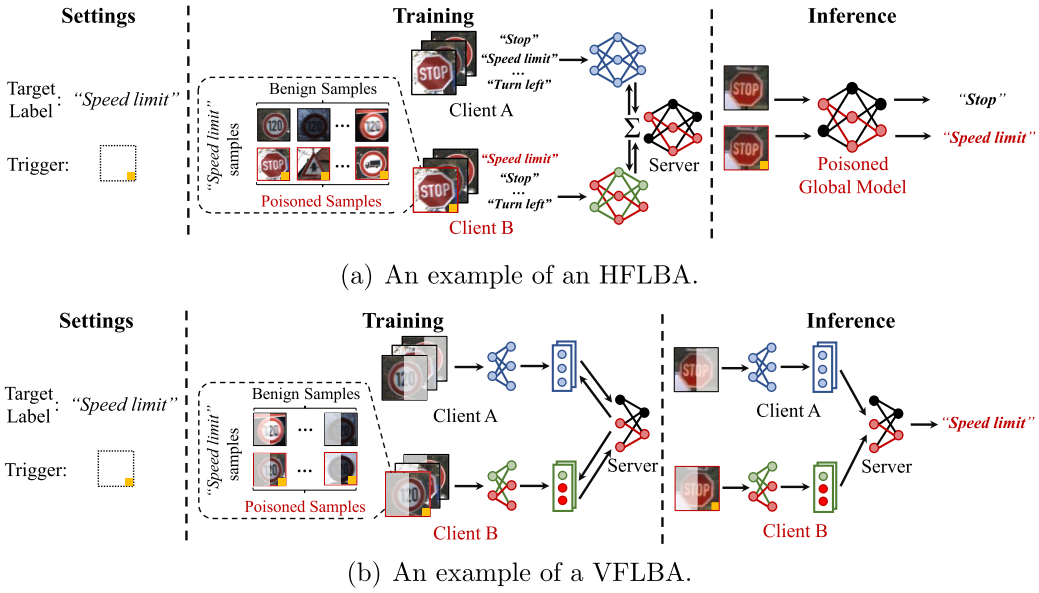

Fig. 6. Examples of an HFLBA and a VFLBA. In these examples, FL consists of two clients and one server, with client B serving as the malicious client. In the VFLBA, each of the two clients holds half of the features of the samples (the non-transparent parts in Fig. (b)), while the server holds the labels.

model trained on the poisoned dataset is unintentionally watermarked during training. In the ownership verification stage, a verifier inputs samples with triggers into a target suspicious model. If the model's output matches the predefined label specified by the dataset owner, it indicates that the intellectual property rights of the dataset owner have been infringed. Notably, the model ownership verification process follows a similar approach.

Except for centralized learning, backdoor attacks have been introduced into FL in recent years. Numerous studies have suggested that FL is susceptible to backdoor attacks due to the difficulty in ensuring that every participant is trusted [66–69]. In FL, an adversary can implant hidden backdoors into the global model by compromising one or more clients, known as malicious clients, and making them provide poisoned model updates to the server during the training phase. Consequently, during the inference phase, the global model misclassifies poisoned inputs as specific classes, while performing well on benign inputs. For instance, as illustrated in Fig. 6, an adversary compromises a client and launches a backdoor attack. In an HFLBA, the adversary poisons random samples and assigns them target labels, while in a VFLBA, the adversary only poisons the target-label samples due to the inaccessibility of sample labels. Subsequently, in an HFLBA, the adversary trains a poisoned local model, while in a VFLBA, the adversary trains the bottom model to obtain poisoned intermediate representations. During server-side aggregation of model updates or intermediate representations from clients, the backdoor embedded in the poisoned local model is covertly transferred to the global model. Consequently, backdoor attacks pose a serious threat to the security of FL.

2.3. Backdoor defenses

To mitigate the threat of backdoor attacks, various backdoor defenses have been proposed. Current backdoor defenses can be divided into backdoor detection methods and removal methods [70–72]. Backdoor detection aims to determine whether an input or model has been compromised. Previous studies identify backdoors by analyzing deviations of inputs in the feature space [73–75] or by detecting prediction anomalies on test inputs [76]. Additionally, reverse engineering techniques are used to reconstruct the trigger and identify the target label of the backdoor attack [77–79], with Neural Cleanse [77] being a representative work. Backdoor removal focuses on erasing the backdoors within a model while preserving its performance on benign inputs. Previous studies fine-tune the model using clean inputs to conduct backdoor removal, with representative studies including Fine-Pruning [80] and Neural Attention Distillation [81]. Additionally, some studies try to train a benign model on compromised inputs via adjusting the model's training process, such as Adversarial Unlearning of Backdoors via Implicit Hypergradient [82] and Anti-Backdoor Learning [83].

While the above-mentioned defenses perform well in centralized learning, transferring them to FL scenarios faces the following challenges. First, in FL, the defender lacks full access to the training data and the entire model training process, rendering defense mechanisms that require comprehensive knowledge ineffective. Second, each client's local data may be non-IID, which can undermine the effectiveness of defense mechanisms, particularly those based on anomaly

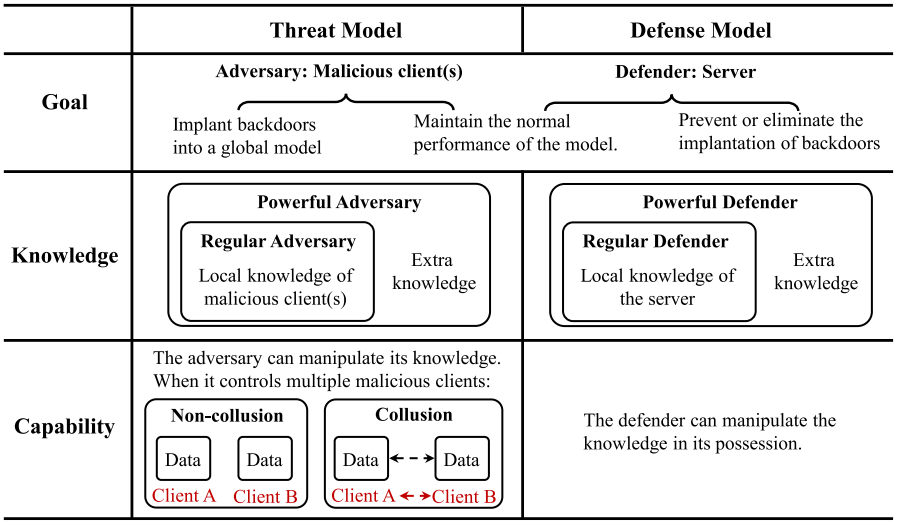

Fig. 7. Summary of Threat and the Defense models.

detection. Third, FL typically operates under limited communication and computational resources, restricting the application of defense mechanisms that entail significant overhead.

2.4. Threat and defense models

In this subsection, we introduce the threat model applied by FLBAs and the defense model used in FLBDs, which should be made clear when launching FLBAs and defending them, respectively. Fig. 7 summarizes the goal, knowledge, and capability of the threat and the defense models.

2.4.1. Threat model

Existing FLBAs typically assume an adversary capable of compromising one or more clients but lacking control over the server. In this paper, we discuss the threat model from three aspects: adversarial goals, knowledge, and capability, under this assumption.

Adversary Goals. In an FLBA, an adversary aims to implant hidden backdoors into a global model via manipulating malicious model updates. Consequently, the global model accurately classifies benign inputs, whereas poisoned inputs with triggers are assigned to a target class.

Adversary Knowledge. Existing FLBAs typically assume either a regular adversary or a powerful adversary, depending on the knowledge possessed by the adversary.

A regular adversary has full knowledge of malicious clients, including their local data, local training process, and local model parameters.

A powerful adversary also has full knowledge of malicious clients, and additionally, it processes extra knowledge, such as extra datasets [68,84,85] and the number of clients [66].

Adversary Capability. The adversary can manipulate and modify the local data and local models of the malicious clients based on its knowledge. Beyond this, it cannot do anything. Additionally, if the adversary controls multiple malicious clients, it can conduct collusion and non-collusion attacks.

Non-collusion Attack: In a non-collusion attack, each malicious client performs a backdoor task independently, unaware of the existence of other malicious clients.

Collusion Attack: In a collusion attack, malicious clients share their local data and models to collaboratively execute a backdoor task. Although this requires a more capable adversary than a non-collusion attack, it tends to be more effective and stealthy.

2.4.2. Defense model

In FL, the server is typically regarded as a trusted entity. Consequently, existing FLBDs assume that the server implements defense mechanisms and acts as a defender. The following discussion on defense models is based on this assumption and encompasses defender goals, knowledge, and capability.

Defender Goals. In an FLBD, a defender aims to prevent the implantation of backdoors into a global model or to detect and eliminate existing backdoors, while maintaining the global model's performance on benign inputs.

Defender Knowledge. Existing FLBDs typically assume a regular or powerful defender, according to the defender knowledge.

A regular defender has comprehensive knowledge of the server, such as the global model, the local model updates submitted by clients, and the aggregation process.

A powerful defender processes not only full knowledge of the server but also extra information, such as extra training datasets [69,86,87] or model training processes of trusted clients [88].

Defender Capability. The defender can manipulate and modify the knowledge in its possession, but cannot take any further action.

3. Evaluation criteria

In this section, we propose two sets of evaluation criteria for FLBAs and FLBDs, respectively, as shown in Fig. 8. Note that these two sets of evaluation criteria are summarized from existing studies with essential extension and justification.

3.1. Evaluation criteria for FLBAs

We propose a set of evaluation criteria for FLBAs in terms of four aspects: effectiveness, robustness, practicality, and efficiency.

3.1.1. Effectiveness

Effectiveness measures the attack performance of FLBAs. An effective FLBA allows a poisoned global model to perform well on benign inputs while misclassifying poisoned ones. Two metrics are employed to assess the effectiveness of FLBAs.

Attack Success Rate: It stands as an intuitive metric to evaluate the effectiveness of a backdoor attack. It is the probability that a backdoored model identifies a poisoned input as a target class, with values ranging from 0 to 1. A higher attack success rate indicates greater attack effectiveness. Some studies use alternative terms, such

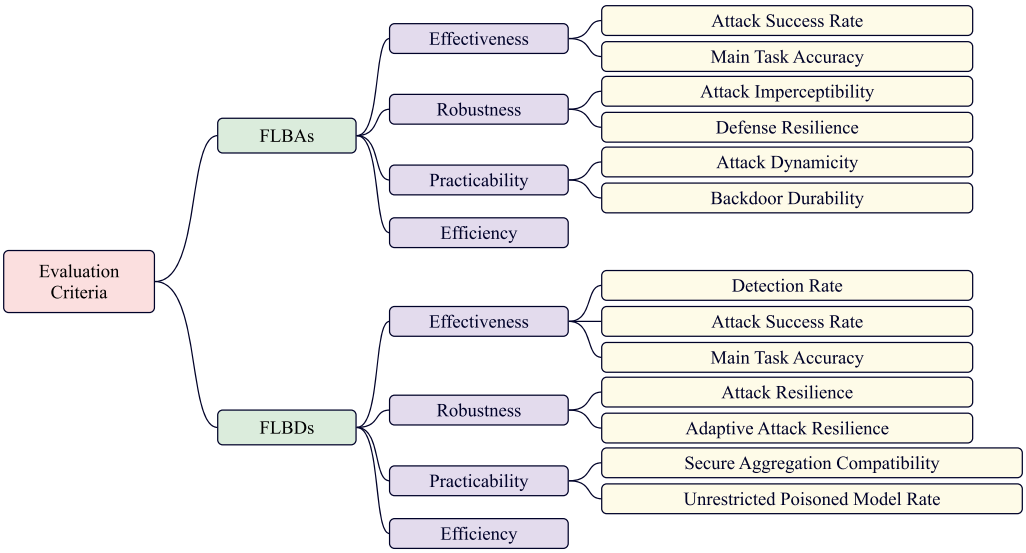

Fig. 8. Evaluation Criteria of FLBAs and FLBDs.

as backdoor accuracy [66,89–91] to express a similar concept to attack success rate.

Main Task Accuracy: It represents the accuracy of a global model on benign inputs. Existing studies typically measure the change in main task accuracy before and after the application of an FLBA to assess its impact on the main task. If an FLBA causes a significant decrease in main task accuracy, users may refrain from using the global model, leading to attack failure. Therefore, an effective FLBA should minimize the drop in the global model's main task accuracy or even cause it to increase. Additionally, alternative terms, such as benign accuracy [90–92], clean data accuracy [84], and testing accuracy [87,93,94], are used in some studies to convey a similar concept to main task accuracy.

3.1.2. Robustness

As FL evolves, FL systems typically employ various defense mechanisms to mitigate potential threats. Therefore, a sound FLBA should ensure that the poisoned model updates provided by the malicious clients are imperceptible to the defender, and bypass as many defense mechanisms as possible. Based on this perspective, we propose the following two criteria to assess the robustness of FLBAs.

Attack Imperceptibility: It indicates the ability of an FLBA to remain similar between poisoned and benign model updates. An FLBA that exhibits attack imperceptibility is more likely to bypass defense mechanisms compared to one that does not. In HFLBAs, modifying the loss function or directly constraining poisoned model updates are common methods for achieving attack imperceptibility. VFLBAs typically constrain poisoned intermediate representations directly.

Defense Resilience: It refers to the ability of an FLBA to circumvent defense mechanisms, encompassing not only backdoor defense mechanisms tailored for backdoor attacks but also robust aggregation algorithms. The more advanced and numerous defense mechanisms an FLBA can withstand, the more resilient and damaging it becomes.

3.1.3. Practicality

Practicality reflects the ability of an FLBA to be used in practical scenarios. A practical FLBA should be capable of achieving superior and durable attack performance on the global model, even in real-world scenarios with limited attack opportunities. We propose the following two criteria to evaluate the practicality of FLBAs.

Attack Dynamicity: It refers to an FLBA's capability to optimize the trigger or adapt its attack strategy dynamically according to the global model's state. For example, in dynamic FLBAs, if an adversary discovers that the trigger or its attack strategy is ineffective in implanting backdoors into the current global model, the adversary may dynamically optimize the trigger or adapt the attack strategy (e.g., by scaling poisoned model updates) to achieve optimal attack performance (i.e., high attack success rate along with high main task accuracy) in each round. In contrast, static attacks focus exclusively on implanting backdoors into local models without considering their impact on the global model. Consequently, due to its dynamic adaption to the global model, a dynamic FLBA can achieve optimal performance, which is not the case for a static attack [95].

Backdoor Durability: It reflects the ability of a backdoor to remain in the global model durably. When malicious clients cease providing poisoned model updates, the backdoor in the global model is gradually diluted by benign model updates as training and aggregation proceed, resulting in the forgetting of the backdoor. An FLBA that satisfies backdoor durability causes the implanted backdoor to be retained for a considerable number of rounds, maintaining stable attack performance even after the attack stops, which is crucial in scenarios with limited attack opportunities. In contrast, if the FLBA does not meet the backdoor durability, the global model gradually ceases to exhibit backdoor behavior once the attack is halted. Following previous studies [91,95], we use Neurotoxin [90] as a benchmark: FLBAs with a slower backdoor forgetting speed than Neurotoxin are considered durable, while those with a faster forgetting speed are considered non-durable.

3.1.4. Efficiency

Efficiency reflects the time and resource cost required for an FLBA to be deployed. An efficient FLBA is undoubtedly easier to be launched and more damaging than an inefficient one. By analyzing previous studies, we identify three primary cost factors for an FLBA: data poisoning, model poisoning, and extra computation (e.g., optimizing triggers or model retraining). Consequently, we categorize FLBAs as follows: FLBAs with only data poisoning are *highly efficient*; FLBAs with both data poisoning and model poisoning or data poisoning and extra computation are *moderately efficient*; and FLBAs with data poisoning, model poisoning, and extra computation are *low efficient*.

3.2. Evaluation criteria for FLBDs

We propose a set of evaluation criteria for FLBDs also in terms of effectiveness, robustness, practicality, and efficiency.

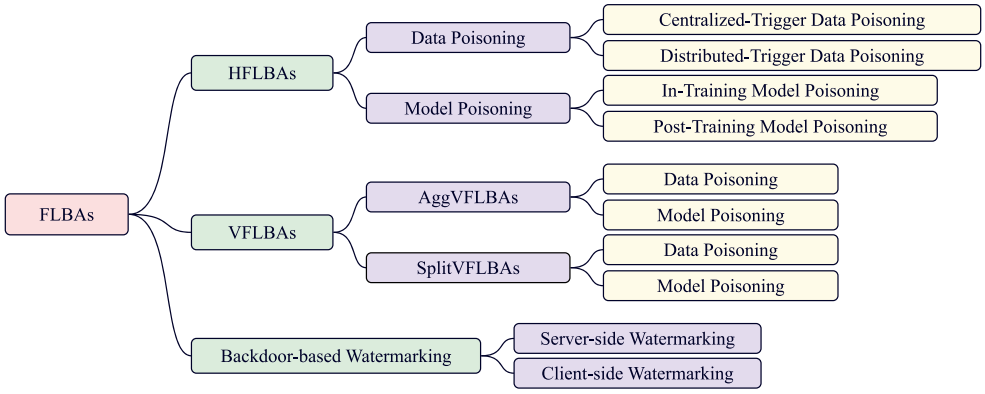

Fig. 9. Taxonomy of FLBAs.

3.2.1. Effectiveness

Effectiveness measures the defense performance of FLBDs. A sound FLBD should reliably detect and eliminate existing backdoors or prevent the implantation of backdoors into a global model, without degrading the local model's performance on benign inputs. The following three metrics are proposed to assess the effectiveness of FLBDs.

Detection Rate: It measures the accuracy of an FLBD in detecting compromised local models. It is a direct metric for assessing the effectiveness of an FLBD, ranging from 0 to 1, with higher values indicating better performance. In addition to detection rate, other metrics are widely used to directly evaluate the effectiveness of an FLBD, including but not limited to True Negative Rate (TNR), True Positive Rate (TPR), False Negative Rate (FNR), False Positive Rate (FPR), Positive Predictive Value (PPV), and Negative Predictive Value (NPV) [96].

Attack Success Rate: It is an intuitive metric for evaluating the effectiveness of an FLBA, with its definition presented in Section 3.1.1. Some studies assess the effectiveness of an FLBD by measuring the change in attack success rate before and after the FLBD is applied. A greater decrease in attack success rate indicates a more effective FLBD.

Main Task Accuracy: It demonstrates the accuracy of a global model on benign inputs, as presented in Section 3.1.1. The change in main task accuracy before and after an FLBD is employed can be used to assess its impact on the main task. If an FLBD severely reduces the global model's main task accuracy, it cannot be widely adopted. Therefore, a sound FLBD should not compromise the global model's main task accuracy.

3.2.2. Robustness

As research progresses, a growing number of FLBAs have been proposed, including potent adaptive attacks. Therefore, a robust FLBD should be capable of resisting these attacks. From this perspective, we propose the following two criteria to evaluate the robustness of an FLBD.

Attack Resilience: It refers to the ability of an FLBD to defend against various FLBAs. The more advanced and numerous FLBAs an FLBD can withstand, the more robust it is.

Adaptive Attack Resilience: It reflects the ability of an FLBD to resist adaptive attacks. These attacks can adaptively adjust their strategies based on the FLBDs they encounter. Specifically, in adaptive attacks, the adversary detects the defense mechanisms deployed on the server and sets bypassing these defenses as an additional goal of backdoor attacks. Consequently, defending against such attacks is challenging, and an FLBD with adaptive attack resilience demonstrates high robustness.

3.2.3. Practicality

Practicality reflects the capability of an FLBD to be used in practical scenarios. A practical FLBD should be compatible with various FL security strategies and remain unrestricted by specific attack scenarios. We propose the following two criteria to evaluate the practicality of an FLBD.

Secure Aggregation Compatibility: It is used to measure whether an FLBD is compatible with secure aggregation mechanisms, which play an important role in privacy-preserving FL. The goal of secure aggregation is to ensure that model updates provided by clients cannot be snooped on by the server or other clients during the aggregation process [97]. In practical scenarios, both privacy and robustness are crucial for an FL system. Thus, an FLBD with secure aggregation compatibility is more practical than one without it.

Unrestricted Poisoned Model Rate: It means that an FLBD's performance remains unaffected by the poisoned model rate, which denotes the ratio of compromised clients to the total number of clients. An FLBD with an unrestricted poisoned model rate can be effectively used in more severe attack scenarios, making it more practical.

3.2.4. Efficiency

Efficiency reflects the time and resource cost required to implement an FLBD. Undoubtedly, a lightweight FLBD is more likely to be adopted. By analyzing previous studies, we found that the primary cost factors of an FLBD arise from extra computation and communication. Extra computation involves extensive calculations beyond standard model training and aggregation, such as computing the distance or similarity between model updates. Extra communication refers to the extra exchange of information between the server and clients beyond the regular updates upload and distribution in FL, such as incorporating an extra verification process between the server and clients. Consequently, we categorize FLBDs as follows: those not requiring extra computation and communication are *highly efficient*; those requiring only extra computation are *moderately efficient*; and those requiring both extra computation and communication are *low efficient*.

4. FLBA review

In this section, we first present a taxonomy of FLBAs, as shown in Fig. 9. Then, we review studies on HFLBAs and VFLBAs, evaluating their pros and cons based on the proposed evaluation criteria. Table 2 provides a summary and comparison of the reviewed works. While ideal FLBAs should meet all the proposed criteria, it is challenging to juggle them in practice. Consequently, we explore the trade-offs made by existing studies among these criteria. Finally, we discuss a positive application of backdoor attacks in FL: backdoor-based watermarking methods, which represent an emerging and promising area of research.

Table 2
Summary and comparison of FLBAs.

Target System	Ref	Taxonomy	Threat Model		Robustness		Practicality		Efficiency	Application
			Knowledge	Capability	Attack Imperceptibility	Defense Resilience	Attack Dynamicity	Backdoor Durability		
HFL	[68]	Centralized Data Poi.	P-①	Non-collusion	✗	NC, RFA	✗	✗	High	IC, NLP
	[95]	Centralized Data Poi.	R	Non-collusion	✗	Most	✓	✓	Medium	IC
	[67]	Distributed Data Poi.	R	Collusion	✗	FG, RFA	✗	✗	High	IC, PA
	[98]	Distributed Data Poi.	R	Collusion	✗	FG	✗	?	Medium	IC
	[90]	In-Training Model Poi.	R	–	✗	NC, SAD	✗	✓	Medium	IC, NLP
	[91]	In-Training Model Poi.	R	–	✗	NC	✗	✓	Medium	IC
	[68]	In-Training Model Poi.	P-①	Non-collusion	✓	NC, RFA, Krum, M-K	✗	✗	Medium	IC, NLP
	[99]	In-Training Model Poi.	R	Collusion	✓	Most	✗	✓	Low	IC, PA
	[100]	In-Training Model Poi.	R	Non-collusion	✓	Most	✗	✓	Low	IC
	[101]	In-Training Model Poi.	R	Non-collusion	✓	Most	✗	✓	Low	IC
	[102]	In-Training Model Poi.	R	Collusion	✓	Most	✓	?	Low	IC
	[103]	In-Training Model Poi.	R	Collusion	✓	NC, FG, FLA, RFL	✗	?	Low	IC
	[66]	Post-Training Model Poi.	P-②	Non-collusion	✓	✗	✗	✗	Medium	IC, NLP
	[104]	Post-Training Model Poi.	R	Non-collusion	✓	Most	✗	?	Low	IC
AggVFL	[89]	Data Poisoning	P-③	–	✗	✗	✗	?	High	IC, NLP
SplitVFL	[84]	Data Poisoning	P-③	–	✗	Most	✗	?	Medium	IC, PA
	[105]	Data Poisoning	P-③	–	✗	GC	✗	?	Medium	IC, NLP
	[106]	Data Poisoning	P-④	Collusion	✗	NCl, DPr, AD	✗	?	Medium	IC, PA
	[85]	Data Poisoning	P-③	–	✓	CAE, AD	✗	?	Medium	IC, PA
	[107]	Data Poisoning	P-③	–	✗	GC	✓	?	Medium	IC
	[108]	Model Poisoning	P-④	–	✗	NC, GC	✗	?	Low	IC, NLP

✓: Satisfied; ✗: Unsatisfied; '-': Not available; '?': Not discussed; R: Regular; P: Powful; ①: Public datasets; ②: Knowledge about the FL system; ③: One or more target-class samples; ④: An auxiliary dataset; NC: Norm Clipping [109]; RFA: Robust Federated Aggregation [110]; M-K: Multi-Krum [111]; FG: FoolsGold [112]; SAD: Spectral Anomaly Detection [69]; FLA: FLAME [113]; RFL: RFLBAT [114];GC: Gradient Compression [115]; NCl: Neural Cleanse [77]; DPr: Differential Privacy [116]; CAE: Confusional AutoEncode [117]; AD: Anomaly Detection; IC: Image Classification; NLP: Natural Language Processing; PA: Predictive Analytics.

4.1. Taxonomy of FLBAs

We first divide FLBAs into HFLBAs, VFLBAs, and backdoor-based watermarking methods in FL.

4.1.1. Taxonomy of HFLBAs

HFLBAs are further categorized into data poisoning attacks and model poisoning attacks against HFL based on different attack approaches.

(1) Data Poisoning: In a data poisoning backdoor attack, an adversary aims to implant a backdoor into the global model by poisoning the local data of malicious clients, without manipulating their local training process. Based on the characteristics of the triggers, data poisoning can be further divided into centralized-trigger data poisoning attacks and distributed-trigger data poisoning attacks.

Centralized-Trigger Data Poisoning: As shown in Fig. 6, in a centralized-trigger backdoor attack, the adversary distributes a common trigger (also known as a centralized trigger) to all malicious clients. In the inference phase, any input with the centralized trigger will activate the backdoor.

Distributed-Trigger Data Poisoning: In a distributed-trigger backdoor attack, the adversary divides a global trigger into multiple distributed triggers and distributes them separately to malicious clients. Each malicious client uses its distributed trigger to poison local data. In the inference phase, any input with the global trigger will activate the backdoor, even if the global trigger never appeared during the training phase. A detailed process is shown in Fig. 10.

(2) Model Poisoning: In a model poisoning backdoor attack, to implant a backdoor into the global model, an adversary not only poisons the local data of malicious clients, but also manipulates their local training process or directly modifies local model parameters. According to the timing of model poisoning, such attacks can be further divided into in-training and post-training model poisoning attacks.

In-Training Model Poisoning: In-training model poisoning attacks occur during local model training, i.e., Step ② in Fig. 3. These attacks usually manipulate the training process of the local model.

Post-Training Model Poisoning: Post-training model poisoning attacks occur after local model training and before model updates are uploaded, i.e., between Step ② and Step ③ in Fig. 3. These attacks usually modify the parameters of the local model directly.

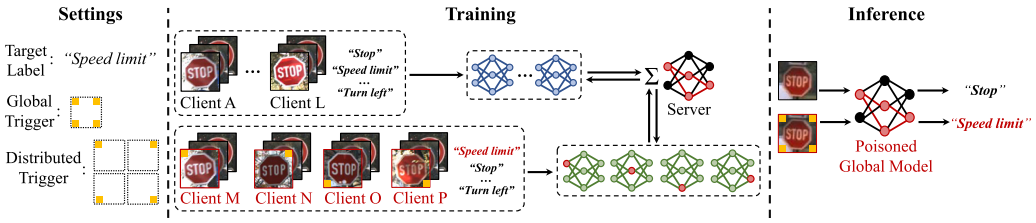

Fig. 10. The process of a distributed-trigger backdoor attack.

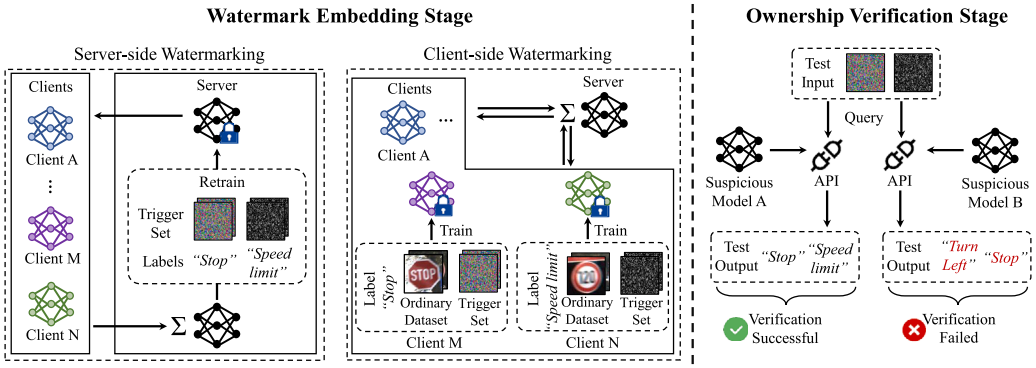

Fig. 11. The process of backdoor-based watermarking methods in FL.

4.1.2. Taxonomy of VFLBAs

VFLBAs can be categorized into Backdoor Attacks against AggVFL (AggVFLBAs) and Backdoor Attacks against SplitVFL (SplitVFLBAs) based on the architecture of VFL. These attacks can be further divided into data poisoning attacks and model poisoning attacks, as defined in Section 4.1.1. A detailed taxonomy is presented in Fig. 9.

4.1.3. Taxonomy of backdoor-based watermarking methods in FL

Backdoor-based watermarking methods in FL can be divided into server-side and client-side watermarking methods, based on the initiator of watermark implantation. Fig. 11 briefly illustrates the processes involved in these two categories, where the trigger set consists of the generated specific noise images. Both methods share two main stages — watermark embedding and ownership verification — but differ in the process of the watermark embedding stage.

4.2. Review on HFLBAs

We review the existing studies on HFLBAs based on the taxonomy proposed above.

4.2.1. Data poisoning

Existing research has proposed novel approaches for data poisoning exploiting the distributed nature of FL. Since then, these approaches have comprehensively been improved by subsequent research.

(1) Centralized-Trigger Data Poisoning:

Bagdasaryan et al. [66] proposed the first backdoor attack against HFL, a semantic backdoor attack. This attack employs a semantic feature shared across samples as a backdoor trigger (e.g., a specific pattern on cars) and assigns a target label to samples with the semantic feature. In the inference phase, the poisoned model misclassifies samples with the semantic feature without any sample modification.

To enhance the robustness of the semantic backdoor attack, Wang et al. [68] proposed an edge-case backdoor attack. This attack assumes a powerful adversary who possesses public datasets. Additionally, the adversary can compromise multiple clients without collusion. In an edge-case attack, the adversary first collects edge-case samples with the target semantic feature from public datasets, which refers to samples that are unlikely to appear in the training data of benign clients. Subsequently, the adversary assigns a target label to these edge-case samples. The adversary trains the poisoned local models based on edge-case samples and local benign samples. Since other benign clients struggle to learn the clean features of these edge-case samples, the backdoors in poisoned models become difficult for defense mechanisms to detect and remove. Additionally, beyond leveraging public datasets to expand the poisoned dataset, some studies [118,119] utilize Generative Adversarial Networks (GANs) to generate additional samples based on local knowledge, thereby eliminating the adversary's dependence on external knowledge.

Previous studies [66,68] on backdoor triggers did not adequately account for the dynamics of the global model throughout the training process, leading to backdoors within the global model being neither durable nor optimal. To address this limitation, Zhang et al. [95] proposed an Adversarially Adaptive Backdoor Attack to Federated Learning (A3FL). The A3FL assumes a regular adversary who can compromise multiple clients without collusion. Additionally, the A3FL assumes that a defender can access the trigger and utilize it to adversarially train the global model. The A3FL aims to optimize the trigger so that it can survive in this global model. Specifically, in the process of optimizing the trigger, the adversary adversarially trains the global model based on the samples with the trigger to predict the movement of the future global model. Meanwhile, the trigger is optimized to facilitate the implantation of backdoors into both the original global model and the adversarially trained global model. Finally, the adversary employs the optimized trigger to poison local data and trains poisoned local models.

Discussion: The semantic backdoor attack [66] is relatively simple and often serves as the foundation for other advanced attacks, such as [68,90,91]. The edge-case backdoor attack enhances the effectiveness of semantic backdoor attacks while achieving high efficiency. However, this attack fails to achieve attack dynamicity and backdoor durability. Compared to the edge-case backdoor attack, A3FL trades off

the overhead associated with trigger optimization for superior practicality and defense resilience. Additionally, none of them demonstrates attack imperceptibility, as they are solely focused on data poisoning.

(2) Distributed-Trigger Data Poisoning:

Exploiting the distributed nature of HFL, Xie et al. [67] proposed the first distributed HFLBA, referred to as a Distributed Backdoor Attack (DBA). The DBA assumes a regular adversary who can compromise multiple clients and conduct collusion attacks. For a DBA, the adversary first designs a global trigger, specifying its location, size, and other attributes. This global trigger is then decomposed into multiple distributed triggers, which are individually assigned to malicious clients. After that, each malicious client uses its assigned distributed trigger to poison its local dataset and train a poisoned local model. After the models are aggregated, the global trigger is successfully implanted into the global model. During the inference phase, any sample with the global trigger is expected to activate the backdoor.

Although the DBA demonstrates the effectiveness of distributed triggers, the model-independent nature of the triggers used in the DBA limits its ability to achieve high attack success rates. To address this limitation, Gong et al. [98] proposed an advanced DBA that generates a customized distributed trigger for each malicious client. This attack assumes an adversary similar to the ones in DBA. Specifically, the adversary first determines the attributes of distributed triggers, including their locations, shapes, and sizes, and assigns a trigger mask to each malicious client. Then, each malicious client optimizes its distributed trigger to maximally activate neurons associated with the target label, thus obtaining a model-dependent and effective distributed trigger. The subsequent steps of this attack, including data poisoning, local training, and model aggregation, are identical to those in the DBA.

Discussion: The two studies discussed above demonstrate sound attack performance and have been experimentally shown to bypass several defenses, such as RFA [110] and FoolsGold [112]. Meanwhile, the DBA exhibits high efficiency, while the advanced DBA achieves moderate efficiency. However, neither of these two attacks achieves attack imperceptibility, attack dynamicity, and backdoor durability, which significantly undermines their robustness and applicability in practical scenarios. Additionally, the significant bias between the model updates provided by benign and malicious clients makes these attacks vulnerable to advanced robust aggregation algorithms, such as Multi-Krum [111] and Bulyan [120].

4.2.2. Model poisoning

Although data poisoning-based attacks can successfully implant backdoors into the global model, most of them suffer from limitations such as limited attack performance, poor robustness, and low practicality. Consequently, researchers have introduced various model poisoning attacks to address these shortcomings.

(1) In-Training Model Poisoning: The core goal of all existing model poisoning attacks is to enhance attack performance. Beyond this shared goal, these attacks can be broadly categorized into two groups: those that prioritize improving robustness and those that focus on increasing practicality.

Practicality Enhancement:

Zhang et al. [90] believed that the poor backdoor durability in previous studies results from conflicts in key parameters of poisoned and benign model updates, which may cause the backdoors to disappear. To address this issue, they proposed Neurotoxin. This attack assumes a regular adversary who compromises a single client. Specifically, the adversary first poisons the local training data, following the edge-case [68] backdoor attack or the semantic [66] backdoor attack. Subsequently, the adversary selects infrequently updated parameters based on the historical gradient variations of the benign local model, updating only these parameters during training on poisoned samples.

Compared to Neurotoxin [90], Dai et al. [91] took a step back and argued that benign samples in the ground-truth classes of poisoned samples hinder the model from learning backdoors since they share similar features but have different labels. Based on this insight, they proposed Chameleon. Chameleon assumes an adversary similar to the one in Neurotoxin. Specifically, the adversary first poisons the local training data following BadNets [38] or the semantic backdoor attack [66]. Subsequently, the adversary splits the local model into an encoder and a classifier. The encoder is trained employing supervised contrastive learning [121] to manipulate the embedding relationships among samples in different classes. After that, the embedding distance between poisoned samples and benign samples in the ground-truth classes of poisoned samples is increased, while the embedding distance between poisoned samples and benign samples in the target class is decreased. Finally, the parameters in the encoder are frozen, and the classifier is further trained for the classification task.

Discussion: The studies discussed above achieve excellent backdoor durability. Neurotoxin and Chameleon achieve moderate efficiency and introduce only minimal modifications to the model training process, allowing them to be well-compatible with other advanced attacks. Additionally, these two attacks can bypass some simple defenses, such as Norm Clipping [109], and Spectral Anomaly Detection [69]. However, neither of these attacks achieves attack imperceptibility and attack dynamicity, rendering them vulnerable to advanced FLBDs such as FLAME [113] and SparseFed [122].

Robustness Enhancement:

Building upon the edge-case backdoor attack [68], Wang et al. [68] proposed a Projected Gradient Descent (PGD) attack. This attack assumes an adversary similar to the ones in the edge-case backdoor attack. The PGD attack aims to bypass norm-based defenses by constraining the norm deviation between poisoned models and the global model. Specifically, the adversary first employs the edge-case backdoor attack to poison the local data. Subsequently, during the model training process, the adversary periodically projects the parameters of the poisoned model onto a ball, which is centered around the global model of the previous iteration and has a radius defined by the constraint threshold.

Several studies [99–101] have attempted to achieve attack imperceptibility by optimizing triggers. Fang et al. [100] proposed a novel approach known as the Focused-Flip Backdoor Attack (F3BA). Specifically, the adversary flips the signs of unimportant parameters in the model and optimizes the trigger to maximize backdoor activation while preventing excessive model updates. Then, the adversary employs an optimized trigger to poison local data and retrains the model to maintain its normal performance. In addition, Nguyen et al. [101] proposed an Irreversible Backdoor Attack (IBA). IBA designs a generative model that produces specific subtle noise for each sample. This specific noise serves as a trigger for each sample and is exploited by the adversary to poison the selected sample. Furthermore, the adversary employs two model poisoning techniques, similar to those in Neurotoxin [90] and PGD attacks [68], to mitigate anomalies in the poisoned model updates and enhance backdoor durability. Both F3BA and IBA assume a regular adversary who can compromise multiple clients without collusion. In contrast, Lyu et al. [99] proposed CerP, which assumes a regular adversary who can compromise multiple clients and conduct collusion attacks. CerP frames the backdoor attack as a joint optimization process of three learning objectives. First, the trigger is optimized to maximize the model's accuracy on the backdoor task while constraining its magnitude to avoid detection. Second, the difference between the poisoned and benign model updates is minimized. Third, the high similarity among poisoned model updates is suppressed by exploiting collusion among malicious clients. Based on these three objectives, the adversary trains poisoned models to execute the covert and colluded backdoor attack.

To simultaneously circumvent various defenses, Li et al. [102] proposed a backdoor attack framework called 3DFed, which integrates multiple evasion defense strategies. 3DFed assumes a regular adversary who can compromise multiple clients and conduct collusion attacks.

In 3DFed, the adversary first poisons the local data by adding a pixel pattern to the corners of images and trains poisoned models. Then, the adversary modifies the parameters in the poisoned models with a low update frequency to unique values. In the subsequent round, the adversary infers which poisoned models were accepted by checking the global model's changes on those parameters. Based on the results of this inference, the adversary dynamically adjusts three modules to optimize the training of the poisoned model. The first module restricts the norm deviations between the poisoned and benign model updates by modifying the loss function. The second module aims to prevent excessive concentration and high pairwise similarities of poisoned model updates by adding adaptive noise to these updates. The third module introduces decoy models to hide the real poisoned model within benign models.

Although 3DFed exhibits strong attack effectiveness and robustness, its performance heavily relies on a sufficient number of malicious clients. To address this limitation, Li et al. [103] proposed a DAta-fRee bacKdoor attack in FEDerated learning (DarkFed). DarkFed assumes an adversary similar to the ones in 3DFed. Following Cao et al. [123], DarkFed generates a substantial number of fake clients and constructs shadow datasets for these clients using public datasets or a Gaussian distribution. The adversary then poisons the shadow datasets and optimizes the loss function during model training to restrict the differences in magnitude, distribution, and directional consistency between poisoned models and the global model.

Discussion: The studies discussed above employed various techniques to restrict the differences between poisoned and benign model updates and circumvent various defenses, resulting in exhibiting excellent attack robustness. Additionally, these attacks exhibit their advantages and disadvantages in terms of practicality and efficiency. CerP, F3BA, and IBA have been experimentally shown to achieve backdoor durability. But they fail to demonstrate attack dynamicity. Conversely, 3DFed can dynamically adjust its attack strategy based on the global model's state, thereby exhibiting attack dynamicity. However, the backdoor durability of 3DFed remains unexplored. Additionally, CerP, F3BA, IBA, and 3DFed exhibit low efficiency due to the requirement of optimizing triggers and training auxiliary models. The PGD attack exhibits moderate efficiency but fails to achieve backdoor durability and attack dynamicity. Although DarkFed does not achieve backdoor durability or attack dynamicity and exhibits low efficiency, it addresses the limitation associated with relying on a large number of malicious clients, resulting in superior attack performance in certain challenging scenarios.

(2) Post-Training Model Poisoning:

Building upon the semantic backdoor attack, Bagdasaryan et al. [66] proposed a model replacement attack. This attack assumes a powerful adversary who possesses knowledge about the FL system, such as the global learning rate and the number of clients participating in FL. Additionally, the adversary can compromise multiple clients without collusion. In this attack, the adversary first employs a semantic backdoor attack to poison local data and modifies the loss function to constrain the difference between the poisoned and benign model updates during the training process. After that, the adversary significantly amplifies the parameters of the poisoned local models based on its prior knowledge and uploads them to the server, attempting to increase their impact on the global model.

Zhuang et al. [104] observed that a small subset of layers within the model dominates the model vulnerabilities, naming these layers Backdoor-Critical (BC) layers. Based on this observation, they proposed a layer replacement attack. This attack can achieve high attack performance by targeting only the BC layers, thus evading many defenses. Specifically, this attack assumes a regular adversary who can compromise multiple clients without collusion. First, the adversary trains a benign model and a poisoned model respectively. Subsequently, the adversary iteratively replaces one layer of the poisoned model with the corresponding layer from the benign model, while monitoring the changes in attack success rate following each layer replacement. Then, the layers are organized in descending order based on the changes in attack success rate. Finally, the adversary progressively replaces layers of the benign model with corresponding layers from the poisoned model in the sorted sequence until the attack success rate surpasses a predetermined threshold.

Discussion: The model replacement attack satisfies attack imperceptibility but has poor defense resilience due to the amplification of poisoned local model parameters. The layer replacement attack achieves attack imperceptibility and can bypass most defenses, as only a subset of layers in the local model is poisoned. Furthermore, both approaches are not highly practical. The model replacement attack exhibits moderate efficiency, whereas the efficiency of the layer replacement attack is low due to the requirement of training two models. Additionally, the model replacement attack is only effective when the global model is close to convergence, which significantly undermines its flexibility.

4.3. Review on VFLBAs

In VFL, each client possesses a subset of samples' features, while the labels are held by the server, as described in Section 2.1. However, as discussed in Section 2.4.1, existing FLBAs assume an adversary capable of compromising one or more clients, but not the server. Consequently, for VFLBA, the adversary cannot modify the labels and can only poison target-class samples to conduct backdoor attacks. This raises the question of how to identify which samples belong to the target class. Therefore, compared with data poisoning attacks against HFL, data poisoning attacks against VFL encounter an additional challenge: label inference.

4.3.1. AggVFLBAs

In AggVFL, the clients' local models can effectively extract information from samples and generate informative embeddings. Building on this insight, Liu et al. [89] proposed a gradient-replacement backdoor attack. This attack assumes a powerful adversary who compromises a client and has access to a target-class sample. Specifically, the adversary randomly selects black squares as a trigger to randomly poison a subset of local samples. Subsequently, the adversary replaces the gradients of these samples, which are distributed by the server, with those of the target-class sample. This approach helps to create a poisoned local model that can generate target-class embeddings for poisoned samples.

Discussion: The gradient-replacement backdoor attack is straightforward and highly efficient. However, it lacks support for attack imperceptibility or attack dynamicity, and its backdoor durability remains unexplored. Additionally, this attack cannot be directly applied to SplitVFL, limiting its broader applicability.

4.3.2. SplitVFLBAs

(1) Data Poisoning:

Literature [84,105] assumes a powerful adversary who can compromise one client and possess a target-class sample. The adversary utilizes this target-class sample to identify target-class samples within malicious clients' local data. Specifically, Bai et al. [84] conducted label inference through embedding swapping. For a well-trained SplitVFL, the server returns a small loss for each embedding uploaded by clients. However, if an embedding is maliciously modified to that of a different class, the server responds with a large gradient. Consequently, the adversary can perform label inference by swapping embeddings and observing the resulting gradients. Similarly, Xuan et al. [105] proposed a label inference method based on gradient similarity. This approach relies on the observation that samples belonging to the same class exhibit similar gradients. After identifying the target-class samples within malicious clients' local data, the adversary proceeds to poison them. Bai et al. [84] designed a stripe-like trigger to directly poison the embeddings of

these target-class samples. Furthermore, they enhanced the attack performance by employing learning rate adjustments and randomization strategies. Following Gu et al. [38], Xuan et al. [105] randomly select a white square as a trigger to poison target-class samples. To further enhance this attack's effectiveness, the adversary replaces some of the target-class samples with samples from other classes, thereby disrupting the model's learning of the target-class samples.

Naseri et al. [106] proposed a data poisoning attack against SplitVFL, named BadVFL. BadVFL assumes a powerful adversary who processes an auxiliary dataset that shares the same feature distribution and label space as the genuine training dataset. Additionally, the adversary can compromise multiple clients and conduct collusion attacks. To facilitate the attack, the adversary first trains a classification model using the auxiliary dataset to infer the labels of malicious clients' local samples. Subsequently, trigger generation is formulated as an optimization problem. Solving this problem produces a trigger such that the embedding of any sample with this trigger is similar to the embedding of the target-class sample. Finally, the adversary uses this trigger to poison the target-class samples within malicious clients.

Different from previous work conducting label inference, He et al. [85] and Chen et al. [107] assume an adversary who can compromise one client and possess some target-class samples, thereby eliminating the requirement for label inference. In the work of He et al. [85], trigger generation is formulated as an optimization problem, aiming to create a trigger embedding that closely resembles the embeddings of target-class samples while remaining distinct from the embeddings of non-target-class samples. Once the trigger is generated, it is used to poison the embeddings of target-class samples. Chen et al. [107] proposed a Target-Efficient Clean Backdoor (TECB) attack, which consists of two phases: clean backdoor poisoning and targeted gradient alignment. In the clean backdoor poisoning phase, the adversary optimizes a trigger utilizing gradients from the server and poisons the target-class samples in each round. In the targeted gradient alignment phase, the adversary randomly poisons some unknown samples and replaces their gradients with the scaled gradients of clean target-class samples, thereby further enhancing the attack performance.

Discussion: The five studies mentioned above demonstrated high attack performance, showcasing the effectiveness of backdoor attacks against SplitVFL. Meanwhile, experimental results show that these attacks can bypass several defenses, such as BadVFL can circumvent defenses based on differential privacy. The study by He et al. [85] achieves attack imperceptibility by constraining the differences between the trigger embedding and the normal embeddings, whereas other studies do not focus on attack imperceptibility. The attack proposed by Chen et al. [107] achieves attack dynamicity, as its trigger is dynamically optimized during the data poisoning process based on gradients from the server. However, attack dynamicity is not focused in other studies. Additionally, since these attacks are in their initial stage, they do not account for backdoor durability. All five studies require additional computations beyond data poisoning, resulting in moderate efficiency.

(2) Model Poisoning:

Gu et al. [108] proposed a Latent Representations-based Backdoor Attack (LR-BA). LR-BA assumes a powerful adversary who possesses an auxiliary dataset that shares the same feature distribution and label space as the genuine training dataset. Meanwhile, the adversary can compromise only one client. LR-BA is a post-training attack, occurring after the completion of the VFL protocol. Initially, the adversary uses the malicious client's local model to obtain the embeddings of the auxiliary dataset. These embeddings are then used to train a classifier capable of accurately predicting the class of any unknown embeddings. Subsequently, the adversary optimizes a backdoored embedding targeting a predefined label using the trained classifier. Finally, the malicious client poisons its local samples using a specified trigger and fine-tunes its local model with the poisoned samples and the backdoored embedding, thereby implanting a backdoor into the local model.

Discussion: Experimental results show that LR-BA can effectively withstand Norm Clipping and Gradient Compression [115]. However, LR-BA does not achieve attack imperceptibility or attack dynamicity, and its backdoor durability remains unexplored. Since LR-BA requires data poisoning and model fine-tuning, its efficiency is low. Additionally, Gu et al. [108] pointed out that the attack performance of LR-BA on multi-classification tasks is unstable and its effectiveness heavily relies on the performance of the classifier.

4.4. Review on backdoor-based watermarking methods in FL

We review existing studies on backdoor-based watermarking methods in FL based on the taxonomy proposed earlier. Notably, some studies [65] utilize both backdoor-based and feature-based watermarking techniques. We focus on the former, as the latter falls outside the scope of this paper. For further details on watermarking, please refer to [124].

4.4.1. Server-side watermarking methods

Server-side watermarking methods usually assume that the server initiates FL training and is trusted. In some studies [65,125], the server embeds a backdoor-based watermark into the global model to safeguard its intellectual property rights. In other studies [126], the server incorporates a distinct watermark into the global model distributed to each client, enabling the identification of client-specific models.

Tekgul et al. [125] proposed the first server-side backdoor-based watermarking approach for FL, named WAFFLE. Specifically, the server generates an image trigger set containing random patterns with a noisy background and labels each pattern with a different class. After each aggregation round, the server retrains the global model using this trigger set, thereby embedding these triggers as a watermark into the global model. In addition to protecting model ownership, Yu et al. [126] proposed Decodable Unique Watermarking (DUW) to locate the infringer of a leaked model. In this method, the server first pretrains an encoder to generate a unique trigger set for each client. Then, this trigger set is embedded into a randomly chosen dataset, along with client-wise unique keys. The backdoor watermark is embedded into the model through training on this dataset. During verification, the ownership of the model is verified, and the client of a leaked model is traced based on the client-unique key. Additionally, Shao et al. [65] proposed FedTracker, which combines a backdoor-based global watermark with multi-bit parameter-based local watermarks. In FedTracker, the backdoor-based global watermark is controlled by the server side, while the local watermarks are controlled by the client side. FedTracker addresses the issue of catastrophic forgetting of the main task caused by retraining the model on the trigger set in WAFFLE. Specifically, the server first generates a trigger set using the method in WAFFLE. Then, the server employs continual learning to retrain the global model on the trigger set, thereby reducing forgetting of the main task.

Discussion: Server-side and client-side watermarking each have advantages and disadvantages. For server-side watermarking, watermark conflicts are not an issue, because watermarks are embedded into the global model solely by the server. Additionally, server-side watermarks can be used to track the client of a leaked model, as demonstrated by methods like DUW [126]. However, these methods often require retraining the model on a trigger set independent of the training data, which inevitably introduces side effects on the model's normal performance. FedTracker [65] has partially alleviated this issue. Moreover, existing server-side watermarking methods are primarily designed for image classification tasks and have not been extended to other types of tasks. Taking NLP tasks as an example, constructing a text-based trigger set without any knowledge of the training data poses a significant challenge.

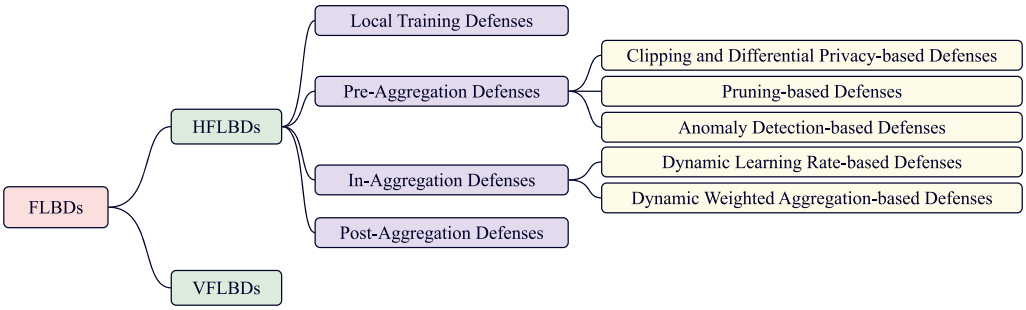

Fig. 12. Taxonomy of FLBDs.

4.4.2. Client-side watermarking methods

For client-side watermarking methods, each client participating in FL training aims to implant its watermark into the global model as proof of its ownership and contribution.

Liu et al. [127] argued that the server is not entirely trusted. Therefore, they proposed a client-side watermarking method. In this method, the client independently generates a noise-based trigger set and embeds the backdoor-based watermark into the model during local training using this trigger set. Building upon this, Yang et al. [128] suggest that the trigger based on random noise could be easily forged by malicious parties. To address this, they designed a non-ambiguous trigger set based on a permutation-based secret key and noise-based patterns to enhance the robustness of the watermark. Similarly, to further improve robustness through the optimization of the trigger set, Nie et al. [129] introduced a scheme called FedCRMW, which constructs trigger sets for watermark embedding using client-specific identifiers and exclusive logos. Additionally, FedIPR proposed by Li et al. [130] aims to mitigate conflicts between watermarks across different clients and enhance the watermark robustness. In FedIPR, clients independently embed both feature-based and backdoor-based watermarks into their local models. In the backdoor-based watermarking, adversarial samples are adopted as triggers, which are generated from original data with the PGD method.

Discussion: Due to the characteristics of FL, client-side watermarking is compatible with most FL security strategies. Moreover, the local knowledge of clients allows them to design robust trigger sets to mitigate the side effects of backdoor-based watermarking on the model, as exemplified by methods such as FedCRMW [129] and FedIPR [130]. However, client-side watermarking faces such challenges as watermark conflicts and difficulties in tracking the client of a leaked model. Additionally, existing backdoor-based watermarking methods solely focus on HFL, while their application in VFL or FTL has not been explored. Given that clients' local knowledge and training processes vary across different FL scenarios, client-side watermarking in VFL and FTL introduces a new set of challenges.

5. FLBD review

In this section, we first present a taxonomy of FLBDs, as illustrated in Fig. 12. Then, based on the taxonomy and evaluation criteria proposed in Section 3.2, we thoroughly review existing FLBDs and assess their pros and cons.

5.1. Taxonomy of FLBDs

Based on the target system, FLBDs can be divided into HFLBDs and VFLBDs. A detailed taxonomy of HFLBDs is provided as follows. VFLBDs are not further classified due to the limited number of studies focused on this topic.

5.1.1. Taxonomy of HFLBDs

HFLBDs can be categorized into local training defenses, pre-aggregation defenses, in-aggregation defenses, and post-aggregation defenses based on the stage at which they operate.

(1) Local Training Defenses: Local training defenses occur during the clients' local training phase, i.e., Step ② in Fig. 3(a). These defenses typically manipulate the local data or the local models' training processes of trusted clients to produce clean and robust local models, ultimately facilitating the creation of clean global models.

(2) Pre-Aggregation Defenses: Pre-aggregation defenses occur after local model updates are uploaded and before model updates are aggregated., i.e., between Step ③ and Step ④ in Fig. 3(a). These defenses usually modify or remove suspicious local model updates. Based on the techniques employed, these defenses can be divided into clipping and differential privacy-based defenses, pruning-based defenses, and anomaly detection-based defenses.

Clipping and Differential Privacy-based Defenses: These defenses encompass two components: constraining the norm of model updates and adding noise to the constrained updates. This approach modifies suspicious model updates to mitigate their impact on the global model.

Pruning-based Defenses: Pruning-based defenses focus on detecting and removing parameters in model updates that are closely associated with backdoors. The pruning technique modifies local model updates to hinder the implantation of backdoors.

Anomaly Detection-based Defenses: These defenses detect and remove suspicious local model updates before aggregation, preventing the implantation of backdoors into the global model.

(3) In-Aggregation Defenses: In-aggregation defenses occur during aggregation of local model updates, i.e., Step ④ in Fig. 3(a). These defenses typically adjust the global model's learning rate or the aggregation strategy to mitigate the impact of potential backdoors. Based on the different objects being adjusted, these defenses can be further divided into dynamic learning rate-based defenses and dynamic weighted aggregation-based defenses.

Dynamic Learning Rate-based Defenses: These defenses dynamically adjust the learning rate distributed by the server in each round, thereby hindering potential backdoor attacks.

Dynamic Weighted Aggregation-based Defenses: These defenses typically assess the suspiciousness of each local model update and assign different aggregation weights to them, mitigating the impact of suspicious local model updates on the global model.

(4) Post-Aggregation Defenses: Post-aggregation defenses occur after the aggregation of local model updates and before the distribution of the global model, i.e., between Step ④ and Step ① in Fig. 3(a). These defenses typically either directly modify the global model or discard the suspicious global model.

5.2. Review on HFLBDs

We review the existing studies on HFLBDs according to the taxonomy proposed above. Table 3 summarizes and compares the reviewed studies on HFLBDs. Fig. 13 briefly illustrates the characteristics of different categories of HFLBDs.

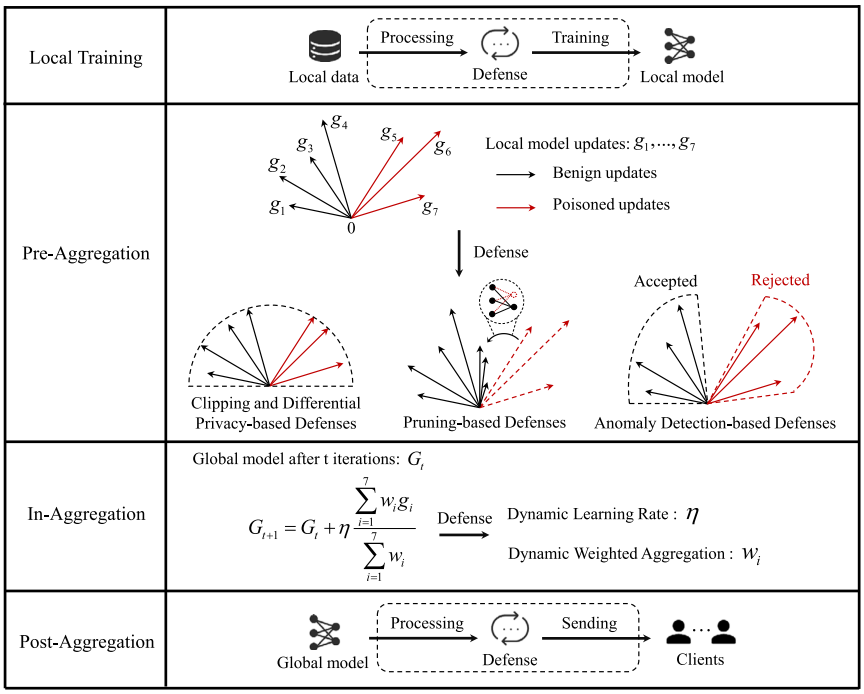

Fig. 13. An illustration of different categories of HFLBDs.

Table 3
Summary and comparison of HFLBDs.

Taxonomy		Ref	Defense Model	Robustness		Practicality		Efficiency	Application
				Attack Resilience	Adaptive Attacks Resilience	Secure Aggregation Compatibility	Unrestricted Poisoned Model Rate		
Local Training		[88]	P-①	MR, DBA	✓	✓	✗	Medium	IC
Pre-Aggregation	Clipping and Differential Privacy	[109]	R	MR	✗	✗	✓	High	IC
	Pruning	[92]	R	MR, DBA, Neur	✓	✗	✗	High	IC
	Anomaly Detection	[69]	P-②	MR	?	✗	✓	Medium	IC, NLP
		[113]	R	MR, DBA, PGD, Edge	✓	✗	✗	Medium	IC, NLP, NIDS
		[94]	R	MR, DBA	✓	✗	✓	Medium	IC
		[96]	R	MR, DBA, Edge	✓	✗	✗	Medium	IC, NLP, NIDS
		[131]	R	MR, Edge	✓	✗	✗	Medium	IC, NLP, NIDS
		[132]	R	EP	✓	✗	✗	Medium	NLP
		[133]	R	MR, Edge	✓	✗	✗	Medium	IC
		[134]	R	MR, DBA, PGD, Edge	?	✗	✗	Medium	IC, NLP, PA
In-Aggregation	Dynamic Learning Rate	[93]	R	DBA	?	✗	✗	High	IC
	Dynamic Weighted Aggregation	[112]	R	MR	✗	✗	✓	Medium	IC, NLP
		[86]	P-②	MR	✓	✗	✓	Medium	IC
		[87]	P-②	MR, DBA, Neur	✓	✗	✓	Medium	IC
		[135]	R	MR, DBA, Edge, Neur	?	✗	✓	Medium	IC
		[136]	R	MR, DBA, Edge, PGD	✓	✗	✓	Medium	IC
Post-Aggregation		[137]	R	MR, Edge	?	✓	✓	Medium	IC, PA
		[138]	R	MR	✓	✓	✗	Low	IC

✓: Satisfied; ✗: Unsatisfied; '?': Not discussed; R: Regular; P: Powerful; ①: Knowledge of trusted clients; ②: An additional dataset; MR: Model Replacement [66]; DBA: Distributed Backdoor Attacks [67]; Edge: Edge-case [68]; PGD: The Projected Gradient Descent attack [68]; Neur: Neurotoxin [90]; EP: Embedding Poisoning [139]; IC: Image Classification; NLP: Natural Language Processing; PA: Predictive Analytics; NIDS: Network Intrusion Detection System.

5.2.1. Local training defenses

Zhang et al. [88] proposed a Federated Learning Provable Defense framework (FLIP), which erases potential backdoors in the global model by adversarially training local models of trusted clients. FLIP assumes a powerful defender who has full knowledge of the server and several trusted clients. During the local training phase, the defender employs a trigger inversion technique [77] to generate a universal trigger for each trusted client and adds this trigger to a random subset of local samples without altering their labels. Subsequently, the trusted clients train robust local models on the modified local data. These robust local models can significantly reduce the impact of backdoors introduced by poisoned models on the global model.

Discussion: Experimental results show that FLIP achieves adaptive attack resilience and effectively defends against the model replacement attack [66] and the DBA [67]. Additionally, it can be compatible with secure aggregation. However, the FLIP requires restricting the number of malicious clients to ensure a sufficient presence of trusted clients. Training a universal trigger for each trusted client results in moderate efficiency for FLIP. Additionally, the performance of FLIP is heavily dependent on the constructed universal trigger, thus it may be ineffective if an adversary employs a trigger that significantly deviates from the universal trigger.

5.2.2. Pre-aggregation defenses

(1) Clipping and Differential Privacy-based Defenses:

To defend against FLBAs based on amplified poisoned model updates [66], Sun et al. [109] proposed a simple but effective defense mechanism. This defense assumes a regular defender. Before the server aggregates local model updates, the defender clips the norms of these updates within a predefined threshold. Subsequently, a small amount of Gaussian noise is added to the clipped model updates based on differential privacy techniques. This approach significantly weakens and disrupts potential poisoned model updates.

Discussion: This defense remains effective regardless of the number of malicious clients and can effectively counter the model replacement attack [66]. Additionally, this defense exhibits high efficiency, leading to it often being integrated as a defensive component within complex defense frameworks [96,113,137]. However, this defense is ineffective against adaptive attacks and is incompatible with secure aggregation. Additionally, setting appropriate thresholds for norm clipping and noise amount is challenging.

(2) Pruning-based Defenses:

Observing that key parameters updated in local models differ between malicious and benign clients, Huang et al. [92] proposed a defense mechanism called Lockdown based on the pruning approach. Lockdown assumes a regular defender. The defender prohibits all clients from updating parameters that contribute less to the main task. Specifically, the defender assigns each client a mask that designates which model parameters can be updated. This mask is dynamically adjusted based on the statistical frequency of parameter updates. Consequently, Lockdown effectively prunes suspicious parameter updates before aggregation, thereby preventing the introduction of backdoors into the global model.

Discussion: Experimental results show that Lockdown can effectively defend against various attacks, including the model replacement attack [66], DBA [67] and Neurotoxin [90], and even adaptive attacks. However, Lockdown is not compatible with secure aggregation. Meanwhile, the performance of Lockdown is significantly affected by the number of malicious clients. Additionally, Lockdown modifies the training strategy of clients, which may introduce new privacy protection concerns and new attack surfaces.

(3) Anomaly Detection-based Defenses:

Research on anomaly detection typically focuses on designing a strategy to effectively separate poisoned and benign model updates. These defenses typically assume a regular defender, except the defense presented in [69].

Li et al. [69] observed that poisoned and benign model updates are represented very differently in low-dimensional latent space. Based on this observation, they proposed an AutoEncoder-based defense mechanism. This defense assumes a powerful defender who knows a clean public dataset. Specifically, the defender first trains a model multiple times on the public dataset to get multiple benign model updates. After that, these model updates are used to train an AutoEncoder [140], in which the encoder encodes model updates into low-dimensional embeddings and the decoder reconstructs the model updates from these embeddings. Subsequently, the defender utilizes the AutoEncoder to reconstruct the local model update of each client. If a reconstructed model update is far away from its original one, this model update should be poisoned.

Thienet et al. [113], Zhang et al. [94], Rieger et al. [96], Kumari et al. [131] and Zhang et al. [132] explored the differences between poisoned and benign model updates across various features. Their work often involves calculating the differences between model updates across the features they introduced and clustering model updates based on these differences to detect and remove anomalous updates. Thienet et al. [113] proposed a defense framework named FLAME, which detects poisoned model updates by measuring angular deviation between model updates. In addition, FLAME introduced adaptive clipping and noising strategies, offering an improvement over the defense based on clipping and differential privacy [109]. Zhang et al. [94] developed FLDetector, a defense mechanism that leverages the consistency of local model updates from a client. This mechanism first predicts each client's model updates using its local historical updates and then detects poisoned updates by analyzing the differences between predicted and actual model updates for each client. Rieger et al. [96] observed that the presence of numerous mislabeled samples in malicious clients causes significant differences in the parameters of benign and poisoned local models, as well as in their prediction vectors for the same samples. Building upon these observations, Rieger et al. [96] proposed DeepSight, which detects poisoned updates by calculating differences between model parameters and differences between prediction vectors of local models for randomly generated samples. Kumari et al. [131] proposed a defense mechanism called BayBFed, which utilizes the probability distribution of local model updates to detect anomalous updates. Zhang et al. [132] observed that, for NLP tasks, the difference in data divergence between poisoned and benign model updates is more significant than their difference in distance. Consequently, they introduced Fed-FA, a defense specifically designed for NLP tasks. This approach employs F-divergence to calculate the differences between each local model update and the global model update, and removes those updates with large F-divergence values.

Previous studies have typically employed a single metric for detecting anomalous model updates, limiting their effectiveness to specific attacks. Kraußet al. [133] and Huang et al. [134] proposed defense mechanisms that integrate multiple detection metrics. Kraußet al. [133] introduced six metrics to measure the distance between local and global model updates, as well as variations in model parameters. Huang et al. [134] introduced three metrics to assess the distance between local and global model updates. The primary difference between these two approaches lies in how to use multiple metrics: the former detects anomalous model updates using each metric individually, while the latter consolidates the three metrics into a unified metric and detects anomalous model updates based on the unified metric.

Discussion: Anomaly detection-based defenses are a crucial part of HFLBDs and have demonstrated good performance across various applications. Additionally, most of them [94,96,113,131–133] are able to bypass adaptive attacks, significantly enhancing their robustness. However, since these defenses typically involve processing and computing model updates, such defenses tend to exhibit moderate efficiency and are incompatible with secure aggregation. Additionally, the studies

by Li et al. [69] and Zhang et al. [94] rely on an additional dataset and historical model updates from clients respectively, rather than on statistics from clients, which allows their defenses to remain effective regardless of the model poisoning rate. In contrast, although other studies [96,113,131–134] do not depend on additional knowledge or historical information, they typically require a poisoned model rate of less than 50%.

5.2.3. In-aggregation defenses

(1) Dynamic Learning Rate-based Defenses:

Inspired by an insight that the updating direction of poisoned and benign models differ in certain dimensions, Ozdayi et al. [93] proposed a defense mechanism that dynamically adjusts the learning rate of the model distributed by the server. This mechanism assumes a regular defender. Specifically, because the signs of benign model updates exhibit high consistency in each dimension, the defender maintains a normal learning rate in the dimensions where the signs of model updates exhibit high consistency. Conversely, as the signs of poisoned model updates typically differ from those of benign model updates, the defender flips the signs of the learning rate in the dimensions where there is a disagreement in signs of model updates. This strategy aims to maximize the loss associated with backdoor learning.

Discussion: This defense mechanism is lightweight and achieves high efficiency. Experimental results demonstrate that it effectively defends against DBA [67]. However, the resilience of this defense against adaptive attacks remains unexplored. Since it adjusts the learning rate based on statistical methods, it becomes ineffective in the presence of a large number of malicious clients. Additionally, this defense is incompatible with secure aggregation.

(2) Dynamic Weighted Aggregation-based Defenses:

The research based on dynamic weighted aggregation seeks to explore a scoring strategy for assessing the suspiciousness of model updates. This strategy assigns different weights to model updates during aggregation based on their respective scores, thereby mitigating the impact of anomalous updates on the global model. These defenses typically assume a regular defender, except the defenses presented in [86] and [87].

Fung et al. [112] argued that poisoned model updates exhibit high similarity, as malicious clients share the same backdoor task. Building upon this idea, they proposed FoolsGold, which assigns the aggregation weights of model updates based on the maximum cosine similarity between updates. Model updates that are more similar to others are assigned smaller weights. Yang et al. [135] proposed a defense mechanism named RoseAgg. To defend against collusion attacks, RoseAgg first aggregates multiple updates that exhibit high similarity into a single update and then employs principal component analysis [141] to extract a benign principal component from the model updates. Subsequently, the aggregation weights of local model updates are assigned based on their projection values onto the clean principal components. The smaller the projection value, the smaller the weight of the local model update. Cao et al. [86] considered that existing robust aggregation algorithms [111,142] are ineffective when there are numerous malicious clients, as they rely solely on local model updates. Therefore, they proposed a defense mechanism based on a root of trust, named FLTrust. FLTrust assumes a powerful defender who has an additional small training dataset. FLTrust assigns the aggregation weight of each local model update based on the difference between the local model update and the benign model update (i.e., the trust root) trained on this dataset. The greater the difference, the smaller the weight of the local model update. Jia et al. [87] proposed a game-theory-based defense mechanism named FedGame, which assigns the aggregation weights of local model updates through a minimax game between an adversary and a defender. FedGame assumes a powerful defender who has an additional small training dataset, and an adversary who is aware of the defender's defense strategy. The adversary estimates the weights of the poisoned models adjusted by the defender using local knowledge and optimizes the backdoor attack strategy to increase the weights of these models. The defender first reverses engineering a trigger and a target class based on the global model and uses them to poison the training dataset. Then, the defender calculates the backdoor accuracy of each local model on the poisoned dataset and adjusts the weights of the local models based on this accuracy. The higher the backdoor accuracy, the smaller the weight of the model. Additionally, Huang et al. [143] introduced a method for evaluating client trustworthiness by analyzing their behavioral information at the classification and feature layers. This trust evaluation is used to assign different aggregation weights to different clients, effectively suppressing the implantation of backdoors.

Unlike previous studies' model training and aggregation processes, Zhang et al. [136] decompose a complete model into an extractor and a classifier, which are trained and aggregated independently. Building on this, they introduce a backdoor defense framework called FLPurifier, designed to disrupt the strong correlation between the trigger features and the target label. During the local training phase, each client first trains the extractor on label-removed samples via supervised contrastive learning [144] and then retrains the entire model on labeled samples. In the aggregation phase, the server performs average aggregation on the clients' extractors and applies weighted aggregation to the clients' classifiers, as the extractors are clean. The server then adjusts the aggregation weights based on the difference between each local classifier and an average classifier representing the mean of all classifiers. The larger the difference, the smaller the weight of the classifier.

Discussion: The studies discussed above can achieve sound defense performance regardless of the poisoned model rate. FLTrust [86], FedGame [87], and FLPurifier [136] have been experimentally shown to be effective against adaptive attacks, significantly enhancing their robustness. However, FoolGolds is ineffective against such attacks and the resilience of RoseAgg against adaptive attacks remains unexplored. Additionally, these defenses require knowledge of each local model update to adjust the aggregation weights, making them incompatible with secure aggregation. Lastly, these studies achieve moderate efficiency, as they require additional computations of the differences between model updates.

5.2.4. Post-aggregation defenses

Xie et al. [137] proposed the first general framework for training certifiably robust FL models against backdoor attacks, named Certifiably Robust Federated Learning (CRFL). CRFL assumes a regular defender and operates during the training and inference phases. During the training phase, the defender clips the norm of the aggregated global model and adds noise to it. During the inference phase, the defender smooths the global model with randomized parameter smoothing and makes predictions based on the smoothed global model. CRFL utilizes clipping and smoothing techniques on model parameters to regulate the smoothness of the global model, thereby providing a sample-wise robustness certification against backdoors with limited magnitude.

Andreina et al. [138] proposed a Backdoor detection via Feedback-based Federated Learning (BaFFLe). BaFFLe assumes a regular defender. BaFFLe introduces a validation phase after each round of aggregation. During this phase, the server sends the current global model and the previous global models to each client. After receiving these global models, the clients calculate the differences in misclassification rates between each pair of global models on local data. Based on these differences, the clients justify whether the current global model has been compromised and report results to the server. Based on the feedback from the clients, the server determines whether to accept the current global model by statistical methods.

Discussion: The studies discussed above occur after aggregation, thus they are compatible with secure aggregation. CRFL [137] has been demonstrated to be effective regardless of the poisoned model rate, resulting in superior practicality. However, although CRFL provides a

robustness certification against backdoors with limited magnitude, its resilience to attacks using triggers exceeding the threshold and adaptive attacks has yet to be investigated. Additionally, CRFL achieves moderate efficiency due to the additional computations required during the inference phase. BaFFLe [138] can defend against adaptive attacks but requires a poisoned model rate of less than 50%. Additionally, BaFFLe requires extra computation and communication during the validation phase, resulting in low efficiency.

5.3. Review on VFLBDs

Currently, there is little research on VFLBDs. Most of them were proposed in studies of VFLBAs to evaluate the robustness of the attacks.

Liu et al. [89] introduced three defense mechanisms to defend against the gradient replacement attack that is an AggVFLBA. The first defense adds additional training layers to the server's model to prevent the leakage of label information. The second defense employs differential privacy to mitigate the potential attack impact by introducing noise into the gradients. The third defense is gradient compression, which restricts the server to send only gradients with significant magnitudes to the clients, thereby preventing the leakage of label information.

Zou et al. [117] also proposed a defense mechanism against the gradient replacement attack [89]. The defense is based on a label disguise technique, termed Confusional AutoEncoder (CAE). CAE consists of an encoder and a decoder, where the encoder takes the true labels as input and outputs fake labels, while the decoder takes fake labels as input and restores the original true label. During the VFL training phase, the active party uses the fake labels generated by the encoder and collaborates with the passive party for training. In the VFL inference phase, the active party transforms the predicted labels back using the trained decoder. CAE can effectively prevent label information leakage and defend against the gradient replacement attack by confusing the true gradients and labels.

Bai et al. [84] and Naseri et al. [106] applied backdoor defenses in centralized learning to the SplitVFL context, including Neural Cleanse [77], Model Pruning [80], Adversarial Neuron Pruning [145], Sample Preprocessing Defense [146], and Anti-Backdoor Learning [83]. Additionally, Bai et al. [84] developed an adaptive defense against VILLAIN they proposed, which neutralizes the unknown trigger by convolutional operations. Naseri et al. [106] performed anomaly detection over the feature embeddings of each class to detect backdoors. However, experimental results show that none of these defenses effectively countered their proposed attack.

He et al. [85] designed two anomaly detection methods to evaluate the robustness of their proposed attack. The first method is to filter out local embeddings that exhibit anomalous in the distributions of length and the element values. The second method is based on reverse engineering. This method constructs reversed triggers for each class. Then, it performs anomaly detection for the reversed triggers. If one or more triggers deviate from the distribution of other reversed triggers, the model should be infected.

Discussion: The defenses discussed above can be broadly categorized into two types: those targeting intermediate computation results, such as embeddings and gradients, and those focusing on the model itself. However, these defenses demonstrate limited effectiveness in countering existing VFLBAs, which may be attributed to the following factors. First, in VFL, since both training and inference rely heavily on intermediate computation results from parties, defenses targeting these intermediate results often involve a trade-off between model performance and defense effectiveness. Second, defenses focusing on the model are typically adapted from those proposed for centralized learning. However, due to the split nature of models in VFL, the effectiveness of these defenses may be significantly compromised. Additionally, while CAE [117] is suitable for the VFL setting and avoids compromising intermediate results, its effectiveness is limited to defending against the gradient replacement attack [89].

6. Open issues and future research directions

In this section, according to the above literature review on FLBAs and FLBDs, we respectively summarize open issues on these two lines of studies. Subsequently, we propose potential future research directions by analyzing the underlying causes of these issues.

6.1. FLBAs

6.1.1. Attack practicality

(1) Open Issues: Existing research has paid limited attention to the practicality of attacks, such as the attack dynamicity and the backdoor durability. Dynamic attacks aim to dynamically execute optimal attack strategies based on the global model's state, and attacks with durable backdoors seek to maintain stable attack performance even after the attack has ceased. Achieving both criteria in an attack typically results in superior practicality. Given that most current studies achieve strong attack performance, these two criteria related to attack practicality become especially significant. However, among the studies reviewed, only one [95] achieves both attack dynamicity and backdoor durability. The reason these criteria are often overlooked may stem from the assumption that the attack scenarios in existing research are generalized, overlooking extreme attack conditions — such as the adversary having a limited number of attack attempts and the server deploying various dynamic defense mechanisms. Yet, such extreme attack scenarios are still possible in real-world environments.

(2) Future Research Directions: As a result, developing FLBAs with attack dynamicity and backdoor durability represents a significant direction in future research. Regarding attack dynamicity, Future research could utilize techniques such as reinforcement learning or game theory to dynamically adjust attack strategies. Additionally, future research could focus on developing techniques that infer potential defense mechanisms based on dynamic changes in the global model. This approach would enable malicious clients to strategically employ attack strategies to effectively bypass these defenses. As for backdoor durability, future work should explore the underlying factors affecting backdoor durability and utilize these insights to guide the development of effective and stable FLBAs. Moreover, in the context of FLBD, dynamic defenses and promoting backdoor unlearning could offer a novel perspective for mitigating the impact of advanced attacks.

6.1.2. Efficiency and robustness

(1) Open Issues: The trade-off between efficiency and robustness presents a significant challenge. A review of existing studies indicates that highly robust attacks often suffer from low efficiency. This is primarily because achieving strong robustness typically requires considerable time and resource overhead, such as minimizing the differences between poisoned and benign model updates across various features [66,99,100] or integrating multiple defense evasion strategies [102]. In contrast, some simple yet effective attacks [67,68] exhibit high efficiency but may lack robustness. Currently, there is no flexible strategy that balances efficiency and robustness.

(2) Future Research Directions: There is an urgent need to explore strategies that can effectively balance efficiency and robustness. A practical approach for balancing efficiency and robustness is to adopt appropriate attack strategies according to the actual local environment. Specifically, malicious clients with abundant computational resources (e.g., large institutions or enterprises) are capable of launching robust and complex backdoor attacks. Conversely, lightweight attacks are mostly adopted by resource-constrained malicious clients. Furthermore, collusion attacks present another viable strategy: when a large number of malicious clients are compromised, their coordinated efforts can significantly enhance both the efficiency and robustness of attacks.

6.1.3. Attack imperceptibility

(1) Open Issues: Attack imperceptibility is often overlooked in VFLBAs. To date, only one study [85] has achieved attack imperceptibility in VFLBAs. In VFL, backdoor attacks may cause malicious clients to generate significantly anomalous intermediate representations due to manipulated sample features or models. Without proper constrained, these anomalies could lead to the detection of malicious clients, resulting in attack failure. Therefore, ensuring imperceptibility in VFLBAs is of paramount importance.

(2) Future Research Directions: In VFL, the intermediate representations generated by clients evolve dynamically during training. Therefore, imperceptible attacks can be achieved by dynamically constraining poisoned intermediate representations. For instance, an adversary could restrict the norm of the poisoned representations to fall within the median range of all benign intermediate representations, effectively hiding them among benign ones. Furthermore, triggers could be adaptively adjusted based on intermediate representations to enhance both the effectiveness and imperceptibility of the attack. Additionally, increasing the diversity of poisoned intermediate representations could further improve attack imperceptibility.

6.1.4. Backdoor attacks on FTL

(1) Open Issues: There is a research gap regarding backdoor attacks on FTL. In FTL, datasets from different participants have neither the same sample IDs nor common feature spaces. Therefore, malicious participants encounter significant challenges in obtaining sufficient knowledge to execute backdoor attacks in FTL. Consequently, exploring how to execute backdoor attacks in FTL represents a novel open issue. Additionally, research on FLBAs across various applications, such as natural language processing and speech recognition, remains limited.

(2) Future Research Directions: Investigating how to implement backdoor attacks in FTL represents a novel research topic in the future. Future studies could draw insights from attack strategies in HFL and VFL, and explore their applications in FTL. Furthermore, researching FLBAs for different applications is a beneficial direction for future studies. Researchers could leverage techniques used for backdoor attacks in centralized learning for different applications to redesign approaches suitable for FL.

6.2. FLBDs

6.2.1. Secure aggregation compatibility

(1) Open Issues: None of the existing pre-aggregation and in-aggregation defense mechanisms are compatible with security aggregation techniques. Secure aggregation enhances the privacy of FL systems by encrypting local model updates, effectively defending against privacy threats such as member inference attacks [147] and model inversion attacks [148]. However, current pre-aggregation and in-aggregation defense mechanisms rely on statistical analysis or modification of local model updates in plaintext, which renders them incompatible with secure aggregation.

(2) Future Research Directions: Given that both robustness and privacy are crucial for FL systems, it is imperative to develop FLBDs that are compatible with secure aggregation. One intuitive approach to achieve this goal is to develop local training or post-aggregation defense mechanisms that avoid analyzing or modifying local model updates. Additionally, future research should focus on developing Privacy-Preserving Federated Learning (PPFL) frameworks that are compatible with FLBDs. For instance, Ma et al. [149] proposed a privacy-preserving defense strategy called ShieldFL, which utilizes two-trapdoor homomorphic encryption to resist encrypted model poisoning without compromising privacy in PPFL.

6.2.2. Restriction on the poisoned model rate

(1) Open Issues: Most existing HFLBDs are only effective at low poisoned model rates, limiting their applicability in practical scenarios. This limitation arises because these defense mechanisms rely on statistical methods to identify poisoned model updates, rendering them ineffective at high poisoned model rates.

(2) Future Research Directions: Developing defense mechanisms that are independent of the poisoned model rate is a promising research direction. Future research could focus on identifying poisoned model updates by analyzing the historical model updates of each client, thereby avoiding the requirement for statistical analyses of all model updates. For example, the direction or magnitude of a client's historical model updates could be monitored. A model update that deviates significantly from historical updates in either direction or magnitude may be indicative of an anomaly. Additionally, future research could focus on eliminating potential backdoors in the global model by modifying the global model itself, rather than concentrating on local model updates. Defense mechanisms based on backdoor removal in centralized learning may provide valuable insights for this approach, such as Neural Attention Distillation [81] and Anti-Backdoor Learning [83].

6.2.3. VFLBDs

(1) Open Issues: There is a lack of extensive research on backdoor defense mechanisms specifically tailored for VFL. The existing VFLBDs are neither universal nor effective, rendering them susceptible to the current VFLBAs. Consequently, there is an urgent need for further investigation to develop robust and practical backdoor defenses specifically tailored for VFL. Additionally, due to differing defense strategies employed by VFLBDs and HFLBDs, the evaluation criteria for VFLBDs require further development.

(2) Future Research Directions: Research on VFLBDs is becoming an urgent topic. Previous studies have adapted backdoor defense mechanisms from centralized learning and HFL to VFL. However, their defense performance remains significantly limited. Therefore, there is a critical need for effective defense mechanisms specifically tailored for VFL. For instance, VFLBDs could employ anomaly detection techniques to detect and remove poisoned embeddings. Furthermore, future research could explore the application of backdoor defense strategies on HFLBDs in VFL, including robust aggregation, pruning, and certified robustness.

6.2.4. Attack resilience

(1) Open Issues: Existing research on FLBDs primarily focuses on defending against fixed-trigger and static backdoor attacks [66–68,68], while neglecting the evaluation of defense performance against trigger-optimization attacks [99–101] and dynamic backdoor attacks [95,102]. In addition, the adaptive attacks assumed in existing FLBDs do not encompass trigger-optimization attacks and dynamic attacks. These advanced attacks exhibit strong effectiveness and robustness and have been widely proposed in recent years. The resilience of FLBDs to existing advanced attacks requires further investigation.

(2) Future Research Directions: Future studies should focus on effectively countering advanced attacks such as trigger-optimization attacks and dynamic backdoor attacks. Future research could draw insights from these attacks to inform the design of resilient defense mechanisms. For instance, a prerequisite for an adversary to launch a dynamic backdoor attack is having full knowledge of the global model's dynamic changes. Therefore, a possible defense strategy is to restrict the client's access to non-essential parameters of the global model during training, thereby preventing the adversary from obtaining critical dynamic information.

6.2.5. Defense mechanisms for practical applications

(1) Open Issues: Although existing FLBDs have demonstrated effectiveness against FLBAs, their deployment in practical applications remains uncertain. Practical FL applications, e.g., in wireless communications and social networks, are vulnerable to a wide range of attacks, such as adversarial examples [150,151], poisoning attacks [152,153], and privacy attacks [154,155]. In such scenarios, the defender faces significant challenges, as specific attack strategies are often unknown in advance, and deploying tailored defense mechanisms for every possible attack is impossible due to resource constraints. Consequently, there is an urgent need to deploy general and effective defense mechanisms.

(2) Future Research Directions: Trust evaluation, which quantifies the trustworthiness of an entity by considering trust influencing factors [156,157], may provide a general security solution for practical FL applications. Currently, a wide variety of trust evaluation algorithms have been proposed, demonstrating the ability to accurately assess the trustworthiness of clients in FL systems [158]. By leveraging these algorithms, clients with low trustworthiness can be identified and excluded, thereby mitigating potential security threats. Furthermore, the performance of trust evaluation algorithms in defending against specific attacks can be enhanced by incorporating additional factors that influence trustworthiness. For instance, the direction of model updates provided by clients is a critical factor in detecting backdoor attacks. Integrating this factor into trust evaluation algorithms can significantly improve their effectiveness in defending against backdoor attacks. Therefore, future research could focus on identifying key factors for detecting attacks and integrating them into trust evaluation algorithms to develop general and effective defense solutions for practical FL applications. Trust evaluation on the local models produced by the clients can also help in generating the global model in a trustworthy way.

7. Conclusion

In this paper, we conducted a comprehensive survey on current FLBAs and FLBDs. First, we introduced the basic knowledge related to FL, backdoor attacks, and defense mechanisms, as well as the threat and defense models for FLBAs and FLBDs, respectively. Then, we proposed two sets of evaluation criteria to evaluate the performance of FLBAs and FLBDs, respectively. Subsequently, we proposed taxonomies of FLBAs and FLBDs from different perspectives, respectively. By employing our proposed criteria and taxonomies, we thoroughly reviewed existing studies. Additionally, we discussed backdoor-based watermarking methods in FL. Finally, according to the review, we delved into several open issues and further indicated future research directions to promote the development of trustworthy FL.

CRediT authorship contribution statement

Zhaozheng Li: Writing – original draft. **Jiahe Lan:** Writing – review & editing, Formal analysis. **Zheng Yan:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Erol Gelenbe:** Writing – review & editing.

Declaration of competing interest

Regarding this paper, there is no any conflict of interest to declare.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants U23A20300; in part by the Key Research Project of Shaanxi Natural Science Foundation under Grant 2023-JC-ZD-35; in part by the Concept Verification Funding of Hangzhou Institute of Technology of Xidian University under Grant GNYZ2024XX007; and in part by the China 111 Project under Grant B16037.

Data availability

No data was used for the research described in the article.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2009, pp. 248–255.
[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252.
[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: Proceedings of the International Conference on Learning Representations, ICLR, 2021.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017.
[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv:1810.04805.
[6] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., LLaMA: Open and efficient foundation language models, 2023, arXiv:2302.13971.
[7] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2021.
[8] Y. Zhang, J. Yan, Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting, in: Proceedings of the International Conference on Learning Representations, ICLR, 2023.
[9] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, Itransformer: Inverted transformers are effective for time series forecasting, in: Proceedings of the International Conference on Learning Representations, ICLR, 2024.
[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[11] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, D. Niyato, H.V. Poor, Federated learning for industrial internet of things in future industries, IEEE Wirel. Commun. 28 (2021) 192–199.
[12] G. Li, J. Wu, S. Li, W. Yang, C. Li, Multitentacle federated learning over software-defined industrial internet of things against adaptive poisoning attacks, IEEE Trans. Ind. Informatics 19 (2022) 1260–1269.
[13] Q. Wu, X. Chen, Z. Zhou, J. Zhang, FedHome: Cloud-edge based personalized federated learning for in-home health monitoring, IEEE Trans. Mob. Comput. 21 (2022) 2818–2832.
[14] M. Hao, H. Li, G. Xu, Z. Liu, Z. Chen, Privacy-aware and resource-saving collaborative learning for healthcare in cloud computing, in: Proceedings of the IEEE International Conference on Communications, ICC, 2020, pp. 1–6.
[15] W. Zhang, T. Zhou, Q. Lu, X. Wang, C. Zhu, H. Sun, Z. Wang, S.K. Lo, F.-Y. Wang, Dynamic-fusion-based federated learning for COVID-19 detection, IEEE Internet Things J. 8 (2021) 15884–15891.
[16] E. Gelenbe, B.C. Gul, M. Nakip, DISFIDA: Distributed self-supervised federated intrusion detection algorithm with online learning for health internet of things and internet of vehicles, Internet Things 28 (2024) 101340.
[17] G. Zhu, Y. Wang, K. Huang, Broadband analog aggregation for low-latency federated edge learning, IEEE Trans. Wirel. Commun. 19 (2020) 491–506.
[18] M.M. Amiri, D. Gündüz, Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air, Proc. the IEEE Int. Symp. Inf. Theory (ISIT) 68 (2019) 1432–1436.
[19] T.T. Vu, D.T. Ngo, N.H. Tran, H.Q. Ngo, M.N. Dao, R.H. Middleton, Cell-free massive MIMO for wireless federated learning, IEEE Trans. Wirel. Commun. 19 (2020) 6377–6392.
[20] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, S.Y. Philip, Privacy and robustness in federated learning: Attacks and defenses, IEEE Trans. Neural Networks Learn. Syst. 35 (2022) 8726–8746.
[21] A. Qammar, J. Ding, H. Ning, Federated learning attack surface: Taxonomy, cyber defences, challenges, and future directions, Artif. Intell. Rev. 55 (2022) 3569–3606.
[22] N. Rodríguez-Barroso, D. Jiménez-López, M.V. Luzón, F. Herrera, E. Martínez-Cámara, Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges, Inf. Fusion 90 (2023) 148–173.
[23] K.N. Kumar, C.K. Mohan, L.R. Cenkeramaddi, The impact of adversarial attacks on federated learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 46 (2024) 2672–2691.
[24] X. Gong, Y. Chen, Q. Wang, W. Kong, Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions, IEEE Wirel. Commun. 30 (2023) 114–121.

[25] Q. Chen, Y. Tao, An investigation of recent backdoor attacks and defenses in federated learning, in: Proceedings of the IEEE International Conference on Fog and Mobile Edge Computing, FMEC, 2023, pp. 262–269.

[26] T.D. Nguyen, T. Nguyen, P. Le Nguyen, H.H. Pham, K.D. Doan, K.-S. Wong, Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions, Eng. Appl. Artif. Intell. 127 (2024) 107166.

[27] Y. Wan, Y. Qu, W. Ni, Y. Xiang, L. Gao, E. Hossain, Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey, IEEE Commun. Surv. & Tutorials 26 (2024) 1861–1897.

[28] W. Jeong, J. Yoon, E. Yang, S.J. Hwang, Federated semi-supervised learning with inter-client consistency & disjoint learning, in: Proceedings of the International Conference on Learning Representations, ICLR, 2021.

[29] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2017,

[30] G. Zhu, Y. Wang, K. Huang, Broadband analog aggregation for low-latency federated edge learning, IEEE Trans. Wirel. Commun. 19 (2019) 491–506.

[31] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. & Tutorials 19 (2017) 2322–2358.

[32] H. Jin, Y. Peng, W. Yang, S. Wang, Z. Zhang, Federated reinforcement learning with environment heterogeneity, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2022, pp. 18–37.

[33] F. Chen, M. Luo, Z. Dong, Z. Li, X. He, Federated meta-learning with fast convergence and efficient communication, 2018, arXiv preprint arXiv:1802.07876.

[34] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, Q. Yang, Vertical federated learning: Concepts, advances, and challenges, IEEE Trans. Knowl. Data Eng. 36 (2024) 3615–3634.

[35] M. Chen, Z. Yang, W. Saad, C. Yin, H.V. Poor, S. Cui, A joint learning and communications framework for federated learning over wireless networks, IEEE Trans. Wirel. Commun. 20 (2021) 269–283.

[36] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, VerifyNet: Secure and verifiable federated learning, IEEE Trans. Inf. Forensics Secur. 15 (2020) 911–926.

[37] C. Thapa, P.C.M. Arachchige, S. Camtepe, L. Sun, SplitFed: When federated learning meets split learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 8485–8493.

[38] T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, Badnets: Evaluating backdooring attacks on deep neural networks, IEEE Access 7 (2019) 47230–47244.

[39] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, 2017, arXiv:1712.05526.

[40] S. Li, M. Xue, B.Z.H. Zhao, H. Zhu, X. Zhang, Invisible backdoor attacks on deep neural networks via steganography and regularization, IEEE Trans. Dependable Secur. Comput. 18 (2021) 2088–2105.

[41] Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Invisible backdoor attack with sample-specific triggers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 16463–16472.

[42] A. Turner, D. Tsipras, A. Madry, Label-consistent backdoor attacks, 2019, arXiv:1912.02771.

[43] A. Saha, A. Subramanya, H. Pirsiavash, Hidden trigger backdoor attacks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 11957–11965.

[44] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks, in: Proceedings of the Annual Network and Distributed System Security Symposium, NDSS, 2018.

[45] K. Doan, Y. Lao, W. Zhao, P. Li, LIRA: Learnable, imperceptible and robust backdoor attacks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, CVPR, 2021, pp. 11966–11976.

[46] K. Doan, Y. Lao, P. Li, Backdoor attack with imperceptible input and latent modification, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2021, pp. 18944–18957.

[47] J. Dumford, W. Scheirer, Backdooring convolutional neural networks via targeted weight perturbations, in: Proceedings of the IEEE International Joint Conference on Biometrics, IJCB, 2020, pp. 1–9.

[48] A.S. Rakin, Z. He, D. Fan, TBT: Targeted neural network attack with bit trojan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 13198–13207.

[49] H. Chen, C. Fu, J. Zhao, F. Koushanfar, ProFlip: Targeted trojan attack with progressive bit flips, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 7718–7727.

[50] J. Dai, C. Chen, Y. Li, A backdoor attack against LSTM-based text classification systems, IEEE Access 7 (2019) 138872–138878.

[51] W. Yang, Y. Lin, P. Li, J. Zhou, X. Sun, Rethinking stealthiness of backdoor attack against NLP models, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, ACL-IJCNLP, 2021, pp. 5543–5557.

[52] J. Yan, V. Gupta, X. Ren, BITE: Textual backdoor attacks with iterative trigger injection, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2023, pp. 12951–12968.

[53] S. Koffas, J. Xu, M. Conti, S. Picek, Can you hear it? Backdoor attacks via ultrasonic triggers, in: Proceedings of the ACM Workshop on Wireless Security and Machine Learning, 2022, pp. 57–62.

[54] W. Zong, Y.-W. Chow, W. Susilo, K. Do, S. Venkatesh, TrojanModel: A practical trojan attack against automatic speech recognition systems, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2023, pp. 1667–1683.

[55] J. Lan, J. Wang, B. Yan, Z. Yan, E. Bertino, FlowMur: A stealthy and practical audio backdoor attack with limited knowledge, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2024, p. 148.

[56] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, Y.-G. Jiang, Clean-label backdoor attacks on video recognition models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 14443–14452.

[57] H.A. Al Kader Hammoud, S. Liu, M. Alkhrashi, F. Albalawi, B. Ghanem, Look listen and attack: Backdoor attacks against video action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 3439–3450.

[58] X. Gong, Z. Fang, B. Li, T. Wang, Y. Chen, Q. Wang, Palette: Physically-realizable backdoor attacks against video recognition models, IEEE Trans. Dependable Secur. Comput. 21 (2024) 2672–2685.

[59] Z. Yan, J. Wu, G. Li, S. Li, M. Guizani, Deep neural backdoor in semi-supervised learning: Threats and countermeasures, IEEE Trans. Inf. Forensics Secur. 16 (2021) 4827–4842.

[60] Z. Yan, G. Li, Y. TIan, J. Wu, S. Li, M. Chen, H.V. Poor, Dehib: Deep hidden backdoor attack on semi-supervised learning via adversarial perturbation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 10585–10593.

[61] H. Zhu, S. Zhang, K. Chen, Ai-guardian: Defeating adversarial attacks using backdoors, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2023, pp. 701–718.

[62] Y.-S. Lin, W.-C. Lee, Z.B. Celik, What do you see? Evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1027–1035.

[63] Y. Li, M. Zhu, X. Yang, Y. Jiang, T. Wei, S.-T. Xia, Black-box dataset ownership verification via backdoor watermarking, IEEE Trans. Inf. Forensics Secur. 18 (2023) 2318–2332.

[64] R. Tang, Q. Feng, N. Liu, F. Yang, X. Hu, Did you train on my dataset? Towards public dataset protection with cleanlabel backdoor watermarking, ACM SIGKDD Explor. Newsl. 25 (2023) 43–53.

[65] S. Shao, W. Yang, H. Gu, Z. Qin, L. Fan, Q. Yang, Fedtracker: Furnishing ownership verification and traceability for federated learning model, IEEE Trans. Dependable Secur. Comput. 22 (2025) 114–131.

[66] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2020, pp. 2938–2948.

[67] C. Xie, K. Huang, P.-Y. Chen, B. Li, DBA: Distributed backdoor attacks against federated learning, in: Proceedings of the International Conference on Learning Representations, ICLR, 2020.

[68] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, D. Papailiopoulos, Attack of the tails: Yes, you really can backdoor federated learning, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2020, pp. 16070–16084.

[69] S. Li, Y. Cheng, W. Wang, Y. Liu, T. Chen, Learning to detect malicious clients for robust federated learning, 2020, arXiv:2002.00211.

[70] Y. Li, X. Lyu, X. Ma, N. Koren, L. Lyu, B. Li, Y.-G. Jiang, Reconstructive neuron pruning for backdoor defense, in: Proceedings of the International Conference on Machine Learning, ICML, 2023, pp. 19837–19854.

[71] Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang, J. Ma, Backdoor defense with machine unlearning, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, 2022, pp. 280–289.

[72] Z. Yan, S. Li, R. Zhao, Y. Tian, Y. Zhao, DHBE: Data-free holistic backdoor erasing in deep neural networks via restricted adversarial distillation, in: Proceedings of the ACM Asia Conference on Computer and Communications Security, 2023, pp. 731–745.

[73] B. Tran, J. Li, A. Madry, Spectral signatures in backdoor attacks, 2018,

[74] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, B. Srivastava, Detecting backdoor attacks on deep neural networks by activation clustering, 2018, arXiv:1811.03728.

[75] Y. Liu, G. Shen, G. Tao, Z. Wang, S. Ma, X. Zhang, Complex backdoor detection by symmetric feature differencing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 15003–15013.

[76] Y. Gao, C. Xu, D. Wang, S. Chen, D.C. Ranasinghe, S. Nepal, Strip: A defence against trojan attacks on deep neural networks, in: Proceedings of the Annual Computer Security Applications Conference, 2019, pp. 113–125.

[77] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B.Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2019, pp. 707–723.

[78] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, X. Zhang, ABS: Scanning neural networks for back-doors by artificial brain stimulation, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2019, pp. 1265–1282.

[79] X. Hu, X. Lin, M. Cogswell, Y. Yao, S. Jha, C. Chen, Trigger hunting with a topological prior for trojan detection, in: Proceedings of the International Conference on Learning Representations, ICLR, 2022.

[80] K. Liu, B. Dolan-Gavitt, S. Garg, Fine-pruning: Defending against backdooring attacks on deep neural networks, in: International Symposium on Research in Attacks, Intrusions, and Defenses, 2018, pp. 273–294.

[81] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Neural attention distillation: Erasing backdoor triggers from deep neural networks, in: Proceedings of the International Conference on Learning Representations, ICLR, 2021.

[82] Y. Zeng, S. Chen, W. Park, Z. Mao, M. Jin, R. Jia, Adversarial unlearning of backdoors via implicit hypergradient, in: Proceedings of the International Conference on Learning Representations, ICLR, 2021.

[83] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Anti-backdoor learning: Training clean models on poisoned data, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2021, pp. 14900–14912.

[84] Y. Bai, Y. Chen, H. Zhang, W. Xu, H. Weng, D. Goodman, VILLAIN: Backdoor attacks against vertical split learning, in: Proceedings of the USENIX Security Symposium, 2023, pp. 2743–2760.

[85] Y. He, Z. Shen, J. Hua, Q. Dong, J. Niu, W. Tong, X. Huang, C. Li, S. Zhong, Backdoor attack against split neural network-based vertical federated learning, IEEE Trans. Inf. Forensics Secur. 19 (2024) 748–763.

[86] X. Cao, M. Fang, J. Liu, N.Z. Gong, Fltrust: Byzantine-robust federated learning via trust bootstrapping, in: Proceedings of the Annual Network and Distributed System Security Symposium, NDSS, 2021.

[87] J. Jia, Z. Yuan, D. Sahabandu, L. Niu, A. Rajabi, B. Ramasubramanian, B. Li, R. Poovendran, FedGame: A game-theoretic defense against backdoor attacks in federated learning, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2023, pp. 53090–53111.

[88] K. Zhang, G. Tao, Q. Xu, S. Cheng, S. An, Y. Liu, S. Feng, G. Shen, P.-Y. Chen, S. Ma, et al., FLIP: A provable defense framework for backdoor mitigation in federated learning, in: Proceedings of the International Conference on Learning Representations, ICLR, 2023.

[89] Y. Liu, Z. Yi, T. Chen, Backdoor attacks and defenses in feature-partitioned collaborative learning, 2020, arXiv:2007.03608.

[90] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, J. Gonzalez, Neurotoxin: Durable backdoors in federated learning, in: Proceedings of the International Conference on Machine Learning, ICML, 2022, pp. 26429–26446.

[91] Y. Dai, S. Li, Chameleon: Adapting to peer images for planting durable backdoors in federated learning, in: Proceedings of the International Conference on Machine Learning, ICML, 2023, pp. 6712–6725.

[92] T. Huang, S. Hu, K.-H. Chow, F. Ilhan, S. Tekin, L. Liu, Lockdown: Backdoor defense for federated learning with isolated subspace training, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2023, pp. 10876–10896.

[93] M.S. Ozdayi, M. Kantarcioglu, Y.R. Gel, Defending against backdoors in federated learning with robust learning rate, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 9268–9276.

[94] Z. Zhang, X. Cao, J. Jia, N.Z. Gong, FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2545–2555.

[95] H. Zhang, J. Jia, J. Chen, L. Lin, D. Wu, A3FL: Adversarially adaptive backdoor attacks to federated learning, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2023, pp. 61213–61233.

[96] P. Rieger, T.D. Nguyen, M. Miettinen, A.-R. Sadeghi, DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection, in: Proceedings of the Annual Network and Distributed System Security Symposium, NDSS, 2022.

[97] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2017, pp. 1175–1191.

[98] X. Gong, Y. Chen, H. Huang, Y. Liao, S. Wang, Q. Wang, Coordinated backdoor attacks against federated learning with model-dependent triggers, IEEE Netw. 36 (2022) 84–90.

[99] X. Lyu, Y. Han, W. Wang, J. Liu, B. Wang, J. Liu, X. Zhang, Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 9020–9028.

[100] P. Fang, J. Chen, On the vulnerability of backdoor defenses for federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 11800–11808.

[101] T.D. Nguyen, T.A. Nguyen, A. Tran, K.D. Doan, K.-S. Wong, IBA: Towards irreversible backdoor attacks in federated learning, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2023, pp. 66364–66376.

[102] H. Li, Q. Ye, H. Hu, J. Li, L. Wang, C. Fang, J. Shi, 3Dfed: Adaptive and extensible framework for covert backdoor attack in federated learning, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2023, pp. 1893–1907.

[103] M. Li, W. Wan, Y. Ning, S. Hu, L. Xue, L.Y. Zhang, Y. Wang, DarkFed: A data-free backdoor attack in federated learning, in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2024, pp. 4443–4451.

[104] H. Zhuang, M. Yu, H. Wang, Y. Hua, J. Li, X. Yuan, Backdoor federated learning by poisoning backdoor-critical layers, in: Proceedings of the International Conference on Learning Representations, ICLR, 2024.

[105] Y. Xuan, X. Chen, Z. Zhao, B. Tang, Y. Dong, Practical and general backdoor attacks against vertical federated learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2023, pp. 402–417.

[106] M. Naseri, Y. Han, E. De Cristofaro, BadVFL: Backdoor attacks in vertical federated learning, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2024, p. 12.

[107] P. Chen, J. Yang, J. Lin, Z. Lu, Q. Duan, H. Chai, A practical clean-label backdoor attack with limited information in vertical federated learning, in: Proceedings of the IEEE International Conference on Data Mining, 2023, pp. 41–50.

[108] Y. Gu, Y. Bai, LR-BA: Backdoor attack against vertical federated learning using local latent representations, Comput. Secur. 129 (2023) 103193.

[109] Z. Sun, P. Kairouz, A.T. Suresh, H.B. McMahan, Can you really backdoor federated learning?, 2019, arXiv:1911.07963.

[110] K. Pillutla, S.M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, IEEE Trans. Signal Process. 70 (2022) 1142–1154.

[111] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2017.

[112] C. Fung, C.J. Yoon, I. Beschastnikh, The limitations of federated learning in sybil settings, in: Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses, RAID, 2020, pp. 301–316.

[113] T.D. Nguyen, P. Rieger, R. De Viti, H. Chen, B.B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, et al., FLAME: Taming backdoors in federated learning, in: Proceedings of the USENIX Security Symposium, 2022, pp. 1415–1432.

[114] Y. Wang, D. Zhai, Y. Zhan, Y. Xia, RFLBAT: A robust federated learning algorithm against backdoor attack, 2022, arXiv:2201.03772.

[115] Y. Lin, S. Han, H. Mao, Y. Wang, W. Dally, Deep gradient compression: reducing the communication bandwidth for distributed training, in: Proceedings of the International Conference on Learning Representations, ICLR, 2018.

[116] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2016, pp. 308–318.

[117] T. Zou, Y. Liu, Y. Kang, W. Liu, Y. He, Z. Yi, Q. Yang, Y.-Q. Zhang, Defending batch-level label inference and replacement attacks in vertical federated learning, IEEE Trans. Big Data 10 (2022) 1016–1027.

[118] J. Zhang, B. Chen, X. Cheng, H.T.T. Binh, S. Yu, PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems, IEEE Internet Things J. 8 (2020) 3310–3322.

[119] H. Mei, G. Li, J. Wu, L. Zheng, Privacy inference-empowered stealthy backdoor attack on federated learning under non-iid scenarios, in: Proceedings of the International Joint Conference on Neural Networks, 2023, pp. 1–10.

[120] R. Guerraoui, S. Rouault, et al., The hidden vulnerability of distributed learning in byzantium, in: Proceedings of the International Conference on Machine Learning, ICML, 2018, pp. 3521–3530.

[121] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2020, pp. 18661–18673.

[122] A. Panda, S. Mahloujifar, A.N. Bhagoji, S. Chakraborty, P. Mittal, SparseFed: Mitigating model poisoning attacks in federated learning with sparsification, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2022, pp. 7587–7624.

[123] X. Cao, N.Z. Gong, MPAF: Model poisoning attacks to federated learning based on fake clients, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 3396–3404.

[124] M. Lansari, R. Bellafqira, K. Kapusta, V. Thouvenot, O. Bettan, G. Coatrieux, When federated learning meets watermarking: A comprehensive overview of techniques for intellectual property protection, Mach. Learn. Knowl. Extr. 5 (2023) 1382–1406.

[125] B.G. Tekgul, Y. Xia, S. Marchal, N. Asokan, WAFFLE: Watermarking in federated learning, in: Proceedings of the International Symposium on Reliable Distributed Systems, 2021, pp. 310–320.

[126] S. Yu, J. Hong, Y. Zeng, F. Wang, R. Jia, J. Zhou, Who leaked the model? Tracking IP infringers in accountable federated learning, in: Proceedings of the NeurIPS Workshop on Regulatable ML, 2023.

[127] X. Liu, S. Shao, Y. Yang, K. Wu, W. Yang, H. Fang, Secure federated learning model verification: A client-side backdoor triggered watermarking scheme, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2021, pp. 2414–2419.
[128] W. Yang, S. Shao, Y. Yang, X. Liu, X. Liu, Z. Xia, G. Schaefer, H. Fang, Watermarking in secure federated learning: A verification framework based on client-side backdooring, ACM Trans. Intell. Syst. Technol. 15 (2023) 1–25.
[129] H. Nie, S. Lu, FedCRMW: Federated model ownership verification with compression-resistant model watermarking, Expert Syst. Appl. 249 (2024) 123776.
[130] B. Li, L. Fan, H. Gu, J. Li, Q. Yang, FedIPR: Ownership verification for federated deep neural network models, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2022) 4521–4536.
[131] K. Kumari, P. Rieger, H. Fereidooni, M. Jadliwala, A.-R. Sadeghi, BayBFed: Bayesian backdoor defense for federated learning, in: Proceedings of the IEEE Symposium on Security and Privacy, SP, 2023, pp. 737–754.
[132] Z. Zhang, D. Chen, H. Zhou, F. Meng, J. Zhou, X. Sun, Fed-FA: Theoretically modeling client data divergence for federated language backdoor defense, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2024, pp. 62006–62031.
[133] T. Krauß, A. Dmitrienko, MESAS: Poisoning defense for federated learning resilient against adaptive attackers, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2023, pp. 1526–1540.
[134] S. Huang, Y. Li, C. Chen, L. Shi, Y. Gao, Multi-metrics adaptively identifies backdoors in federated learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2023, pp. 4652–4662.
[135] H. Yang, W. Xi, Y. Shen, C. Wu, J. Zhao, RoseAgg: Robust defense against targeted collusion attacks in federated learning, IEEE Trans. Inf. Forensics Secur. 19 (2024) 2951–2966.
[136] J. Zhang, C. Zhu, X. Sun, C. Ge, B. Chen, W. Susilo, S. Yu, FLPurifier: Backdoor defense in federated learning via decoupled contrastive training, IEEE Trans. Inf. Forensics Secur. 19 (2024) 4752–4766.
[137] C. Xie, M. Chen, P.-Y. Chen, B. Li, CRFL: Certifiably robust federated learning against backdoor attacks, in: Proceedings of the International Conference on Machine Learning, ICML, 2021, pp. 11372–11382.
[138] S. Andreina, G.A. Marson, H. Möllering, G. Karame, BaFFLe: Backdoor detection via feedback-based federated learning, in: Proceedings of the IEEE International Conference on Distributed Computing Systems, ICDCS, 2021, pp. 852–863.
[139] K. Yoo, N. Kwak, Backdoor attacks in federated learning by rare embeddings and gradient ensembling, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP, 2022, pp. 72–88.
[140] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Spec. Lect. IE 2 (2015) 1–18.
[141] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscip. Rev.: Comput. Stat. 2 (2010) 433–459.
[142] D. Yin, Y. Chen, R. Kannan, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: Proceedings of the International Conference on Machine Learning, ICML, 2018, pp. 5650–5659.
[143] W. Huang, G. Li, X. Yi, J. Li, C. Zhao, Y. Yin, SupRTE: Suppressing backdoor injection in federated learning via robust trust evaluation, IEEE Intell. Syst. 39 (2024) 66–77.
[144] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap your own latent-a new approach to self-supervised learning, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2020, pp. 21271–21284.
[145] D. Wu, Y. Wang, Adversarial neuron pruning purifies backdoored deep models, in: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS, 2021, pp. 16913–16925.
[146] Y. Li, T. Zhai, Y. Jiang, Z. Li, S.-T. Xia, Backdoor attack in the physical world, 2021, arXiv:2104.02361.
[147] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P.S. Yu, X. Zhang, Membership inference attacks on machine learning: A survey, ACM Comput. Surv. 54 (2022) 1–37.
[148] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 1322–1333.
[149] Z. Ma, J. Ma, Y. Miao, Y. Li, R.H. Deng, ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning, IEEE Trans. Inf. Forensics Secur. 17 (2022) 1639–1654.
[150] E. Darzi, F. Dubost, N.M. Sijtsema, P.M. van Ooijen, Exploring adversarial attacks in federated learning for medical imaging, IEEE Trans. Ind. Informatics 20 (2024) 13591–13599.
[151] Y. Duanyi, S. Li, X. Ye, J. Liu, Constructing adversarial examples for vertical federated learning: Optimal client corruption through multi-armed bandit, in: Proceedings of the International Conference on Learning Representations, ICLR, 2024.
[152] M. Fang, X. Cao, J. Jia, N. Gong, Local model poisoning attacks to byzantine-robust federated learning, in: Proceedings of the USENIX Security Symposium, 2020, pp. 1605–1622.
[153] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, R.M. Parizi, A robust privacy-preserving federated learning model against model poisoning attacks, IEEE Trans. Inf. Forensics Secur. 19 (2024) 6693–6708.
[154] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, 2019, pp. 2512–2520.
[155] C. Fu, X. Zhang, S. Ji, J. Chen, J. Wu, S. Guo, J. Zhou, A.X. Liu, T. Wang, Label inference attacks against vertical federated learning, in: Proceedings of the USENIX Security Symposium, 2022, pp. 1397–1414.
[156] J. Wang, Z. Yan, H. Wang, T. Li, W. Pedrycz, A survey on trust models in heterogeneous networks, IEEE Commun. Surv. & Tutorials 24 (2022) 2127–2162.
[157] J. Wang, Z. Yan, J. Lan, E. Bertino, W. Pedrycz, TrustGuard: GNN-based robust and explainable trust evaluation with dynamicity support, IEEE Trans. Dependable Secur. Comput. 21 (2024) 4433–4450.
[158] J. Guo, Z. Liu, S. Tian, F. Huang, J. Li, X. Li, K.K. Igorevich, J. Ma, TFL-DT: A trust evaluation scheme for federated learning in digital twin for mobile networks, IEEE J. Sel. Areas Commun. 41 (2023) 3548–3560.