Journal of
*Sensor and Actuator Networks*

MDPI

*Article*

# Building Trusted Federated Learning: Key Technologies and Challenges

**Depeng Chen [1,2], Xiao Jiang [1,2], Hong Zhong [1,*] and Jie Cui [1,2]**

[1] School of Computer Science and Technology, Anhui University, Hefei 230601, China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230026, China
* Correspondence: zhongh@ahu.edu.cn

**Abstract:** Federated learning (FL) provides convenience for cross-domain machine learning applications and has been widely studied. However, the original FL is still vulnerable to poisoning and inference attacks, which will hinder the landing application of FL. Therefore, it is essential to design a trustworthy federation learning (TFL) to eliminate users' anxiety. In this paper, we aim to provide a well-researched picture of the security and privacy issues in FL that can bridge the gap to TFL. Firstly, we define the desired goals and critical requirements of TFL, observe the FL model from the perspective of the adversaries and extrapolate the roles and capabilities of potential adversaries backward. Subsequently, we summarize the current mainstream attack and defense means and analyze the characteristics of the different methods. Based on a priori knowledge, we propose directions for realizing the future of TFL that deserve attention.

**Keywords:** trustworthy federated learning; machine learning; security and defense; privacy protection



**Citation:** Chen, D.; Jiang, X.; Zhong, H.; Cui, J. Building Trusted Federated Learning: Key Technologies and Challenges. *J. Sens. Actuator Netw.* **2023**, *12*, 13. https://doi.org/10.3390/jsan12010013

Academic Editor: Mingjun Xiao

Received: 13 December 2022
Revised: 17 January 2023
Accepted: 18 January 2023
Published: 6 February 2023



## 1. Introduction

Artificial intelligence technology represented by machine learning strongly drives the development of various industries. Machine learning (ML) is a paradigm that learns from past experience and makes accurate predictions about new problems. ML gives machines the ability to learn with little or even without human intervention. In recent years, ML algorithms, represented by deep learning (DL), have achieved great achievement in areas such as image recognition and natural language processing. However, researchers find that high-quality predictive models rely on high-quality training data, yet individuals or groups are often reluctant to contribute data due to privacy concerns, resulting in data silos. At the same time, sufficient attention has been paid to the misuse of privacy, and privacy laws have further regulated access to data.

Federated learning (FL) [1] provides an excellent idea for solving these problems. Unlike distributed machine learning, in federated learning, users update the model rather than their data to obtain a better global model. It assures that data can be utilized without leaving the local area, thus dispelling users' privacy anxiety. Although FL has been partially applied in practice, such as Google, using it to predict the subsequent input of the user's keyboard, we found that the current level of FL is still insufficient to meet its security requirements. Means such as poisoning attacks, and inference attacks, still affect the usability of FL, especially in combination with highly sensitive information areas such as medicine and finance. The large-scale application of FL has been hampered by these problems, and researchers have had to redesign the model to achieve user trustworthiness. Therefore, trustworthy federated learning (TFL) [2], in combination with safety solutions, deserves to be discussed further.

Unlike traditional FL, TFL's goal is to eliminate users' concerns about the security and privacy of the model system and ensure the credibility of the model framework. Typically, researchers choose to use security algorithms [3] or secure architecture [4,5], such

as blockchain technology, to achieve TFL. However, the current research lacks a systematic definition of TFL's requirements. TFL imposes more stringent safety requirements on FL systems, which should have the following basic principles:

- High Confidentiality: Confidentiality is reflected in the fact that malicious adversaries cannot steal sensitive information in FL.
- High Integrity: Integrity is reflected in the fact that private data cannot be maliciously modified without authorization during training.
- High Availability: The model system is required to provide access by authorized users and be used on demand. The model also needs to have a usable accuracy rate as well as efficiency. The cost of trustworthiness cannot be a significant loss of accuracy and a high rate of loss of efficiency.
- Strong Robustness: In addition to following the information security fundamentals, FL should have sufficient resistance in the face of complex scenarios or unknown attacks.
- Provable Security: The security protocols and methods must be rigorously secure based on specific mathematical assumptions.

In response to the above requirements, we survey the current status of FL and look forward to the next more promising development direction of TFL. Analytical work on TFL has been partially studied, but we will look at the threats faced by FL from some new perspectives.

### *1.1. Contributions*

Although similar studies have been conducted to investigate the threat of FL, these efforts still need to provide a more comprehensive summary of existing technologies and a clear indication of future research directions. Our work provides a comprehensive overview of FL, including its definition, threats, and potential future research. This paper will facilitate the construction of usable TFL paradigms and their rapid application to actual production. Our main contributions are as follows:

- We thoroughly investigate the development mapping and critical technologies of FL and meticulously analyze the existing FL research content.
- We assess the threats to FL from an adversary perspective. Furthermore, we summarize mainstream FL-specific attacks from the perspective of security threats and privacy threats.
- We summarise and abstract the approaches to privacy protection in FL and evaluate their strengths and weaknesses. Based on this, we provide some valuable prospects for building TFL.

### *1.2. Paper Organization*

Section 2, we briefly introduce FL and classify the current FL models from different perspectives. In Section 3, we list the primary attack methods that threaten the model's security and privacy and summarize and compare the defense schemes in Section 4. In Section 5, we point out the future promising research directions of TFL. Finally, Section 6 gives the conclusion of our work.

## 2. Federated Learning

Similar to the traditional distributed machine learning paradigm, FL also utilizes the assistance of distributed clients for more complex model training. The target of FL's communication shifts from data to the model. The optimized global model is obtained through the aggregation of multiple local models, which ensures the availability of the global model while making the client's data visible only to itself, eliminating its distrust of the server or external attackers, thus allowing better quality data to participate in the training [6–8]. In addition, FL has likewise been extensively studied for joint learning of heterogeneous data [9,10], which makes it a good prospect for cross-domain machine learning. For example, information from different devices (e.g., images and text) can be used for training at the same time to capture more feature information and provide a

more valuable output model. The work in Ref. [11] uses FL in 6G networks for resource recommendation and scheduling-based propagation analysis.

The traditional FL model is composed of a server and a number of clients. The server aggregates the local model, broadcasts the global model, and ultimately outputs a highly available predictive model. The client obtains the global model, updates it with local data, and finally uploads it to the server. However, this structure requires establishing a mutual trust mechanism between the client and the server, which will pose a severe threat to the system if the server is hijacked or malicious. We will discuss the details in Section 3.

### *2.1. Classification of FL*

FL is in a rapid state of development, and various techniques and methods are being used to enhance its applicability of FL. In order to cope with more complex requirements, FL with different complex morphologies is proposed [7]. Some research works want to apply machine learning models to more significant distributed scenarios, requiring FL to consider communication and data aggregation costs. For example, Ref. [12] focuses on using FL in mobile edge computing. However, some other works require more secure participation in training, which requires FL to pay more attention to security and privacy protection. In ref. [13], the authors focus more on security threats and classify FL in terms of data feature distribution. Other than that, Ref. [14] also classifies FL with technologies from different perspectives. Therefore, classifying FL according to different perspectives is an essential first step in understanding and optimizing FL design.

#### 2.1.1. Centralized/Multi-Center/Decentralized FL

Although FL has been implemented for decentralized training, centralized FL still requires a central server to complete the accusation of aggregation and broadcasting, which we call centralized FL [1,15–18]. The single-server design ensures that the model's rights are centralized in the hands of the server, which helps to manage the whole training process and avoid errors. For example, Google's Gboard for Android keyboard is based on this architecture. However, a centralized server tends to occur a single point of failure, which might destroy the whole FL system. To release this security dependency, a decentralized FL was proposed. It attempts to reduce or even eliminate the server's control over the global model. As demonstrated in Ref. [19], authors proposed two asynchronous peer-to-peer algorithms for the novel setting of decentralized collaborative learning of personalized models. This approach removes the server directly to achieve complete decentralization. However, the time and communication cost of this approach is often huge. Multi-center FL does not require a centralized server, but multiple decentralized edge servers need to manage model updates. This weakens the impact of servers becoming malicious nodes on the global model while ensuring the utility of the model. To solve the above problems, Ref. [20] learns multiple global models from the data, simultaneously derives the best match between users and centers, and proposes an optimization approach, federated SEM, to eliminate the multi-center FL problem effectively.

#### 2.1.2. Horizontal FL/Vertical FL/Federated Transfer Learning

In addition, facing different application scenarios, it is also a common method to design FL according to the characteristics of training data. Depending on the degree of overlap between the feature space and the sample space, FL can be classified into horizontal federated learning (HFL), vertical federated learning (VFL), and federated transfer learning (FTL), respectively [8].

First of all, in business-to-consumer (B-C) FL, clients often use data sets with overlapping features. For example, different banks can provide similar data training models, and their data characteristics are highly coincidental. In this situation, the HFL is more appropriate. The client typically uses stochastic gradient descent ($SGD$) to minimize losses, and the server performs a secure aggregation algorithm (such as FedAvg [1], FedPorx [21]) to obtain a global model. According to different applications, we can further refine HFL

into HFL to businesses (H2B) and HFL to consumers (H2C) [13].

HFL has the advantage of being able to quickly extract features from similar data and obtain a highly credible global model but tends to be weak for data with a little overlap in features. For example, insurance companies rely on banks' credit data to provide customized services, and the two have different feature spaces, so they cannot be trained directly using HFL. Compared with HFL, VFL is relatively more complex but applicable to a wider range of scenarios and has stronger practical value. However, VFL also faces some problems, such as low efficiency, and related work has also been studied in this respect. For example, Zhang et al. [22] designed a new backward updating mechanism and bilevel asynchronous parallel architecture to solve the problem of low efficiency caused by synchronous calculation in practical applications.

It is worth noting that the first two perform well for supervised learning but struggle with weakly supervised or unsupervised data. FTL introduces the idea of migration learning to cope with the need for tiny overlapping samples. For the data with low correlation, the direct use of the first two models for training results in poor effectiveness because the aggregation algorithm is difficult to extract similar effective features. Transfer learning uses the similarities between the target domain and the source domain so that the migration model can learn from the data with big differences. Thus, a federated learning model incorporating transfer learning gains the ability to learn from heterogeneous data [23], which provides a feasible solution for collaborative modeling.

## 3. Threats in FL

In this section, we will introduce the mainstream attacks confronted in FL. Before discussing what threats FL faces, we first introduce a new perspective to analyze where these threats may originate. In the second and third parts, we analyze the possible attacks on FL from the point of view of security and privacy, respectively. The analysis of the source of threats and other sections have been partially studied in previous work [13,14,24–26].

### *3.1. Adversary Status*

To assess the threats to FL, we first need to know what role the adversary can play in the system model. Unlike traditional machine learning, the identity of the adversary in FL is relatively complex. Taking the basic FL as the reference object, the participating entities can be divided into server, clients, and malicious external entities. It is difficult to trace the source of the threat accurately. Assuming they both make malicious attackers, we can distinguish them as internal attackers (Server and Client) and external attackers. At the same time, we cannot ignore the possibility of a collusion attacker.

1. Server: In FL, the server has high privileges. Therefore, once it is malicious, it is an increased threat to the security and utility of the system. Generally, for example, there are attacks such as model poisoning attacks or backdoor attacks. The server can easily break model convergence by poisoning the global gradient. If it is semi-honest, it can also steal the client's privacy through inference or model reversal attacks. Since the server has updated gradient information, membership information can be easily stolen from the client using a gradient change-based membership inference attack. Therefore, it is necessary to let the server obtain as little accurate information as possible based on its duties.
2. Client: On the one hand, the client, as a carrier of private information, is the most vulnerable object of attack, such as membership inference attacks aimed at stealing customers' private information. Homomorphic encryption and differential privacy can be exploited as promising methods to prevent gradient information from leakage. Meanwhile, the introduction of a shuffler mechanism can be used further to mask the user's ID [27]. On the other hand, a malicious client can have a bad impact on model training. It can disrupt the availability of the global model by poisoning [15] or inserting backdoors [28], but unlike the server, the impact of a single malicious client is limited.

3. External attackers: An external attacker aiming at sabotage might hijack the server or bring it down directly, thus completely disrupting the training. External entities that eavesdrop on server and client communication channels also threaten clients' privacy considerably. Homomorphic encryption and differential privacy can limit its access to accurate information, and the combination of the trusted execution environment's FL can shield it from threats.
4. Collusion: Multiple malicious adversaries can collude to launch a joint attack. In practice, a conspiracy attack requires only a tiny amount of secrecy to be divulged by an internal adversary to undermine the availability of most security protocols. For example, HE and SMC-based security schemes rely on the absolute security of keys. They synchronize and upload the colluded malicious parameters to the server for aggregation and perform iterative attacks to disrupt the performance of the model [29]. Furthermore, dishonest clients and servers can conspire to steal confidential information (e.g., private keys), posing a threat to the partially privacy-preserving FL model [12].

In Table 1, we analyze and compare the impact of different malicious entities on the model. The security concern is determined by whether the model can converge adequately. We simply divide the threats confronted by our system from low to high. Since the server has too much information, it is involved in a high threat level of attacks, while the threat to the client depends on the number of attacks involved. Privacy is measured by the threat to the client's private data. Although a single client has limited information and low threat, collusion with the server threatens other users' information. The notable exception is the collusion attack, which is more complex to analyze because of the collusion between different entities [30]. However, because collusion attacks usually involve servers, we set the threat level as high.

**Table 1.** Evaluation of the threat of different attack entities.

| Position | Malicious Entities | Security Treat | Privacy Treat | Reference |
|---|---|---|---|---|
| Internal | Server | high | high | Poisoning [31] et al. |
| Internal | Client | medium | low | Poisoning [32], Backdoor [28] et al. |
| External | Attacker | medium | high | Inference [33] et al. |
| Collusion | Server and Client | high | high | Sybil-based Collusion Attacks [34] et al. |

### *3.2. Security Threats in FL*

In this paper, we believe that security attacks aim to disrupt the availability and robustness of the model. Specifically, a security attack is a possibility of a vulnerability being exploited by a malicious/curious attacker to affect the security of a system and violate its privacy policy. Here we list a few mainstream attack patterns and we summarize the main attacks in Table 2.

#### 3.2.1. Poisoning Attack

Poisoning attacks are one of the most common security attacks in FL. Since each client can impact the global model, maliciously training data and even model weights can directly affect the model's accuracy. A poisoning attack aims at reducing the generalization ability of the model to destroy its usability. Although there are many means of poisoning attacks at present, depending on the target of poisoning, they can be roughly classified into data poisoning and model poisoning. It is worth noting that poisoning attacks can be initiated by different participants. Especially if the server is malicious, it can efficiently execute both types of poisoning attacks during the training progress.

Data poisoning can be roughly divided into two categories: dirty-label attacks and

clean-label attacks. The former tends to misclassify by injecting desired target labels into training datasets. The typical dirty-label attack is a label-flipping attack [35]; it reverses the label of a feature-invariant sample, thus forcing the model to recognize it as another class. In Figure 1, malicious adversaries generate poisonous training samples by label flipping and eventually mislead the global model to generate incorrect classifications. Unlike dirty-label attacks, clean-label attacks correctly classify poisoned labels during training. However, the classification models will classify it into the wrong class. Clean label attacks are more insidious than the former, as most resistance methods based on distribution differences have little impact on them.

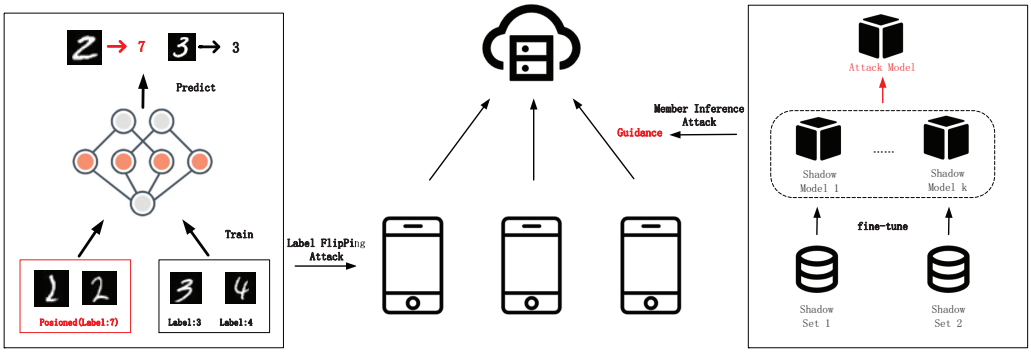


**Figure 1.** Poisoning attack and member inference attack in FL.

Unlike data poisoning, model poisoning usually requires sophisticated technical skills and high computational resources. However, it is also relatively more destructive to models. Model poisoning aims to make the model misclassify for selective inputs with high confidence. The work in [36] was carried out by an adversary controlling a small number of malicious agents to cause the global model to misclassify a set of chosen inputs with high confidence. It is a common method to misjudge the model by noise. In the work of [37], noise with different intensities and distributions was used to detect the pixel space of opposite images.

#### 3.2.2. Backdoor Attack

Whereas a poisoning attack changes the correct decision boundaries through datasets with different boundaries, a backdoor attack systematically controls the decision boundaries of the model by implanting triggers. The attacker inserts a hidden backdoor into the model and triggers the hidden backdoor in the prediction phase to complete the malicious attack. Specifically, the adversary inserts a backdoor into the data of the specified label and participates in the training of FL. When the global model classifies the backdoor data, the backdoor will be triggered, and the classifier will output a specified malicious output. Since the malicious model has the same accuracy, identification is difficult. In Ref. [38], the authors demonstrated that by designing backdoor implantation for low-probability or edge-case samples, it becomes difficult for the system to detect malicious samples.

The benefit of this loophole is that the backdoor attack replaces the original uploaded local model with a model so that the attacker controls how the model performs on an attacker-chosen backdoor subtask [28]. This method scales the model weights $(\gamma = n/\eta)$ to retain the backdoor mean. Instead, Ref. [39] found that canonical cropping and weak differential privacy can mitigate backdoor attacks and tested different attacks on the non-iid dataset. Backdoor attacks are relatively undetectable and perform superiorly, therefore representing a serious security risk to FL.

#### 3.2.3. Free-Rider Attack

The free-rider attack is covert and less damaging. A malicious client may participate in obtaining a joint model without actually contributing any data during the training process [40,41]. This can discourage participants with high-quality data from getting

enough revenue and negatively affect FL training. Typically, fewer free riders are less harmful, but this is unfair to other clients. Currently, the use of FL models based on contribution value estimation can alleviate this part of the problem.

### *3.3. Privacy in FL*

Attacks against user privacy undermine the confidentiality of the FL model. Although FL needs to share model parameters instead of sharing local data, there are still ways to steal user local information.

#### 3.3.1. Inference Attack

Member inference attacks [42] assist in generating an attack model to determine whether the target is a member of the original data by training the shadow model to mimic the behavior of the target model. Inference attacks can be divided into black-box and white-box attacks depending on whether the gradient parameters can be stolen.

In a white-box attack, the adversary can save snapshots of FL model parameters and perform attribute inference by exploiting differences between successive snapshots, which is equivalent to aggregated updates from all participants minus the adversary [13]. Black box attacks require relatively little knowledge to be gained and are, therefore, more practical. Generally, an adversary will test the global model using specific inputs and generate confidence scores based on differences in the distribution, which is used to infer sensitive information. On this basis, Ref. [43] further introduces a label-only attack that does not need confidence scores and, simultaneously, no loss of attack efficiency, thus further reducing the prior knowledge required for an attack.

Against the target of the inference, inference attacks can be classified as a category inference attack (CIA) [44], feature inference attack (FIA) [45], label inference attack (LIA) [46] and member inference attack (MIA). All of these lead to unintended additional information leakage to the adversary. Amongst them, MIA has received the most attention. MIA in FL models aims to infer whether a data record was used to train a target FL model. In Figure 1, the method of using shadow models to train attack models is the most common. The shadow model behaves similarly to the target model, such as supervised training using a verifiable dataset. The attack model uses the shadow model to identify behavioral differences in the target model and uses them to distinguish between members and non-members of the target model. However, this method relies on stealing intermediate gradients and is, therefore, still a white-box model. Prior work [47] has developed transfer attacks and borderline attacks to reduce the knowledge required to attack label-only and can achieve remarkable performance.

#### 3.3.2. Model Inversion Attack

In contrast to inference attacks, model inversion attacks tend to obtain a certain level of statistical information. They are used to train inversion models to reconstruct the client's original data from the received preliminary information on the model [48–50]. Here, an honest and semi-trusted server is an attacker. The server can reconstruct the user's original data from intermediate activations by training the inversion model. For example, Ref. [51] notes that split federated learning (SFL) is vulnerable to an MI attack. The server can reconstruct the original numbers of the team doctor client just by the accepted intermediate activations. MI resistance at the training time is significantly more difficult because the server node can access any intermediate activation. However, this is not unsolvable. Related work presents the loss of sensitive data during communication by minimizing the distance correlation between the original data and the intermediate representation [49] to reduce the usability of the model inversion model.

#### 3.3.3. GANs

GANs have been hugely successful in the image field. It can generate a large amount of high-quality fake data through gaming methods. Therefore, it is an enhancement from

both an offensive and a defensive point of view. On the one hand, techniques based on GANs can enhance the ability to poison and infer attacks. The fake data generated by GANs facilitates poisoning attacks. The work in [52] achieves over 80% accuracy in both the poisoning and the main task by generating data through GAN. The work in [53] considers the use of GAN to generate enriched attack data for the shadow model, which in turn improves the accuracy of member inference attacks to 98%. Due to the nature of GANs, the system cannot predict all possible threats based on them. Therefore, it is more difficult to prevent attacks based on GANs. On the other hand, mechanisms combined with GAN can also improve the robustness of the FL model [54]. The work in [55] shares the client's generator with the server to aggregate the client's shared knowledge and thus improve the performance of each client's local network.

**Table 2.** Summary of the main attacks.

| Category | Attack | Description | Method | Initiators | Hazards | Ref. |
|---|---|---|---|---|---|---|
| Security | Poisoning | The attacker injects malicious data to corrupt the output model. | Data Pos, Model Pos | Client, Server | Availability, Robustness | [35–37] |
| | Backdoor | Prediction by implanting backdoor control models. | Backdoor | Client | Integrity, Robustness | [28] |
| | Free-rider | The attacker obtains a high-value training model with low-value data. | Random weights attack | Client | Availability, Fairness | [40,41] |
| Privacy | Inference | High confidence sensitive information deduced by means of attacks. | Member Inf, Class Inf, Feature Inf, Label Inf | Server, Attacker | Confidentiality | [35] |
| | Model Inversion | Using leaked information to reverse model analysis to obtain private information | Map Inversion | Client | Confidentiality | [49] |
| | GANs | The attacker obtains a high-value training model with low-value data. | Random weights attack | Client | Confidentiality | [52] |

## 4. Defense

### *4.1. Defense Mechanism*

Based on our analysis of security and privacy issues in federated learning above, two main perspectives are worth considering to improve the security of FL: FL needs to identify and deal with possible security threats at any stage of training. In addition, FL should ensure mutual trust between all entities, which helps to attract more quality data to participate in training.

For the first problem, we usually use some proactive defenses. These methods are expected to detect and eliminate threats as they arise. This is typically cost-effective, but it is limited in the number of threats it can handle. For the second problem, the key problem is to keep sensitive information from being transmitted directly. The usual method is to encrypt sensitive information or use a secure transmission channel. Such approaches tend to be reactive, and data are not monitored once processed.

#### 4.1.1. Anomaly Detection

Anomaly detection entails statistical and analytical methods to identify events that do not conform to expected patterns or behaviors. Target-based detection models can be roughly divided into anomaly detection for the server and anomaly detection for clients. On the server side, anomaly detection methods such as parametric threshold-based, feature-based, and smart contract-based have been proposed for screening poisoned

clients. By testing the outlier degree of data points, the server can effectively reduce the damage of poisoned data to the global model, a common means to actively defend against poisoning attacks. For example, the work in [15] designed an outlier data point detection mechanism that can effectively eliminate tag reversal and backdoor-based poisoning attacks. Li et al. [56] use a pre-trained anomaly detection model to test whether users deviate from the FL training regulations. In addition, by saving incremental updates in the blockchain distributed ledger, the server can detect and audit the updates of the model [57]. Another aspect of anomaly detection methods such as BAFFLE [58] on the client side allows the detection to be decentralized to the client, with the server simply analyzing the results of the participant's determination. At the same time, anomaly detection methods based on participant parameter distributions and energy anomalies can be constructed to cope with free-rider attacks.

#### 4.1.2. Blockchain

Blockchain is based on a peer-to-peer network. Blockchain ensures secure storage and data traceability through a combination of chain, tree, and graph structures. In addition, the blockchain achieves tamper-evident data through the consensus mechanism of proof of work (POW). Blockchain and FL are complementary to each other. Blockchain is a natural fit for development alongside FL as an inherently secure distributed system. Combined with FL, we can make all its data copied, shared, and distributed on multiple servers. As in Figure 2, FL can build a trusted third party and complete some trusted operations on the chain, thus reducing the trust anxiety on the server.

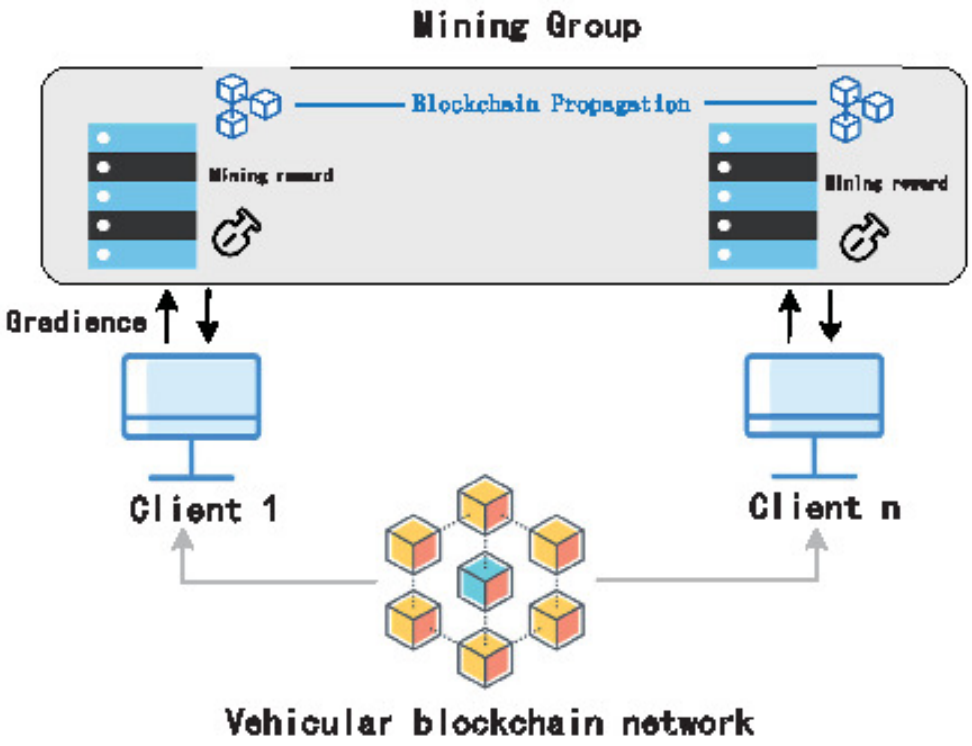


**Figure 2.** Blockchain in VFL.

The blockchain's decentralization can weaken the server's authority, while the distributed ledger provides secure verification for FL [59]. In addition to its verifiable nature, the blockchain can enhance FL's fairness. The transparent and verifiable distribution of incentives can be improved through smart contracts, thereby achieving that all customers receive incentives that meet their values [60]. Ref. [61] uses blockchain to facilitate transparent processes and enforce regulations. The server is independent of blockchain computing, enabling a trust boundary with the user.

#### 4.1.3. Differential Privacy

Since, in FL, the a priori knowledge of the attack is often gradient information in transmission, it is important to mask the authenticity of the information. Differential privacy wishes to obscure the actual query by adding a specified perturbation. Differential privacy was first used to encrypt database information. Blurring the privacy boundary between similar data subsets can satisfy both security and unpredictability. At the same time, DP execution's time consumption is minimal compared to other methods, making it outstanding in scenarios with time performance requirements. Ref. [62] applies differential privacy in deep learning for the first time and can maintain high accuracy guarantees for the model while introducing privacy moments to measure privacy loss.

The FL model, in combination with DP, can be broadly divided into the curator model, the local model, and the shuffle model. The curator model (CDP) has high accuracy but weak security. On the contrary, the local model (LDP) adds perturbation in local training progress to increase security but sacrifices more accuracy. The shuffle model (SDP) is a compromise between the two. Therefore, considering privacy, accuracy, and efficiency in the DP model is a research hotspot. In Figure 3, we tested this on the MNIST dataset and more intuitively represented the utility loss.

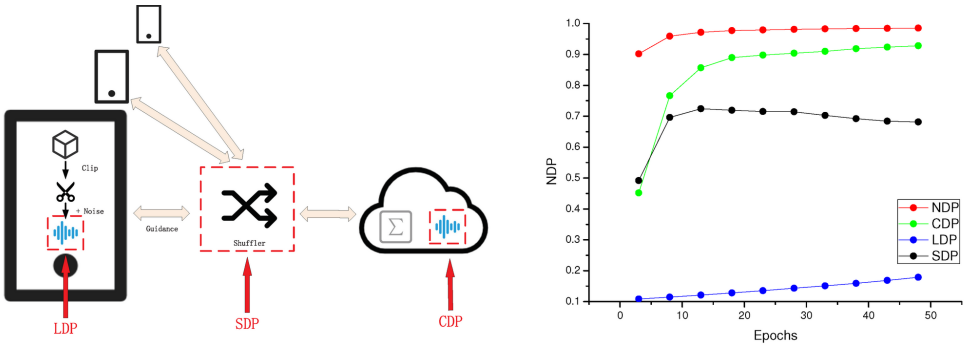


**Figure 3.** We experimentally compared the performance of different DP-FL schemes on the MNIST dataset under the same privacy budget.

In combination with DP, FL has been extensively studied in academia. Although DP inevitably suffers from the loss of accuracy, some ideas are put forward to compensate for this loss. Renyi differential privacy [63] uses Renyi entropy to define DP more broadly and further shrink the upper bound of the privacy budget. It blurs the line between slack DP and strict DP. The newest discrete Gaussian DP [64] is proposed to alleviate the contradiction between utility and security. In addition, privacy amplification with the help of a third-party shuffler is also worthy of attention.

#### 4.1.4. Homomorphic Encryption

HE converts the plaintext computation to the ciphertext calculation and decrypts it to obtain the same result. This solves the trust-building and key-transfer problems faced in traditional cryptography, as the operator is only concerned with the ciphertext and does not need to know any decryption information (decryption key $K$). Therefore, combined with HE, a secure aggregation of gradient information can be achieved.

A safe homomorphic system should satisfy:

$$\forall m_1, m_2 \in M, Enc_{pk}(m_1 \odot_M m_2) \leftarrow Enc_{pk}(m_1) \odot_C Enc_{pk}(m_2)$$

Here, $M$ and $C$ denote the plaintext space and ciphertext space, respectively, $\odot$ denotes the operator. Operations on ciphertext can be overloaded as addition and multiplication.

In Paillier [65], addition can be extended to multiplication, and multiplication can be extended to subdivision operations.

$$\textbf{Addition} : Dec_{sk}(m_1 \odot_C m_2) = Dec_{sk}(m_1 + m_2)$$
$$\textbf{Scalar multiplication} : Dec_{sk}(m_1 \odot_C m_2) = Dec_{sk}(m_1 * m_2)$$

Generally speaking, the functionality of homomorphic encryption is proportional to its complexity. Based on this, HE can be divided into two categories: semi-homomorphic encryption and fully homomorphic encryption. The former satisfies finite operations but is efficient, and the latter satisfies arbitrary operations but is inefficient. Based on this, Ref. [66] proposed BatchCrypt for Cross-silo federated learning, reducing the communication and time cost of HE with almost no loss of accuracy. Ref. [67] designed FedML based on the Paillier homomorphic encryption algorithm to implement the federation matrix factorization in semi-honest scenarios. HE in FL still has some limitations, mainly high interaction overhead and loss of accuracy.

#### 4.1.5. Secure Multiparty Computing

Secure multiparty computing (SMC) usually encrypts the communication process and hides the input information from the output side. Specifically for each party involved, the output value can only be known based on its input, and no other knowledge is available. With secure SMC, multiple parties cooperate to compute functions of common interest without revealing their private inputs to other parties [68]. As multiple clients in FL interact with the server, secure aggregation is a central concern, and SMC is the most suitable.

SMC can be implemented through three frameworks: secret sharing [69] and inadvertent transmission [70], where secret sharing is at the heart of secure multiparty computing. The work in [71] uses secret sharing to build a lightweight FL framework for the IoT. Different carefully designed local gradient masks and an additional mask reuse scheme are used to reduce the overhead of communication. Ref. [72] designed VerifyNet, a verifiable FL framework that uses a double masking mechanism to secure private information on a secret sharing mechanism. This approach is also privacy guaranteed for clients that drop out during training.

#### 4.1.6. Trusted Execution Environments

TEE provides assurance of integrity and confidentiality for handling sensitive code and data [73] on computers. One of its main design purposes is to solve the problem of secure remote computing, which is what TFL needs. Taking Intel SGX as an example, it provides a secure container, which constrains the sensitive information uploaded by remote users in the container. It ensures the confidentiality of calculation and intermediate data.

Based on this concept, Ref. [74] designed an efficient privacy-preserving federated learning framework called FLATEE, which can handle malicious parties without privacy leaks. In FLATEE, TEE generates the symmetric encryption key and the public key. The client performs the privacy algorithm (DP and encryption) in the secure enclave TEE, and the server performs privacy aggregation in the aggregator secure enclave TEE. However, due to resource sharing, SGX still has many attack surfaces, such as page tables, cache, CPU internal structures, etc. For example, side-channel attacks are common TEE attacks. Facing these vulnerabilities, Ref. [75] designed a ShuffleFL with TEE, in which a randomized grouping algorithm is used to dynamically organize all participants into a hierarchical group structure, combined with intra-group gradient segmentation aggregation against opponents. The specific process can be found in Figure 4.

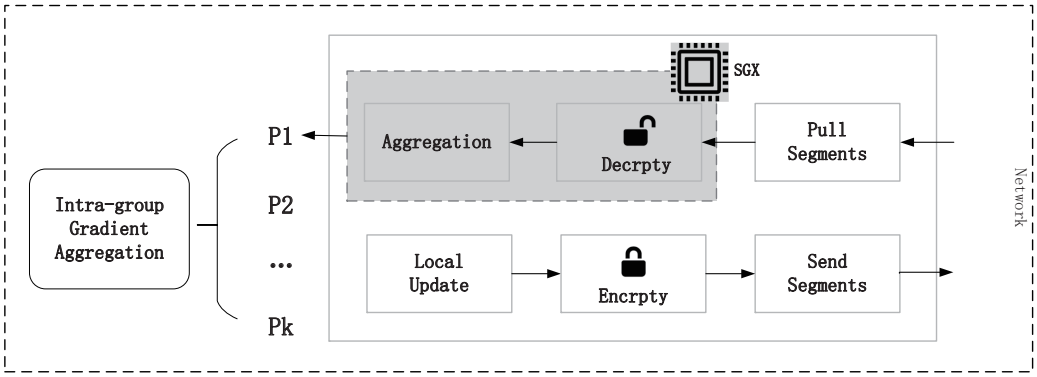


**Figure 4.** ShuffleFL with TEE.

#### 4.1.7. Hybrid

On the one hand, in the face of complex security needs, sometimes maintaining data localization alone often fails to achieve sufficient privacy guarantees. The work in [76] combined DP and SMC in the database using DP fuzzy secrets and used a secret sharing mechanism to slice and dice the restructuring of parameters and answers for computational and output privacy. Inspired by the hybrid methodology, Refs. [76,77] combine it with FL. In other areas, Ref. [78] introduces a novel strategy that combines differential privacy methods and homomorphic encryption techniques to achieve the best of both worlds. Ref. [64] has designed an efficient and secure aggregation scheme and uses distributed DP for privacy enhancement.

On the other hand, there are inherent limitations to a single-defense approach. For example, differential privacy has an inherent loss of accuracy, and HE and SMC perform poorly in terms of efficiency. In combination with SMC, DP in [77] can achieve a small increase in noise while guaranteeing a predefined trust rate. This reduces the negative impact of DP on the utility of the model. However, the underlying security mechanism of [77] is additive homomorphic encryption and is therefore accompanied by a longer training time and higher transmission costs. Ref. [79] improved security policy using function encryption and designed HybridAlpha to achieve shorter time and communication costs. In Ref. [80], HE and TEE were used jointly to achieve an accurate analysis of genomic data. This combined solution provided a good compromise regarding efficiency and computational support for safe statistical analysis.

### *4.2. Security Evaluation*

In Table 3, we compare the aspects of protection capability, model accuracy, scheme efficiency, model robustness, scalability, and generalization for different methods.

**Table 3.** The horizontal comparison of security solutions.

| Scheme | Protection | Precision | Efficiency | Robustness | Scalability | Generalizability | Ref. |
|---|---|---|---|---|---|---|---|
| Anomaly Detection | medium | high | high | medium | high | low | [15,56,57] |
| Data Sanitization | medium | high | medium | low | high | high | [81] |
| Blockchain | high | high | low | medium | high | medium | [59,60] |
| Differential Privacy | high | low | high | high | high | high | [62–64] |

**Table 3.** *Cont.*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Homomorphic Encryption | high | high | low | high | low | high | [66] |
| Secure Multiparty Computing | high | high | medium | high | low | medium | [68] |
| Trusted Execution Environments | medium | high | high | medium | low | high | [73,75] |

Regarding system security protection, AD, TEE, and DS cannot work against internal malicious nodes. The related means based on perturbation and cryptography hide the intermediate variables of the computation with solid theoretical security. Regarding accuracy impact, DP introduces noise to mask critical information, which affects the final model convergence accuracy. This loss of accuracy is often unacceptable under LDP. In terms of efficiency performance, the DP-based FL has the lowest time consumption thanks to its streamlined algorithm. The intermediate information masking method represented by DP has good generalization for different types of FL. In contrast, schemes such as AD need to be designed specifically for different kinds of FL and have certain bureaus.

Homomorphic encryption is usually inefficient in the face of high-dimensional vectors in FL, which take a lot of time to encrypt and decrypt. AD and DS exhibit weak robustness to new attacks and need to be dynamically updated on time. The scalability of FL is reflected in the performance in complex scenarios such as large-scale node distribution and unexpected user dropouts. The HE and SMC participating nodes are also computational and heavily burdened with computation and communication. It is worth noting that HE and SMC are cryptographically provably secure, which is significant for the interpretability of the construction of TFL. With a large number of level nodes, training efficiency is significantly reduced. At the same time, limited local computing resources limit the use of TEE. However, directly executing the training process in the TEE environment will significantly decline performance. Take Intel SGX as an example; it only supports CPU operation, which limits the model's efficiency (CNN, DNN) and relies on GPU training. At the same time, when the memory exceeds the limit, it will induce a lot of paging overhead.

## 5. Future Research

The continuing fire of federal learning research, new forms of attack, and scenario demands have raised the bar even higher. However, the investigation into TFL is still in its infancy. In this section, the main existing threats and means of defense in FL are combined to suggest future directions worthy of attention.

### *5.1. Security Metrics*

There still needs to be a uniform security metric in FL. From a global perspective, researchers need to assess the program's level of security accurately. Establishing a consistent metric will facilitate the refinement of the system's rating metrics and the assessment of the availability of attacks and defenses. Establishing sound security metrics enables the selection and optimization of defense technologies.

Relevant research remains limited. The work in [82] presents a method to choose privacy metrics based on nine questions that help identify the right ones for a given scenario. Ref. [83] presents the need for careful model design for both performance and cost. For differential privacy [62,63] proposes privacy accounters to measure privacy loss. From the local perspective, the security assessment of the system needs further refinement, for example, how to assess the security risk of each entity or parameter. Ref. [17] performs a fine-grained trust analysis of the different entities involved in the training using trust separation and trust boundaries. However, there is still a lack of data and parameter sensi-

tivity analysis. As can be seen, the vast majority of the work is a metric for the refinement of specific privacy paradigms, but it does measure the means of the global model.

### 5.2. Model Interpretability

The interpretability of the model is to understand the model's decision-making from the perspective of human beings. More specifically, interpretable means that researchers should clarify the causal relationship between each part and output. As a unique distributed machine learning, FL's interpretability can subdivide the explanation of machine learning and algorithms. The interpretability of machine learning has been widely discussed in academic circles [84–86], but a specific definition still needs to be provided. The interpretability of algorithms includes aggregation algorithms and security algorithms. Ref. [87] demonstrates that heterogeneous data slow down the convergence of the FedAvg algorithm and proves the need for learning rate decay. The specific security algorithm should be analyzed in detail; taking DP as an example, Ref. [88] does the first work that rigorously investigates theoretical and empirical issues regarding the clipping operation in FL algorithms. The interpretability of FL determines whether users can trust it, which is necessary for TFL. It also helps to optimize the utility of the model.

### 5.3. The Tradeoff between Safety, Precision, and Utility

Typically, as the security level of a system increases, it will inevitably increase the algorithm's complexity, resulting in additional computational and communication overheads. Thus, safety, precision, and efficiency are often mutually constrained. In the case of DP, which is currently frequently studied, the larger the perturbation added, the safer the intermediate variables, and conversely, the less accurate the model [89].

Part of the approach has been proposed to balance these three aspects. Recently, FL models that introduce the Shuffle mechanism have been heavily researched. Ref. [90] first demonstrates that shuffle can achieve privacy amplification without compromising accuracy and gives a strict upper boundary for privacy amplification. In terms of communication efficiency, gradient compression [91] and sparsification [92] are used to reduce the size of the transmitted information.

### 5.4. Decentralization

A centralized FL can give the server too much power to create a trust crisis. However, an utterly decentralized server would also be inconvenient to manage and audit. It is worth thinking about effectively spreading the risk while ensuring the model can be handled safely. Blockchain can help decentralize FL [60], where the server no longer acts as the core of auditing and verification but only performs aggregation algorithms and where the associated auditing and verification can be achieved through smart contracts on the blockchain. However, Block-FL suffers from both efficiency and expense problems.

### 5.5. Trusted Traceability

Current FL cannot backtrack during their lifecycle. When a malicious node launches an attack, it is difficult for the system to identify the source. Part of the work did preliminary work to verify the availability of gradient information [72]. The technology combined with blockchain is worthy of attention in this regard, and the nature of blockchain offers the possibility of traceability. Prior works [93,94] achieve authentication of user identity with the help of blockchain.

## 6. Conclusions

FL facilitates the free flow of data and offers the possibility of machine learning cross-domain applications. However, it is necessary to remove user trust anxiety and facilitate the commercial deployment of TFL. However, research on TFL is still in its infancy. In this article, we clearly define trustworthy federated learning. By summarizing and analyzing the security and privacy threats faced by FL, we hope to provide new research perspectives

to community researchers. Finally, we provide some helpful research directions for the top-level design of TFL. TFL is an enhanced framework designed for market needs, and our research aims to provide some references for its design.

**Author Contributions:** Conceptualization, D.C.; Methodology, D.C.; Investigation, X.J.; Writing—original draft, D.C. and X.J.; Supervision, J.C.; Project administration, H.Z. and J.C.; Funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The University Synergy Innovation Program of Anhui Province: GXXT-2022-049.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. McMahan, H.; Moore, E.; Ramage, D.; Hampson, S.; Aguera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, MA, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282. Volume 54.
2. Yang, Z.; Shi, Y.; Zhou, Y.; Wang, Z.; Yang, K. Trustworthy Federated Learning via Blockchain. *IEEE Internet Things J.* **2023**, *10*, 92–109. [CrossRef]
3. Lin, X.; Wu, J.; Li, J.; Zheng, X.; Li, G. Friend-as-learner: Socially-driven trustworthy and efficient wireless federated edge learning. *IEEE Trans. Mob. Comput.* **2023**, *22*, 269–283. [CrossRef]
4. Bugshan, N.; Khalil, I.; Rahman, M.S.; Atiquzzaman, M.; Yi, X.; Badsha, S. Toward Trustworthy and Privacy-Preserving Federated Deep Learning Service Framework for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1535–1547. [CrossRef]
5. Zhang, Q.; Ding, Q.; Zhu, J.; Li, D. Blockchain empowered reliable federated learning by worker selection: A trustworthy reputation evaluation method. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Nanjing, China, 29 March 2021; IEEE: Piscataway, NJ, USA, 2021.
6. Yang, Q.; Liu, Y.; Cheng, Y.; Kang, Y.; Chen, T.; Yu, H. Federated learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2019**, *13*, 1–207 .
7. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Roselander, J. Towards federated learning at scale: System design. In Proceedings of Machine Learning and Systems, Standford, CA, USA, 31 March–2 April 2019; pp. 374–388.
8. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *Acm Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
9. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019-2019 IEEE international conference on communications (ICC), Shanghai, China, 20–24 May 2019; IEEE: Piscataway, NJ, USA; pp. 1–7.
10. Xu, C.; Qu, Y.; Xiang, Y.; Gao, L. Asynchronous federated learning on heterogeneous devices: A survey. *arXiv* **2022**, arXiv:2109.04269.
11. Ahmed, S.T.; Kumar, V.V.; Singh, K.K.; Singh, A.; Muthukumaran, V.; Gupta, D. 6G enabled federated learning for secure IoMT resource recommendation and propagation analysis. *Comput. Electr. Eng.* **2022**, *102*, 108210. [CrossRef]
12. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
13. Lyu, L.; Yu, H.; Yang, Q. Threats to federated learning. *Lect. Notes Comput. Sci.* **2020**, *12500*, 3–16.
14. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [CrossRef]
15. Liu, X.; Li, H.; Xu, G.; Chen, Z.; Huang, X.; Lu, R. Privacy-enhanced federated learning against poisoning adversaries. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4574–4588. [CrossRef]
16. Yue, S.; Ren, J.; Xin, J.; Zhang, D.; Zhang, Y.; Zhuang, W. Efficient federated meta-learning over multi-access wireless networks. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1556–1570. [CrossRef]
17. Liu, R.; Cao, Y.; Chen, H.; Guo, R.; Yoshikawa, M. Flame: Differentially private federated learning in the shuffle model. *AAAI Conf. Artif. Intell.* **2021**, *35*, 8688–8696. [CrossRef]
18. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **2021**, *5*, 1–19. [CrossRef] [PubMed]
19. Vanhaesebrouck, P.; Bellet, A.; Tommasi, M. Decentralized collaborative learning of personalized models over networks. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 509–517.
20. Xie, M.; Long, G.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J.; Zhang, C. Multi-center federated learning. *arXiv* **2020**. arXiv:2005.01026.
21. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. *Mach. Learn. Syst.* **2020**, *2*, 429–450.
22. Zhang, Q.; Gu, B.; Deng, C.; Huang, H. Secure bilevel asynchronous vertical federated learning with backward updating. *AAAI Conf. Artif. Intell.* **2021**, *35*, 10896–10904. [CrossRef]

23. Liu, Y.; Kang, Y.; Xing, C.; Chen, T.; Yang, Q. A secure federated transfer learning framework. *IEEE Intell. Syst.* **2020**, *35*, 70–82. [CrossRef]
24. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]
25. Jere, M.S.; Farnan, T.; Koushanfar, F. A taxonomy of attacks on federated learning. *IEEE Secur. Priv.* **2020**, *19*, 20–28. [CrossRef]
26. Lyu, L.; Yu, H.; Ma, X.; Sun, L.; Zhao, J.; Yang, Q.; Yu, P.S. Privacy and robustness in federated learning: Attacks and defenses. *arXiv* **2020**, arXiv:2012.06337.
27. Girgis, A.; Data, D.; Diggavi, S.; Kairouz, P.; Suresh, A.T. Shuffled model of differential privacy in federated learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Online, 13–15 April 2021; pp. 2521–2529.
28. Bagdasaryan, E.; Veit, A.; Hua, Y.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics,Online, 26–28 August 2020; pp. 2938–2948.
29. Taheri, R.; Shojafar, M.; Alazab, M.; Tafazolli, R. FED-IIoT: A robust federated malware detection architecture in industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 8442–8452. [CrossRef]
30. Ranjan, P.; Corò, F.; Gupta, A.; Das, S.K. Leveraging Spanning Tree to Detect Colluding Attackers in Federated Learning. In Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 2–5 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–2.
31. Li, Y.; Chen, C.; Liu, N.; Huang, H.; Zheng, Z.; Yan, Q. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Netw.* **2020**, *35*, 234–241. [CrossRef]
32. Cao, D.; Chang, S.; Lin, Z.; Liu, G.; Sun, D. Understanding distributed poisoning attack in federated learning. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 233–239.
33. Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P.S.; Zhang, X. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
34. Xiao, X.; Tang, Z.; Li, C.; Xiao, B.; Li, K. SCA: Sybil-based Collusion Attacks of IIoT Data Poisoning in Federated Learning. *IEEE Trans. Ind. Inform.* **2022**. [CrossRef]
35. Fung, C.; Yoon, C.J.; Beschastnikh, I. Mitigating sybils in federated learning poisoning. *arXiv* **2018**, arXiv:1808.04866.
36. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 634–643.
37. Tabacof, P.; Valle, E. Exploring the space of adversarial images. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 14–19 July 2016; IEEE: Pisataway, NJ, USA, 2016; pp. 426–433.
38. Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.Y.; Papailiopoulos, D. Attack of the tails: Yes, you really can backdoor federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16070–16084.
39. Sun, Z.; Kairouz, P.; Suresh, A.T.; McMahan, H.B. Can you really backdoor federated learning? *arXiv* **2019**, arXiv:1911.07963.
40. Lin, J.; Du, M.; Liu, J. Free-riders in federated learning: Attacks and defenses. *arXiv* **2019**, arXiv:1911.12560.
41. Fraboni, Y.; Vidal, R.; Lorenzi, M. Free-rider attacks on model aggregation in federated learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Online, 13–15 April 2021; pp. 1846–1854.
42. Nasr, M.; Shokri, R.; Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In Proceedings of the 2019 IEEE symposium on security and privacy (SP) , San Francisco, CA, USA, 19–23 May 2019: IEEE: Piscataway, NJ, USA, 2019; pp. 739–753.
43. Choquette-Choo, C.A.; Tramer, F.; Carlini, N.; Papernot, N. Label-only membership inference attacks. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; pp. 1964–1974.
44. Gao, J.; Hou, B.; Guo, X.; Liu, Z.; Zhang, Y.; Chen, K.; Li, J. Secure aggregation is insecure: Category inference attack on federated learning. *IEEE Trans. Dependable Secur. Comput.* **2021**, *20*, 147–160. [CrossRef]
45. Luo, X.; Wu, Y.; Xiao, X.; Ooi, B.C. Feature inference attack on model predictions in vertical federated learning. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 181–192.
46. Fu, C.; Zhang, X.; Ji, S.; Chen, J.; Wu, J.; Guo, S.; Wang, T. Label inference attacks against vertical federated learning. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022.
47. Li, Z.; Zhang, Y. Membership leakage in label-only exposures. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Online, Republic of Korea, 15–19 November 2021; ACM: New York, NY, USA, 2021; pp. 880–895.
48. Fredrikson, M.; Jha, S.; Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; ACM: New York, NY, USA, 2015; pp. 1322–1333.
49. Vepakomma, P.; Singh, A.; Gupta, O.; Raskar, R. NoPeek: Information leakage reduction to share activations in distributed deep learning. In Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 17–20 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 933–942.
50. He, Z.; Zhang, T.; Lee, R.B. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet Things J.* **2020**, 8, 9706–9716. [CrossRef]

51. Li, J.; Rakin, A.S.; Chen, X.; He, Z.; Fan, D.; Chakrabarti, C. ResSFL: A Resistance Transfer Framework for Defending Model Inversion Attack in Split Federated Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2022; pp. 10194–10202.
52. Zhang, J.; Chen, J.; Wu, D.; Chen, B.; Yu, S. Poisoning attack in federated learning using generative adversarial nets. In Proceedings of the 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 5–8 August 2019; IEEE: Piscataway, NJ , USA, 2019; pp. 374–380.
53. Zhang, J.; Zhang, J.; Chen, J.; Yu, S. Gan enhanced membership inference: A passive local attack in federated learning. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
54. Ghonima, R. Implementation of GANs Using Federated Learning. In Proceedings of the 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 142–148.
55. Wu, Y.; Kang, Y.; Luo, J.; He, Y.; Yang, Q. Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2021; pp. 2334–2340.
56. Li, S.; Cheng, Y.; Liu, Y.; Wang, W.; Chen, T. Abnormal client behavior detection in federated learning. *arXiv* **2019**, arXiv:1910.09933.
57. Preuveneers, D.; Rimmer, V.; Tsingenopoulos, I.; Spooren, J.; Joosen, W.; Ilie-Zudor, E. Chained anomaly detection models for federated learning: An intrusion detection case study. *Appl. Sci.* **2018**, *8*, 2663. [CrossRef]
58. Andreina, S.; Marson, G.A.; Möllering, H.; Karame, G. Baffle: Backdoor detection via feedback-based federated learning. In Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), Washington, DC, USA, 7–10 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 852–863.
59. Kim, H.; Park, J.; Bennis, M.; Kim, S.L. Blockchained on-device federated learning. *IEEE Commun. Lett.* **2019**, *24*, 1279–1283. [CrossRef]
60. Rückel, T.; Sedlmeir, J.; Hofmann, P. Fairness, integrity, and privacy in a scalable blockchain-based federated learning system. *Comput. Netw.* **2022**, *202*, 108621. [CrossRef]
61. Miao, Y.; Liu, Z.; Li, H.; Choo, K.K.R.; Deng, R.H. Privacy-Preserving Byzantine-Robust Federated Learning via Blockchain Systems. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2848–2861. [CrossRef]
62. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
63. Mironov, I. Rényi differential privacy. In Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 263–275.
64. Kairouz, P.; Liu, Z.; Steinke, T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In Proceedings of the International Conference on Machine Learning, Xiamen, China, 8–11 November 2021; pp. 5201–5212.
65. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238
66. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In Proceedings of the 2020 USENIX annual technical conference (USENIX ATC 20), Boston, MA, USA, 15–17 July 2020; pp. 493–506.
67. Chai, D.; Wang, L.; Chen, K.; Yang, Q. Secure federated matrix factorization. *IEEE Intell. Syst.* **2020**, *36*, 11–20. [CrossRef]
68. Mugunthan, V.; Polychroniadou, A.; Byrd, D.; Balch, T.H. Smpai: Secure multi-party computation for federated learning. In Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services, Vancouver, BC, Canada, 9–14 December 2019; MIT Press: Cambridge, MA, USA, 2019.
69. Li, C.; Li, G.; Varshney, P.K. Communication-efficient federated learning based on compressed sensing. *IEEE Internet Things J.* **2021**, *8*, 15531–15541. [CrossRef]
70. Hauck, E.; Loss, J. Efficient and universally composable protocols for oblivious transfer from the CDH assumption. *Cryptology* **2017**, 1011.
71. Wei, Z.; Pei, Q.; Zhang, N.; Liu, X.; Wu, C.; Taherkordi, A. Lightweight Federated Learning for Large-scale IoT Devices with Privacy Guarantee. *IEEE Internet Things J.* **2021**. [CrossRef]
72. Xu, G.; Li, H.; Liu, S.; Yang, K.; Lin, X. Verifynet: Secure and verifiable federated learning. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 911–926. [CrossRef]
73. Mondal, A.; More, Y.; Rooparaghunath, R.H.; Gupta, D. Poster: FLATEE: Federated Learning Across Trusted Execution Environments. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; pp. 707–709.
74. Mondal, A.; More, Y.; Rooparaghunath, R.H.; Gupta, D. Flatee: Federated Learning Across Trusted Execution Environments. *arXiv* **2021**, arXiv:2111.06867.

75. Zhang, Y.; Wang, Z.; Cao, J.; Hou, R.; Meng, D. ShuffleFL: Gradient-preserving federated learning using trusted execution environment. In Proceedings of the 18th ACM International Conference on Computing Frontiers, Online, 11–13 May 2021; ACM: New York, NY, USA, 2021; pp. 161–168.
76. Pettai, M.; Laud, P. Combining differential privacy and secure multiparty computation. In Proceedings of the 31st Annual Computer Security Applications Conference, Los Angeles, CA, USA, 7–11 December 2015; ACM: New York, NY, USA, 2015; pp. 421–430.
77. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; ACM: New York, NY, USA, 2019; pp. 1–11.
78. Kim, M.; Lee, J.; Ohno-Machado, L.; Jiang, X. Secure and differentially private logistic regression for horizontally distributed data. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 695–710. [CrossRef]
79. Xu, R.; Baracaldo, N.; Zhou, Y.; Anwar, A.; Ludwig, H. Hybridalpha: An efficient approach for privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; ACM: New York, NY, USA, 2019; pp. 13–23.
80. Sadat, M.N.; Al Aziz, M.M.; Mohammed, N.; Chen, F.; Jiang, X.; Wang, S. Safety: Secure gwas in federated environment through a hybrid solution. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2018**, *16*, 93–102. [CrossRef] [PubMed]
81. Shen, Y.; Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In Proceedings of the International Conference on Machine Learning , Long Beach, CA, USA, 9–15 June 2019; pp. 5739–5748.
82. Wagner, I.; Eckhoff, D. Technical privacy metrics: A systematic survey. *ACM Comput. Surv.* **2018**, *51*, 1–38. [CrossRef]
83. Majeed, I.A.; Kaushik, S.; Bardhan, A.; Tadi, V.S.K.; Min, H.K.; Kumaraguru, K.; Muni, R.D. Comparative assessment of federated and centralized machine learning. *arXiv* **2022**, arXiv:2202.01529.
84. Koh, P.W.; Liang, P. Understanding black-box predictions via influence functions. In Proceeding of the International Conference on Machine Learning, Sydney, Australia, 7–9 August 2017; pp. 1885–1894.
85. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 1135–1144.
86. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef]
87. Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv* **2019**, arXiv:1907.02189.
88. Zhang, X.; Chen, X.; Hong, M.; Wu, S.; Yi, J. Understanding Clipping for Federated Learning: Convergence and Client-Level Differential Privacy. In Proceedings of the International Conference on Machine Learning, PMLR, MA, Baltimore, MD, USA, 17–23 July 2022; pp. 26048–26067.
89. Kim, M.; Günlü, O.; Schaefer, R.F. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2011; IEEE: Piscataway, NJ, USA, 2021; pp. 2650–2654.
90. Balle, B.; Bell, J.; Gascón, A.; Nissim, K. The privacy blanket of the shuffle model. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2019; Springer: Cham, Switzerland, 2019; pp. 638–667.
91. Li, Z.; Kovalev, D.; Qian, X.; Richtárik, P. Acceleration for compressed gradient descent in distributed and federated optimization. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020; pp. 5895–5904.
92. Cheng, A.; Wang, P.; Zhang, X.S.; Cheng, J. Differentially Private Federated Learning with Local Regularization and Sparsification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2022; pp. 10122–10131.
93. Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A.; Choo, K.K.R. Blockchain-enabled authentication handover with efficient privacy protection in SDN-based 5G networks. *IEEE Trans. Netw. Sci. Eng.* **2019**, *8*, 1120–1132. [CrossRef]
94. Li, Y.; Tao, X.; Zhang, X.; Liu, J.; Xu, J. Privacy-preserved federated learning for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8423–8434. [CrossRef]