mathematics

MDPI

Article

Challenges and Countermeasures of Federated Learning Data Poisoning Attack Situation Prediction

Jianping Wu [1], Jiahe Jin [2] and Chunming Wu [1,*]

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China; wjpself@zju.edu.cn

[2] Key Laboratory of Key Technologies for Open Data Fusion in Zhejiang Province, Hangzhou 310007, China; jinjh@zj.gov.cn

* Correspondence: wuchunming@zju.edu.cn

Abstract: Federated learning is a distributed learning method used to solve data silos and privacy protection in machine learning, aiming to train global models together via multiple clients without sharing data. However, federated learning itself introduces certain security threats, which pose significant challenges in its practical applications. This article focuses on the common security risks of data poisoning during the training phase of federated learning clients. First, the definition of federated learning, attack types, data poisoning methods, privacy protection technology and data security situational awareness are summarized. Secondly, the system architecture fragility, communication efficiency shortcomings, computing resource consumption and situation prediction robustness of federated learning are analyzed, and related issues that affect the detection of data poisoning attacks are pointed out. Thirdly, a review is provided from the aspects of building a trusted federation, optimizing communication efficiency, improving computing power technology and personalized the federation. Finally, the research hotspots of the federated learning data poisoning attack situation prediction are prospected.

Keywords: federated learning; data poisoning; situation prediction; privacy protection

MSC: 68P27

check for updates

Citation: Wu, J.; Jin, J.; Wu, C. Challenges and Countermeasures of Federated Learning Data Poisoning Attack Situation Prediction. *Mathematics* **2024**, *12*, 901. https://doi.org/10.3390/math12060901

Academic Editor: Shaomin Wu

Received: 31 January 2024
Revised: 14 March 2024
Accepted: 16 March 2024
Published: 19 March 2024

1. Introduction

In recent years, rapid development in machine learning has led to significant achievements in computer vision [1], natural language understanding [2], and large language models [3,4] in the artificial intelligence community. Machine learning involves training models with extensive data. However, as information techniques become more prevalent, incidents of personal data leakage have become increasingly common, raising people's awareness of data security and privacy protection. Consequently, many countries have introduced laws and regulations to safeguard data privacy. To achieve a balance between privacy protection and data silos, federated learning has emerged as a promising solution. It aims to train a centralized federated model using decentralized data sources while ensuring the privacy of the original data throughout the training process [5].

According to relevant research efforts [6], although federated learning solves the privacy computing and data island problems in traditional machine learning, there are still many security threats due to the existence of malicious participants. It is of great significance to study various attacks against federated learning systems, discover the vulnerabilities of federated learning, and promote the research of related defense methods to build a more secure federated learning system [7].

At present, existing research efforts on federated learning security attacks mainly focus on model poisoning attacks [8–10]. The attacker corrupts the global model by constructing malicious model updates. However, model poisoning requires the attacker to control single or multiple parties. With the expansion of federated learning deployment applications, the number of compromised participants has gradually decreased, and the application space for model poisoning has become increasingly narrow. Data poisoning attacks [11–13] mean that the attacker adds a small number of carefully designed poisoning samples to the training data set of the model, and uses the training or fine-tuning process to poison the local model, thereby affecting the training and performance of the global model. In practical applications, data poisoning attacks have lower requirements on the attacker's ability and knowledge and a wider range of implementation scenarios than model poisoning. Moreover, data poisoning attacks are harder to detect in large-scale training data sets. However, the current research on data poisoning is still relatively superficial and only stays at the stage of simply verifying the feasibility of the attack.

In response to poisoning attacks, a series of defenses have been proposed, which can be summarized into two categories: passive defenses and active defenses [14]. Passive defenses mainly start from the aggregation server side, design relevant aggregation model strategies, and eliminate poisoning models, thereby improving the global model performance [15]. The active defenses eliminate the impact of the poisoning model on the global model by detecting the performance of the local model and eliminating the poisoning model [13]. At present, active defenses is the trend and main direction of studying federated learning poisoning attack detection solutions. It can put potential risks in front and achieve the goal of timely stop loss.

Currently, academic circles have conducted the relevant research on detecting and mitigating data poisoning attacks in federated learning. Steinhardt et al. [16] proposed a data cleaning method, that is, cleaning the training set, filtering and eliminating poisoning data, to achieve a defense against poisoning attacks. However, such a method cannot work in the form of a local model poisoning attack where the adversary directly tamperes with the model parameters [17]. Feng et al. [18] proposed a data poisoning defense strategy based on a logistic regression classifier, which removes samples whose abnormality exceeds the threshold by detecting outliers. However, this method assumes that the server can know the proportion of poisoning samples in the training data in advance, which cannot be implemented in actual applications of federated learning. Zhao et al. [19] first proposed a method of using Generative Adversarial Network (GAN) [20] to generate detection samples. However, because only accuracy is used as the detection indicator, this solution cannot accurately detect targeted attacks. Jagielski et al. [21] proposed a detection method in which the server collects some local training samples and trains the comparison model, and iteratively estimates the residual values of the comparison model and the local model.This method can effectively resist poisoning attacks on training data; however, when the local training set contains many malicious samples, the detection effect of this method is poor. At the same time, this method requires users to upload private training data when constructing the training set used, which violates the original intention of not leaving the local training data in federated learning. At the same time, Li et al. [22] proposed the reasons that limit the large-scale application of the federated learning technology in the real world, namely expensive communication, system heterogeneity, data privacy issues and algorithm complexity.

How to dynamically perceive data poisoning attacks in federated learning and implement a defense at the lowest cost is important research content. Data security situational awareness is a method that assesses data security risks and situations by monitoring and analyzing data traffic, user behaviors, security events and other information in the network in real time. It can help organizations discover data security threats and vulnerabilities in a timely manner and take appropriate measures to protect data security. The defense idea of federated learning is basically the same as the data security situational awareness method, and both build analysis models based on behavior and data flow. This article takes the federated learning data poisoning attack situation prediction as the theme, puts forward improvement suggestions around problems and difficulties, and provides a reference for improving the ability of federated learning to prevent data poisoning attacks. The main contributions of this article are as follows:

1. From the perspective of technical principles, the types of data poisoning based on federated learning are summarized, and the advantages and disadvantages of various technologies are analyzed in detail.
2. Regarding the prediction of federated learning data poisoning attack situations, challenges such as system architecture vulnerability, communication efficiency shortcomings, computing resource consumption, and prediction robustness are raised.
3. Suggestions are put forward to build a federated learning data poisoning attack situation prediction system to help organizations discover and respond to data poisoning threats in a timely manner.

2. Related Work

2.1. Federated Learning

Federated Learning is a distributed machine learning framework proposed by Google's McMahan et al. [23] in 2017 and implemented in the language prediction model on smartphones, achieving the goal of not leaking user personal data. A unified machine learning model is trained using data sets distributed across multiple mobile phones. The user's mobile phone downloads the prediction model from the server, performs training and fine-tuning based on local user data, and uploads the fine-tuned model parameters to continuously optimize the global model of the server. In addition, federated learning is also widely used in fields such as finance, medical care, and the Internet of Things [24]. WeBank Yang et al. [25] and others expanded the federated learning model proposed by Google and extended it to various privacy protection learning scenarios.

The architecture of the federated learning system includes two types of roles: multiple participants and an aggregation server. Each participant has a complete data set of data features, and there is no intersection or a small intersection between data samples. They can join together to train a unified global model with better performance parameters. The training process of this system usually includes the following steps: Model initialization: the aggregation server generates an initial global model.

Model broadcast: the aggregation server shares the initial global model to all participants.

Model training: Based on the shared global model, participants use local data sets to train local models.

Collect parameters: Participants upload updated model parameters.

Model aggregation: The server aggregates the model parameters of each participant.

Update global model: The server broadcasts the aggregation.

The model parameters are continuously iterated in steps (2–6) until the global model loss function converges [24,26], thus completing the entire training process, as shown in Figure 1.

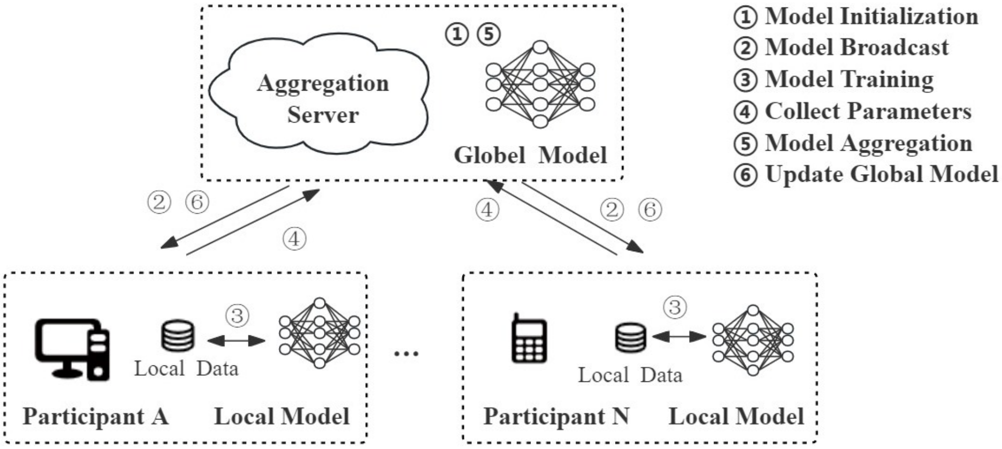


Figure 1. Federated learning model training process [25,27].

2.2. Federated Learning Attack Types

In federated learning scenarios, including the two processes of model training and model inference, the attacker's main goal is usually to destroy the training model or infer private information. Model training uses distributed computing methods, which brings great challenges to the security of the entire system. The academic community has conducted extensive research. According to different attack purposes, they can be divided into poisoning attacks and inference attacks. Poisoning attacks performed by malicious actors aim to affect the global model by controlling the local model behavior. Malicious actors can implement poisoning attacks targeting the training phase by controlling local model parameters and local training data. Poisoning attacks can be divided into data poisoning attacks [11–13] and model poisoning attacks [8–10]. Data poisoning attacks mainly focus on label reversal attacks and backdoor attacks. The purpose of inference attacks is to infer other information in the federated learning system through attack methods such as adversarial samples. This part is beyond the scope of this article.

Currently, the research on federated learning system attacks mainly focuses on the training phase. Malicious participants attack the federated learning system during the training phase, causing the global model failure and privacy leakage of participants. The federated learning model mainly faces security risk issues such as data poisoning, model poisoning and backdoor attacks during the training phase, as shown in Figure 2.

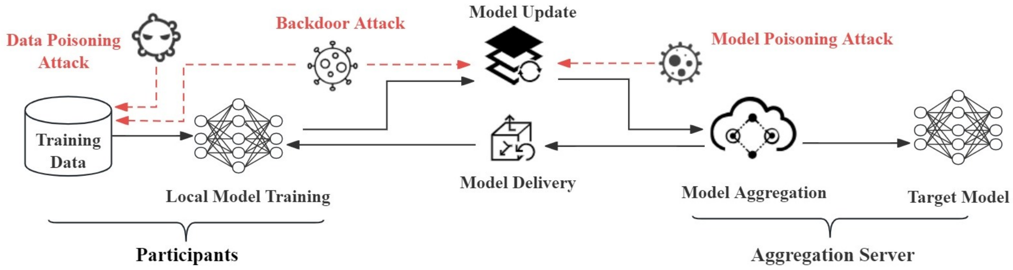


Figure 2. Security attacks faced during the training phase of the federated learning model [21].

The data poisoning attack (DPA) was first proposed by Biggio et al. [11]. In a federated learning scenario, attackers can implement data poisoning attacks by controlling participants or modifying participants' training data sets (such as adding forged data or modifying existing data, etc.), thereby reducing the accuracy of the model. However, aggregation algorithms weaken the impact of data poisoning on the global model. Depending on the attacker's purpose, data poisoning can be divided into two types: targeted and non-targeted. A non-targeted attack [27] is one in which the attacker aims to induce the model to produce as many incorrect predictions as possible, regardless of the category of data in which the errors occur, that is, a purely destructive behavior. A targeted attack means that the attacker intends to change the model's classification results of certain known test samples without pursuing the impact on the test results of other samples. Table 1 shows the effects of untargeted data poisoning attacks and targeted data poisoning attacks.

The model poisoning attack (MPA) refers to an attacker modifying the weight parameters of the model during the model update stage, which impacts the performance and reliability of the model [8–10]. As the aggregation server cannot verify the authenticity of model updates uploaded by participants, it creates opportunities for attackers to carry out model poisoning attacks. A malicious party can construct arbitrary model updates and send them to the server, thereby compromising the aggregated global model.

Table 1. Analysis of the Effect of Federated Learning Data Poisoning Attack [26].

Literature	Attack Type	Attack Methodology	Attack Evaluation Index	DataSet	Training Settings/ (Number–Items)	Result%
[28]	Untargeted poisoning	Utilizing the projected stochastic gradient ascent algorithm to maximize the experience loss of the target node	Model error rate	EndAD	6:*:no-iid	Base: 6.88 ± 0.52 Result: 28.588 ± 3.74
[11]	Untargeted poisoning	Predict changes in the SVM decision function caused by malicious input and use this ability to construct malicious data	Model error rate	MNIST	*:*:*	Base: 2–5 Result: 15–20
[13]	Targeted poisoning	Aim for successful poisoning in the final rounds and choose the right tag to flip	Maximum recall loss	CIFAR-10	50:*:iid	Base: 0 Result: 2:1.42; 20:25.4
[29]	Targeted poisoning	Utilize GAN technology to generate data and implement label flipping	Poisoning task accuracy rate	MNIST	10:*:no-iid	Base: 0 Result: 20:60±; 40:80±; 60:85±

Note: In the "Training Settings column", use the format x:y:z, x represents the participant, y represents the amount of data each participant has, z represents the division of the dataset (iid or no-iid), and "*" indicates unknown.

A backdoor attack (BA) involves injecting a backdoor into the target model. By activating a preset trigger, an attacker can make the model output specific labels when processing data with triggers, without affecting the inference results of normal data. In federated learning, attackers can contaminate the training set and upload malicious model updates to insert backdoors. Consequently, backdoor attacks in federated learning can be executed through data poisoning or model poisoning. It is important to note that triggers in backdoor attacks are primarily embedded through data poisoning attacks and model poisoning attacks. Therefore, privacy protection methods employed to defend against poisoning attacks can also be utilized to safeguard private information from backdoor attacks [30,31].

2.3. Data Poisoning Attack Methods

Data poisoning attacks can be divided into methods such as label flipping, target optimization, gradient optimization, and clean labeling based on technical implementation methods.

Label flipping [11–13] data poisoning by directly modifying the label information of the training data of the target category, while the characteristics of the data remain unchanged. Attackers can poison data by modifying data and data labels. Fung et al. [32] train a softmax classifier across ten honest clients, each holding a single digit partition of the original ten-digit MNIST dataset. Attackers achieve data poisoning attack goals by manipulating data labels, such as deliberately labeling the number 1 as a 7. Figure 3 shows how to control the data labeling process.

Optimization-based data poisoning will aim to solve a series of max/min problems [12,13,33]. In data poisoning, there is usually a target problem of making the most effective poisoning sample. This problem can be used to calculate the optimal set of data points for label poisoning, and can also be used to find the most efficient data modification scheme. Obviously, the performance of the attack mainly depends on the construction and solution strategy of the optimization problem.

Gradient-based data poisoning [34] makes the poisoned samples move in the direction of the gradient against the objective function L, thereby achieving the maximum poisoning effect. Figure 4 [11] is the result of a poisoning experiment using two numbers extracted from the MNIST handwritten digit data set and a poisoning ratio of 30%. After the attack, the overall accuracy of the model dropped significantly, the classification error rate increased by 10–30%, and the error rate increased with the number of iterative trainings.

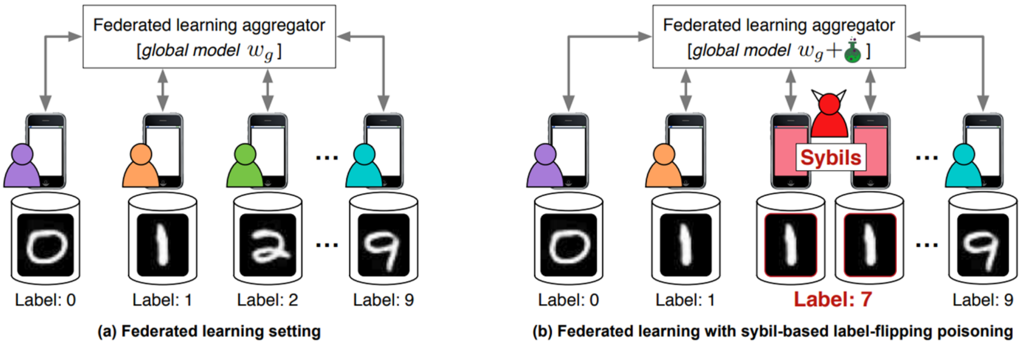

Federated learning aggregator
[global model w_g]
Label: 0
Label: 1
Label: 2
Label: 9
(a) Federated learning setting
Federated learning aggregator
[global model w_g+]
Sybils
Label: 0
Label: 1
Label: 7
Label: 9
(b) Federated learning with sybil-based label-flipping poisoning

Figure 3. Federated learning with and without colluding sybils mounting a sybil-based poisoning attack. In the attack (**b**), two sybils poison the model by computing over images of 1 s with the (incorrect) class label 7 [32].

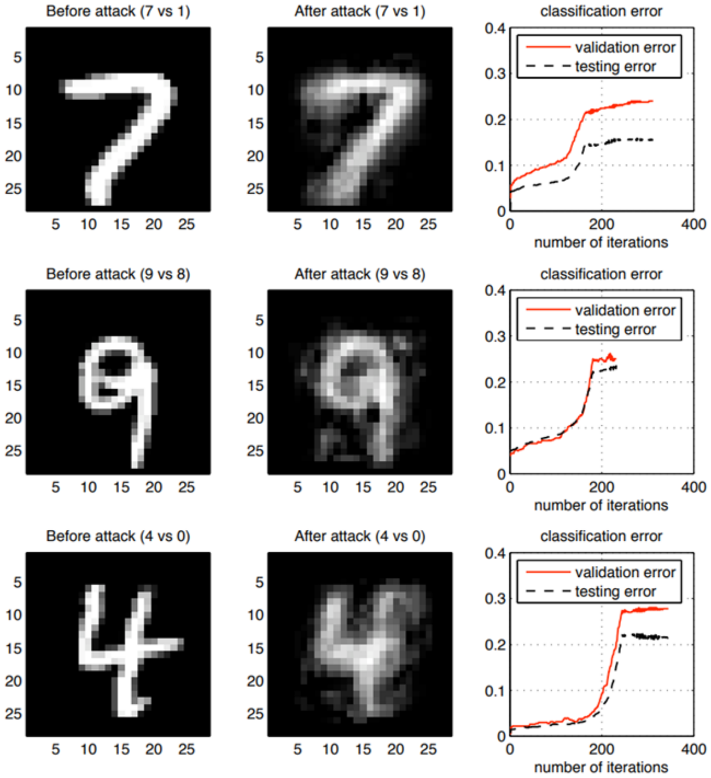


Figure 4. Modifications to the initial (mislabeled) attack point performed by the proposed attack strategy, for the three considered two-class problems from the MNIST dataset. The increase in validation and testing errors across different iterations is also reported [11].

Data poisoning with clean labels [35–37] will cause the label of the poisoned image to be consistent with the visual sense, but the test image will be misclassified. The attack accuracy of this method is very high, and the infection rate is also very low. Only a very small amount of pictures need to be poisoned to significantly increase the attack success rate. Traditional label flip poisoning is easy to detect, for example: base image (dog) + perturbation = poisoned image, but the poisoned image still looks like a dog to the human eye. If the poisoned image is tagged as a fish, it will be easily discovered by data cleaners. In the clean label data poisoning attack, the label of the poisoned image is consistent with the visual sense. Even if the dog image is added with noise, the poisoned image obtained can still be labeled as a dog.

2.4. Federated Learning Privacy Protection Technology

Currently, federated learning utilizes perturbation technology and encryption technology to safeguard privacy. Perturbation technology, specifically differential privacy technology, is employed to offer enhanced privacy protection. Encryption technology, on the other hand, utilizes techniques such as secure multi-party computation and homomorphic encryption to achieve privacy preservation. These techniques have been widely utilized for privacy protection in traditional machine learning.

Differential privacy (DP) is an output privacy protection model initially proposed by Dwork et al. [38] in 2006. It quantifies and restricts the leakage of personal information. The fundamental concept of differential privacy is to prevent attackers from extracting individual information from the dataset by obfuscating query results. This makes it impossible for attackers to discern an individual's sensitivity from the query results. In essence, the output of the function remains unaffected by any specific record in the dataset. Hence, differential privacy is effective in countering membership inference attacks. Compared to encryption-based technology, differential privacy technology reduces the communication overhead and enhances the transmission efficiency.

Homomorphic encryption (HE) is a technology proposed by Rivest et al. [39] that performs algebraic operations on data to obtain encrypted results. By decrypting this result, the obtained outcome is consistent with performing the same operation on the original plaintext. This technology is significant as it addresses the confidentiality issue when entrusting data and operations to a third party. It finds extensive application in various cloud computing scenarios.

Secure multi-party computation (SMPC) was formally introduced in 1982 by Yao Qizhi et al. [40], a Turing Award winner and academician of the Chinese Academy of Sciences. Its purpose is to collectively compute the results of a function using private inputs from each party, without revealing these inputs to others. A secure multi-party computation ensures that participants can obtain accurate calculation results while keeping their private inputs confidential.

2.5. Data Security Situational Awareness

Situational awareness [41] originated from the U.S. military's research in military confrontation. It involves extracting elements within a specific time and space range, understanding their meaning, and predicting their potential effects. Situational awareness can be observed as a cognitive process [42] that utilizes past experience and knowledge to identify, analyze, and comprehend the current system situation. In the military domain [43], it enables us to understand both ourselves and the enemy, thereby increasing our chances of survival in numerous battles. As can be observed in Figure 5 [44], the perception is a kind of cognitive mapping. The so-called cognitive mapping means that decision makers use related technologies such as data fusion, risk assessment, and visualization to denoise and integrate information in different formats obtained from different locations to obtain more accurate and comprehensive information. Subsequently, the decision-maker continuously extracts semantics from this information and identifies the elements that need attention and their intentions. Decision makers can effectively assess their impact on the system in real time.

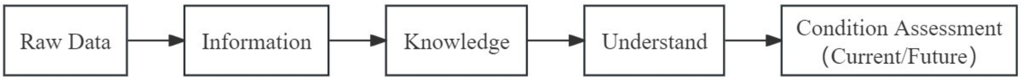


Figure 5. Situation awareness cognitive mapping process [44].

The objective of data security situational awareness [45] is to apply the theory and methods of situational awareness to the field of data security. This enables data security personnel to effectively monitor the security status of the entire data in a dynamically changing environment and provide decision-making support for management. The key to achieving data security situational awareness is by proactively monitoring security risks related to operations, accounts, hosts, and shares through database logs collected via interfaces, Syslogs, deployment probes, etc. These logs can provide an early warning for potential security threats, as shown in Figure 6.

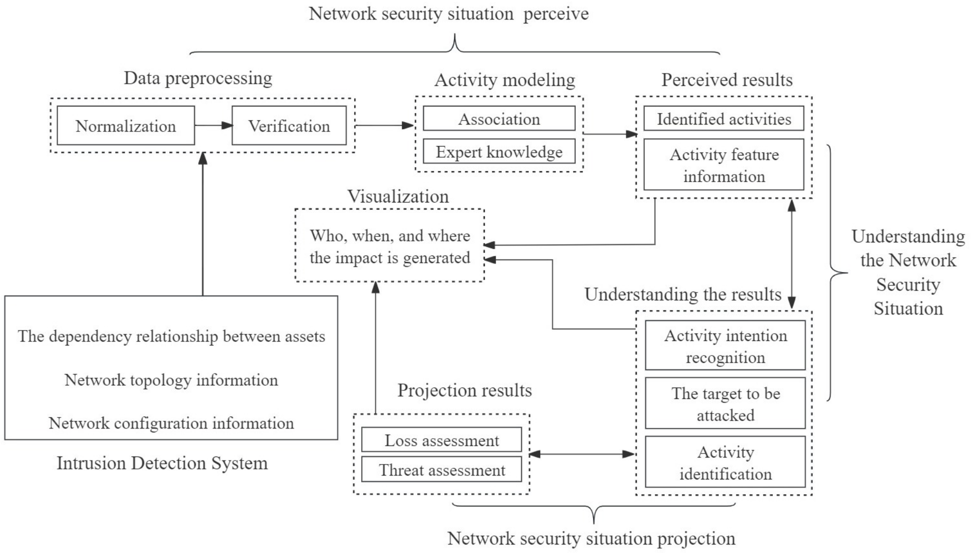

Network security situation perceive
Data preprocessing
Normalization
Verification
Activity modeling
Association
Expert knowledge
Perceived results
Identified activities
Activity feature information
Visualization
Who, when, and where the impact is generated
Understanding the Network Security Situation
Understanding the results
Activity intention recognition
The target to be attacked
Activity identification
The dependency relationship between assets
Network topology information
Network configuration information
Intrusion Detection System
Projection results
Loss assessment
Threat assessment
Network security situation projection

Figure 6. Network situation awareness model [45].

3. The Challenge of Predicting Data Poisoning Attack Situation

The proposal of federated learning data poisoning attacks has attracted widespread attention in related industries [46–48]. However, there are still many challenges that need to be addressed in the situation prediction. The core issues include system architecture fragility, communication efficiency shortcomings, computing resource consumption and situation prediction robustness. These problems seriously restrict the further development and application of federated learning.

3.1. System Architecture Vulnerability Issues

Participants in the federated learning model need to continuously communicate and collaborate with the aggregation server. According to the research results, vulnerabilities exist in aggregation servers, participants, and communication protocols [24]. This article only focuses on aggregation servers and parties involved in data poisoning attacks.

(1) The aggregation server is the nerve center of the federated learning model. It is responsible for initializing model parameters, aggregating model updates of participants and distributing the global model. If the server is compromised, the attacker can immediately disable the situation prediction ability and release malicious participants at will, thereby conducting data poisoning attacks and affecting the quality of the global model [49]. In addition, the aggregation server can utilize the maximum a posteriori principle (MAP) technology of the model inversion attack (MIA) to reconstruct participants' training data, sensitive attributes or input data, potentially compromising their privacy [50]. Zhu et al. [51] introduced a label-only model reverse attack method, which uses labels to estimate the true confidence. The attack process is illustrated in Figure 7.

(2) Participants refer to the nerve endings of the federated learning model. Once a data poisoning attack occurs, they can destroy the aggregated global model by uploading updates to the model [11]. Currently, the participants in federated learning applications are mainly individual users. Compared with aggregation servers, individual users have weaker security protection measures, and the cost of attack is relatively low [52]. Attackers can easily join the federated learning training process by invading ordinary users or registering new users. They can attack the global model by forg-

ing local data or modifying model updates [25]. Wang et al. [46] proposed a data poisoning attack framework for the federated learning autonomous driving steering control (ATT-FLAV) based on the Bandit algorithm, which is used for dynamic data poisoning attacks against nonlinear regression models, as shown in Figure 8. In addition, attackers can also join forces with other malicious parties to launch attacks to enhance the attack effect. At the same time, individual users, as participants, also have vulnerabilities and unavailability when deploying data poisoning attack situational awareness probes. Therefore, participants can be said to be the most vulnerable link in a federated learning system [53].

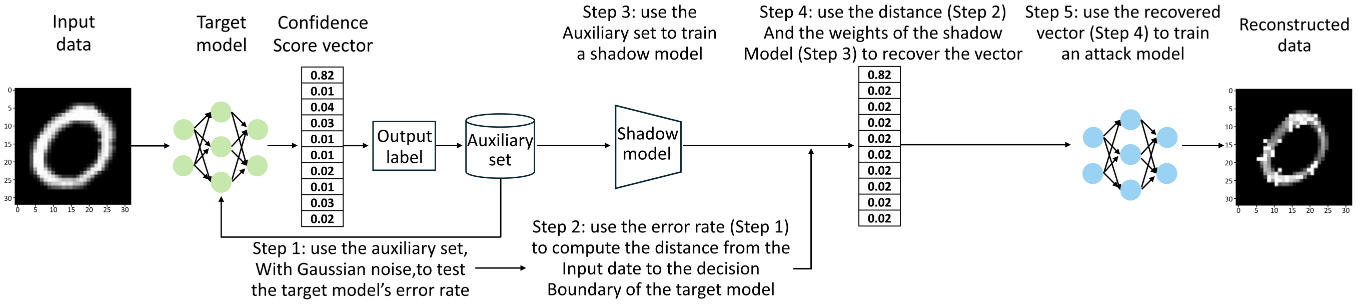

Input data
Target model
Confidence Score vector
0.82
0.01
0.04
0.03
0.01
0.01
0.02
0.01
0.03
0.02
Output label
Auxiliary set
Step 3: use the Auxiliary set to train a shadow model
Shadow model
Step 4: use the distance (Step 2) And the weights of the shadow Model (Step 3) to recover the vector
0.82
0.02
0.02
0.02
0.02
0.02
0.02
0.02
0.02
0.02
Step 5: use the recovered vector (Step 4) to train an attack model
Reconstructed data
Step 1: use the auxiliary set, With Gaussian noise,to test the target model's error rate
Step 2: use the error rate (Step 1) to compute the distance from the Input date to the decision Boundary of the target model

Figure 7. Overview of label-only attack method [51].

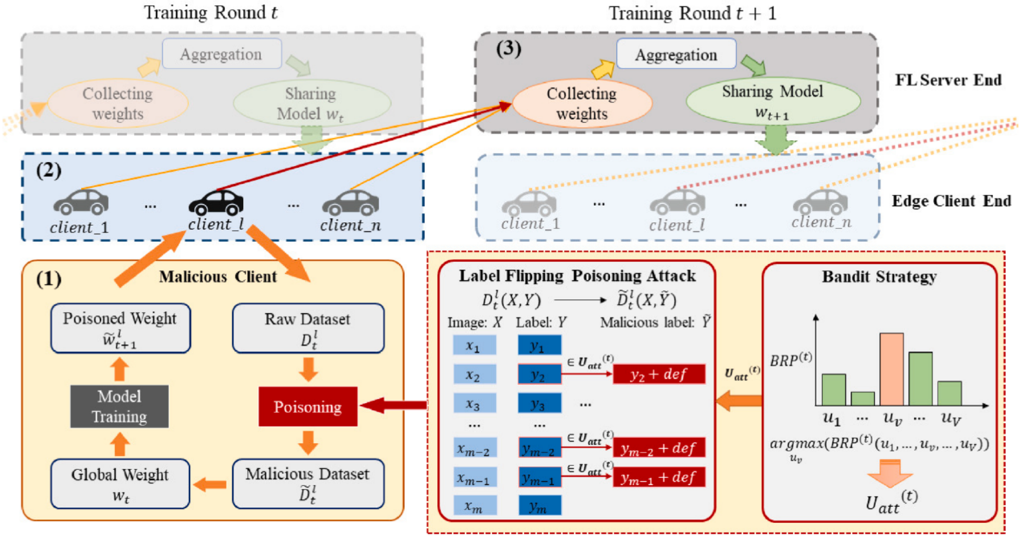

Training Round t
Aggregation
Collecting weights
Sharing Model w_t
Training Round $t+1$
(3)
Aggregation
Collecting weights
Sharing Model w_{t+1}
FL Server End
(2)
client_1
client_l
client_n
Edge Client End
(1)
Malicious Client
Poisoned Weight $\tilde{w}^l_{t+1}$
Raw Dataset D^l_t
Model Training
Poisoning
Global Weight w_t
Malicious Dataset $\tilde{D}^l_t$
Label Flipping Poisoning Attack
$D^l_t(X,Y) \longrightarrow \tilde{D}^l_t(X,\tilde{Y})$
Image: X
Label: Y
Malicious label: $\tilde{Y}$
x_1 y_1
x_2 y_2 $\in U_{att}{}^{(t)}$ $y_2 + def$
x_3 y_3
x_{m-2} y_{m-2} $\in U_{att}{}^{(t)}$ $y_{m-2} + def$
x_{m-1} y_{m-1} $\in U_{att}{}^{(t)}$ $y_{m-1} + def$
x_m y_m
$U_{att}{}^{(t)}$
Bandit Strategy
$BRP^{(t)}$
u_1 ... u_v ... u_V
$\underset{u_v}{argmax}(BRP^{(t)}(u_1, \ldots, u_v, \ldots, u_V))$
$U_{att}{}^{(t)}$

Figure 8. Framework of Data Poisoning Attack on Federated Learning for Autonomous Steering Angle Control System. In the first step, malicious vehicles are subjected to a label-flipping data poisoning attack that alters some of the training data labels (1). Once the label-flipping attack is completed in the first step, the malicious vehicles use the poisoned training data to train their own models, while honest vehicles train their models with normal data (2). After training, all client models are uploaded to the server end and merged to create a new global model, which initializes client models in the next round of federated learning, as depicted in the numbered Block (3) at the top [46].

3.2. Communication Efficiency Shortcomings

In federated learning, the aggregation server and remote participants need to frequently communicate to interact with and update the model. Training involves multiple devices, which may cause huge bandwidth pressure on the communication network [54]. However, if the aggregation server cannot detect malicious participants in time, it can easily lead to global model contamination or privacy leakage issues. The training time of the global model is mainly composed of the data processing time and communication transmission time. As the computing power of the computer equipment increases, the data processing time gradually decreases, and the communication transmission efficiency of federated learning becomes the main factor limiting the training speed [55]. In an Internet environment, updating and uploading a large number of local models will cause excessive communication overhead on the aggregation server and cannot meet normal application requirements. Furthermore, adjacent model updates may contain many duplicate updates or updates that are not related to the global model [56].

Anomaly detection is crucial for the safety of the training process and predicting situations. Paudice et al. [57] suggested using data pre-filtering and outlier detection to prevent poisoning attacks and mitigate their impact on the system. However, the centralized anomaly detection that detects client data can pose significant privacy risks and escalate computational communication expenses. In order to predict data poisoning attacks in federated learning, monitoring modules must be installed in all participants, and they must maintain a constant interaction with the aggregation server. This leads to an increased communication overhead.

3.3. Computing Resource Consumption Problem

Existing federated learning defense mechanisms generally do not take into account the problem of limited computing resources. Some validation mechanisms consume a lot of resources to validate all updates, causing time delays and affecting the entire training process. Anomaly detection is an important means to ensure the security of the training process and achieve a situation prediction. Fang et al. [9] demonstrated the poisoning of local models by launching a poisoning attack on the local model of the client, which resulted in a significant increase in the test error rate of the global model. Regarding the necessity of defense, Zhao et al. [19] deployed a generative adversarial network (GAN) on the server side to generate client model parameters for auditing a data set, and used the data set to check the accuracy of the participant model and determine whether there was a poisoning attack. However, the above detection algorithm needs to consume a large amount of computing resources of the aggregation server to review the local models of the participants, which causes the participants to waste a lot of resources in federated learning.

3.4. Situation Prediction Robustness Issues

Federated learning commonly suffers from heterogeneous data, heterogeneous models, and insufficient generalization capabilities; this results in poor robustness in predicting data poisoning attack situations.

(1) Data heterogeneity and model heterogeneity: In federated learning scenarios, the data of the participants are usually heterogeneous, that is, they may come from different data sources, with different data structures, data types and data distributions. Wu et al. [58] summarized the challenges faced by federated learning into three aspects. First, there is heterogeneity in the storage, computing and communication capabilities of various participants. Secondly, the non-independent and identically distributed local data of each participant raises the problem of data heterogeneity. Data heterogeneity may reduce the accuracy and generalization ability of the model. Finally, the models required by each participant according to their application scenarios lead to the problem of model heterogeneity. Model heterogeneity refers to the differences in model structures, parameters, and hyperparameters used by different participants, which affects the performance and effect of the model. Both data heterogeneity and model heterogeneity may lead to the non-convergence of the global model [59], which may have an impact on the situation prediction effect and performance of federated learning data poisoning attacks.

(2) Insufficient generalization ability. The current research on defense mechanisms against poisoning attacks in federated learning primarily focuses on the Byzantine robust aggregation algorithm [60], which is designed based on a central server to identify and eliminate potential poisoning participants. Other studies [61–64] utilize clustering algorithms or weight functions implemented by the server to mitigate or remove po-

tential poisoning participants who deviate from the majority of participants' updates. Another approach that is discussed in the literature [65] is the FoolsGold algorithm, which assumes that malicious updates have lower randomness compared to normal updates, providing defense against poisoning attacks where over half of the data from poisoned participants exceeds half. While these methods offer some protection against poisoning attacks, accurately assessing model updates submitted by participants is challenging due to the absence of real data sets on the server. As a result, these approaches may struggle to address poisoning attacks when the number of poisoned participants surpasses that of normal participants, ultimately leading to a reduced robustness in the situation prediction.

4. Countermeasures for Predicting Data Poisoning Attack Situation

By analyzing federated learning data poisoning attacks, we can conclude that the main security threats to federated learning currently come from the participating parties. However, due to the large number, wide range, strong heterogeneity, and difficulty of controlling the participants, the existing research mainly focuses on improving the aggregation algorithm of the server. Having a safe and reliable aggregation algorithm can ensure that the global model in the system can converge correctly even if there are malicious nodes [24].

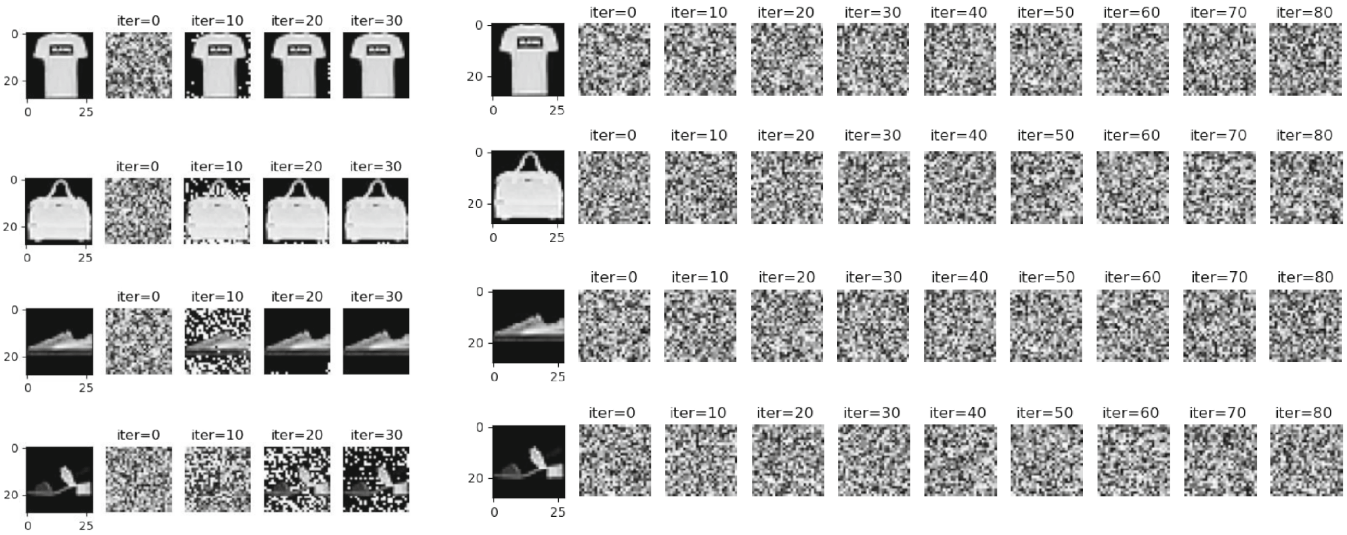
4.1. Build a Trusted Federated Learning System

Judging from the development trends in the past two years, the federated learning research focuses on how to balance data privacy protection, model performance and learning efficiency, which is the core issue of trustworthy federated learning [66]. Trusted federated learning is an enhanced federated learning method.

(1) Optimize the federated learning model structure. To prevent data poisoning attacks, some researchers have taken inspiration from the security measures of centralized learning. They propose modifying the structure of the federated learning model to increase the model's robustness and decrease the damage caused by contaminated data. In [67], a method of removing the aggregation server is proposed, and the corresponding tasks are handed over to the participating nodes; the blockchain replaces the removed aggregation server as a component of the model and information source. In a decentralized federated learning system, participants communicate with each other without the coordination of an aggregation server. Lu et al. [68] proposed a decentralized federated learning (DFL) method to defend against gradient reversal attacks, and demonstrated its security capabilities in a depth gradient leakage (DLG) environment, as shown in Figure 9. Li et al. [69] performs a cluster analysis on model parameters to distinguish good and bad models, and then detect potential malicious participants. Such a defense idea can be applied to detect malicious aggregation servers, and determine through a comparative analysis whether the global model update issued by the aggregation server after each iteration is under attack. Chang et al. [70] proposed a new federated learning framework called Cronus. This framework replaces model parameters with data labels, solving security risks caused by sharing parameters and enabling knowledge transfer.

(2) Improve the anomaly detection capabilities of participants. For data poisoning attacks, there are differences between poisonous samples and normal samples. The most intuitive situational awareness defense strategy is to detect and reject poisonous input samples. Liu et al. [71] used an anomaly detection algorithm to detect toxic samples and rejected their identification. Behavior-based defense ideas were proposed in the literature [15,62,72,73]. This idea identifies potential malicious participants by analyzing the behavioral characteristics of the models uploaded by participants, such as the similarity between local updates and global updates and the error rate after the aggregation of partial models. Udeshi et al. [74] proposed to build a trigger interceptor using the dominant colors in the input image and use it to detect and block the corresponding backdoor trigger. Kieu et al. [75] proposed a method for detecting

anomalies in time series datasets based on recursive autoencoders that reduces the impact of overfitting on outliers. To further achieve robust and efficient anomaly detection in time series under unsupervised settings, a variational recurrent encoder model [76] can separate anomalies from normal data without relying on anomaly labels. Table 2 shows the federated learning data poisoning defense effect.



(a) attack results on CFL

(b) attack results on DFL

Figure 9. The case study of an image restoration attack on the fashion-MNIST dataset. The picture above shows the image restoration result of DLG on CFL and DFL. Here, the attacker restores four images of the target t-shirt, bag, sneaker, and sandal. In each subfigure, the first column represents the original images, followed by the images restored by the DLG model as the number of DLG training rounds increases. From figure (**a**), it can be found that the CFL method is vulnerable to an DLG attack, where the training data are recovered clearly within 30 rounds. By comparison, the DFL method can resist the image restoration attack by DLG much better. Figure (**b**) shows that the original image is still not restored after 80 rounds with the DFL model. This case study demonstrates that our DFL method achieves better privacy against DLG attacks [68].

Table 2. Analysis of federated learning data poisoning defense effect [26].

Literature	Attack Type	Defensive Thinking	Defense Mode	Defensive Indicators	Defense Result
[62]	Untargeted poisoning	Behavior based	Aggregating models using robust distributed gradient descent algorithms	Model error rate	Base: 10± Attack: 60± After: 10±
[15]	Targeted poisoning	Behavior based	Aggregating models using robust distributed gradient descent algorithms	Model accuracy	Base: 94.3± Attack: 77.3± After: 90.7±
[69]	Targeted poisoning	Based on clustering	Use clustering algorithms to identify malicious models	Model accuracy	Base: 78± Attack: 76±; 74.5± After: 78±; 77.5±
[72]	Targeted poisoning	Behavior based	Determine the malicious model based on the cosine similarity between the local model and global model	Model error rate	Base: 2.80 ± 0.12 Attack: unknown After: 2.99 ± 0.12±, 2.96 ± 0.15; 3.04 ± 0.14
[73]	Targeted poisoning	Behavior based	Determine the malicious model based on the cosine similarity between the local model and global model and combined with the reputation mechanism	Model accuracy	Base: unknown Attack: unknown After: 83.11; 81.23

4.2. Strengthen Data Traffic Monitoring

When abnormal traffic and data leakage are detected in federated learning scenarios, situational warnings and closed-loop disposals can be triggered promptly for real-time monitoring and analysis of data traffic. The federated learning is affected by heterogeneous equipment among participants and limited network bandwidth, causing computing and communication efficiency to become the biggest challenges hindering its implementation. The main factor affecting the efficiency of federated learning algorithms is the communication cost of passing parameters between the client and central service. The current research mainly reduces the single communication cost and total communication times through model compression, reducing the model update frequency and client selection method [77], and thereby reducing the communication complexity of the algorithm.

(1) Based on model compression methods, model compression [78] refers to the method of streamlining the model, which is carried out on the premise of ensuring the accuracy of the model. After model compression, the amount of network parameters and calculations are usually reduced. Model compression can decrease communication overhead and optimize federated learning at the cost of model performance [77]. Xu et al. [79] proposed a federated ternary quantization algorithm, which optimizes the learning model in the client through self-learning. This algorithm aims to solve the problem of updating a large number of redundant parameters in the federated learning process. The authors proved that the convergence of this algorithm is improved. Shah et al. [80] consider the compression techniques of the server model to address the downstream communication and compression techniques of the client model to solve upstream communication, both of which play a crucial role in the development and maintenance of sparsity across communication cycles and have proven to be effective. On the basis of model compression, Caldas S et al. [81] proposed a federated random deactivation (dropout) method to select a subset of the global model to update parameters. Compared to existing work, communication between server and client is reduced by 14× and client-to-server communication is reduced by 28× for EMNIST. Figure 10 shows that for CIFAR-10, server-to-client communication is saved 10 times and client-to-server communication is saved 21 times.

(2) Methods for reducing model update frequency primarily enhance performance by increasing participant calculations and improving parallel computing capabilities. The FedAvg algorithm proposed by Mcmahan et al. [23] combines a local stochastic gradient descent with a server that performs model averaging. The client first iterates local updates multiple times and then sends the local iteration results to the server. This algorithm satisfies the independent and simultaneous data distribution. Good training results can be achieved under distribution assumptions. However, the FedAvg algorithm only has an obvious optimization effect when the data are independent and identically distributed, and its performance is poor when the data are not independently and identically distributed. The FedProx algorithm proposed by Li et al. [82] can dynamically update the number of local calculations required by different clients in each round. It does not require the participants to unify the number of calculations in each update. Therefore, this algorithm is more suitable for non-IID joints. Regarding modeling scenarios, Zhou et al. [83] started from the perspective of the algorithm framework, parallelized communication and training, and proposed the Overlap–FedAvg algorithm based on the set hierarchical computing strategy, data compensation mechanism and Nesterov Accelerated Gradient algorithm.

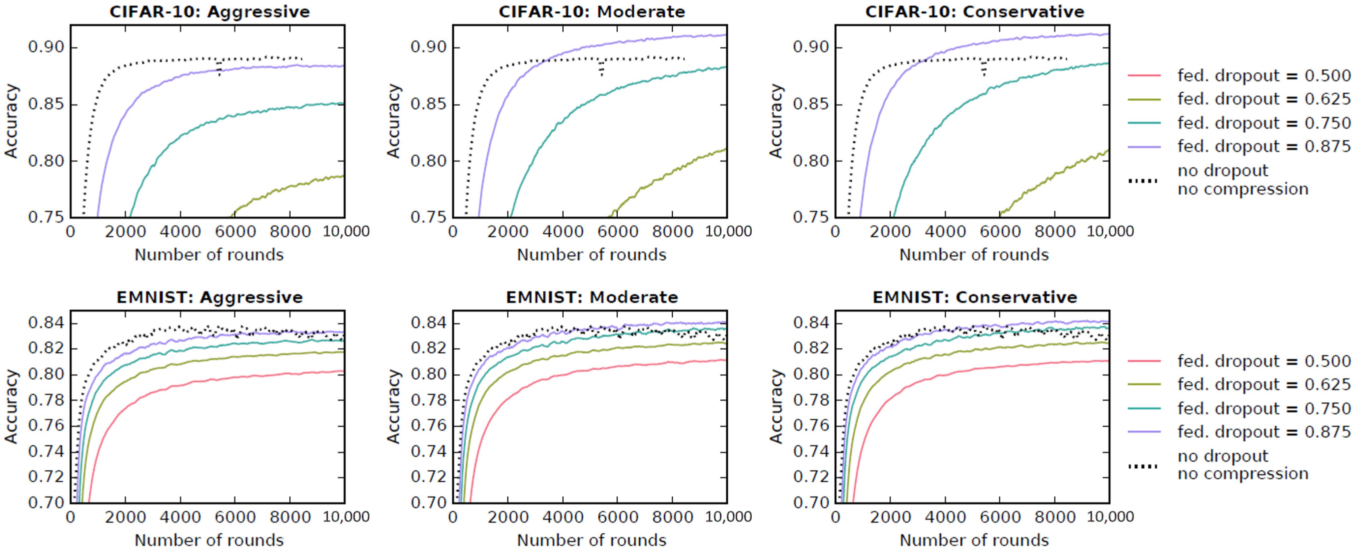


Figure 10. Effect of using both compression and Federated Dropout on CIFAR-10 and EMNIST [81].

This algorithm can be orthogonal to many other compression methods; in order to maximize the use of the cluster, the experimental results are shown in Table 3. This overlapping FedAvg framework is parallel and can greatly speed up the federated learning process while maintaining almost the same final accuracy as FedAvg. It is particularly useful for large models and can handle scenarios where the client's network connection is slow or unstable. Additionally, it is robust against imbalanced and non-IID data distributions and can reduce the number of communication rounds needed to train deep networks on dispersed data.

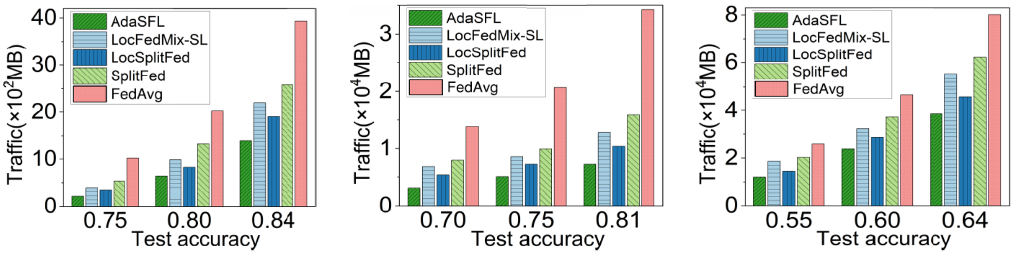
Table 3. Comparison of average wall-clock time of Overlap–FedAvg and FedAvg for one iteration [84].

Model	Dataset	Parameters	FedAvg	Overlap–FedAvg
MLP	Mnist	199,210	31.2	28.85 (↓7.53%)
MnistNet	Fmnist	1,199,882	32.96	28.31 (↓14.11%)
MnistNet	Emnist	1,199,882	47.19	42.15 (↓10.68%)
CNNCifar	Cifar10	878,538	48.07	45.33 (↓5.70%)
VGG	Cifar10	2,440,394	64.4	49.33 (↓23.40%)
ResNet	Cifar10	11,169,162	156.88	115.31 (↓26.50%)
ResNet	Cifar100	11,169,162	156.02	115.3 (↓26.10%)
Transformer	Wikitext-2	13,828,478	133.19	87.9 (↓34.0%)

(3) When dealing with a large number of clients, the federated learning algorithm's communication with each client can result in low algorithm efficiency based on client selection methods [77]. The existing research addresses this problem by selecting a certain number of clients among many clients and training them as representatives to reduce the communication overhead and optimize the algorithm efficiency. Huang et al. [85] dynamically select clients in each round based on multi-armed bandits; Lai et al. [86] further implemented a federated learning client selection algorithm based on the exploration–exploitation strategy.

4.3. Explore Collaborative Support to Ensure Computing Power

To mitigate the issue of compromised Federated Learning data poisoning detection capabilities due to insufficient computational power, optimization techniques such as Model Parallelism and Zero Redundancy Optimizers (ZeRO) [87] can be employed. These distribute extensive model parameters across multiple GPUs. To alleviate the GPU burden, techniques like tensor offloading and optimizer offloading [88] make use of cost-effective CPU and memory resources. Under the premise of protecting the user privacy, these can reduce the computational/communication burden on resource-constrained end devices. Integrating data parallelism and model parallelism in Edge Computing, Split Federated Learning (SFL) is becoming a practical and popular method for distributed data model training. To address the heterogeneity and node dynamism of federated learning systems, an efficient SFL method (AdaSFL) was proposed in reference [89]. It integrates control over the local update frequency and data batch size to enhance the model training efficiency. Compared to baseline methods, AdaSFL can reduce the completion time by about 43% and decrease the network traffic consumption by approximately 31% while achieving a similar test accuracy, as shown is Figure 11.



(a) EMNIST (b) CIFAR-10 (c) IMAGE-100

Figure 11. Network traffic consumption of AdaSFL and the baselines when achieving different target accuracies on the three datasets in experiments [89].

4.4. Improve the Robustness of Situational Awareness

The robustness of data poisoning attack situational awareness is based on a federated learning-based framework. At present, there are many related research results in the academic community, mainly focusing on two aspects: personalized federation optimization and the optimization of defense against data poisoning attacks.

(1) Regarding personalized federation optimization, reference [84,90] proposed an effective method to alleviate the heterogeneity of data and models through a personalization at the device, data and model levels, and provide high-quality personalized models for each device, namely personalized federated learning. This method is widely used in personalized smart medical care [91], smart home services [92], location-aware recommendation services [93] and other scenarios, so personalized federated learning has attracted much attention.

In terms of device heterogeneity, Xie et al. [94] proposed a new asynchronous federated optimization algorithm (FedAsync), and proved the convergence of FedAsync through a theoretical analysis and experimental verification. For strongly convex, non-strongly convex problems, and restricted non-convex problems, this method can converge linearly to the global optimal solution. Unlike FedAvg, delayed feedback updates are not deleted and the central server can receive updates from client devices at any time. When the delay is small, FedAsync converges much faster than FedAvg. When the delay is severe, FedAsync still has a similar performance to FedAvg.

In terms of the model heterogeneity, Kulkarnit et al. [95] divide different personalized federated learning methods into Adding User Context [96], Transfer Learning [97], Multi-task Learning [98], Knowledge Distillation [99], Meta-Learning [100], Base + Personalization Layers [101] and Mixed global and local models [102]. Chen et al. [103] proposed a framework called PFKD, which solves the problem of model heterogeneity through knowledge distillation technology and solves the problem of data heterogeneity through personalized algorithms to achieve more personalized federated learning. Dinh et al. [104] proposed a personalized federated learning algorithm called pFedMe. This algorithm introduces Moreau Envelopes as a client regularization loss function, which separates the personalized model optimization from global model learning. This allows pFedMe to update the global model in a similar order to FedAvg while also optimizing the personalized model in parallel based on the local data distribution of each client. Through synthetic data set experiments, as shown in Figure 12, the pFedMe personalized model pFedMe-PM has an accuracy rate higher than its global models pFedMe-GM, Per-FedAvg and FedAvg by 6.1%, 3.8% and 5.2% respectively.

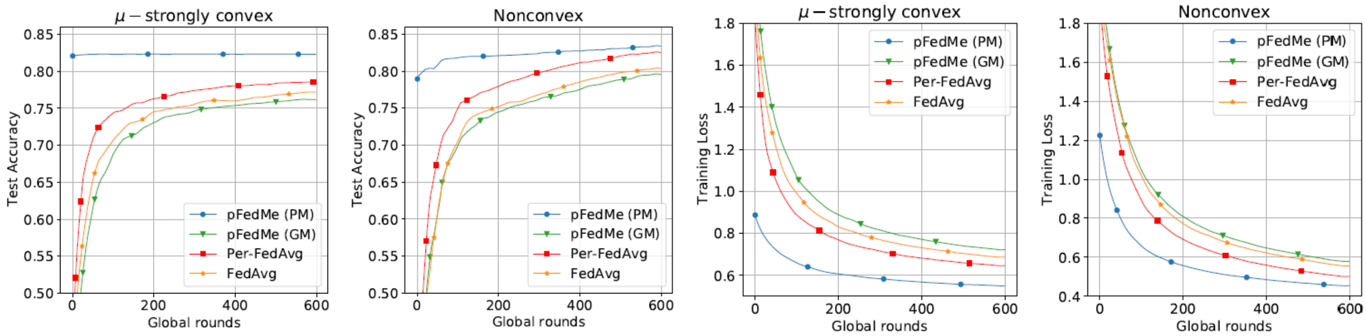


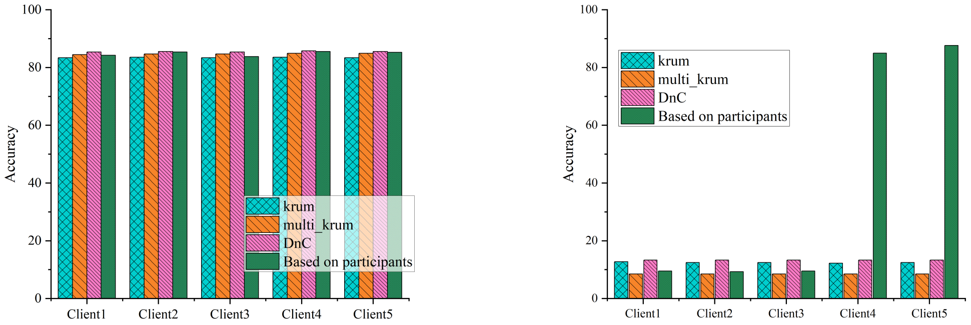
Figure 12. Performance comparison of pFedMe, FedAvg and Per-FedAvg in μ-strongly convex and nonconvex settings using Synthetic [104].

In terms of data heterogeneity, Shen et al. [105] proposed a new federated mutual learning framework, allowing each client to train a personalized model that takes into account the data heterogeneity. When solving the problem of model heterogeneity, a memetic model is introduced as an intermediary between the personalized model and the global model, and the knowledge distillation technology of deep mutual learning is used to transfer knowledge between the two heterogeneous models.

In terms of the comprehensive optimization, Wu et al. [58] proposed a collaborative cloud edge framework, PerFit, for personalized federated learning. Intelligent IoT applications benefit from a personalized federated learning framework that resolves equipment issues. By solving data and model heterogeneity issues, PerFit can be ideal for complex IoT environments while ensuring user privacy by default for large-scale real-world deployments.

(2) Regarding the optimization of defense against poisoning attacks, the distributed training structure of federated learning is vulnerable to poisoning attacks. Existing methods mainly design security aggregation algorithms for the central server to defend against poisoning attacks, but they require the central server to be trustworthy and the number of poisoned participants to be lower than normal participants. Liu et al. [60] proposed a poisoning attack defense method based on federated learning participants. The main idea is to regard participants as independent executors of defense strategies under the framework of the FedAvg algorithm. During local training, the participant uses a difference calculation function (such as mean square error) to determine the difference loss weight between their local model and the global model parameters. This difference loss weight and function is then integrated into the training loss function. This allows for adaptive personalized training that uses the difference between the global model and the participant's local model.The federated learning training accuracy of this algorithm is better than poisoning attack defense methods such as Kurm, multi-Kurm, and DnC, as shown in Figure 13. When the proportion of poisoned participants exceeds half, normal participants can still defend against poisoning attacks.

In response to serious security challenges such as single points of failure and the lack of privacy that centralized federated learning frameworks still face, Wang et al. [106] proposed a personalized federation algorithm based on permissioned blockchain. By conducting experiments on the MNIST data set, it is proven that high-precision protection against poisoning attacks can be achieved and applied to edge computing scenarios. Bitoye [107] et al. suggested that adding differential privacy and self-normalization layers to the local model of each client is sufficient for federated learning without requiring any changes to the communication protocol or optimization algorithm. Specifically, the differential privacy noise is first used to increase the randomness and uncertainty of the model and reduce the impact of adversarial samples [108]; then, self-normalization technology is used to maintain the stability and convergence of the model and improve the generalization ability of the model. Finally, a simple and scalable defense solution is implemented, which effectively improves the robustness of the model.



(a) The accuracy of partial participant in poisoning is 40%

(b) The accuracy of partial participant in poisoning is 60%

Figure 13. Liu et al. [60] examines the accuracy of some participants on the MNIST dataset, when there are varying proportions of poisoned participants. It was found that when the proportion of poisoned participants is 40%, the Krum, Multi-Krum, and DnC algorithms are comparable to the algorithm proposed in this article. The accuracy of each participant model is more than 60%, resulting in better results.

5. Future Research Directions

Through the above review, it is found that the implementation of industries such as autonomous driving, smart medical care and location recommendation services has raised new challenges for federated learning research. There are several factors that could influence the prediction of federated learning data poisoning attacks. Additionally, these factors may be correlated with one another. More research is needed to strike a balance between federated learning and accurately predicting data poisoning attacks. To be more specific, we can begin by focusing on the following areas.

(1) We can enhance the prediction capability of data poisoning attacks by utilizing generative artificial intelligence (Generative AI). Generative AI can generate synthetic training data, strengthen attack detection models, simulate data distribution and heterogeneous data, and evolve attack strategies. These capabilities have a potential application value and can help improve the ability of federated learning to predict data poisoning attacks. For example, using generative AI models such as GANs can generate large amounts of synthetic training data, including both normal data and malicious poisoning data. This helps train more robust federated learning models to better identify and defend against real-world poisoning attacks. Generative AI can also generate samples under different data distributions to simulate attack scenarios in heterogeneous environments. This helps federated learning models better adapt to data diversity and remain robust in the face of unknown attacks.

(2) We can utilize large language models to enhance our capabilities in predicting data poisoning attacks. Large language models have demonstrated excellent performances in the field of natural language processing (such as Generative Pre-trained Transformer). In recent years, this type of model has also been used in research fields such as federated learning data poisoning attack pattern recognition, federated learning participant trust assessment and secure communication protocol design; it aims to improve its capabilities in data poisoning attack prediction and defense. For example, large language models can analyze and understand communication and update patterns in federated learning networks, training the model to identify the differences between normal and abnormal updates. Utilizing the powerful word processing capabilities of large language models, potential attack patterns can be discovered from

model update logs submitted by participants, thereby identifying and preventing data poisoning attacks in advance.

(3) Improve the prediction ability of data poisoning attacks through a graph neural network. Graph neural networks have obvious advantages in processing complex network structure data and have been widely used in many fields in recent years. GNN has demonstrated potential in enhancing federated learning data poisoning attack prediction capabilities. This can be achieved by identifying abnormal communication patterns, modeling participant networks, analyzing global/local structures, and more. Currently, the project team is conducting research on federated learning data poisoning attacks based on graph neural networks. Taking the graph neural network analysis federated learning architecture as an example, GNN can not only capture the local characteristics of each node, but also understands the global network structure by aggregating the neighbor information. This capability allows GNN to assess the security of the network as a whole, as well as reveal signs of data poisoning attacks in detail.

6. Conclusioins

This article elaborates on the concepts of federated learning technology, attack types, data poisoning methods, etc. Four major problems in federated learning were identified: fragile system architecture, low communication efficiency, large consumption of computing resources, and poor situation prediction robustness. Response strategies for each type of problem were introduced.

In terms of building the trusted federated learning system, a comprehensive data poisoning attack anomaly detection capability is established by deploying strategies or optimization algorithms on several critical elements (e.g., aggregation servers, participants, model updates and model parameters). For the optimization of communication efficiency, many technologies are developed, i.e., model compression, reducing model update frequency and client selection, etc., or finding a balance between the three. The use of optimization techniques such as computing power, model parallelism, and other technologies is necessary to provide the computing power foundation required for each node to deploy data poisoning attack predictions. For improving robustness, personalized federated learning is used to perform personalized processing, in which strategies from the device, data and model levels are attempted to alleviate heterogeneity problems. In addition, efforts are made to optimize algorithm accuracy and lightweightness. This method will be simple, yet robust, and widely applicable in the real world.

Currently, the author's team is working on efficient, verifiable and privacy-preserving federated learning (Re_useVFL), which will support the verification of the integrity of parameters, the correctness of cloud server aggregation results, and the consistency of cloud server distribution results, which will boost the performance of the data poisoning attack prediction.

Finally, this paper explains the main research directions for the data poisoning attack prediction in federated learning, serving as a reference for the researching community.

Author Contributions: Methodology, J.W. and C.W.; Formal analysis, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2024 Key R&D Program of Zhejiang Province, China (No. 2024C01212).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Voulodimos, A.; Doulamis, N.; Doulamis, A. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [CrossRef]

2. Young, T.; Hazarika, D.; Poria, S. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
3. Anil, R.; Dai, A.M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2305.10403.
4. Radford, A.; Wu, J.; Child, R. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
5. Chen, M.X.; Zhang, J.B.; Li, T.R. A review of federated learning attack and defense research. *Comput. Sci.* **2020**, *49*, 310–323.
6. Zhang, C.; Xie, Y.; Bai, H. A survey on federated learning. Knowledge-Based Syst. *Comput. Sci.* **2021**, *216*, 106775.
7. Wang, B.; Dai, X.R.; Wang, W. Adversarial sample poisoning attack for federated learning. *Chin. Sci. Inf. Sci.* **2023**, *53*, 471–482.
8. Baruch, M.; Baruch, G.; Goldberg, Y. A little is enough: Circumventing defenses for distributed learning. In Proceedings of the 33rd Int'l Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; ACM: Red Hook, NY, USA, 2019; p. 775.
9. Fang, M.; Cao, X.; Jia, J.; Gong, N. Local model poisoning attacks to byzantine-robust federated learning. In Proceedings of the 2020 USENIX Security Symposium, Boston, MA, USA, 12–14 August 2020; USENIX Association: Berkeley, CA, USA, 2020; pp. 1605–1622.
10. Shejwalkar, V.; Houmansadr, A. Manipulating the byzantine:optimizing model poisoning attacks and defenses for federated learning. In Proceedings of the 2021 NDSS, Virtual, 21–25 February 2021; ISOC: Rosten, VA, USA, 2021; pp. 21–24.
11. Biggio, B.; Nelson, B.; Laskov, P. Poisoning attacks against support vector machines. In Proceedings of the 29th Int'l Conf. on Machine Learning, Edinburgh, UK, 26 June–1 July 2012; ACM: Edinburgh, UK, 2012; pp. 1467–1474.
12. Zhang, J.L.; Chen, B.; Cheng, X.; Binh, H.; Yu, S. PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.* **2021**, *8*, 3310–3322. [CrossRef]
13. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data poisoning attacks against federated learning systems. In Proceedings of the 25th European Symp. on Computer Security, Guildford, UK, 14–18 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 480–501.
14. Chen, Q.; Chai, Z.; Wang, Z.l. Poisoning Attack Detection Scheme in Federated Learning Based on Generative Adversarial Networks. Available online: http://kns.cnki.net/kcms/detail/51.1307.TP.20230522.1041.004.html (accessed on 27 October 2023).
15. Yin, D.; Chen, Y.; Kannan, R. Byzantine-robust distributed learning:Towards optimal statistical rates. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; JMLR: San Diego, CA, USA, 2018; pp. 5650–5659.
16. Steinhardt, J.; Koh, P.W.; Liang, P. Certified defenses for data poisoning attack. In Proceedings of the 31st International Conference on Neural Information Proceedings Systems, Long Beach, CA, USA, 4–9 December 2017; NIPS: La Jolla, CA, USA, 2017; pp. 3520–3532.
17. Bhagoji, A.N.; Chakraborty, S.; Mittal, P. Analyzing federated learning through an adversarial lens. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; JMLR: San Diego, CA, USA, 2019; pp. 634–643.
18. Feng, J.; Xu, H.; Mannor, S. Robust logistic regression and classification. In Proceedings of the 27th International Conference on Neural Information Proceeding Systems, Bangkok, Thailand, 18–22 November 2020; NIPS: La Jolla, CA, USA, 2014; pp. 253–261.
19. Zhao, Y.; Chen, J.; Zhang, J. Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks. *Concurr. Comput. Pract. Exp.* **2020**, *34*, e5906. [CrossRef]
20. Goodfellow, I.; Pouget-abadie, J.; Mirza, M. Generative adversarial nets. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
21. Jagielski, M.; Oprea, A.; Biggio, B. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In Proceedings of the 39th IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 21–23 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 19–35.
22. Li, T.; Sahu, A.K.; Talwalkar, A. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
23. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.Y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; JMLR: Fort Lauderdale, FL, USA, 2017; pp. 1273–1282.
24. Gu, Y.H.; Bai, Y.M. Research progress on security and privacy of federated learning models. *J. Softw.* **2023**, *34*, 2833–2864.
25. Yang, Q.; Liu, Y.; Chen, T.J.; Tong, Y.X. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 12. [CrossRef]
26. Chen, X.H.; Ren, Z.Q.; Zhang, H.Y. Overview of Security Threats and Defense Measures in Federated Learning. Available online: http://kns.cnki.net/kcms/detail/51.1307.TP.20230731.1744.024.html (accessed on 23 October 2023).
27. Li, M.H.; Wan, W.; Lu, J.R. Shielding federated learning: Mitigating by zantine attacks with less constraints. In Proceedings of the 18th IEEE International Conference on Mobility, Sensing and Networking, Guangzhou, China, 14–16 December 2022; IEEE: Piscataway, NJ, USA, 2022; p. 178185.
28. Sun, G.; Cong, Y.; Dong, J.; Wang, Q.; Lyu, L.; Liu, J. Data poisoning attacks on federated machine learning. *IEEE Internet Things J.* **2021**, *9*, 11365–11375. [CrossRef]

29. Zhang, J.; Chen, J.; Wu, D.; Chen, B.; Yu, S. Poisoning attack in federated learning using generative adversarial nets. In Proceedings of the 2019 IEEE International Conference on Big Data Science and Engineering, Los Angeles, CA, USA, 9–12 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 374–380.
30. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the 2020 International Conference on Artificial Intelligence and Statistics, Virtual, 26–28 August 2020; PMLR: New York, NY, USA, 2020; pp. 2938–2948.
31. Wang, H.J.; Liang, Y.N.; Li, L.; Li, R. Overview of privacy protection mechanisms in federated learning. *Mod. Comput.* **2022**, *28*, 1–12.
32. Fung, C.; Yoo, C.J.M.; Beschastnikh, I. Mitigating Sybils in Federated Learning Poisoning. *arXiv* **2018**, arXiv:1808.04866.
33. Han, X.; Huang, X.; Claudia, E. Adversarial label flips attack on support vector machines. In *ECAI 2012*; IOS Press: Amsterdam, The Netherlands, 2012; pp. 870–875.
34. Shi, L.; Chen, Z.; Shi, Y.C.; Zhao, G.; Wei, L.; Tao, Y.; Gao, Y. Data Poisoning Attacks on Federated Learning by Using Adversarial Samples. In Proceedings of the 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), Shijiazhuang, China, 22–24 July 2022; pp. 158–162.
35. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. *arXiv* **2023**, arXiv:1312.6199.
36. Wang, H.Y.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.Y.; Lee, K.; Papailiopoulos, D. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. *arXiv* **2023**, arXiv:2007.05084.
37. Sha, F.H.; Huang, W.R.; Na, J.B.M.; Suciu, O.; Studer, C.; Dumitras, T.; Goldstein, T. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. *arXiv* **2018**, arXiv:1804.00792.
38. Dwork, C.; Mcsherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
39. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On data banks and privacy homomorphisms. *Found. Secur. Comput.* **1978**, *4*, 169–180.
40. Yao, A.C. Protocols for secure computations. In Proceedings of the 23rd Annual Symposium on Foundations Of Computer Science (SFCS 1982), Chicago, IL, USA, 3–5 November 1982; IEEE: Piscataway, NJ, USA, 1982; pp. 160–164.
41. Endsley, M.R. Toward a theory of situation awareness in dynamic system. *Found. Secur. Comput.* **1995**, *37*, 32–64. [CrossRef]
42. Franke, U.; Brynielsson, J. Cyber situational awareness a systematic review of the literature. *Comput. Secur.* **2014**, *46*, 18–31. [CrossRef]
43. Lenders, V.; Tanner, A.; Blarer, A. Gaining an edge in cyberspace with advanced situational awareness. *Secur. Priv. IEEE* **2015**, *13*, 65–74. [CrossRef]
44. Bass, T. Intrusion Detection Systems and Data Fusion. *Commun. ACM* **2000**, *43*, 99–105. [CrossRef]
45. Gong, J.; Zang, X.D.; Su, Q. A review of network security situational awareness. *J. Softw.* **2017**, *28*, 1010–1026.
46. Wang, S.; Li, Q.; Cui, Z.; Hou, J.; Huang, C. Bandit-based data poisoning attack against federated learning for autonomous driving models. *Expert Syst. Appl.* **2023**, *227*, 120295. [CrossRef]
47. Talpur, A.; Gurusamy, M. GFCL: A GRU-based Federated Continual Learning Framework against Data Poisoning Attacks in IoV. *arXiv* **2022**, arXiv:2204.11010.
48. Shahid, A.R.; Ahmed, I.; Shahriar, B.; Md, Z.H. Assessing Wearable Human Activity Recognition Systems Against Data Poisoning Attacks in Differentially-Private Federated Learning. In Proceedings of the 2023 IEEE International Conference on Smart Computing (SMARTCOMP), Nashville, TN, USA, 26–30 June 2023; pp. 355–360.
49. hong, L.T.; Aono, Y.; Hayashi, T.; Wang, L.H.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1333–1345.
50. Fredrikson, M.; Lantz, E.; Jha, S. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In Proceedings of the USENIX Security Symposium, Philadelphia, PA, USA, 19–20 June 2014; USENIX Association: Berkeley, CA, USA, 2014; pp. 17–32.
51. Zhu, T.Q.; Ye, D.Y.; Zhou, S. Label-only model inversion attacks: Attack with the least information. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 991–1005. [CrossRef]
52. Guo, J.J.; Liu, J.Z.; Ma, Y.; Liu, Z.Q.; Xiong, Y.P.; Miao, K.; Li, J.X.; Ma, J.F. Federated learning backdoor attack defense method based on model watermark. *J. Comput. Sci.* **2024**, *47*, 622–676.
53. Jere, M.S.; Farnan, T.; Koushanfar, F. A taxonomy of attacks on federated learning. *IEEE Secur. Priv.* **2021**, *19*, 20–28. [CrossRef]
54. Zhou, C.X.; Sun, Y.; Wang, D.G. A review of federated learning research. *J. Netw. Inf. Secur.* **2021**, *7*, 77–92.
55. Konecny, J.; Mcmahan, H.B.; Yu, F.X. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
56. Wang, L.; Wang, W.; Bo, L.I. CMFL: Mitigating Communication Overhead for Federated Learning. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; IEEE: Piscataway, NJ, USA, 2019.
57. Paudice, A.; Muñoz-González, L.; Gyorgy, A. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv* **2018**, arXiv:1802.03041.
58. Wu, Q.; He, K.W.; Chen, X. Personalized federated learning for intelligent iot applications: A cloud-edge based framework. *arXiv* **2020**, arXiv:2002.10671.

59. Long, Y.C. *Research on Adversarial Attacks and Robustness of Vertical Federated Learning*; Guangzhou University: Guangzhou, China, 2023.
60. Liu, J.Q.; Zhang, Z.; Chen, Z.D. A poisoning attack defense method based on federated learning participants. *Comput. Appl. Res.* **2023**, *7*, 0340.
61. Liu, B.; Zhang, F.J.; Wang, W.X. Byzantine Robust Federated Learning Algorithm Based on Matrix Mapping. *Comput. Res. Dev.* **2021**, *58*, 2416–2429.
62. Blanchard, P.; El, M.E.M.; Guerraoui, R. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Proceedings of the Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; NIPS Foundation: San Diego, CA, USA, 2017; pp. 119–129.
63. Lu, Y.; Fan, L. An Efficient and Robust Aggregation Algorithm for Learning Federated CNN. In Proceedings of the 2020 3rd International Conference on Signal Processing and Machine Learning, Beijing, China, 22–24 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–7.
64. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust Aggregation for Federated Learning. *IEEE Trans. Signal Process.* **2019**, *70*, 1142–1154. [CrossRef]
65. Fung, C.; Yoon, C.J.M.; Beschastnikh, I. The Limitations of Federated Learning in Sybil Settings. In Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID 2020), Virtual, 14–16 October 2020; USENIX Association: Berkeley, CA, USA, 2020; pp. 301–316.
66. Chen, D.; Jiang, X.; Zhong, H.; Cui, J. Building Trusted Federated Learning: Key Technologies and Challenges. *J. Sens. Actuator Netw.* **2023**, *12*, 13. [CrossRef]
67. Li, L.X.; Yuan, S.; Jin, Y. Overview of federated learning technology based on blockchain. *Comput. Appl. Res.* **2021**, *38*, 3222–3230.
68. Lu, G.X.; Xiong, Z.B.; Li, R.N. Decentralized Federated Learning: A Defense Against Gradient Inversion Attack. In Proceedings of the International Wireless Internet Conference 2022, Virtual, 17 November 2022; pp. 301–316.
69. Li, D.; Wang, W.E.; Wang, W.; Yao, Y.; Chau, M. Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and K-means. In Proceedings of the 2021 International Conference on Dependable Systems and Their Applications (DSA), Yinchuan, China, 11–12 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 551–559.
70. Chan, H.Y.; Shejwalkar, V.; Shokri, R.; Houmansadr, A. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv* **2019**, arXiv:1912.11279.
71. Liu, Y.; Xie, Y.; Srivastava, A. Neural Trojans. In Proceedings of the 2017 IEEE International Conference on Computer Design (ICCD), Boston, MA, USA, 5–8 November 2017; pp. 45–48.
72. Muñoz-González, L.; Co, K.T.; Lupu, E.C. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv* **2019**, arXiv:1909.05125.
73. Awan, S.; Luo, B.; Li, F. Contra: Defending against poisoning attacks in federated learning. In Proceedings of the 2021 European Symposium on Research in Computer Security, Virtual, 4–8 October 2021; Springer: Cham, Switzerland, 2021; pp. 455–475.
74. Udeshi, S.; Peng, S.; Woo, G.; Loh, L.; Rawshan, L.; Chattopadhyay, S. Model agnostic defence against backdoor attack in machine learning. *arXiv* **2019**, arXiv:1908.02203.
75. Kieu, T.; Yang, B.; Guo, C. Outlier detection for time series with recurrent autoencoder ensembles. In Proceedings of the International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 2725–2732.
76. Kieu, T.; Yang, B.; Guo, C. Anomaly detection in time series with robust variational quasi-recurrent autoencoders. In Proceedings of the IEEE International Conference on Data Engineering, Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 1342–1354.
77. Yang, Q.; Tong, Y.X.; Wang, Y.S. A review of federated learning algorithms in swarm intelligence. *Chin. J. Intell. Sci. Technol.* **2022**, *4*, 29–44.
78. Li, J.; Zhao, Y.; Xue, Z.; Cai, Z.; Li, Q. A review of deep neural network model compression. *J. Eng. Sci.* **2019**, *41*, 1229–1239.
79. Xu, J.J.; Du, W.L.; Jin, Y.C.; He, W.; Cheng, R. Ternary compression for communication-efficient federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1162–1176. [CrossRef] [PubMed]
80. Shah, S.M.; Lau, V.K. Model Compression for Communication Efficient Federated Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 5937–5951. [CrossRef] [PubMed]
81. Caldas, S.; Konecny, J.; Mcmahan, H.B. Expanding the reach of federated learning by reducing client resource requirements. *arXiv* **2018**, arXiv:1812.07210.
82. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization for heterogeneous networks. *arXiv* **2021**, arXiv:1812.06127.
83. Zhou, Y.H.; Ye, Q.; Lv, J.C. Communication-efficient federated learning with compensated Overlap-FedAvg. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 192–205. [CrossRef]
84. Bellet, A.; Guerraoui, R.; Taziki, M.; Tommasi, M. Personalized and private peer-to-peer machine learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Playa Blanca, Spain, 9–11 April 2018; pp. 473–481.
85. Huang, T.S.; Lin, W.W.; Wu, W.T.; He, L.; Li, K.; Zomaya, A.Y. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1552–1564. [CrossRef]
86. Lai, F.; Zhu, X.F.; Madhyastha, H.; Chowdhury, M. Oort: Informed participant selection for scalable federated learning. *arXiv* **2020**, arXiv:2010.06081.

87. Rajbhandari, S.; Rasley, J.; Ruwase, O.; He, Y. ZeRO: Memory optimizations Toward Training Trillion Parameter Models. In Proceedings of the SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA, 9–19 November 2020; pp. 1–16.
88. Chen, C.C.; Feng, X.H.; Zhou, J.; Yin, J.W.; Zheng, X.L. Federated Large Language Model: A Position Paper. *arXiv* **2023**, arXiv:2307.08925.
89. Liao, Y.; Xu, Y.; Xu, H.; Yao, Z.; Wang, L.; Qiao, C. Accelerating Federated Learning with Data and Model Parallelism in Edge Computing. *IEEE/ACM Trans. Netw.* **2024**, *32*, 904–918. [CrossRef]
90. Vanhaesebrouck, P.; Bellet, A.; Tommasi, M. Decentralized collaborative learning of personalized models over networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; pp. 509–517.
91. Jin, T.; Pan, S.; Li, X.; Chen, S. Metadata and Image Features Co-Aware Personalized Federated Learning for Smart Healthcare. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 4110–4119. [CrossRef]
92. Rasti-Meymandi, A.; Sheikholeslami, S.M.; Abouei, J.; Plataniotis, K.N. Graph Federated Learning for CIoT Devices in Smart Home Applications. *IEEE Internet Things J.* **2023**, *10*, 7062–7079. [CrossRef]
93. Ye, Z.; Zhang, X.; Chen, X.; Xiong, H.; Yu, D. Adaptive Clustering based Personalized Federated Learning Framework for Next POI Recommendation with Location Noise. *IEEE Trans. Knowl. Data Eng.* **2023**, *10*, 1–14. [CrossRef]
94. Xie, C.; Koyejo, S.; Gupta, I. Asynchronous Federated Optimization. *arXiv* **2019**, arXiv:1903.03934.
95. Kulkarni, V.; Kulkarni, M.; Pant, A. Survey of personalization techniques for federated learning. *arXiv* **2020**, arXiv:2003.08673.
96. Mansour, Y.; Mohri, M.; Ro, J.; Suresh, A.T. Three approaches for personalization with applications to federated learning. *arXiv* **2020**, arXiv:2002.10619.
97. Schneider, J.; Vlachos, M. Mass personalization of deep learning. *arXiv* **2019**, arXiv:1909.02803.
98. Smith, V.; Chiang, C.K.; Sanjabi, M.; Talwalkar, A.S. Federated multi-task learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4424–4434.
99. Li, D.; Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv* **2019**, arXiv:1910.03581.
100. Jiang, Y.; Konecny, J.; Rush, K.; Kannan, S. Improving federated learning personalization via model agnostic meta learning. *arXiv* **2019**, arXiv:1909.12488.
101. Arivazhagan, M.G.; Aggarwal, V.; Singh, A.K.; Choudhary, S. Federated learning with personalization layers. *arXiv* **2019**, arXiv:1912.00818.
102. Hanzely, F.; Richtarik, P. Federated learning of a mixture of global and local models. *arXiv* **2020**, arXiv:2002.05516.
103. Chen, X.B.; Ren, Z.Q. PFKD: A personalized federated learning framework that comprehensively considers data heterogeneity and model heterogeneity. *J. Nanjing Univ. Inf. Sci. Technol. (Natural Sci. Ed.)* **2023**, *32*, 1–10.
104. Shen, T.; Zhang, J.; Jia, X.K.; Zhang, F.; Lv, Z.; Kuang, K.; Wu, C.; Wu, F. Federated mutual learning: A collaborative machine learning method for heterogeneous data, models and goals (English). *Front. Inf. Technol. Electron. Eng.* **2023**, *24*, 1390–1403. [CrossRef]
105. Dinh, C.T.; Tran, N.H.; Nguyen, T.D. Personalized federated learning with moreau envelopes. *arXiv* **2020**, arXiv:2006.08848.
106. Yuan, B.; Qiu, W. Personalized Federated Learning System Based on Permissioned Blockchain. In Proceedings of the 2021 International Conference on Intelligent Computing, Automation and Systems (ICICAS), Chongqing, China, 29–31 December 2021; pp. 95–100.
107. Ibitoye, O.; Shafiq, M.O.; Matrawy, A. DiPSeN: Differentially Private Self–Normalizing Neural Networks For Adversarial Robustness in Federated Learning. *arXiv* **2021**, arXiv:2101.03218.
108. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.