electronics

MDPI

Review

A Detailed Survey on Federated Learning Attacks and Defenses

Hira Shahzadi Sikandar [1], Huda Waheed [1], Sibgha Tahir [1], Saif U. R. Malik [2,*] and Waqas Rafique [3]

[1] Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan
[2] Cybernetica AS Estonia, 13412 Tallinn, Estonia
[3] Department of Computer Science, University College London, London WC1E 6BT, UK
* Correspondence: saif@cyber.ee

Abstract: A traditional centralized method of training AI models has been put to the test by the emergence of data stores and public privacy concerns. To overcome these issues, the federated learning (FL) approach was introduced. FL employs a privacy-by-design architecture to train deep neural networks utilizing decentralized data, in which numerous devices collectively build any machine learning system that does not reveal users' personal information under the supervision of a centralized server. While federated learning (FL), as a machine learning (ML) strategy, may be effective for safeguarding the confidentiality of local data, it is also vulnerable to attacks. Increased interest in the FL domain inspired us to write this paper, which informs readers of the numerous threats to and flaws in the federated learning strategy, and introduces a multiple-defense mechanism that can be employed to fend off threats.

Keywords: federated learning (FL); machine learning (ML); FL attacks; defensive mechanisms

check for updates

Citation: Sikandar, H.S.; Waheed, H.; Tahir, S.; Malik, S.U.R.; Rafique, W. A Detailed Survey on Federated Learning Attacks and Defenses. *Electronics* **2023**, *12*, 260. https://doi.org/10.3390/electronics12020260

Academic Editor: Aryya Gangopadhyay

Received: 24 October 2022
Revised: 20 December 2022
Accepted: 21 December 2022
Published: 4 January 2023

1. Introduction

Currently, as computing devices grow more commonplace due to widespread adoption of technology, people continually create enormous volumes of information. Such data collection and storage in a centralized storage area is expensive and time-consuming. Conventional processes incorporating AI machine learning (ML) cannot handle such expansive arrangements and applications due to fundamental problems including constrained bandwidth, conflicting organization linkage, and severe inertness constraints [1]. AI is a widely used approach with many applications, among which two examples are spam sifting and estimating the value of gaseous gasoline. For these applications, the consistency or security of the AI framework, especially that of attackers, has been a serious concern. To be more explicit [2], traditional AI data-processing models frequently utilize basic models of data transactions in which one side collects and passes data towards the other party, which cleans and merges the data. The third party would utilize the pooled data to create models. Models frequently offer finished products or services. The above-mentioned new data norms and legislation pose obstacles to this old practice. Furthermore, since consumers may be unsure about the models' future usage, the transactions breach rules such as the General Data Protection Regulation (GDPR).

The traditional centralized training of AI models is encountering efficiency and privacy concerns as data are kept in diverse silos and societies become more conscious of data privacy risks. In response to this new reality, federated learning (FL) has recently emerged as a viable alternative approach. This approach educates a system without transferring datasets across several decentralized endpoints or hosts, and retains local datasets. This strategy differs from standard centralized machine learning methods, where all local samples are posted to a single server, as well as more traditional decentralized alternatives, which frequently presume that localized datasets are uniformly distributed. This enables devices to cooperatively develop common forecasting models while maintaining all the trained information in the system, sparing machine learning the requirement of having to save information in the cloud. This extends the above applications of model parameters for system prediction. Data privacy and system robustness may be at risk from existing FL protocol designs that can be exploited by attackers from within or outside the system. It is crucial to develop FL systems that provide privacy guarantees and are resilient against many sorts of attackers, in addition to training global models. As the number of sophisticated computer devices and applications increases, the amount and variety of information created at the organization's edge will continue to grow at unprecedented rates [3,4]. This stream of information is innately decentralized and heterogeneous, and when amassed, it might incorporate experiences and disclosures with the possibility to drive logical and mechanical advancement. There are two variants of assaults on machine learning (ML) methods, namely causal attacks and exploratory attacks. Exploratory attacks impact learning by influencing training data, and the attack might employ classifications with any modifying instruction.

However, security threats related to information ownership are swiftly becoming a concern for the general population. The contention between great administrations and client protection is producing a need for new exploration and innovation to permit ways to obtain experiences from information without uncovering private data. However, federated learning is increased by a configuration that enables a large number of users to maintain their unique information on their own devices, such as cell phones, while collectively learning a model by only exchanging local boundaries with the server. A few ongoing exploration endeavors have shown that FL's default security is insufficient to shield underlying preparing information against protection spillage attacks through slope-based remaking.

Federated learning [5,6] is prone to software/hardware faults and adversarial attacks because of its large-scale and decentralized implementation. Some clients in particular might become defective owing to software defects or even be hacked during training, delivering arbitrary or harmful information to the server and substantially reducing the overall convergence performance. In addition, as seen in [7], federated learning represents a possibly hazardous plan tradeoff: customers, who were beforehand exclusively latent information providers, may now observe the middle-of-the-road model state and make changes as a feature of the decentralized preparation process. This permits malicious customers to adjust the preparation interaction with negligible limitations. In FL [8], devices are subject to data and model poisoning assaults in terms of robustness. Malicious participants may actively undermine the global model's convergence, alter their local data (data poisoning) or upload gradients, or install backdoored inputs on the global model—which is known as model poisoning. Model poisoning threats also be characterized as: (a) Byzantine assaults, wherein the attacker wants to introduce secret triggers to the global model to exploit it; and (b) backdoor attacks, in which the attacker desires to introduce hidden triggers into the global model to exploit it. Recent articles [9] have established a "gradient inversion attack," which allows an adversary to listen in on a client's communications with the server to start reconstructing the client's sensitive information.

In this study, we have discussed recent improvements for mitigating dangers to FL's privacy and their countermeasures. Current attacks are harmful material throughout the data training set before the commencement of the training procedure, whereas the training process is considered to be integrity-preserving. As a result, these assaults are sometimes called data poisoning attacks.[6]. We have provide information about all possible attacks on federated learning and their defenses by focusing on insider and outsider attacks, as compared to [2], which only focused on insider attacks. In addition, we discuss data reconstruction, model inversion, and backdoor attacks, which were not discussed in [2]. The purpose of reviewing these attacks is to provide knowledge about assaults and their countermeasures so that one can avoid using these materials. We also present the taxonomies of attacks and their possible defenses. The report is structured in the following manner: Section 2 is about the overview of federated learning. In Section 3, different types of federated learning are discussed. Subsequently, Section 4 provides attacks performed on federated learning along with their defense mechanisms. At the end of Section 5, this approach and its challenges are the addressed in an open discussion.

2. Overview of Federated Learning

In its most basic version, the FL architecture comprises a director or server that organizes instructive events [10]. FL represents the advent of a new revolution in machine learning that is employed when training data are dispersed. This enables numerous customers to construct a shared machine learning technique, despite protecting the confidentiality of their data. This strategy differs from conventional machine learning, which needs training data to be centralized in a single data center or repository. The fact that FL can train a model without centralizing client datasets has garnered significant interest in machine learning. The majority of clients are edge devices which may count in the millions. These devices connect to the server no less than two times in each training cycle. Clients receive individual parameters for the current global model from the server, which trains the global model using their local data. Clients then submit the updated parameters to the aggregator (server) [11]. This iteration cycle will repeat until either a predefined period or an accuracy condition is fulfilled. In the federated averaging algorithm, aggregation is an averaging procedure [12]. That is all there is to FL model training. We hope that the key element in this procedure is obvious: rather than transferring the original data all around, model weights are conveyed, as shown in Figure 1.

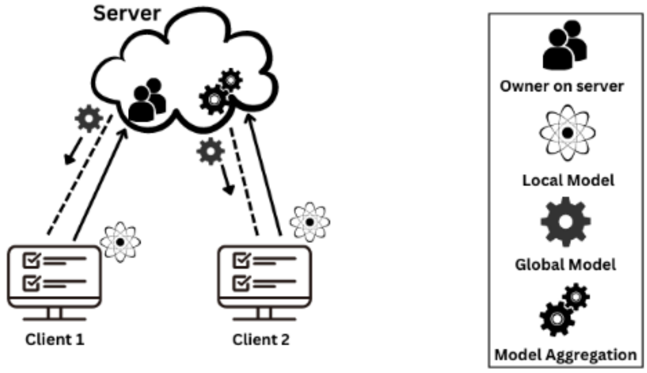

Figure 1. Federated Learning Architecture [13].

By utilizing participant-provided local training data, FL aids in creating a shared global model with improved performance [14]. Some of the most significant applications and rapidly expanding trends in the use of FL in practice include loan status prediction, health condition assessment, and next-word prediction while typing [15]. FL [16] is a rapidly expanding field of study because of its decentralized data method and characteristics relating to privacy and safety that strive to meet the needs of contemporary security in terms of customer data regulations. On the other hand, it is vulnerable to a variety of assaults and the training and deployment of the system have been plagued by errors at every stage [17].

3. Types of Federated Learning

When training an algorithm for machine learning, federated learning—also known as collaborative learning—allows for devices or servers in the network's periphery to work together without exchanging data samples. Federated learning involves numerous individuals remotely sharing data in order to train a single deep learning model and iteratively improve it. Both sides obtain their respective models from a cloud-based datacenter, with the latter often providing a pre-trained foundation model. After training it on secret information, the model's updated settings are summed up and encrypted. The cloud receives the encrypted model updates, decrypts them, takes an average, and incorporates them into the master model. The collaborative training process is repeated again until the model is completely trained. Moreover, the FL framework has several types that are described in this section.

- **Horizontal FL (HFL) :** Horizontal FL is appropriate for datasets with the same feature but located on various devices. HFL is divided into horizontal FL to Horizontal to business (H2B) and horizontal FL to horizontal to consumers (H2C) . Traditionally, H2B has a small group of members. They are often chosen throughout FL training. Participants often have high processing capacity and strong technical skills. hundreds or perhaps millions of potential participants under H2C. Only a subset of them gets trained in each cycle.
- **Vertical FL (VFL):** Different vertical federations employ different feature sets, the term vertical federated learning can also be abbreviated to heterogeneous federated learning. When two datasets have the same sample ID space but different feature spaces, a technique known as vertical federated learning or feature-based federated learning may be used. In deliberately vertical FL cases, datasets are identical but have distinct characteristics, as shown in [18]. VFL is primarily directed towards corporate players. Thus, VFL individuals have similarities to H2B participants [19].
- **Federated Transfer Learning (FTL):** Federated FTL is similar to classical machine learning by being used to augment a model that has been pre-trained with a new feature. However, the descriptions given for federated transfer learning are more involved, comprising intermediate learning to map to a common feature subspace, as opposed to the convolutional neural network transfer techniques, which are essentially dropping the last few layers from a network trained on big data, and then re-tuning the model to recognize labels on a small dataset [20]. The greatest example would be to expand vertical federated learning to additional sample cases not available in all partnering organizations.
- **Cross-Silo FL:** When the count of participating machines is constrained, they are available for all rounds, and cross-silo federated learning is employed. The training architecture for cross-silo FL differs significantly from the one used in an example-based context. Clients may trade certain intermediate outcomes rather than model parameters, depending on the details of the training process, to aid other parties' gradient calculations, which may or may not include a central server as a neutral party [21]. The training data might be in FL format, either horizontal or vertical. Cross-silo is often utilized in circumstances involving corporations.
- **Cross-Device FL:** Cross-device FL is used in scenarios involving a large number of participating devices. Learning across several user devices using data generated by a single user is the focus of cross-device configurations. Cross-device federated learning was first used when Google used GBoard user data to build next-word prediction models [22]. Client selection and incentive design are two significant strategies facilitating this sort of FL.

4. FL Attacks and Their Defenses

Machine learning has the potential to be one of the most disruptive technologies in decades. Almost every business may benefit from artificial intelligence (AI) applications, and adoption rates reflect a popular belief in the technology's promise [23]. However, as machine learning becomes more prevalent, the danger of all forms of assault becomes increasingly serious [24]. While technologies hold enormous potential for good, their potential for damage has grown as a growing number of enterprises depend on them. Attacker want to capitalize on such potential. Attacks against the integrity of machine learning are increasingly complicated and possibly more dangerous. Figure 2 demonstrates the different types of attacks on the FL framework.

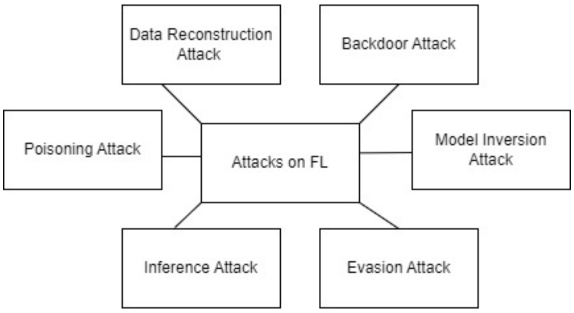

Figure 2. Attacks on FL.

The majority of a database is unaffected, apart from an unnoticed backdoor that allows attackers to manipulate it. The model seems to perform as planned, but with one catastrophic fault, such as always classifying one file type as innocuous.

4.1. Poisoning Attacks and Defenses

4.1.1. Data Poisoning Attacks and Defenses

Tampering with ML training data creates unwanted results [25]. An attacker will break into a machine learning database and introduce false or misleading data. As the algorithm learns from this tainted data, it will reach unanticipated and perhaps destructive conclusions [26]. This strike can do a lot of damage with very little effort. The major disadvantage of AI is that its effectiveness is nearly directly related to the quality of its data. Poor-quality data will give substandard results regardless of how complex the model is, and history demonstrates that it does not take much to do so [27]. Figure 3 shows the process of a data poisoning attack.

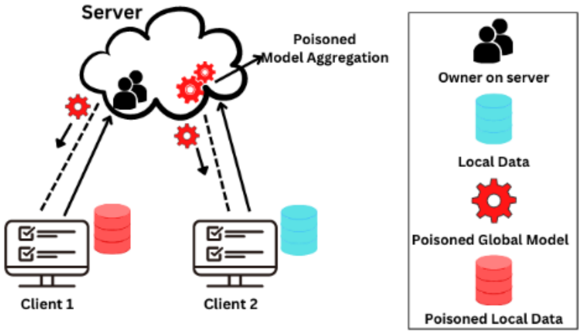

Figure 3. Data Poisoning Attack.

Adversaries may implant a collection of tainted and infected data during training to produce a result in categorization at the moment of inference in current poisoning attacks. There are two categories of poisoning assaults now in use: provision and targeted attacks [28]. In availability attacks, specific test cases are meant to be mistakenly classified, and targeted assaults, as outlined in [29], jeopardize the model's overall accuracy [29] and are defined in [30] as those that target accessibility and those that target honesty. Target accessibility is a group in which the enemy is intended to diminish the overall model performance.

The original target distribution is data poisoning attacks which are broadly classified into two types: First is clean-label attacks (CLA) [31–33], where the attacker cannot change the mark of any preparation information, because there is a technique that ensures that information has a place with the correct class and data sample poisoning must be unnoticeable. The second type of attack is the use of derogatory labels [13,34] whereby the attacker might introduce different sample data into the training data to obtain the model to mistakenly classify the data with the desired target label. Data tampering attacks are open to all FL participants. Attacks using poisoned data [35] because they deploy a distributed type of FL are frequently less successful than model poisoning attempts. The impact of the FL model depends on how much the backdoor players participate in the attacks and the extent to which training data have been contaminated [36]. Figure 4 and Table 1 show different defense mechanisms.

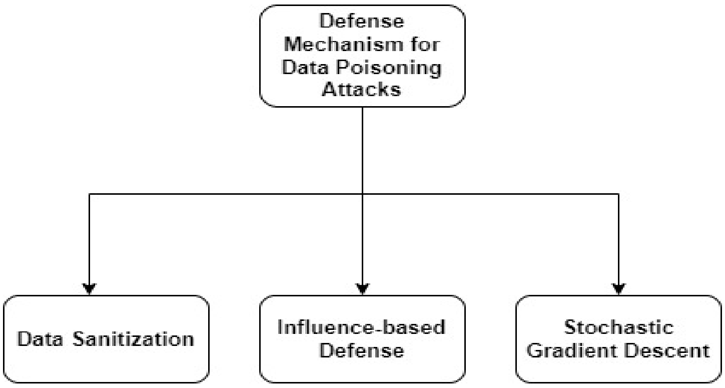
Defense Mechanism for Data Poisoning Attacks
Data Sanitization
Influence-based Defense
Stochastic Gradient Descent

Figure 4. Defense Mechanism for Data Poisoning Attacks.

Defense Mechanism 1: Data sanitization is a well-known security strategy against data poisoning attacks, in which suspect data are screened away before they enter the training process. The term data sanitization refers to the act of wiping or erasing information from a storage medium in such a way that it cannot recover. Even with sophisticated forensic equipment, once a sensor has been cleaned, there are no repairable data left on it. Data sanitization is accomplished through complete eradication, cryptography ablation, and data removal [37]. One of the defenses used in Slab is that of powerful data sanitization protection against a wide range of threats [38].

Defense Mechanism 2: In paper [39], the authors discussed the addition of an influence-based defensive method to the slab defense, to lessen the impact of poisoned data on the learner's model. The damage that poisoned training data may do to a learner's model in a live setting can be mitigated through the use of influence-based protection measures. A traditional method in robust statistics, namely that of an influence function, is used in this approach. Additionally, current data sanitization procedures may be used in conjunction with it to further remove some of the contaminated data. By creating an estimated upper limit on the loss, defense and data-dependent defenses for certified defense enable the analysis of the oracle [40].

Defense Mechanism 3: In [41], the authors analyzed how the algorithm stochastic gradient descent which provides robustness against data poisoning attacks. The component defending against such assaults seeks to identify malevolent members based on the ratio of the number of the times that the model refreshes to the total number in each round of learning [42]. SGD is significantly faster than Batch GD because it chooses a "random" instance of training data at each step and then computes the gradient. Furthermore, other mechanisms are discussed in Table 1 below.

Table 1. Comparing Defenses used in multiple research papers.

Related Papers	Defense Mechanism	Model Accuracy
[43]	AUROR	70%
	EE-Detection (elliptic envelope)	85%
[44]	Deep k-NN	99%
	One-class SVM	37%
	Random point eviction	12%
[32]	CONTRA	84%
	FoolsGold	79%
	Krum	68 %

4.1.2. **Model Poisoning Attacks and Defenses**

The term "model poisoning" describes a variety of tactics used to influence the federated learning algorithm [45] due to the distributed type of FL, which are frequently less effective than model poisoning attacks. The FL model's impact is determined by both the volume of contaminated training data and the contribution of backdoor players to the assaults [43]. To degrade the performance of the global model, adversaries may change the local model gradients, for instance, by lowering the overall accuracy. Direct gradient manipulation enabled us to set up covert global model backdoors [44]. Rule manipulation during model training is another. According to several other papers including [45], model poisoning is explained via model training rules, which is significantly more effective than model poisoning via data poisoning [46]. Under some circumstances, changing a single local model may jeopardize the global model. The authors used a complex training rule adjustment in this example [47]. By adding a penalty component to the objective function, we effectively close the gap between malicious and benign weight update distributions and performed a covert targeted model poisoning assault.

Targeted model poisoning, as shown in Figure 5, aims to convince the FL model to confidently misclassify a particular group of inputs. It should be noted that assaults in this situation are not confrontational [48], as these inputs are not altered during testing to cause misclassification. Poisoning attacks on model updates, where only a fraction of the updates transmitted to the server at any given iteration is safe, have received a lot of attention recently. A backdoor could be inserted into a model with a single-shot attack. These contaminated updates [49] may be created by inserting hidden backdoors.

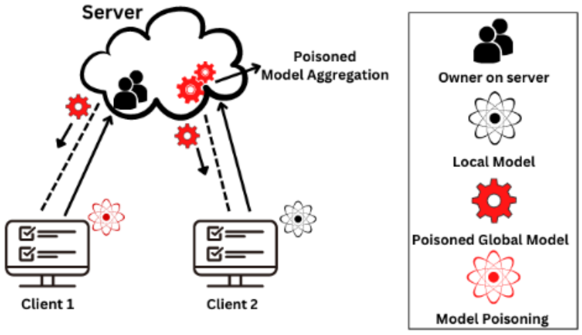

Figure 5. Model Poisoning Attack.

We discovered that model poisoning attacks are significantly more successful than data poisoning assaults in FL situations [50]. We investigated a targeted model poisoning attack wherein a single non-colluding hostile actor tries to compel the model to erroneously categorize a set of specified inputs, and we found that model poisoning attacks are substantially more successful than data poisoning attacks under FL circumstances. These employ parameter estimation for the benign players' updates and the alternating minimization technique to alternately optimize for the training loss and the hostile aim to increase the assault stealth and escape detection. This adversarial model poisoning technique could be used to poison the FL model covertly. Figure 6 and Table 2 show the different defense mechanisms to address the data poisoning attack.

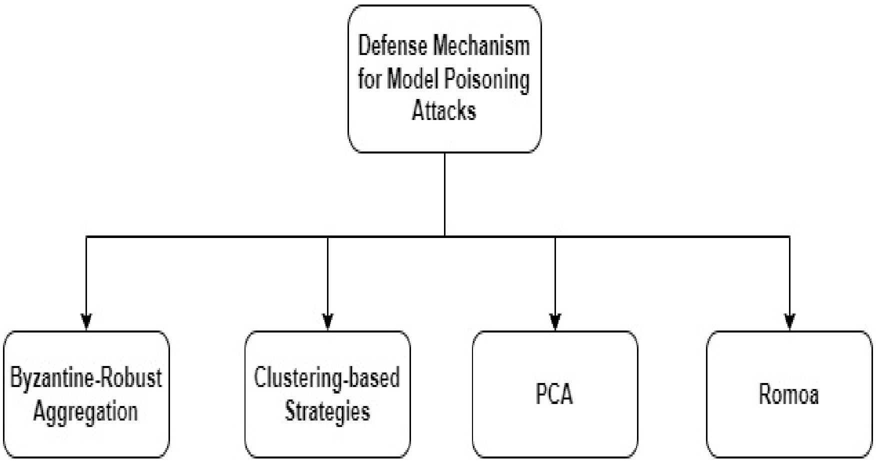
Defense Mechanism for Model Poisoning Attacks
Byzantine-Robust Aggregation
Clustering-based Strategies
PCA
Romoa

Figure 6. Defense Mechanism for Model Poisoning Attacks.

Defense Mechanism 1: Byzantine robust aggregation is a system that does well against untargeted poisoning assaults and focused model poisoning assaults using enhanced learning rates [51], but it does less well against flexible poisoning assaults, which reduces the accuracy of the global model test. Analyzing the resilience against Byzantine attacks, in which the attackers arbitrarily corrupt the parameters, Byzantine robust aggregation is a flexible and robust aggregation approach known as an auto-weighted geometric median (AutoGM) [52].

Defense Mechanism 2: According to alternative processes such as clustering-based procedures, model updates should be reviewed at the aggregate and then separated into two groups, for instance, using dimensionality reduction approaches such as head part analysis [53]. The most popular clustering method is called K-means clustering. As the simplest unsupervised learning method, it uses a centroid to make predictions. The goal of this approach is to reduce within-cluster variability. However, these techniques also count on the impartiality and uniform distribution of the training data.

Defense Mechanism 3: Other mechanisms including clustering-based procedures propose that model updates are examined at the aggregator and then segregated into two groups when utilizing dimensionality reduction approaches such as head part analysis [54]. It may be used independently or as a jumping-off point for additional dimension-reduction techniques. The data are transformed via PCA, a projection-based approach, into a collection of orthogonal axes. However, for these tactics to be effective, the training data need to be neutral and equally dispersed.

Defense Mechanism 4: To prevent federated learning from being compromised by model poisoning assaults, the authors of [55] suggested a new method called Romoa, which aggregates models for protection. In contrast to earlier research, Romoa can handle both specific and general poisoning assaults with a single strategy. By using a novel similarity metric, Romoa can improve fairness for federated learning participants and enable more accurate attack detection. We suggest that Romoa may offer sufficient protection against model poisoning attacks, particularly those assaults that break Byzantine-robust federated learning methods, based on an extensive examination of standard datasets.

Table 2. Comparison of aggregation rules as mentioned in [7].

Related Papers	Defense Mechanism	Model Accuracy
Krum	Krum + ERR	38%
	Krum + LFR	42%
	Krum + union (ERR+LFR)	52%
Trimmed mean	Trimmed mean + ERR	83%
	Trimmed mean + LFR	82 %
	Trimmed mean + union (ERR+LFR)	82%
Median	Median + ERR	79 %
	Median + LFR	80%
	Median + union (ERR+LFR)	81 %
Romoa	Romoa + similarity + union	93%

A different protective strategy is immune to all poisons [56]. They established a cap on the severity of poison attacks and governmental guarantees about the defense's convergence when it is used. A thorough analysis of their attacks and countermeasures was performed on three genuine datasets from the healthcare, loan assessment, and real estate industries. The results are shown in Table 3.

Table 3. Comparison of the RONI and TRIM poisoning rates on the bases of MSE.

Related Papers	Defense Mechanism	Model Accuracy
RONI [18]	With 12%	3%
	With 20%	6%
TRIM [18]	With 12%	0%
	With 20%	0%

4.2. *Inference Attacks and Defenses*

Gradient trading may result in significant protection spillage during FL preparation [57–59]. Because deep learning algorithms seem to internally find various data qualities that are unrelated to the core aim [60], model updates may expose more information about undesired traits to adversarial players, as depicted in Figure 7.

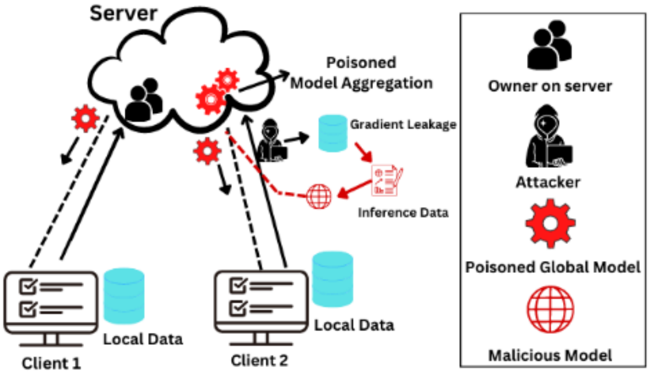

Figure 7. Inference Attack [13].

An attacker can also examine the changes between subsequent snapshots, which are equal to the aggregated updates from all participants minus the adversary, to determine attributes. The use of generative adversarial networks (GANs) has been recommended to tackle deep FL models [61]. It is possible that one player willfully compromises another. Figure 2 shows different mechanisms that can be used to address these attacks.

The attacker may also use the FL model parameters screenshot to determine attributes by examining the changes between subsequent snapshots, which are equal to the aggregated updates from all participants minus that of the adversary. The use of generative adversarial networks (GANs) was recommended to tackle deep FL models [62]. The GAN attack presupposes that just one participant provided the training data for the entire class and that only in that case are the representations produced by the GAN accurate duplicates of the training data. In typical ML contexts, this is akin to model inversion attacks [63].

For instance, information about preparation information may, under some conditions, be inferred from model changes made throughout the learning system [64,65]. To learn more about the preparation of other members unrelated to the elements that constitute the classes in the FL model, an adversary may employ both passive and active property forecasting attacks [66]. Deep leakage from gradient (DLG), a novel optimization approach, can quickly generate training inputs and labels [37]. The relationship between the labels and symbols of the linked gradients is used by the analytical method known as improved deep leakage from gradient (iDLG) to obtain tags from those gradients [38]. Figure 8 and Table 4 show the defense mechanisms of inference attacks.

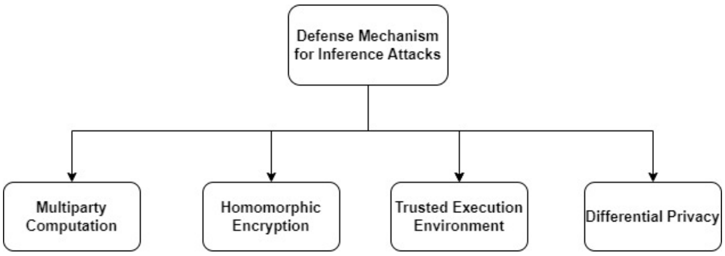
Defense Mechanism for Inference Attacks
Multiparty Computation
Homomorphic Encryption
Trusted Execution Environment
Differential Privacy

Figure 8. Defense Mechanism for Inference Attacks.

Defense Mechanism 1: Secure Multiparty Computation (SMC) is a fundamental technique used in secure computation [12]. A minimum of two groups must agree to the SMC method for it to be used, and the results must be made available to a subset of participants. In addition, in secure SMC, two processes were used; single masking, which protects participants' information, and chained communication, which enables the transfer of masked information between participants through a serial chain frame.

Defense Mechanism 2: Differential privacy (DP). By introducing noise to the clipped model parameters before model aggregation in differential privacy systems, the user's participation is hidden [39]. This mechanism secures data to some extent but it is difficult to achieve higher accuracy as some accuracy is lost when noise is introduced into the model parameters.

Defense Mechanism 3: A trusted environment for execution. The Trusted Execution Environment (TEE) provides a secure environment for performing the federated learning process with a low computational cost as compared to other safe computing approaches [40]. The existing TEE environment was virtually tuned for CPU devices. This mechanism works well for CPU-enabled devices but struggles to configure with other small devices.

Defense Mechanism 4: Homomorphic encryption executes estimations on inputs that have been encoded without first being decoded. When information is encrypted using a homomorphic algorithm, the result is ciphertext that behaves exactly like the original data when it is decrypted. With homomorphic encryption, sensitive information may undergo extensive mathematical processing without risking decryption. It is now impossible to break the encryption, and experts believe that it is even secure against quantum computers. However, assaults on the encryption process are not impossible. Therefore, private information may be sent and examined without fear of disclosure, provided the recipient has the right decryption key.

Table 4. Defense mechanism against inference attacks as mentioned in [46].

Defense Mechanism			**Attack**
Differential privacy	**SMC**	**Homomorphic encryption**	
Effective	Ineffective	Effective	**Loss function**
Effective	Limited effectiveness	Effective	**Deep leakage gradient**
Context-dependent effectiveness	Effective	Effective	**mGAN**
Effective	Ineffective	Ineffective	**GAN**
Effective	Ineffective	Ineffective	**Adversarial example**

4.3. Backdoor Attacks and Defenses

The attack prediction model poisoning falls under the category of backdoor attacks, as shown in Figure 9. However, in comparison, these are less distinct. They work by retaining the accuracy of the main task while introducing covert backdoors into the global model [31]. Compromised FL participating machines may establish a backdoor by training the model on specified backdoor data [13]. Finding backdoor models becomes more challenging when they do not diverge from other models [62]. By using an objective function that rewards model correctness and penalizes it for deviating from what the aggregator defense considers to be within its acceptance threshold, anomaly detection is avoided [63].

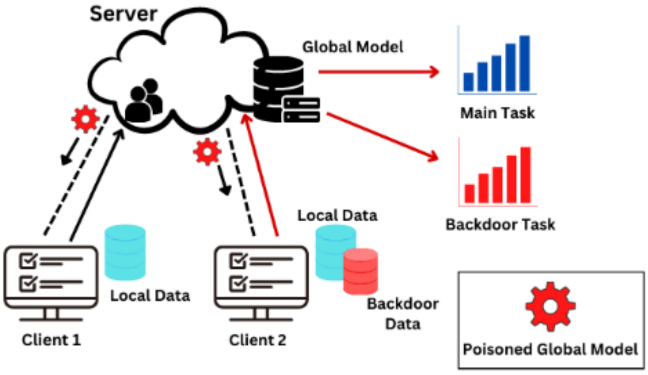

Figure 9. Backdoor Attack.

Backdoor attacks, as shown in Figure 9, might be disastrous since they can predict false positives with a high degree of precision. Additionally, sophisticated backdoor attacks [13] in FL could be able to circumvent the catastrophic forgetting problem and keep the backdoor from being forgotten while training is taking place. Secure averaging in federated learning enables devices to maintain their anonymity during the model updating procedure. A device or group of devices may add the backdoor capability to the global federated learning model using the same capabilities [31]. Without affecting the model's overall accuracy, an opponent might employ a backdoor to incorrectly categorize some jobs. An attacker, for instance, might pick a certain label for a data instance with a particular set of traits. Backdoor attacks are often referred to as targeted assaults [12].

A backdoor attack usually focuses on a single input that was incorrectly classified as belonging to the attacker's class. [64]. A backdoor model may misclassify input from any class marked with the trigger into the target class, which largely determines the attack. A backdoor attack can be classified as class-specific or class-agnostic [64] depending on whether the trigger effect is dependent on the target classes. A backdoor model may misclassify input into the target class from any class marked by the existence of the trigger. A backdoor model may misclassify the input from specified classes stamped with the trigger into the targeted class. Thus, the assault is determined by the existence of a trigger, together with the class of the target.

Defense Mechanism: Norm thresholds or weak differential privacy are updated to prevent backdoor attacks. Even if this impairs the performance of the global model, participant-level differential privacy might act as a form of defense against such attacks. Differential privacy ensures that hackers cannot reverse-engineer data pieces to identify people, even if they obtain access to data containing such information. This decreases the danger of individual data compromise even when the source data itself is compromised [39]. However, enhanced model parameters can be eliminated from models using norm thresholding. Due to the capabilities of the deep learning model and secure aggregation techniques, it can be difficult to identify malicious individuals even with these safeguards. Being a distributed system, the FL framework makes it more difficult to regulate randomly malfunctioning components. Table 5 compares the defense mechanisms of different studies.

Table 5. Backdoor Countermeasure summary [64].

Domain	Work	Model Access	Poisoned Data Access
Blind backdoor removal	Fine pruning	White-box	Inapplicable
	Suppression	Black-box	Inapplicable
	RAB	White-box	Applicable
Offline data inspection	Activation clustering	White-box	Applicable
	Gradient clustering	White-box	Applicable
	Differential privacy	White-box	Applicable
Offline model inspection	DeepInspect	Black-box	Inapplicable
	Meta classifier	Black-box	Inapplicable
Online input inspection	STRIP	Black-box	Inapplicable
	Epistemic classifier	White-box	Inapplicable
	NNoculation	White-box	Inapplicable
Online model inspection	ABS	White-box	Inapplicable

4.4. Evasion Attacks and Defenses

In evasion assaults [47,48], an adversary carefully manipulates the information tests conducted in the sent model to avoid detection. The term "antagonistic instances" refers to evasion attacks that use altered tests that, to a human, appear realistically consistent with the original information tests. They are, however, specifically designed to deceive a classifier. Evasion attacks [13] are not new or uncommon in FL, but they do have specific flaws in such environments. Perturbations are created by using limited optimization techniques, such as projected gradient ascent, to maximize the loss function while keeping a norm restriction in mind, as described in the white box. In a black-box scenario, adversaries can take the place of models trained on analogous data. By allowing the attacker to access the model and the local training loss function, FL facilitates various attacks. One of the most frequent examples is when a photo's pixels are changed before being uploaded, which prevents the image recognition system from correctly categorizing the result. In truth, the confrontational situation has the power to trick people. In terms of their understanding of the target device, attackers can carry out three different types of attacks. White-box, gray-box, and black-box are the different classifications [49]. It is assumed in that, in white-box [47], the assailant is knowledgeable in all fields. They could, for example, be a bank intern who has gathered all of the necessary information. Even though this is a firm premise, evaluating fraud detectors in the worst-case scenario is beneficial. This scenario can also be used as an upper bound when compared to other, more constrained scenarios.

Additionally, in gray-box [50], the attacker is only vaguely familiar with the detection system. The fraud detector in particularly is aware of the data collection process used to determine the features but is unaware of the training data, learning process, or learned hyperparameters. The attacker is not aware of the detection method or training data in the black box [49,50]. For bank transactions, however, we assume that the attacker had a month's worth of earlier transactions to estimate the aggregated features. Figure 10 and Table 6 shows the adversarial defense mechanisms of evasion attack.

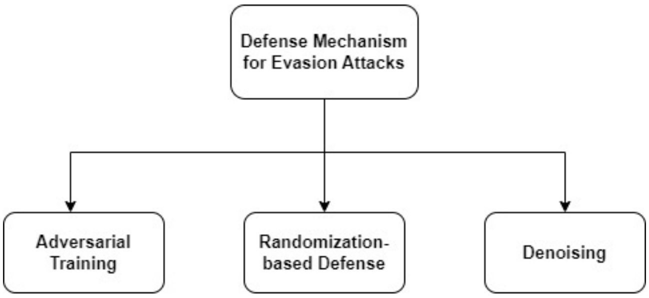
Defense Mechanism for Evasion Attacks
Adversarial Training
Randomization-based Defense
Denoising

Figure 10. Defense Mechanism for Evasion Attacks.

Defense Mechanism 1: Adversarial training provides a natural defensive strategy against performing malicious samples by exposing a neural network against the aggressive provision of guidance [11]. To mitigate adversarial assaults, malicious samples must be created and incorporated into the training phase. This method trains a system to survive typical malicious sample production approaches. Adversarial training is one of the best defenses against adversarial assaults, according to recent research in the references [51–53].

Defense Mechanism 2: The majority of DNNs are not concerned with randomization-based defense [11] seeking to transform adversarial consequences into random effects. Although randomization-based defenses have equal effectiveness in both gray-box and black-box settings, the EoT technique [54] may undermine the majority of them in white-box environments by integrating the randomization process into the attack process.

Defense Mechanism 3: Adversarial training, which exposes a neural network to perform malicious samples during training, provides a natural defensive strategy against aggressive samples [11]. According to recent research in references [51–53], providing guidance is one of the best defenses against adversarial assaults.

Table 6. Adversarial defense mechanisms [11].

Defense Mechanism		Accuracy
Adversarial training	FGSM adversarial training [56]	Model accuracy up to 83%
	PGD Adversarial training [56]	Model accuracy up to 88.56% under white-box attacks
	Adversarial Logit pairing [57]	Accuracy goes from 1% to 27.9% under white-box
Randomization	Random input transformation [58]	Accuracy is 60% for gray-box and 90% for black-box attacks
	Random noising [59]	Model accuracy up to 86 %
	Random feature pruning [60]	Accuracy increases to 16% depending on perturbation size
Denoising	GAN-based input cleansing [61]	Error rate evaluated up to 0.9%
	Feature denoising	50% model accuracy under white-box attacks

4.5. Model Inversion Attacks and Defenses

While training a supervised neural network, a secret dataset is kept from the public eye. Model inversion attacks are attempts by a hostile user to retrieve that dataset. Ref. [67] provided a probabilistic explanation for model inversion assaults and developed a variational target that takes into consideration both variety and precision. To achieve this variational goal, we selected a variational family defined inside the code space of a deep generative model and trained it on a publicly available auxiliary dataset that is structurally comparable to the target dataset. On empirical datasets consisting of photos of faces and chest X-rays, our technique significantly outperforms the state of the art in terms of target attack accuracy, sample realism, and variety. An attacker with access to both the training set and the additional dataset A may perform a model inversion attack and retrieve certain variables from training set B for those people. All of the variables from A and some from B are interconnected here, and thus the new individual dataset in question would have to include both sets of variables. Data retrieved from individuals in the training dataset will be more accurate than characteristics merely inferred from individuals who were not included in the training dataset due to the possibility of making a mistake and an inaccuracy in the latter.

Defense Mechanism: Similarly, [68] suggested a revolutionary model inversion attack (MIA) search technique, in which a pre-trained deep generative model capable of producing a face picture from a random feature vector is utilized to reduce the dimensionality of the search space from images to feature vectors. As a result, the MIA procedure can quickly find the low-dimensional feature vector matching the face picture with the highest confidence score. Another study compared the PCA technique to random seed GAN-integrated MIA, DCGAN-integrated MIA (DCGAN-MIA), and standard MIA in an experimental setting, using two objective criteria and three subjective factors. The results of the tests show that the suggested method is effective and superior in creating clones of people's faces that are almost indistinguishable from the originals.

4.6. Reconstruction Attacks and Defenses

The reconstruction attack refers to any technique that pieces together a private dataset using only publicly available data. In most cases, the dataset includes personally identifiable information that must be shielded from prying eyes. While the attacker may not have access to the dataset itself, they may have access to public aggregate statistics about the datasets, which may be accurate or corrupted in various ways (such as the addition of noise) [69]. An attacker may successfully reassemble a significant chunk of the original private data if the public statistics are not sufficiently skewed. Reconstruction attacks are relevant for the study of private data because they demonstrate the need to properly distort any released statistics to protect even a very limited idea of individual privacy. Many of the first studies on differential privacy were driven by the discovery of reconstruction attacks, which directly led to the notion of differential privacy.

Defense Mechanism: This attack can be secured by the safety limit of gradient-based data reconstruction via a microscopic perspective on neural networks using rectified linear units (ReLUs), the de facto standard for activation functions. The threshold for a successful data reconstruction attack is measured in terms of the proportion of a training set that consists of exclusively activated neurons. The number of ExANs in a training set is inversely proportional to the risk of a data reconstruction attack, and vice versa [70]. Numerous attack techniques significantly surpass their predecessors and come to rebuild training batches located around the neural network's vulnerable border. Meanwhile, exclusivity reduction techniques are developed to increase the safe area around training batches that are already within the secure border. This is done as a kind of mitigation since unique reconstruction is impossible in this case. An adversarial training-based system with three modules—adversarial reconstruction, noise regularization, and distance correlation minimization—provides protection against a reconstruction assault. Because they are decoupled from one another, those modules may be used alone or in tandem. This methodology has been shown to be successful in safeguarding input privacy while maintaining the model's functionality via extensive trials on a large-scale industrial Internet advertising dataset.

5. Discussion and Open Research Directions

Healthcare and transportation both employ federated learning. Although FL frameworks offer more privacy protection than other ML frameworks, they are not susceptible to several attacks. Additionally [12,71], the distributed form of the system makes it far more challenging to put defensive mechanisms in place. The following are some of the difficulties of federated learning:

- **Communication Costs and Variations in the System:** Due to the high number of devices, local processing is significantly slower in federated networks [65] (e.g., millions of smartphones). Communication to cross the distortion could be significantly more expensive than it is in conventional data centers. To fit a model to data provided by the federated network, it is crucial to develop communication-efficient algorithms that repeatedly transmit brief messages or model changes as part of the training process rather than sending the entire dataset across the network.
  Due to variances in hardware (CPU, RAM), network connection (3G, 4G, 5G, WiFi), and power (battery level), each device in a federated network may have varied storage, computation, and communication capabilities [3]. Due to network capacity and system-related restrictions, it is also typical to only observe a small portion of the network's devices active at any given moment. In a network of a million devices, it is feasible that only a few hundred are actually active [30]. Any active device may become inactive at any time for a variety of reasons. These system-level characteristics account for why issues such as stragglers and fault tolerance are more important than in typical data centers [65].
- **Diversity in statistics:** Devices on the network frequently produce and gather non-identically scattered data [65,72]. For instance, mobile phone users may utilize a range of slang and jargon when asked to predict the next phrase [30]. More significantly, there might be a foundational structure that depicts the connection between devices and the dispersion of data points between them. It is possible that distributed optimization will experience hiccups as a result of this data production paradigm, and modeling, analysis, and evaluation will all be more challenging.
- **Robustness to adversarial attacks:** It has been shown that neural networks are vulnerable to a wide variety of adversarial attacks, such as undetected adversarial samples [3]. As neural networks are more frequently employed in federated settings, more people will be exposed to them. While the issue of adversarial robustness [30] is still under investigation, there are a few recommendations to address it, including the following: 1) developing newer robustness metrics for images, text, and audio; 2) including robustness audits in the deployment process; and 3) continuously testing deployed models against unidentified adversaries. Federated learning is still in its infancy, but it will be a popular topic in research for a very long time. As the game goes on, FL's attack strategies will alter [8,73]. Designers of FL systems should be aware of current assaults so that they may take protective measures when developing new systems. This survey offers a concise and easily readable analysis of this subject to better comprehend the threat situation in FL. Global cooperation on FL is being promoted via an increasing number of seminars at significant AI conferences [66]. A multidisciplinary effort spanning the entire research community will be necessary to develop a general-purpose defensive mechanism that can withstand a wide range of assaults without degrading model performance.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, H.S.S., H.W.; methodology, H.S.S.; software, H.S.S.; validation, W.R., S.U.R.M.; formal analysis, H.S.S., S.T.; investigation, H.S.S.; resources, H.W.; data curation, S.T.; writing—original draft preparation, H.S.S.; writing—review and editing, H.S.S.; visualization, H.S.S., H.W.; supervision, S.U.R.M.; project administration, S.T.; funding acquisition, W.R. All authors reviewed the results and approved the final version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data has been collected from all the sources that are presented here in references section.

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

References

1. Sun, G.; Cong, Y.; Dong, J.; Wang, Q.; Lyu, L.; Liu, J. Data Poisoning Attacks on Federated Machine Learning. *IEEE Internet Things J.* **2021**, 1. http://doi.org/10.1109/jiot.2021.3128646.
2. L. Lyu et al., "Privacy and Robustness in Federated Learning: Attacks and Defenses," in IEEE Transactions on Neural Networks and Learning Systems, 2022, doi: 10.1109/TNNLS.2022.3216981..
3. Jere, M.; Farnan, T.; Koushanfar, F. A Taxonomy of Attacks on Federated Learning. *IEEE Secur. Priv.* **2021**, *19*, 20–28. http://doi.org/10.1109/msec.2020.3039941.
4. Tolpegin, V.; Truex, S.; Gursoy, M.; Liu, L. Data Poisoning Attacks Against Federated Learning Systems. *Comput.-Secur. Esorics* **2020**, *2020*, 480–501. http://doi.org/10.1007/978-3-030-58951-624.
5. Fung, C.; Yoon, C.; Beschastnikh, I. Mitigating sybils in federated learning poisoning. *arXiv* **2018**, arXiv:1808.04866.
6. Alfeld, S.; Zhu, X.; Barford, P. Data Poisoning Attacks against Autoregressive Models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
7. Huang, Y.; Gupta, S.; Song, Z.; Li, K.; Arora, S. Evaluating gradient inversion attacks and defenses in Federated learning. *arXiv* **2021**.
8. Fang, M.; Cao, X.; Jia, J.; Gong, N. Local model poisoning attacks to Byzantine-robust federated learning. *arXiv* **2019**.
9. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning. *Acm Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19, 2019. http://doi.org/10.1145/3298981.
10. Huang, L.; Joseph, A.; Nelson, B.; Rubinstein, B.; Tygar, J. Adversarial machine learning. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence—AISec '11, Chicago, IL, USA, 21 October 2011. http://doi.org/10.1145/2046684.2046692.
11. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. http://doi.org/10.1109/tnnls.2018.2886017.
12. Federated Learning: A Step by Step Implementation in Tensorflow, Medium, 2022. [Online]. Available online: https://towardsdatascience.com/federated-learning-a-step-by-step-implementation-in-tensorflow-aac568283399 (accessed on 18 January 2022).
13. Melis, L.; Song, C.; Cristofaro, E.D.; Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706, http://doi.org/10.1109/SP.2019.00029.
14. Goldblum, M.; Tsipras, D.; Xie, C.; Chen, X.; Schwarzschild, A.; Song, D.; Madry, A.; Li, B.; Goldstein, T. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv* **2020**.
15. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated Optimization: Distributed machine learning for on-device intelligence. *arXiv* **2016**.
16. Abdulrahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond. *IEEE Internet Things J.* **2021**, *8*, 5476–5497. http://doi.org/10.1109/JIOT.2020.3030072.
17. Exclusive: What Is Data Poisoning and Why Should We Be Concerned?—International Security Journal (ISJ), International Security Journal (ISJ), 2022. [Online]. Available online: https://internationalsecurityjournal.com/what-is-data-poisoning/ (accessed on 26 January 2022).
18. Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; Li, B. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 19–35, doi: 10.1109/SP.2018.00057.
19. Awan, S.; Luo, B.; Li, F. CONTRA: Defending Against Poisoning Attacks in Federated Learning. *Comput. Secur. Esorics* **2021**, *2021*, 455–475. http://doi.org/10.1007/978-3-030-88418-522.
20. Phong, L.; Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1333–1345. http://doi.org/10.1109/tifs.2017.2787987.
21. Su, L.; Xu, J. Securing Distributed Gradient Descent in High Dimensional Statistical Learning. *Acm Meas. Anal. Comput. Syst.* **2019**, *3*, 1–41. http://doi.org/10.1145/3322205.3311083.
22. Chen, X.; Liu, C.; Li, B.; Lu, K.; Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv* **2017**.
23. Gu, T.; Dolan-Gavitt, B.; Garg, S. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv* **2017**.
24. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. *arXiv* **2018**.
25. Cretu, G.F.; Stavrou, A.; Locasto, M.E.; Stolfo, S.J.; Keromytis, A.D. Casting out Demons: Sanitizing Training Data for Anomaly Sensors. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 18–21 May 2008; pp. 81–95, http://doi.org/10.1109/SP.2008.11.
26. Steinhardt, J.; Koh, P.W.; Liang, P. Certified defenses for data poisoning attacks. *arXiv* **2017**, arXiv:1706.03691.

27. Seetharaman, S.; Malaviya, S.; Kv, R.; Shukla, M.; Lodha, S. Influence based defense against data poisoning attacks in online learning. *arXiv* **2021**.
28. Li, Y. Deep reinforcement learning: An overview. *arXiv* **2017**, arXiv:1701.07274.
29. Wang, Y.; Mianjy, P.; Arora, R. Robust Learning for Data poisoning attacks. In Proceeding of the Machine Learning Research, Virtual, 18–24 July 2021.
30. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *arXiv* **2019**.
31. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statististics, Online, 26–28 August 2020; pp. 2938–2948.
32. Shafahi, A.; Huang, W.R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; Goldstein, T. Poison frogs! Targeted clean-label poisoning attacks on neural networks. *arXiv* **2018**.
33. Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E.C.; Roli, F. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017. http://doi.org/10.1145/3128572.3140451
34. Turner, A.; Tsipras, D.; Madry, A. Clean-Label Backdoor Attacks, OpenReview, 2022. [Online]. Available online: https://openreview.net/forum?id=HJg6e2CcK7 (accessed on 31 January 2022).
35. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep Models Under the GAN. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3November 2017. http://doi.org/10.1145/3133956.3134012.
36. 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. http://doi.org/10.1145/2810103.2813677.
37. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from gradients. *arXiv* **2019**.
38. Zhao, B.; Mopuri, K.; Bilen, H. IDLG: Improved Deep Leakage from gradients. *arXiv* **2020**.
39. Geyer, R.; Klein, T.; Nabi, M. Differentially private federated learning: A client level perspective. *arXiv* **2017**.
40. Mo, F.; Haddadi, H. Efficient and private federated learning using tee; In EuroSys, Dresden, Germany, 25–28 March 2019.
41. Mammen, P. Federated learning: Opportunities and challenges. *arXiv* **2021**.
42. Miao, C.; Li, Q.; Xiao, H.; Jiang, W.; Huai, M.; Su, L. Towards data poisoning attacks in crowd sensing systems. In Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Los Angeles, CA, USA, 26–29 June 2018; pp. 111–120.
43. Bouacida, N.; Mohapatra, P. Vulnerabilities in Federated Learning. *IEEE Access* **2021**, *9*, 63229–63249. http://doi.org/10.1109/ACCESS.2021.3075203.
44. Peri, N.; Gupta, N.; Huang, W.R.; Fowl, L.; Zhu, C.; Feizi, S.; Goldstein, T.; Dickerson, J.P. Deep k-NN Defense Against Clean-Label Data Poisoning Attacks. In Proceedings of the Computer Vision – ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020; pp. 55-70, 2020. http://doi.org/10.1007/978-3-030-66415-24.
45. A Study of Defenses against Poisoning Attacks in a Distributed Learning Environment—F-Secure Blog, F-Secure Blog, 2022. [Online]. Available online: https://blog.f-secure.com/poisoning-attacks-in-a-distributed-learning-environment/ (accessed on 26 January 2022).
46. Enthoven, D.; Al-Ars, Z. An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies. *Fed. Learn. Syst.* **2021**, 173–196. http://doi.org/10.1007/978-3-030-70604-38.
47. Carminati, M.; Santini, L.; Polino, M.; Zanero, S. Evasion attacks against banking fraud detection systems. In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses, San Sebastian, Spain, 14–15 October 2020; pp. 285–300.
48. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Srndic, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion Attacks against Machine Learning at Test Time. *Adv. Inf. Syst. Eng.* **2013**, 387–402. http://doi.org/10.1007/978-3-642-40994-325.
49. "How to Attack Machine Learning (Evasion, Poisoning, Inference, Trojans, Backdoors)", Medium, 2022. [Online]. Available online: https://towardsdatascience.com/how-to-attack-machine-learning-evasion-poisoning-inference-trojans-backdoors-a7cb5832595c (accessed on 26 January 2022).
50. Demontis, A.; Melis, M.; Pintor, M.; Jagielski, M.; Biggio, B.; Oprea, A.; NitaRotaru, C.; Roli, F. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In Proceedings of the 28th USENIX Security Symposium (USENIX Security 19), Santa Clara, CA, USA, 14–16 August 2019; pp. 321–338.
51. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**.
52. Xie, C.; Wu, Y.; Maaten, L.; Yuille, A.; He, K. Feature Denoising for Improving Adversarial Robustness, *arXiv* **2019**, arXiv:1812.03411.
53. Carlini, N.; Katz, G.; Barrett, C.; Dill, D. "Ground-Truth Adversarial Examples", OpenReview, 2022. [Online]. Available online: https://openreview.net/forum?id=Hki-ZlbA- (accessed on 31 January 2022).
54. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing robust adversarial examples. *arXiv* **2017**.

55. Mao, Y., Yuan, X., Zhao, X., Zhong, S. (2021). Romoa: Robust Model Aggregation for the Resistance of Federated Learning to Model Poisoning Attacks. In: Bertino, E., Shulman, H., Waidner, M. (eds) Computer Security – ESORICS 2021. ESORICS 2021. Lecture Notes in Computer Science(), vol 12972. Springer, Cham. https://doi.org/10.1007/978-3-030-88418-5_23
56. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**.
57. Kannan, H.; Kurakin, A.; Goodfellow, I. Adversarial Logit Pairing. *arXiv* **2018**.
58. Guo, C.; Rana, M.; Cisse, M.; Maaten, L.v. Countering Adversarial Images using Input Transformations. *arXiv* **2017**.
59. Liu, X.; Cheng, M.; Zhang, H.; Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *Computer Vision – ECCV 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 381–397.
60. Dhillon, G.S.; Azizzadenesheli, K.; Lipton, Z.C.; Bernstein, J.; Kossaifi, J.; Khanna, A.; Anandkumar, A. Stochastic Activation Pruning for robust adversarial defense. *arXiv* **2018**.
61. Shen, S.; Jin, G.; Gao, K.; Zhang, Y. APE-GAN: Adversarial perturbation elimination with GAN. *arXiv* **2017**.
62. Liu, K.; Dolan-Gavitt, B.; Garg, S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. *Res. Attacks Intrusions Defenses* **2018**, 273–294. http://doi.org/10.1007/978-3-030-00470-513.
63. Jiang, Y.; Wang, S.; Valls, V.; Ko, B.J.; Lee, We.; Leung, K.K.; Tassiulas, L. Model pruning enables efficient federated learning on edge devices. *arXiv* **2019**.
64. Gao, Y.; Doan, B.G.; Zhang, Z.; Ma, S.; Zhang, J.; Fu, A.; Nepal, S.; Kim, H. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv* **2020**.
65. Li, T. "Federated Learning: Challenges, Methods, and Future Directions", Machine Learning Blog | ML@CMU | Carnegie Mellon University, 2022. [Online]. Available online: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/ (accessed on 31 January 2022).
66. 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, 24–26 October 2016. http://doi.org/10.1145/2976749.2978318.
67. Wang, K.-C.; Fu, Y.; Li, K.; Khisti, A.; Zemel, R.; Makhzani, A. Variational Model Inversion Attacks. Advances in Neural Information Processing Systems. *Adv. Neural Inf. Process. Syst.* **2022**, *34*, 9706–9719.
68. Khosravy, M.; Nakamura, K.; Hirose, Y.; Nitta, N.; Babaguchi, N. Model Inversion Attack by Integration of Deep Generative Models: Privacy-Sensitive Face Generation from a Face Recognition System. *IEEE Trans. Inf. Forensics Secur.* **2022**, *67*, 9074–9719.
69. Garfinkel, S.; Abowd, J.M.; Martindale, C. Understanding database reconstruction attacks on public data. *Commun. ACM* **2019**, *62*, 46–53.
70. Lyu, L.; Chen, C. A novel attribute reconstruction attack in federated learning. *arXiv* **2021**, arXiv:2108.06910.
71. Xie, C.; Huang, K.; Chen, P.; Li, B . Dba: Distributed backdoor attacks against federated learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019 .
72. Wei, W.; Liu, L.; Loper, M.; Chow, K.; Gursoy, M.; Truex, S.; Wu, Y. A framework for evaluating gradient leakage attacks in federated learning. *arXiv* **2020**.
73. Biggio, B.; Nelson, B.; Laskov, P. Poisoning attacks against Support Vector Machines. *arXiv* **2012**.