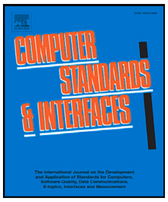
Computer Standards & Interfaces 86 (2023) 103723

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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



A robust analysis of adversarial attacks on federated learning environments

Akarsh K. Nair, Ebin Deni Raj, Jayakrushna Sahoo [*]

Indian Institute of Information Technology, Kottayam, India

ARTICLE INFO

Keywords:
Federated learning
Distributed learning
Machine learning
IoT
Artificial intelligence
Security issues in federated learning

ABSTRACT

Federated Learning is a growing branch of Artificial Intelligence with the wide usage of mobile computing and IoT technologies. Since this technology uses distributed computing paradigm to do the learning part, most of the participating components are mobile devices and come outside the range of protection offered by a centralized system. As a result, several security issues such as data leakage, communication issues, poisoning, system manipulation via the backdoor, and so on arise with the usage of such a methodology. These sorts of attacks are categorized into various categories concerning their modus operandi. In this study, we review such attacks, namely poisoning attacks, inferencing attacks, their types, and working in a Federated Learning environment in detail. This study will give a precise idea of security issues faced in Federated Machine Learning and possible solutions.

1. Introduction

Federated Learning (FL) is an emerging concept in Machine Learning (ML) where the training happens in a highly decentralized manner with many client devices performing collaborative operations which are collectively managed by a central server device [1]. The major peculiarity of such a training scenario is that the training data remains decentralized, i.e. the training happens at the location of the data and not the other way around as in traditional ML approaches. This methodology substantiates the basic principles of targeted collection and data minimization, alleviating several privacy risks as well as overheads associated with the traditional approach of centralized ML [2]. In recent times, the area has been receiving considerable attention from all angles of the industry as well as academia. FL had opened up several research opportunities in AI as well. With the widespread of AI integrated chips, the computing capabilities of mobile devices have increased considerably. As a result, the AI model training methods are also on a gradual shift, i.e. from a centralized fashion to a decentralized one.

The formal proposal of FL was initially made by Google, describing it as a distributed ML paradigm that uses user interaction along with multiple mobile devices to facilitate ML procedures [3]. The peculiarity of FL is that, unlike traditional ML models, the FL methodology does not rely on an individual system for training but multiple smaller systems are used for the purpose. It enables mobile devices to collectively learn a shared model without sharing the data required for training to a server or another device, stopping the usual practice of storing complete data as a single entity in the training device [4]. These trained models work by making changes to an existing model with the aid of training done at the device end with private data. It improves the whole outlook of learning procedures and the changes observed while training at the mobile devices will be summarized as updates. Such updates are sent to the central device via protected channels where those will be aggregated and needed improvements will be made to the shared model. For the aggregation purpose, a commonly used technique is the "Federated Averaging" approach and several new optimized variants of this approach are now made available. The major advantage of such a system is that the training data need not be shared and it always remains at the device itself ensuring privacy [5].

Federated Learning has been under extensive research for the last couple of years and several newer terminologies are arising with respect to the domain. On a baseline, FL can be divided into two categories i.e. Model Centric FL and Data Centric FL [6]. Model centric FL as the name suggests works on a model-oriented basis whereas data centric FL functionalities are more inclined towards data. Similarly, depending upon the type of devices participating in the FL training, FL can be classified into cross device FL [7] and cross silo FL [8]. Cross device implies training done among smaller devices such as mobile devices, IoT devices, or small scale computer systems and cross-silo implies FL among a group of larger entities usually institutions such as hospitals, banks and so on or even high-end data centers. Another classification of FL also exists based upon the pattern of data distribution, i.e. Horizontal FL and Vertical FL [1]. Unsurprisingly, data distribution happens in a horizontal manner in horizontal FL with identical feature space but dissimilar space in the sample. In vertical FL, the cases need to have

[*] Corresponding author.
E-mail address: jsahoo@iiitkottayam.ac.in (J. Sahoo).

https://doi.org/10.1016/j.csi.2023.103723
Received 28 December 2021; Received in revised form 20 June 2022; Accepted 31 December 2022
Available online 6 January 2023

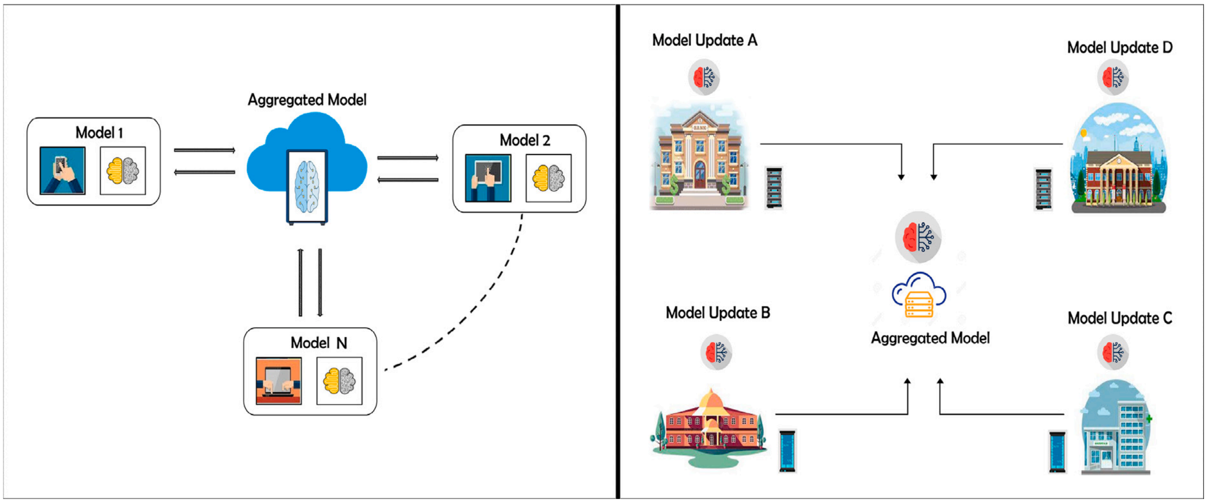


Fig. 1. (a) Overview of FL training on devices (b) Overview of FL training on Silos.

identical sample space but dissimilar feature space [9]. Fig. 1 provides an overview of the basic implementation of FL in different setups.

When compared to traditional methodologies, FL models are usually highly efficient in the way they perform with reduced latency and power efficiency with assured privacy. For having a feasible practical implementation, many hurdles need to be crossed on the algorithmic as well as the technical side. A typical optimization algorithm used in ML is not that much effective when it comes to FL. Most of those algorithms are accustomed to working with a homogeneous portioned data set in a server environment whereas, in an FL environment, data distribution is extremely irregular covering a large number of devices [4]. So the prime focus is to make use of high-end processors in mobile devices capable of generating better updates with increased accuracy. With the increasing accuracy of model updates, the system requires a smaller number of iterations to generate an efficient model. Even though FL has the capability to entirely change the way traditional learning methodologies work, the state-of-the-art work in Federated Learning has barely reached anywhere near the expected [10]. As of now, FL has not reached a stage where it is capable of solving all ML-related problems.

Moreover, some of the major challenges faced in an FL setting are related to communication, the diversity of the participating devices, and privacy concerns. In the case of communication, federated networks face an intensive bottleneck [11]. Whenever an additional dimension of privacy issues gets added to it, the system entails local storage to generate data and does not advise the transfer of raw data. Federated networks will literally be constituting a huge number of mobile devices and due to this, the speed of communication inside the network will be manifold slower when compared to local processes. Limited availability of resources like bandwidth and power adds to the complexity even more [12]. So, the need for developing an efficient communication method is the prime consideration, also keeping in mind that it should be capable of repetitively sending small messages or model upgrades during the process of training apart from communication [13].

During the learning process, FL makes use of a large number of devices having diverse characteristics. They may differ from one another in every aspect, starting from storage, computational and communication capabilities, connectivity to power, and so on [12]. As the network size becomes huge, the chances of active devices willing to participate in the training procedures may also vary highly. In a network with thousands of devices, only a few may be active participants. In addition to that, the possibility for an active device to suddenly go offline at some point of processing also exists [11]. Also, in distributed environments, straggler mitigation and fault tolerance are usual issues but the above-mentioned conditions escalate them even more. Thus issues such as less number of participants, tolerance for discrepancy in hardware, and robustness towards changing availability of components of the network are prevalent.

Another challenge faced in FL is statistical heterogeneity. The devices in the network will be constantly generating and collecting data in a very much dissimilar and distributed manner. Also, the count of these data points in a device may change considerably and a hidden statistical structure may exist which can depict the connection between devices and their related distributions [14]. Similar to other methodologies, privacy is the prime concern in FL as well. Thus, the system prefers the transfer of updated models rather than raw data as an extended privacy preservation strategy. During the training process, a constant update of the model is provided which has the possibility of revealing valid information to an outside source [15]. For enhancing privacy, recent proposals suggest the usage of techniques such as secure multiparty computation (SMC) [16] or differential privacy [17] but the fact is that such a goal can only be achieved by sacrificing system efficiency or performance. Finding a balance in both theoretical and empirical methods is a prime issue to be sorted out while understanding private FL systems [18].

In an FL environment, the processors are collaborative and there is constant interaction between the participating devices. It is operating in such a way that the participating devices are capable of observing intermediary states of the model and coming up with random updates for the globally shared model [19]. All the existing approaches assume that the participating devices in FL are unmalicious and provide legitimate data for training and all the updates are generated based on this data. However, in reality, it need not be the case. The possibility of malicious users breaking the system is real and existing [20]. If such malicious users enter the system and provide faulty updates to the server, the entire training process gets compromised. In systems related to high-end organizations, this may lead to huge losses in terms of resources as well. For instance, in the case of locking systems based on biometric authentication, a backdoor created to it can grant attackers the privilege to mislead the system into authorizing the entry of random people. Such attacks can be classified into various types depending upon their modus operandi.

A number of comparative studies and review articles have been published covering the privacy preservation aspect of FL. For instance, Truong et al. [21] have published a review article where the authors present privacy preservation in Federated Learning from a General Data Protection Regulation (GDPR) perspective. The study heeds various FL-based system frameworks, attack models, and types and presents a set of existing solutions. Mothukuri et al. [22] have also presented a comparative study on the security and privacy of federated learning. They provide classification and overview of various techniques alongside examining vulnerabilities and threats in FL systems. A similar review paper is published by Lyu et al. [23] discussing the privacy and robustness concerns in FL environments.

Moreover, more prominent surveys on federated learning related to adversarial attack scenarios have been conducted [24–28]. Lyu et al. [24] have surveyed the various threats in federated learning. The authors have provided a fundamental introduction to the concept of FL and a unique taxonomy covering threat models and elemental attacks on FL systems. Bouacida et al. [25] have also reviewed the vulnerabilities of FL. Tolpegin, et al. [26] have done a comparative study exclusively on Data poisoning attack scenarios in FL. Even though the study is fascinating, the work limits itself to one type of attack, neglecting other methodologies. A similar survey concentrating on back door attacks in FL systems is presented by Sun et al. [27]. The authors have accorded backdoor attack scenarios, methodology, and other factors in detail. Jera et al. [28] also completed a survey discussing the taxonomy of attacks in FL systems.

The focus of this study is entirely different from the existing surveys. Most existing works either concentrate on privacy preservation or adversarial attack scenarios. Even though works related to adversarial attack scenarios show some degree of similarity, they mainly focus on a single mode of attack or category and analyze those attacks only. Thus, this study differentiates itself as we present a robust analysis of all the different attack scenario in FL and present it under a single roof, which from our knowledge, is one of the first attempts of the same. This study presents the following:

- The study concentrates on two major types classified with respect to their attack patterns, the Poisoning method, and the Inference method.
- The article discus the various subcategories under each attack methodology stating their motivation and motto of attacks as well.
- The article presents a comparative study on different literature available on the same and evaluates it in detail.
- Research gaps related to adversarial attacks on Federated Learning have also been discussed briefly.
- In addition, this study identifies open issues and provides a few possible paths to solutions for further research in adversarial attacks on Federated Learning

The study is divided into 6 sections. A detailed overview of threats in FL will be given in Section 2. Section 3 contains a detailed study about Poisoning attacks and Section 4 goes through the Inferencing attacks. We will be discussing the open issue for further research in Section 5 and Section 6 will be concluding the work.

2. Overview of threats in FL

In the FL scenario, the participating devices are diverse in nature and take up all forms ranging from high-end devices such as workstations, computer systems, etc. to resource constraint devices such as mobile phones, IoT devices, and so on. Due to this heterogeneity, the chances of systems getting compromised are always present and this has driven the system to take up the strategy of shared training models and formation of the global model via aggregation or averaging. The data is always kept at these local devices and there exists a central server that controls and coordinates the learning process. It is in charge of combining such model updates from the individual devices, which were previously trained locally using an individual data set. Since the data is always kept private, the clients have the privilege to control the data and local models randomly [29]. This may result in malicious participants leveraging harmful algorithms for performing different types of attacks on such systems.

In a broader sense, security attacks can be classified into two categories, targeted attacks, and untargeted attacks. Such attacks which make the system deviate from its mentioned tasks and have malicious purposes are also referred to as adversarial tasks [30]. In targeted attacks, the attacker mainly focuses on manipulating the labels of certain tasks but in the case of targeted attacks, the attacker's prime motive is to compromise the performing efficiency of the shared FL model. To safeguard the system from these attacks, the security measures call for either the detection and removal of malicious participants from attaining more training updates or procedures to abrogate the influences caused by the malicious users on the shared FL model [31,32]. Usually, untargeted attacks are easier to detect as they reduce the system performance noticeably. To put it another way, attacks that affect the universal model performance are easier to detect as they will not be able to surpass the usual security check of the system. On a wider scale, threats in FL can be classified into the following:

2.1. Insider vs. Outsider

In the FL scenario, the systems are prone to adversarial attacks from within the network or outside sources. Insider attacks are basically performed by the FL server or the participating devices of the system. Outsider attacks are majorly performed by non-participating devices and it includes acts like monitoring of communication channels of participating devices and server [33]. It can also be performed by final users of the model at the time of deployment as a service. Generally, insider attacks do more harm to the system when compared with outside attacks as it considerably increases the attacking potentiality of the attacker [34]. Due to this reason, more studies are focused on insider attacks which can be of any one of these general forms:

2.1.1. Byzantine attacks

The byzantine machines usually do not obey any protocols and send random messages to the server machine. These attackers may possess detailed cognition about the system and algorithms used in learning and conspire with one another [35]. This class of attacks is trendy as it facilitates individual clients to cause convergence issues to the model or even generate a faulty model through falsified convergence [36]. The major peculiarity of such attack types is the possibility for a single client to seriously affect the whole model and sabotage the model accuracy.

2.1.2. Sybil attacks

The Sybil attackers in an FL environment simulate several fake participant devices or use already compromised devices to build even more potent attacks on the global shared model [37,38]. Sybil attacks are usually successfully executed in systems where client devices have a free pass to join and leave the system at any given time. In such instances, an adversary may join the system and poison or influence several other client devices, finally leading to a colluding adversarial scenario.

2.2. Semi honest vs. Malicious

Depending upon the intent of the adversary accessing the system, adversarial attack scenarios can be divided into mainly two types. They are the following:

2.2.1. Semi-honest attack scenario

Attackers are more passive in such a mechanism or can even be considered as honest-but-curios. They do not diverge from the protocols of the system but try to acquire knowledge about the individual states of fellow devices. The attackers do only monitor the acquired data like parameters or aggregated gradients of the shared model [39].

2.2.2. Malicious attack scenario

Malicious attacks are usually referred to as active attacks as the attacker tries to acquire knowledge about the private states of other genuine participating devices and deviates randomly from the FL system protocols by altering, playback, or deleting messages [40]. These attack models are highly malicious, allowing adversaries to execute highly potent attacks successfully.

2.3. Training phase vs. Inference phase

Another categorization can be derived depending upon the stage of the system when it is prone to adversarial attacks. They can be classified into the following two categories:

2.3.1. Training phase

Attacks occurring during the training phase mainly try to gain knowledge, manipulate or even falsify the FL model as a whole [41]. The adversary aiming to jeopardize the training data will run a data poisoning attack [19] or use a model poisoning attack [25] to jeopardize the learning process. A vast variety of inference attacks can also be used on private updates or on a combined update of all devices.

2.3.2. Inference phase

These categories of attacks are also called evasion or exploratory attacks. The usual aim of these attacks is not to change the targeted model but to deceive them into making erroneous predictions or acquiring proof substantiating model attributes [42]. For such attack models, the efficacy of the attacks is directly dependent upon the level of model information that the adversary can gather. In contrast to inference attack scenarios in centralized ML, the model broadcasting feature in FL makes it prone to evasion attacks at server end models and allows access to random malicious clients.

Depending upon the level of access an attacker has to a particular system, inference phase attacks can be classified into two categories, white-box attacks and black-box attacks [43]. White-box attacks have complete access to the FL model and network parameters whereas black-box attacks are confined to performing basic querying of the FL model and have no accessibility to the parameters. Inference attacks are not limited to individual systems, global models also come across such attacks similar to the conventional ML model whence deployment as a service is done [44]. Particularly, the broadcasting nature of the FL global model makes it more susceptible to white-box attacks and thus requires additional efforts to ensure protection from such evasion attacks. Even though evasion attacks look similar to backdoor attacks, the reality is they are entirely different entities. Evasion attacks make use of the decision boundaries learned by the genuine model to build the adversarial sample which will be misclassified by the model whereas, with backdoor attacks, the decision boundaries are shifted deliberately due to unsuccessful training procedures which eventually leads to the misclassification of samples [45].

Next, we present an overview of different attacks discussed in Fig. 2. Based on the attacking methodology, FL attacks are divided into various categories. In our study, we will be discussing three different types, namely Insider–Outsider attacks [2.1], Semi-honest and Honest attacks [2.2] and Training Phase-Inference Phase attacks [2.3] . Later on, we move on to a detailed study of Poison attacks [3] and Inference attacks [4] and their subcategories. Inference and Poisoning attacks are two broad classifications done based upon the modus-operandi of the attacks and multiple subcategories come under them which can be further classified into one of the three types mentioned at the beginning. Generally poisoning attacks are classified into Data Poisoning attacks [3.1] and Model poisoning [3.2] . The third category of poisoning attack is also described referred to as Backdoor attacks [3.3] which are usually the result of successful poisoning attacks. We also present the various subcategories of all three attacks in detail. In Inference attacks, we deal with 5 major types of attack types. They are Membership Inference attacks [4.1], Reconstruction attacks [4.2], Model inversion attacks [4.3], Property inference attacks [4.4], and Attribute inference attacks [4.5]. Each type of attack is explained in detail along with its subcategories wherever applicable.

3. Poisoning attack

In the FL scenario, poisoning is a type of attack that has the highest possibility to be implemented successfully as every client device has complete control over their training data maintaining a sense of secrecy with the server, implying that data authenticity is void [46]. Poisoning attacks are one of the most powerful forms of active attacks where the attacker is an “insider” system and is capable of tampering with private data or local models to directly influence the performance of the global model. In FL systems, the poisoning attack begins with the attacker downloading the global model parameters for updating the local model. Generally, the adversary makes use of poisoned data to train a local model which helps in gaining control over the global model and updates it to the server [47]. Once model averaging is done, the presence of synthetic parameters in the global model helps it to achieve high results for poisoned data without affecting the main tasks. While implementing poisoning attacks, the adversary usually tries to compromise the basic integrity of the system adversary by breaking into the learning process leading to system crashes, or by creating a backdoor which grants system control to the attacker.

To increase the robustness of models, FL model updates are gathered from a large group of clients increasing the chances of the system getting poisoned from one or the other client and thus increasing the severity of the attack [48]. Depending upon their targeted entity, poisoning attacks make use of wide array of techniques to accomplish their tasks. Based on the applied methodologies, attacks can be of various types such as data flipping attacks, back door attacks, and so on. A brief discussion of the different types of poisoning attacks will be presented in the coming section. Fig. 3 provides an illustration of poisoning attacks in general.

3.1. Data poisoning

In the context of centralized ML, the idea of data poisoning was first proposed by Biggio et al. in [49]. In the proposed model, the adversaries aimed at the vulnerabilities of the support vector machine technique and tries to merge malicious data points with clean data during the training phase in an attempt to maximize misclassification. Although the FL setting facilitates clients to perform active contributions of training data and model parameters, it also opens up the chances for malicious users to poison the global shared model via training process manipulation. Data poisoning attacks occur when the adversary is possessing a part of the training data which is used for the learning process. By infecting the data with malicious points, the learning system is made contaminated thus making it possible for the attacker to make changes to the system [50]. Traditional data poisoning attacks just alternate the labels of samples used for training present in the training class.

A major issue that occurs with the FL setting is the sheer volume of updates during the training process and thus the inability to filter out adversarial and benign updates as data remains remote in both cases. The limited accessibility of the FL server to gradient updates only complicates the problem even more. This opens up the possibility of training nodes easily getting compromised and manipulated. In such instances, the compromised nodes can also actively participate in model training contributing their part to the global model ultimately achieving the adversarial motto. This major pitfall of FL models is considered as one of the prime motivations for data poisoning adversarial attacks. Data poisoning is the most forthright way to deliver a poisoning attack in a learning model. The motto of such attacks is, ‘poison the data, poison the model’. Basically, an adversary puts erroneously labeled or

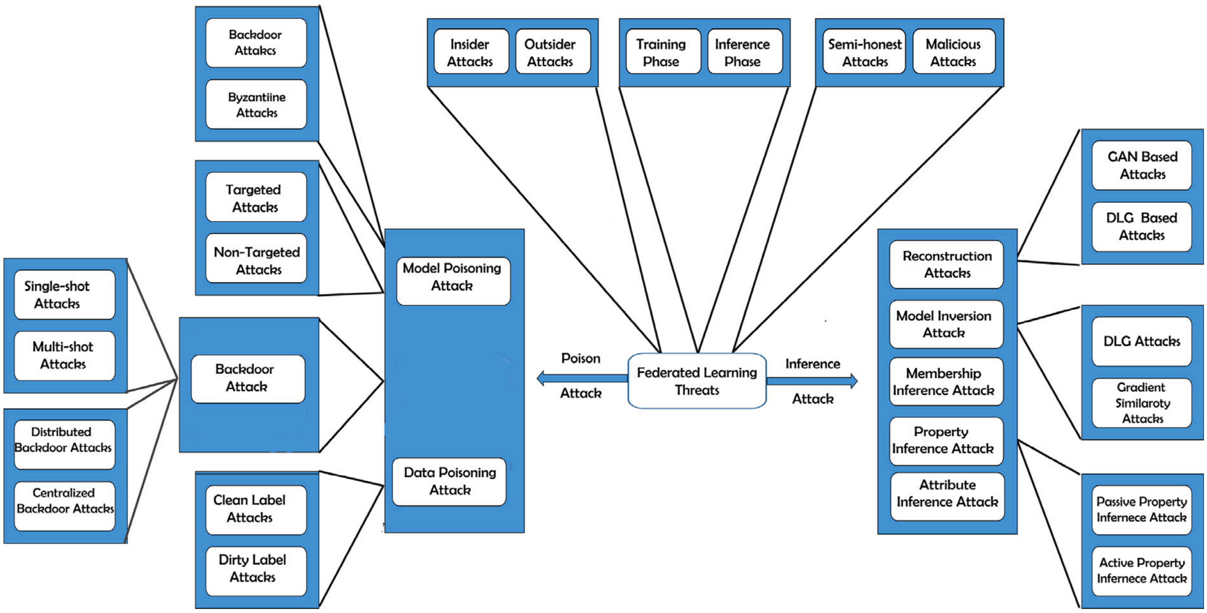


Fig. 2. Different types of attacks faced in FL.

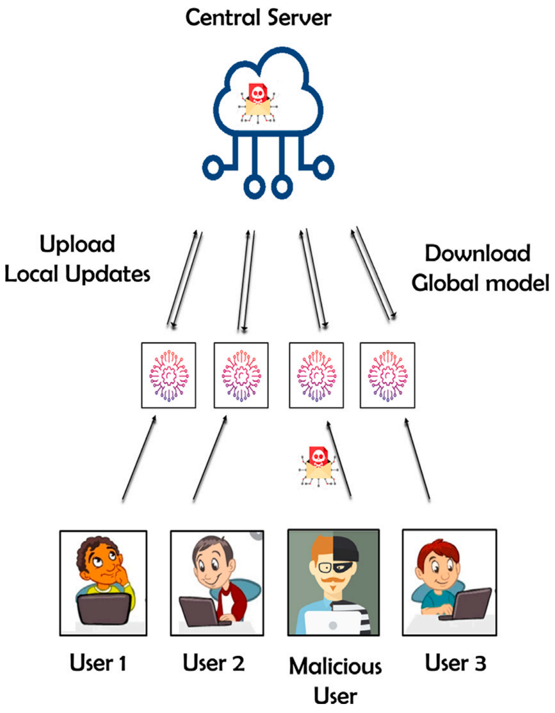


Fig. 3. Poisoning attacks against FL.

false data into the training data set. Data poisoning attacks can be classified into two broad categories, i.e. clean label [51] and dirty label attacks [52].

Clean label attacks assume that the attacker does not have the ability to alter the labels of training data. An authentication process exists by which data are ascertained as being part of the precise class and for poisoned data samples, it becomes a hard barrier to bypass [53]. Whereas in dirty label poisoning, the attacker can add multiple data samples as it wishes aiming to wrongly classify the labels in the training data set [54].

In clean label attacks, poisoned images maintain their malicious characteristics even after correct labeling. These attacks open the possibility for a distinctive threat model where the adversary can poison the data set by just adding malicious data on the source and these poisoned data when acquired by a system will automatically poison the system data set as well [55]. A novel data poisoning approach termed as “Meta poison” is discussed by Huang et al. in [56]. As stated by the authors, the proposed model identifies some disturbances in the data set that are capable of determining the model behavior on particular targets. It is said that the proposed method achieved better performance on fine-tuned models when compared with previous methods. The poisoning factor is real in such an approach and is proven on multiple ML models.

Clean label backdoor poisoning is another genuine attack model presented by Zhao et al. in [57]. The attacker will be able to alter discrete features or small portions of the training data set to attach backdoors into the model making it work as per the attacker’s intent. The backdoor does not affect the working of clean inputs in any way and it performs as normal thus making such an attack hard to detect.

There are two major reasons for the wide usage of dirty-label data poisoning techniques. Firstly, in an FL environment, it is assumed that data is always kept and only models shared. This makes the attacker least concerned about the susceptibility issues related to data credibility. The second thing is the absence of access to global parameter vectors in the FL environment which is not presumed in clean-labeled data poisoning. Label flipping attacks discussed in article [37] is a commonly used attack belonging to the dirty label attack type. In this model, the attacker works by flipping genuine labels on a particular class to another while keeping data featured unaltered. A successful instance of such an attack creates a model which is not capable of performing efficient classification tasks. Chen et al. in [58] has proposed a backdoor attack using dirty label data poisoning and they have proved the high efficiency of the model as well. Also, the authors have experimentally proved that by adding a minute quantity of poisoned data to the training data set, the model can be made to perform classifications tasks according to the attacker’s purpose with high accuracy.

Another novel data poisoning approach based upon generative adversarial networks (GAN) has been proposed by Zhang et al. in [59]. The adversary behaves like an honest participant and uses a GAN to mimic data samples of other participants. This generated sample helps the adversary to poison updates and compromise the global model by sending these poisoned updates in a scaled manner to the server. The author states that this model performs efficiently when compared to other data poisoning methods and the global model performs efficiently for both poisonings as well as other tasks. In conventional FL attacking methodology, the presumption is that the attacker has already acquired part of other users' training data. The authors have also presented an improvised version of the work in [60] where repeatedly updated global model parameters are used to create pseudo data which have a similar distribution pattern of training data as other devices. Making use of the newly proposed methodology, another novel attacking model is presented termed as generative poisoning attack particularly aiming at FL environments. The authors state that the experimental result guarantees effective compromising of the global model using this approach.

Shejwalkar et al. in [61] presents one of the first data poisoning attack models that does systematic poisoning of data. The proposed model is built on the standard label flipping attack [37] which was primarily devised for traditional ML. In FL data poisoning attacks, it was previously observed that an increased amount of label flipping was accompanied by increased loss and norms of resultant update. Thus the proposed model makes use of this observation to produce poisoned updates capable of reducing the accuracy of the global model. An issue with such attacks is that the excessive amount of label flipping on data, resulting in updates generated deviating far from the genuine updates. This may lead to the generated update getting classified as malicious updates by the aggregation technique. To surpass this issue, the authors propose adjusting the amount of label flipping trying to maintain it at an optimal position where it can evade getting detected by the aggregation technique.

The high impracticality and ineffectiveness of normal data poisoning attacks had led the authors to devise such an alternative approach. As stated by Fang et al. [62], even in a simple FL setting with highly intensive data poisoning, the model performs barely as much as a label flipping attack does. The authors also make use of label flipping methodology stating that the larger number of flipped data leads to higher deviation. They experimentally prove the claims using standard datasets combined with various aggregation techniques as well.

Another data poisoning attack in the FL setting has been proposed by Zhang et al. in [63] with an "honest-but-curious" adversary [23] who only has access to limited portions of data. The adversary makes use of intermediate output along with the poisoned data to train its decoder to fetch the victim's private data. The decoder is designed in such a way that the reconstruction loss is to be minimalized meaning that with adequate poisoned data and corresponding output, the decoder is able to reconstruct the input for whatever intermediate values provided. Using that, the sensitive data of the victims can be accessed. Based upon the mode of decoder training, the authors propose two types of attacks methodologies, static and dynamic. Static mode simply means that the total sum of poisoned data collection and decoder training is done on an iterative basis. When it comes to poison data being distributed, the adversary requires to collect them over multiple approaches. Thus the decoder training is also done on multiple iterations and such a technique is referred to as the adaptive method. They also propose defensive measures capable of tackling their proposed attack model as well as some of the other common attacks. The authors do several experiments and make use of various matrices to prove the validity of their proposed models.

Summarizing, data poisoning is a mode of attack whose scope is limited to FL insider devices only. Thus, the credibility of a model is dependent upon the degree of engagement of FL system participants and the sheer volume of poisoning of training data. The effectiveness of data poisoning attacks is proportional to the number of participants in the system thus implying that system with fewer participants is least affected by such attacks [24]. When it comes to experimental results, the observed trend shows that a proportional increase is never observed in terms of results with an increasing volume of poisoned samples. Also, the same trend is observed in the case of targeted poisoning in the global model as well. One reason for such an issue is that the malicious agent's update is comparatively small and it needs to be boosted to yield a good result for model poisoning. Additionally, in instances where the update of data poisoning is boosted, it does not occur in an isolated manner and the global model gets boosted as well thus reinstating the previous model [26]. Thus an inference can be made that in the FL environment, data poisoning attacks are less efficient when compared to model poisoning attacks [62].

3.2. Model poisoning

The Model poisoning attack works by affecting the model performance without getting noted. Since FL works in a distributed manner, the attacker aims for individual models and poisons them. These malicious participants will be directly able to alter the model parameters and inject antagonistic attributes to a model update launching a model poisoning attack thus corrupting the global model [64]. The prime motivation behind model poisoning attacks is the fact that all participants in an FL system have direct accessibility to the global model implying that it can manipulate the model in different ways. Primarily by directly altering the weights of the global model via model update and secondly by injecting malicious neurons to the system through various training approaches thus trying to alter the local model update to make the malicious update go unnoticed even under the surveillance of the server system. Even though several other methodologies are used, these two are one of the most widely and efficiently implemented strategies.

The major difference between data and model poisoning methodologies is that model poisoning attacks do not fudge training data. As discussed previously, in FL environments, multiple local models exist and the aim of model poisoning is to influence these models or attack the global model. Such poisoning attacks are pretty forthright. The adversaries try to replace the functioning model using a poisoned one. These poisoned models survive inside the system and open a backdoor for the attacker either for modifying the model or even allowing the model's entire replacement with a poisoned one. In FL, all participants are given direct authority over the joint global model. A malicious user can exploit this and launch potent attacks thus making it much more hazardous than data poisoning attacks. Bagdasaryan et al. [19] state that the aim of model poisoning is to poison locally existing model updates prior to being sent to the server or inserting a hidden backdoor into the global shared model. It is stated that the effectiveness offered by model poisoning attack shows an increasing trend with the increasing size of the FL environment i.e. when the system comprises multiple clients [22]. The ability of malicious participants to alter the updated model prior to being sent to the central device for aggregation makes the global model susceptible to easy poisoning.

Model poisoning can be generally classified into targeted and non targeted attacks. In targeted model poisoning, the attacker's aim is to make the FL model do faulty classifications on a set of data with excessive confidence [24]. The peculiarity of such an attacks methodology is that the inputs are not tampered with at test time as in the case of other adversarial attack models. Adversarial manipulations of the training process are done in these attack types thus helping in achieving misclassification. A further classification can be brought into model poisoning attacks that is, byzantine attacks and backdoor attacks [23]. In byzantine attacks usually, the attacker tries to ruin the convergence of smaller models as well as bring down the performance of the global model [65]. As with backdoor attacks, the adversaries focuses to lodge a backdoor trigger into the global model deceiving it into making a constant prediction of an adversarial class on a smaller task still maintaining better performance on main tasks [53]. Moreover, in most cases of model poisoning attacks implemented via backdoors, the usual practice is to make use of data poisoning to access the parameter updates used for poisoning [20]. In the FL setting, model poisoning and data poisoning can never be stated as exclusive of one another because data poisoning attacks in due course works by changing a smaller batch of updates sent to the global model at some particular iteration [37].

Researchers have shown that model poisoning attacks provide better effectiveness when compared to data poisoning attacks in the FL setting. Bhagoji et al. in [53] have experimentally proved the statement by examining a targeted model poisoning attack in an FL environment with a single malicious device where the aim of the attacker is to make the model perform faulty classification. To operate secretly without getting detected, the authors suppose using the alternating minimization strategy for covering up the training loss along with the attacker's objectives and using parametric estimations for updates provided by genuine participants. Such an adversarial model poisoning can lead to an undiscovered poisoning attack for an FL model.

The authors in [19] have proposed a backdoor attack model making use of model replacement techniques. Prior research had revealed that with the convergence of the global model, the deviations in the sum of updates provided by genuine client devices will approximate to zero. In an attempt to ensure that the backdoor survives the convergence stage, the authors propose that the attackers can scale up the weights of the poisoning model using a specified value and replace it with the shared global model. This process may execute in any given round but the ideal result is obtained close to model convergence. Model replacement makes sure that the attacker's part does survive averaging and becomes part of the global model. This attack is a type of single-shot attack implying that the global model will show high precision on backdoor tasks very shortly once poisoning is done. One issue accompanied with such a methodology is that whence the learning rate gets boosted, it will acutely bring change to the weight distribution which can lead to detection from centralized server [27].

Bhagoji et al. [64] explains in detail model poisoning attacks in the context of FL performed by a single non collaborative malicious agent where the major aim of the adversary is to perform faulty classification of data over a specified input data. A number of different proposals are put forward to achieve the same [66]. Boosting the infected model update to nullify the impact of other device's updates is the primary proposal [67]. When it comes to improving stealth, another proposal made is to use an alternating minimization strategy [68]. The strategy iteratively shifts between minimizing training loss and enhancing updates to reach attackers' objectives. This helps the system to reach high performance in malicious as well as genuine updates. Additionally, estimation done on other agents' updates has shown an improving trend in the success rate of attacks in general. The authors try to give a more solid explanation of the techniques mentioned in [64] via their work in article [53].

The authors in [53] propose a model which alternatively minimizes benign and malicious training in an attempt to enhance the stealth of the attack. Similar to the attack model discussed in [19], the methodologies presented in [53] are also not much effective in the case of single-shot attacks since the interference of the main server may result in the ineffective implementation of adversarial tasks. Alternative minimization here simply means that the model work by continuously optimizing the main task and adversarial task prior to the model convergence. The model works assuming the attackers have control over a very small number of clients (generally one) and have no visibility of other private models. The malicious user optimizes model updates at every round with respect to the set objectives and performs explicit boosting to negate the convergence effect of genuine participants. Even though the model ensures stealth when the poisoning sample is low, it is at risk when the compromised client devices make use of larger data sets for poisoning.

Another methodology of implementing model poisoning in FL environments via a novel optimization based technique is proposed in [69]. The prime focus of this model is in regard to effectiveness, persistence, and stealth of attacks which are tremendous issues for model poisoning attacks in FL. In earlier works, it was shown that only a small division of neurons undergo changes at the training stage whereas most of the other neurons are close to zero. Those neurons are known as redundant space and our model proposal aims at embedding poisoning data into that redundant space. Such embedding is done under the optimizer's guidance and the aim is to regulate stealth and persistence. Such neurons are related to adversarial tasks only and have no influence on main tasks. Performance degradation is not an issue with the model thus increasing stealth. Experimental results show that the method is capable of bypassing existing defensive measures with ease and offers a high attack success rate [69]. Hossain et al. [70] have also proposed a novel model poisoning which the authors refer to as an unprecedented DP-exploited stealthy model poisoning attack. As the name suggests, the authors make use of differential privacy principles to develop and launch the attacks. They claim the attack to be highly stealthy and potent rendering the final model useless. The basic approach is to manipulate the noise applied during differential privacy in an uncontrolled manner thus producing a model having high amounts of noise thus becoming the least accurate.

Fang et al. [62] presents an elite analysis of model poisoning attacks in FL scenarios through their work. The authors attempt to present a novel model poisoning attack methodology capable of compromising the integrity of the learning procedures during the training phase itself. It is an offline mode of attack methodology meaning that the attacker poisons the compromised client just before the commencement of the FL training procedures. The proposed attack model is built on the basic label flipping techniques and it completely relies on the previous observations that with the increase in the amount of label flipping, a considerable increase in the loss, as well as the norm, can be observed. This helps to deliver poisoning attacks capable of reducing the global model's accuracy to a higher degree. The attacks operate with a prior assumption that the adversary had already manipulated a few client devices and thus, it uses the local model parameters to tamper with the model comprising the global model resulting in large values of test error rates. Since the nature of the attacks is based upon the local model, the authors refer to such an attack as local model poisoning attacks. They further evaluate the possibilities of the attack and suggest extensive countermeasure as well. The author states that in general, data poisoning attacks prove to be less efficient due to their time-consuming nature even in simple FL environments. Additionally, they also asserts that basic label flipping attacks have the same potency as compared to the data poisoning attack methodologies.

Another similar model poisoning attack is discussed in [71]. Baruch et al. propose a novel non-omniscient attack on distributed learning scenarios. The attack functions by adding a trivial quantity of noise to every single dimension of the average of the benignant gradients. Such embedded noise is peculiarly selected in the way that even though they will be large and capable enough to negatively impact the global model, they will also be small enough to be detected by secure aggregation algorithms. From the gradients that the adversary possesses, it calculates the average and the standard deviation. Additionally, it also computes a coefficient value derived from the total number of clients, malicious and genuine users combined. Using these three values, the adversary is able to compute the poisonous update by adding the average value to the product of the coefficient term and the standard deviation. The major peculiarity of such a method is that the adversarial attack performed is independent of "aggregation principle knowledge" meaning that it would function efficiently without having any idea of the aggregation technique used. Thus by adding a minute amount of noise, such attacks can evade detection from all sorts of aggregation techniques. The authors state that their attack methodology is capable of not just preventing model convergence but also capable of effective backdoor creations. They also claim that 20% of malicious users are sufficient to degrade a model performance by 50%. Through extensive experimentation on standard datasets, the claims are clearly established and justified.

Even though referred to as a state-of-the-art technique, a major disadvantage of such an approach is its ineffectiveness due to the minute amount of noise it uses for poisoning. As an alternative for that, an AGR-agnostic attack approach was presented by Shejwalkar and Houmansadr [72] which was stated to overcome all the disadvantages posed by the previous models discussed in [62,71]. The model claims to work even with no knowledge of the hidden aggregation techniques.

Authors in [72] propose an optimized model poisoning attack methodology. In the proposed model, the attacker launches the attack by first trying to create a reference aggregate with limited genuine updates they possess and then follows it with the generation of a “malicious perturbation”. The final poisoned model updates are computed via perturbation of the malicious update to the highest extent possible towards the opposite direction of the idealistic model. While doing all the procedures, the system keeps a keen eye on evading getting detected by aggregation algorithms. From the proposed model, the authors also generate a universal optimization model which can be mounted on model poisoning attacks for higher efficiency with applicability for a wide variety of settings. Along with the model, they propose an algorithm for calculating an effective coefficient value for model poisoning attacks built on the proposed optimization method. As an alternative for the absence of “aggregation technique” knowledge, the authors propose making use of key intuition techniques used in robust aggregation methods. The idea behind the key intuition technique is that in case an update moves away, isolating itself from a group of genuine updates, the system marks it as malicious content. Thus, our agnostic attack can restrain itself from doing a wide search and limit itself to a performing search over a very small area which the authors refer to as “a ball of small radius around the clique of the benign updates” [72]. The authors have proved the experimental validity of all their claims via comparative experimentation with state-of-the-art models on standard datasets and also propose a countermeasure for such powerful poisoning attacks.

Shejwalkar et al. [61] propose another variance of model poisoning attacks inspired from [72]. The authors refer to the attack as projected gradient ascent or PGA attack. The PGA attack draws its base from the common optimization problems faced in poisoning attacks and the aim is to generate a poison attack with a resultant model having high cross-entropy loss. Additionally, the level of malicious activity in the poisoned update is regulated so that it can surpass the corresponding aggregation technique. For ensuring the survival of the malicious update post aggregation, the model makes use of an approach they refer to as “ball of radius”. In this method, the update is plotted in a circular pattern around the origin with the circle having a radius value equal to the average of norms of the genuine updates. It is assumed that the adversary had knowledge of the global model parameters and is capable of directly altering the updates of the malicious participants. Thus, the proposed attack makes use of a stochastic gradient ascent algorithm (SGA) and later fine-tunes the global model in an attempt to increase the loss of the genuine data to generate a poisonous model as the final result. In PDA, two different methodologies are proposed to implement the SGA algorithm. The first one is to make use of the opposite direction approach and the second one is to make use of a malicious gradient direction approach which is similar to the traditional label flipping attack [62]. The authors critically evaluate the existing attacks models along with their proposed models and prove the proposed model’s efficacy and robustness in various FL settings.

Chang et al. [73] proposes an attack specifically targeting aggregation methods making use of weighted principles. The weighted aggregation technique works by assigning weights with respect to the distance of the data point from the aggregate implying that the greater the distance, the higher the weight change. The authors perform their discussion based upon two such weighted techniques, i.e. multiple weight update (MWU) with averaging [74] and MWU with optimization [75]. The MWU with averaging is an algorithmic approach through which the user aims to perform an interactive decision-making task and generate an allied payoff as well. The ultimate aim in such cases will be to attain a global payoff that corresponds with the payoff of that particular decision which also peaks global payoff with the addition of discernment as well. Similarly, the MWU with an optimization approach turns out to be an effective solution for a large class of continuous optimization problems.

At any random epoch, the above-mentioned aggregation technique starts with dispatching equal weight values to all clients and the updation of weights will be performed on grounds of the degree of change in the client’s update compared to the global average. This very same fact is the point of attack used by the adversary. The OFOM attack primarily generated a couple of malicious updates, the first one positioned away from the genuine mean value and obtained via summing of an arbitrary large vector with the mean of the genuine updates. The second update will be positioned directly close to the empirical mean of the genuine updates and the previously generated malicious update. By using such a methodology in MWU aggregation, once the very first epoch comes to an end, the adversary will be able to assign weights (approximately 1) to the clients attached with the second malicious update.

Similarly, in the case of MWUAvg and MWUOpt, all the genuine clients are allotted with minute weight values which directly leads to the loss of accuracy during aggregation. In such a methodology, the adversary needs to possess just two malicious clients to be completely able to poison and destroy the integrity of the model. The authors experimentally prove the efficiency of OFOM attacks in reducing the accuracy of aggregation methods to a random value on multiple standard datasets. They also propose a novel defensive strategy referred to as “Cronus” which is proved to be effective against the proposed as well as some of the other state-of-the-art poisoning techniques.

3.3. Back-door attacks

Even though backdoor attacks cannot be described as an independent attack methodology, they do come frequently in the context of poisoning attacks. A backdoor can be referred to as an input that opens a hidden entry into a system that the programmer is not conscious of, but gives access to the attacker enabling him to make changes to the system according to their wish [19]. Usually, backdoor formations are the result of successful poisoning attacks. Attackers design a backdoor attack in such a way that it is capable of misleading the trained model into predicting a “particular label” on any provided input that has an embedded field added by the adversary and this embedded value is referred to as the trigger. When compared to attacks that prevent the model convergence in terms of accuracy such as Byzantine attacks [76], backdoor attacks in the context of FL manipulate local models and embed them with the primary task as well as the adversarial task via the backdoor. By doing so, the aim is to make the global model function accurately on legitimate data as well as backdoor attached data. The major motivation behind the backdoor implementation is mainly related to the stealth of the model compared to other approaches. It also enhances the adversarial control in the form of several backdoor subtasks which further increases the potency of the attack [19].

Based on the methodology applied for backdoor creation, such attacks are classified into two categories, centralized backdoor attacks and distributed backdoor attacks (DBA). Generally, the term backdoor attacks are used to refer to centralized backdoor attacks. The major difference between both the methodologies comes down to the way they implement their “trigger”. When it comes to a centralized attack, the adversary makes use of a global trigger while in DBA, the triggers are usually local ones and they will be part of the global trigger. When working with global triggers in backdoor attacks, the same trigger needs to be embedded into all adversarial clients. Depending upon the nature of the trigger initiating the attack, backdoor attacks can be further classified into semantic as well as artificial [61]. Instances, where the trigger is identified to be existing inside the sample by default, are referred to as semantic backdoor attacks [77] and when triggers are supposed to be manually added into the system, it is referred to as artificial backdoor attacks. In artificial backdoor attacks, it is to be noted that the triggers are usually added during test time only [78].

Fung et al. in [37] presents an overview of centralized backdoor poisoning attacks. The authors also propose a novel defensive strategy referred to as "FoolsGold" which is proved to be capable of tackling such centralized attacks. Similarly, Xi et al. in their work [78] presents a detailed study on DBA in an FL setup and performs a novel threat assessment framework as well. DBA breaks down universal trigger patterns and attaches them to individual devices' training data. Through extensive experiments, the authors state that the proposed DBA is highly resolute and effectual accompanied by a greater success rate and flexibility to execute in tough situations. The results open the possibility for DBA to be the attack to be vigilant about in the near future.

Currently, another variant of backdoor attack termed as "Dynamic Backdoor Attack" has been proposed by Huang in his recent work [79]. As stated by the author, in dynamic backdoor attacks, the prime aim of the attacker is to bring down the performance of the model on specific tasks still maintaining efficiency in main tasks. Even though in definition, it is similar to normal backdoor attacks, the dynamic nature is in the way the poison persists in the data. In static backdoors, the idea is that injected poisoning remains static and unchanged whereas the reality is that the dynamic nature of FL gives an additional advantage to the adversary when it comes to updation of the poisoned data to maintain the effect. When it comes to the defensive side as well, the traditional defensive approach considers these changes as newly initiated poisoning meaning that the system requires extensive studies to form a defensive strategy against it giving an advantage to the adversary in the form of added time. In his work, the author has also proposed a novel defensive strategy termed as "symbiosis network" against such attacks and experimentally proven its efficiency as well.

Backdoor attacks operate in two ways with respect to data rounds, that is single-shot and multi-shot attacks. In the FL context, single-shot attacks are referred to as A-S attacks and multi-shot attacks as A-M attacks. For multi-shot attacks, attackers are identified through several rounds and infected updates need accumulation for attacks to succeed. Else, genuine updates will be weakening the backdoor gradually. A single shot attack simply means that only one chance is needed for the attacker to attach a backdoor trigger successfully. Here, scaling is used to dominate the genuine updates and ensures the survival of the backdoor through the aggregating procedures. In multi-shot attacks, one of the most common examples is DBA and it starts poisoning from square one resulting in low accuracy and difficulty in convergence. For single-shot attacks, Bagdasaryan et al. [19] states that attacks implemented late are more effective. At the time of convergence of the global model, the updates made by genuine clients will be containing less common patterns and more private which probably gets omitted during aggregation, leaving less impact on backdoors thus adding weight to the author's statement. Model replacement attack is one the most commonly used form of single-shot backdoor attack [80].

Summarizing, some of the major problems that make poisoning attacks harder to detect in the context of FL are the few ones discussed below. The participants are given a certain degree of privacy making the training process not visible to the server, thus making authentication of an update made by a local device not at all possible [81]. Simple FL uses IID as well as Non-IID data for training. As per the training properties of Non-IID data, the local updates provided by participating devices are entirely dissimilar to one another. In parameter transmission, the secure aggregation protocol is used and it blocks the server from performing an audit of every single participant's update with the universal model.

4. Inference attacks

Inference attacks basically means having unauthorized access to data through which some inferences can be made about the system or process. Such attacks are possible when an attacker is able to derive valued data from irrelevant data about a database or a system without having direct access to it. Inference attacks pose a high threat to privacy in FL environments. In terms of severity of the attack, inference attacks have a high similarity to poisoning attacks since it is highly possible for the malicious participant as well as a malicious server to launch inference attacks [22]. In FL training, privacy leakage occurs at the time of gradient sharing. Inference attacks make use of this to breach the system. With model updates also, they tend to reveal more data than intended relating to an individual's training data thus giving an upper hand to adversarial participants. There are also chances that the adversaries are able to take snaps of FL model parameters and perform property inference by capitalizing on the differences observed between sequential snaps. This is equal to calculating the aggregated updates of all participants excluding the poisoned one [24]. The prime motto behind inference is the fact that FL model training often leaves back or opens up more data than intended due to various reasons at multiple aspects such as at gradient level, membership data level, and so on. This always makes the system prone to powerful inference attacks when such triggers can be efficiently interpreted. Fig. 4 shows the illustration of inference attacks in general.

4.1. Membership inference attacks

The ultimate aim of a membership inference attack is to deduce whether a member is part of the training data set or not [82]. In membership inference attacks, the adversary tries to capitalize on privacy leakage related to private data records used in FL training. The attacker has the ability to ascertain whether a record belongs to the training data set or not. The attacker builds a pseudo model referred to as a shadow model to mimic the original data set. In case the output of the shadow model comes with high confidence values, it can be inferred that the data set possesses a high similarity to the original one.

Melis et al. in their work [83] discuss about collaborative model updation and also demonstrates the unidentified information leakage of fellow participants' training data in such instances. They further present the extent of exploitation caused through inference attacks due to such information leakage. Such inference attacks enable the malicious user to not only infer membership details but also attributes that are specific to individual training data sets and are totally autonomous sharing no commons that the global model aims to catch hold off. In the case of DL models, the system by default recognizes multiple sets of data that are not related to their designated tasks. As a result, these unintended data are automatically leaked during collaborative model updates. Active attacks usually make the joined model infer multiple sets of other participants that the adversary wishes to obtain without any sort of performance issue arising for the global model on main tasks. Data leakage in FL environments can give rise to such powerful inference attacks. The authors also state that existing defensive measures are ineffective and this gives rise to the need of adopting more efficient measures for privacy protection against membership inference attacks.

Nasr et al. [44] investigates a novel white-box membership inferencing attacks in centralized and federated DL environments. The purpose of the attack is to measure data membership leakage in training instances. In their work, the authors prefer not to do a basic extension of the black-box attack to the white-box due to the ineffectiveness of results yielded in such cases. Rather than that, the possible privacy breach of the stochastic gradient descent (SGD) algorithm [84] is utilized for the proposed approach. The study shows that each data points regulate multiple model parameters through the SGD algorithm in an attempt to reduce its contribution to learning loss. It is shown

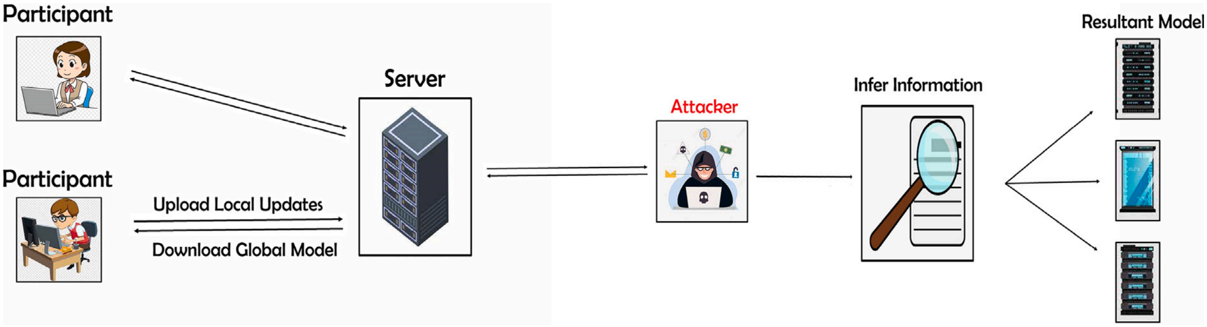


Fig. 4. Inference attacks against FL.

that even well-generalized models are prone to White box membership inferencing attacks.

Furthermore, in an FL scenario, the authors of [85] have shown that a participant device or even a parameter server that can peek into a member's data is capable of performing highly potent membership inference attacks on fellow devices. The claim was established with valid experimental results obtained by using multiple standard data sets. In FL model training, the system performs repetitive parametric updation of the models over multiple epochs on the same dataset. This acts as a catalyst for increasing the accuracy of inference attacks in FL models. The attacker's inputs such as parametric updates are capable of influencing the victim's parameters. Thus, the attackers can capitalize on the leakage in SGD even more to fetch a higher quantity of training data from other devices. It is also worrying that the malicious users in an FL setting are capable of launching active membership attacks on other genuine participants irrespective of the high prediction accuracies shown by the global model.

4.2. Reconstruction attacks

Privacy leakage in FL is not limited to just leakage of data but model updates from gradient client updates or SVM or KNN are also leaked where values are stored. The privacy information is obtained using these methods through a reconstruction attack. The motto is to get the training data itself or even vectors for the data used during ML model training [86]. In the context of FL, gradient update obtained from the client is enough to leak details about the client. Such leakages make the trained models highly susceptible to reconstruction attacks. Such attacks are mainly divided into two categories, GAN based attacks and deep leakage from gradient (DLG) attacks [87]. The attacks based on GAN make use of GANs to regenerate images that are similar to the ones used in training whereas DLG aspires on disclosing the training data completely from gradients. Both the attacks make use of gradient for the purpose of reconstruction thus referred to as gradient based reconstruction Attacks. Unlike other attacks, gradient-based attacks need dedicated defensive measures to be employed for system security such as multi-party computation (MPC) [88] , homomorphic encryption (HE) [89,90] and so on.

4.2.1. GAN based attacks

In FL scenarios, a major type of data reconstruction attack is delivered with the aid of GAN-based techniques [91,92]. In [91] the author discusses information leakage for a collaborative DL environment. The motivation behind the development of collaborative DL is primarily to reduce privacy issues related to sharing of sensitive data. Models are trained locally and only subsets of parameters are shared to maintain privacy.

The authors [91] state and prove that federated DL is a structurally broken system and offers the least protection for genuine participants. The proposed attack makes use of real-time working of the learning process which allows adversaries to develop a GAN that is capable of generating a model of sample similar to the ones that were kept private. The crude idea is to reconstruct data for a particular class using a GAN discriminator which differentiates the source of the data i.e. whether it is from the desired class or synthetically generated. Once such real data is created, it is given to the model under another label. This drives the model into trying to find the differences between the two datasets which in turn enhances the performance of the GAN. Existing defensive measures including differential privacy are ineffective against the proposed model [91]. It is hence stated that collaborative DL is less prudent as an alternative for centralized approaches. They also propose using MPC or HE as an increased privacy protection measure but one of the main motives behind the introduction of federated models was to avoid the usage of such complex measures [93].

Hitaj et al. in [92] also proposes a user-level GAN-based reconstruction method for FL settings named "mGAN-AI". The proposed model is said to be capable of not only reconstructing the real training data but also aiming for a specific device and breaking the user-level privacy for reconstructing private data from that particular device. The proposed method is experimentally proven to be able to deliver stronger privacy breaches compared to general reconstruction. The model framework makes use of a GAN incorporated with a multitask discriminator, which is capable of simultaneously discriminating category, reality, and client identity of input data. The discrimination is done on the client's identity that facilitates the generator to retrieve user-specified individual data. The model does not hinder the main training process in any way and remains hidden. The authors validate their claims through extensive experimentation and prove that the model outperforms currently used algorithms.

Another GAN-based approach is proposed by Sun et al. [94] where the adversary aims to reconstruct the victim's local data using model parameters. The generator and the discriminator networks of the GANs are simultaneously used for launching the proposed attacks. The idea is to reconstruct extremely similar images and data distributions between the two-component networks. The generator generates the bogus data making use of the latest global model parameters and the role of the discriminator is to find the differences between the generated data and the genuine data. These procedures take place iteratively helping the system reduce the prediction loss in turn generating a highly efficient model. The loss of the generator is calculated via the prediction results of the bogus data and the real data and the loss of discriminator through prediction result and ground truth.

4.2.2. DLG attacks

The principal theme of DLG is to create bogus data and labels through match-making using the dummy gradient with the shared gradients. This methodology has been widely used to perform attacks based on data leakage in FL environments.

Zhu et al. [95] proposed an approach that discusses the possibility of obtaining private training data from publicly shared gradients. By using the deep leakage from gradient (DLG) algorithm, the authors propose recreating dummy data and their respective labels with the aid of shared gradients. They begin by initializing the synthetic data and labels. Near enough gradients to the real ones are calculated on the currently shared model using a distributed methodology. By lessening the disparity between the fake and the real gradient, the authors propose the iterative updation of virtual data and labels consistently. Even though DLG performs well, there could be multiple factors affecting the quality of regenerated data in FL. DLG faces difficulty in converging and discovering real labels in a consistent fashion.

An enhanced model of [95] is proposed by Zhou et al. [96]. The authors state that in the previous models, the issue of label data leakage was real during gradient sharing and the authors propose an approach to extract data accurately from gradients. The model is named improved DLG as they state that the model possesses the capacity to extract ground truth labels with certainty that was absent in the previous model. The approach performs well for a differentiable model trained using a cross-entropy loss function over one-hot labeled data. The model was experimentally tested and accuracy was proved. The major issue with this model is that it can identify the ground-truth labels if and only if gradients are provided for every individual sample present in the training data.

An attack of a similar type is presented by Geiping et al. in [97] which also makes use of gradient information for data reconstruction. The authors state that by capitalizing on the magnitude-invariant loss function combined with optimization strategies, the possibility of reconstruction of data aided by their parametric gradient is highly possible. They also state that such a break of privacy is not isolated to a particular type of architecture implying that the vulnerability of a network to adversarial attacks is independent of factors like the depth of the network. Lim et al. [98] also proposes a data reconstruction attack built on deep gradient leakages similar to the one discussed in [96]. The reconstruction is performed by inverting the gradients of the model acquired via data leakage. The magnitude of the gradients directly determines the quality of the data during reconstruction as gradients of larger volumes are presumed to possess higher amounts of information. A major setback of the proposed approach was that the system was also prone to higher possibilities for misclassification as prediction loss increased.

4.3. Model inversion attacks

In the FL setting, the models are constantly updated via queries and model updates from multiple devices. Attackers can easily rebuild the model in case they get hold of these queries. This type of attack is discussed in “equation solving attacks”. In such attacks, the adversary can understand the data used in training as well as models [28].

Model inversion attacks basically do the reconstruction of training data from model parameters (in White-box setting). Model inversion attacks can be categorized into two, DLG attacks and gradient similarity (GS) attacks [99]. In DLG attacks, it is assumed that the server is malicious and its objective is to reconstruct the participant’s data by making use of their uploaded gradient. The reconstructed data is optimized by the server to minimize the Euclidean distance between raw gradients and the ones created during the backpropagation of the reconstructed data. In GS attacks, the idea will be similar to DLG attacks. Rather than Euclidean distance, GS attacks make use of cosine similarity between raw gradients and the synthetic gradients for optimization of reconstructed data through local updates.

He et al. in [100] proposes a novel attacking methodology for compromising inference data privacy in a collaborative DL setting. In such a setting, a single malicious participant is able to recover any random data given as input to the system even when it has no access to fellow participants’ data or computations. For experimental proof, a system with a pair of devices is considered. The target model is split into two out of which one is trusted and the other one is not which means that the data leakage can only occur through the non-trusted device. The same scenario is applicable in an edge-cloud based setting as well where edge devices can be trusted whereas the server may not be. Their proposed model is not just limited to a system with a couple of devices but can be scaled to multiple devices. In such multi-device instances, all adversarial participants belong to the untrusted category whereas the layers ranging from initial to intermediate are the trusted ones. The adversary is assumed to be obeying all the system inferencing protocols and neither compromises the inference process of the trusted devices nor gains knowledge about input or any intermediate values.

The authors have also discussed the applications of the model in three different settings in detail. In the white-box scenario, the adversary gains knowledge of the DNN layers controlled by the trusted devices which include model structure as well as parameters. These model parameters are used to recover the input data without needing to know training data or querying the model. The major difference when it comes to the black-box setting is that here, the attacker learns information about the model indirectly via querying. The adversary may process values or distribution of training data set but not compulsory for recovering sensitive data. A peculiar instance of the black-box model is discussed as the third category known as the “query free” setting where the adversary is incapable of performing model querying. Experimental proofs show that the model inversion attack, in general, is able to recover data with high accuracy even when the adversary has no knowledge of the trusted model or cannot even query the model i.e. the adversary does not need to meet many requirements for the reconstruction process.

Fredrikson et al. [101] have discussed a new class of model inversion attacks that capitalizes on the confidence values disclosed alongside predictions. The authors state that the attacks are applicable in a wide variety of settings. Basically, the algorithm simply finishes the target vector for each possible value of a sensitive attribute and calculates a weighted probability estimate which is said to be the exact value. Usually, the sensitive attributes are considered as genetic markers. A Gaussian error model is used to penalize the sensitive values that drive the prediction far away from the real label. The authors state that their proposed model produces a minimally prejudiced maximum aposteriori (MAP) estimate for sensitive sets whence information is available thus minimizing the misprediction rate of the attacker. Experimental results have established the author’s claims that their inversion algorithm is capable of inferencing sensitive responses from data with no false-positive results at all. The model also proved its mettle in image extraction even from facial recognition data set with high precision.

4.4. Property inference attacks

In the FL setting, the model’s basic aim is to do precise predictions for a peculiar task by learning properties and patterns in the data set. In such cases, there is a chance that the model learns properties that are not directly related to the main task or unnecessary in the given scenario. The property inference attacks make use of such learnings to exploit the model [102]. For sensitive training data, property inferencing attacks lead to severe privacy breaches.

By inferencing properties of a data set, the aim is to infer properties that are independent of a subclass of training inputs and not the whole class. Our main concentration will only be on such properties which are autonomous from the main set of the class [83]. In the FL setting, the relevance of data present determines the contribution of the individual model towards the global model in each iteration. In single-batch property inference, where the idea is to detect properties of data that are private to the batch and no other batches possess. The inference is made even when a property comes out in training data. It poses serious privacy breach issues. For property inference attacks to function properly, the attacker needs to possess auxiliary training data which is labeled precisely with the particular property that needs to be inferred. Property inference attacks can be of two types:

4.4.1. Passive property inference

In passive property inference attacks, the adversary basically notices the updates and does inferencing without making any changes to the local or collective training process. The adversary remains passive thus the name. It is presumed that the attacker possesses auxiliary data which comprises the property of interest and points that do not possess such properties. Such data need to be taken from the target participant's class which makes it unrelatable with other classes [103]. The methodology of implementation of such attacks is derived from the capability of the adversary in leveraging snapshots of the global model to bring forth combined updates based on data having particular properties and updates based on data that does not. Such a process creates label samples enabling the adversary in training a binary batch classifier that is capable of determining whether the updates belong to the data possessing the property or not.

4.4.2. Active property inference

Active adversaries are capable of performing more powerful attacks making use of multi-task learning. The attacker forwards the local model with the aid of an augmented property classifier connected to the last layer. This model will be trained meticulously aiming for the main task and recognizing batch properties. In collaborative training, the attacker uploads updates depending on the joint loss making the shared model learn individually for data possessing those properties and which does not. Thus the gradient formed can be separated too, capacitating the attacker to distinguish whence data possesses certain properties. Such adversaries are still considered "honest-but-curios". The attackers will stick to the system protocols and do not flood the system with faulty messages [23]. The major factor differentiation active and passive inference attack is that in active form of attack, the adversary can perform added local computations and put forward the results to the shared learning protocols.

Shen et al. in their work [104] have discussed data leakage in blockchain-assisted FL environments. The authors also propose novel property inference attacks for such models working on intelligent edge computing environments [105]. To be precise, the attack is active in nature and it learns the property leakage of individual model updates of devices and identifies a group of participants possessing a particular property. For the proposed model, the edge server is presumed to be the adversary and it has white-box access to the FL model and algorithm as well. The adversary can only acquire the combined global model formed from individual updates of several participants for every iteration. The aim is to perform selective inference of subgroups of participants who are more likely to possess the targeted properties. The adversary also needs to possess a set of correctly labeled auxiliary data alike the targeted data having similar gradients. It should also have correct labels of the targeted properties and labels of the main task. It is to be noted that the adversary need not follow the procedures of the main task, rather it is capable of taking active actions aiming to improve the efficiency and accuracy of the proposed attack. A dynamic participant selection methodology is also formulated aiming to accelerate the selection of target users. The efficiency and effectiveness of the model were experimentally proven for property inference without tampering with the performance of main tasks.

4.5. Attribute inference attacks

In attribute inference attack methodology, the intruder tries to familiarize or aim for the private attributes of a record holder and his identity will be deduced from the data in public platforms [106]. In most cases, such data will be processed with the help of some learning algorithms to make the derivations about the persons. Even though widely studied in ML [107,108], in the FL context, attribute inference attacks are not discussed extensively due to their resemblance to other modes of attack making it hard to categorize them as an individual entity extensively. Attribute inference attacks are mainly used in instances where the adversary possesses all the other non-relevant attributes of one particular data record and also has exclusive permission to either a trained model [83] or model embedding but is still in need of determining a missing field of that particular record to make it complete.

Lyu and Chen in their work [6] have proposed a novel attribute reconstruction attack in the context of FL. The author state the ineffectiveness of the current FL practice of gradient sharing after small batch-wise training and suggests the usage of an epoch averaged gradient approach which they claim to be more efficient than previous methods. They perform a detailed analysis of the attribute inference attacks and discuss its potency in cases of epoch averaged gradient sharing approach. In order to perform efficient attribute reconstruction, they also demonstrate the usage of a novel method referred to as the "cos-matching" approach, that makes use of the cosine similarity to estimate the apparent distance between the genuine update provided by the victim user and the simulated update. The authors validate all their claims via extensive experimentations and result comparisons performed with most of the baseline approaches.

Table 1 summarizes the articles we have investigated and provides an elite comparison based upon their attack methodology and mode of implementation.

5. Open challenges

FL itself is an emerging domain and numerous research projects are being done on the same. This gives rise to various open problems in all aspects of FL. This study extensively concentrates on privacy issues in the context of FL. So the discussions also will be oriented toward the open issues in privacy preservation and preventing adversarial attacks in FL.

Privacy preservation of user data is a basic requirement of all learning systems. It is being determined by multiple factors such as what is being computed, how computing is done, and who has the access to processed data. The "What" question can be addressed by employing procedures such as minimization and differential privacy. Employing deferential privacy accounting and privatization methodologies optimally is still an open challenge for real-life scenarios [90,109]. The solution to the "How" question can be attained by including secure MPC, HE, and trusted execution environment (TEE). Even though MPC techniques have been applied in limited applications having federation-crucial functionalities, other than that, most of the other functionalities are still executed in highly computationally complex methods [89]. At the same time, developing a dependable exploit-resistant TEE [110] method is still an open challenge and the needed assisting frameworks and processes are still in their primary stages of development. A possible solution to this issue can be achieved via the application of combinations of MPC, HE, and TEE methodologies anticipating that even in case of failure of one method, the other method would persist. Such technology is referred to as the "Privacy in Depth" concept which is still new in FL context [6]. The methodologies should be devised to enable extensive privacy. Even when a part of the system gets crippled, the privacy measures degradation should be "elegant rather than abrupt". Here, the phrase "elegant rather than abrupt" points to the state of degradation of the system, meaning that even during worst case scenarios, the system should not succumb to external access in a lightning pace rather than undergo a gradual breakdown in a cluster wise or similar manner giving ample triggers and time for the end user to identify and run safety measures. Another problem is related to verifiability. Verifiability simply means that the users can testify that their side of processing has been done genuinely. As of now, zero-knowledge proofs and trusted execution are the common methods used for initiating verifiability [93].

When it comes to open challenges related to adversarial attacks, they are numerous and varying. In some cases, devices participating in training regulate the optimization process which can cause negative

Table 1
Comparative study summary.

Ref.	Attack discussed	Setting	Attack methodology
[19]	Model Poisoning	Federated Learning	A single-shot model replacement attack for a secure backdoor creation is proposed. The idea is to scale up the weights of the poisoning model and replace it with the shared model to ensure the survival of the backdoor through averaging.
[44]	Membership Inference	Federated DL	Proposed model enables user to measure data membership leakage in training instances. The possible privacy breach of the stochastic gradient descent algorithm is utilized for this purpose.
[53]	Model Poisoning	Federated Learning	A model poisoning attack making use of alternative minimization for genuine and poisoned training in an attempt to improve the stealth wise performance of the attack are discussed.
[56]	Data Poisoning	Deep Learning	An approach known as Meta poison capable of identifying disturbances which point to the model behavior on particular targets is proposed.
[58]	Data Poisoning	Deep Learning	A backdoor attack model making use of dirty-label data poisoning is proposed and experimented for high efficiency. A minute quantity of data poisoning in the training set drives closer to attaining the adversarial aims .
[59]	Data Poisoning	Federated Learning	A GAN-based data poisoning approach is proposed. The usage of GAN is to mimic data of fellow participants for producing poisoning updates and such updates are scaled updates compromising the global model.
[60]	Data Poisoning	Edge-based FL	A GAN based data poisoning approach which makes use of repeatedly updated global parameters for pseudo data production resembling training data of fellow participants is proposed.
[78]	Data Poisoning	Federated Learning	A distributed backdoor attacking model also capable of performing a novel threat assessment framework is proposed. DBA aims at breaking down universal trigger patterns and attaching themselves to an individual's training data.
[61]	Data Poisoning	Federated Learning	A data poisoning attacks developed from basic label flipping attacks is proposed. The authors propose using poisoning regulation to ensure the survival of their update through aggregation.
[63]	Data Poisoning	Federated Learning	A poisoning attack via data reconstruction using decoder is proposed. The adversary has limited access to data and using the intermediate result and data, the adversary reconstruct private data of the benign users thereby gradually accessing sensitive data as well.
[64]	Model Poisoning	Federated Learning	A model poisoning attack performed by a single non-collaborative malicious user aiming for faulty classification of input data is discussed. Multiple proposals for the same are put forward and experimented.
[27]	Model Poisoning	Federated Learning	A backdoor attacking model with non-malicious clients possessing correctly labeled samples from the targeted tasks is demonstrated and evaluated in detail. Multiple factors that determine the attack performance is inferred through experimenting.
[69]	Model Poisoning	Federated Learning	A novel optimization-based model poisoning attack aiming for poisoning using redundant space is proposed. The prime concern of the model is in regulating the effectiveness, persistence, and stealth of attacks.
[62]	Model Poisoning	Federated Learning	An offline mode of attack derived from basic label flipping methodology aiming to compromise the integrity of global model at training phase making use of local model parameters. Also referred to as local model poisoning attacks.
[71]	Model Poisoning	Distributed Learning	A novel non-omniscient attack is proposed. The attacks work by adding a small amount of noise to genuine data gradients negatively impacting the global model. Using this data, the adversary fabricates poisonous updates that function independent of aggregation principles used.
[72]	Model Poisoning	Federated Learning	An attacker creates a reference aggregate with limited data they possess and then follows it with the generation of a "malicious perturbation". The final poisoned model updates are computed via perturbation of the malicious update
[73]	Model Poisoning	Collaborative Learning	A weighted aggregation method is proposed. The proposed attack aims for aggregation techniques and uses constantly updated weighted values to poison the model at every epoch. As small as two malicious users are enough for the attack to be implemented successfully.
[61]	Model Poisoning	Federated Learning	A model poisoning attacks referred to as PGA attack is proposed. It makes use of the SGA algorithm to fine-tune the model gradually poisoning it and increasing the total loss. To ensure malicious update survival, they propose a technique known as "ball of radius".
[70]	Model Poisoning	Federated Learning	A novel model poisoning referred to as unprecedented DP-exploited stealthy model poisoning attack is proposed. Makes use of differential privacy mechanism altering the noise used turning the model useless gradually.
[83]	Membership Inference	Collaborative Learning	Inference attacks caused due to unidentified information in a collaborative learning environment is demonstrated. The attack enables an adversary to infer membership details and specific attributes of individual users in the system.
[85]	Membership Inference	Deep Learning	The possibility of complex membership inference attacks being performed by participants capable of peeking into members' data is shown. Multiple factors acting as a catalyst for the attack is discussed.
[91]	Reconstruction	Collaborative DL	Model proposes creating a GAN to reconstruct a model using synthetic data which is similar to private ones. The synthetic data is created and refined using GAN discriminator.
[92]	Reconstruction	Federated Learning	A user-level GAN-based reconstruction attack is proposed. The model facilitates the adversary for selective data reconstruction of individual users.
[95]	Reconstruction	Federated Learning	Uses DLG algorithm and shared gradients for fake data creation similar to real data. By meticulous updation and comparison on gradients of both data, private training data can be obtained.

(*continued on next page*)

impacts on inference-time as with evasion attacks. From the results of various security analyses done on FL, it can be clearly inferred that the currently existing defenses are proving to be absurd with the changing communication and computational constraints [111]. When it comes to failures that are not caused by malicious users at the first glance, it is a still hard task to deal with as the FL setting does not provide access to raw data and such a failure may be the result of some sort of poisoning attacks [24]. Additionally, a major open question in FL is

Table 1 (*continued*).

Ref.	Attack discussed	Setting	Attack methodology
[96]	Reconstruction	Federated Learning	An improved version of DLG is proposed for reducing label data leakage during gradient sharing. Using the improved DLG, ground truth labels was accurately extracted from shared gradients.
[97]	Reconstruction	Federated Learning	Uses optimization techniques along with a magnitude-invariant loss to exploit leaked parametric gradient information for performing data reconstruction attacks.
[98]	Reconstruction	Federated Learning	Makes use of gradient acquired via data leakage and inverts them to perform reconstruction. The magnitude of gradients determines the reconstruction quality as well.
[94]	Reconstruction	Federated Learning	A GAN-based reconstruction attacks making use of model parameters is proposed. It uses the discriminator and generator network to iteratively fine-tune the result of one another eventually reconstructing identical copies of the target data and reducing system performance.
[100]	Model Inversion	Collaborative DL	A model for compromising inference data privacy for performing inversion attacks is discussed. The adversary is able to recover data of fellow users with limited or no access to their data or models updates.
[101]	Model Inversion	Machine Learning	The attack makes use of confidence values disclosed with a prediction for performing model inversion. The gaussian error model is used to drive the model close to identifying real labels and a minimally prejudiced MAP estimate for reduction of misprediction rate.
[104]	Property Inference	Federated Learning	Learns property leakage of individual model updates and identifies a group of participants possessing a particular property in a blockchain based environment

regarding the similitude of data poisoning and model poisoning attacks. As both cannot be exclusive of one another, it gives rise to a discussion on their relationship, similarity, and feasibility as individual entities. Another open question is on the impact of training time attacks on inference-time attack sensitivity [25].

In contrast to ML, in FL, there is no such threshold determining the genuineness of shared models. This gives rise to a long list of participants whose model updates may be misclassified as noise which may indeed be poisoning attacks. This affects the global modal convergence directly but still does not have a clear solution statement [19]. A similar challenging instance is related to the identity protection of users. In all settings, preserving the identity of a user is a fundamental need but this tampers the detection of adversarial users [92]. It is still an open challenge requiring effective solutions as the existing ones are not practically feasible. When it comes to communication-related issues, the future generation networks will be widely employing wireless methods of communication making the model highly susceptible to inference attacks. Thus stronger algorithms are needed to minimize outside interferences [112].

Even though FL systems are similar to some of the classical distributed computing and networking architectures up to a certain degree, some research aspects in FL related to privacy and system security are yet to gain high popularity. Studies revolving around privacy preservation and vulnerability analysis are still going on throughout the research community. As a result, a significant issue associated with such research is the lack of system vulnerability evaluating mechanisms as already available for networking applications [113,114]. For networking applications, a wide range of vulnerability evaluating mechanisms such as network scanning, vulnerability scanning, ethical hacking, and so on are used. When it comes to tools also, a wide array of tools such as intruder, Paessler PRTG [113], acunetix [114] and so on are available. From our bounded knowledge, we identify these shortcomings in FL security as a future scope with ample scope for developmental research.

In summary, some of the very basic open challenges presented here are:

- Challenges related to the optimal employability of differential privacy accounting and similar privatization methodologies.
- Issues faced in the transition of privacy measures from centralized systems to FL systems.
- Ineffectiveness of existing measures in ensuring user verifiability leading to privacy breaches.
- Lack of adaptive defensive techniques for constantly evolving FL environments.
- Incapability of system in differentiating between noisy updates and poisoned updates due to the lack of a well-defined threshold value for shared model authentication.
- The conflict between user identity preservation and adversarial user detection.

6. Conclusion

In this study, we present a detailed survey on literature-based upon privacy threats in FL. Specifically, we discuss different types of attacks, attack scenarios, and their classification based upon their modus operandi. Based on the study, we make classification of major attacks into poisoning attacks, inferencing attacks and discuss their types in detail. We also discuss some of the open issues and current works on their solution-finding. With the exploding trend of edge computing and IoT, we hope that the future of ML will be more into a distributed setting and thus privacy issues related to that will be a serious roadblock needing extensive study and solution finding in the coming era.

CRediT authorship contribution statement

Akarsh K. Nair: Conceptualization, Methodology, Software. **Ebin Deni Raj:** Analysis, Review and editing. **Jayakrushna Sahoo:** Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) http://dx.doi.org/10.1145/3298981.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: A. Singh, J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 1273–1282, URL https://proceedings.mlr.press/v54/mcmahan17a.html.

[3] B. McMahan, D. Ramage, Federated learning: Collaborative machine learning without centralized training data, 2017, URL https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[4] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60, http://dx.doi.org/10.1109/MSP.2020.2975749.

[5] Q. Xia, W. Ye, Z. Tao, J. Wu, Q. Li, A survey of federated learning for edge computing: Research problems and solutions, High-Confidence Comput. 1 (1) (2021) 100008, http://dx.doi.org/10.1016/j.hcc.2021.100008, URL https://www.sciencedirect.com/science/article/pii/S266729522100009X.

[6] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, 2019, arXiv preprint arXiv:1912.04977.

[7] M.H.u. Rehman, A.M. Dirir, K. Salah, E. Damiani, D. Svetinovic, TrustFed: A framework for fair and trustworthy cross-device federated learning in IIoT, IEEE Trans. Ind. Inform. 17 (12) (2021) 8485–8494, http://dx.doi.org/10.1109/TII.2021.3075706.

[8] M.A. Heikkilä, A. Koskela, K. Shimizu, S. Kaski, A. Honkela, Differentially private cross-silo federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2007.05553, arXiv, URL https://arxiv.org/abs/2007.05553.

[9] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, H. Yu, Federated learning, Synth. Lect. Artif. Intell. Mach. Learn. 13 (3) (2019) 1–207, http://dx.doi.org/10.2200/S00960ED2V01Y201910AIM043.

[10] M. Alazab, S.P. RM, P. M, P.K.R. Maddikunta, T.R. Gadekallu, Q.-V. Pham, Federated learning for cybersecurity: Concepts, challenges, and future directions, IEEE Trans. Ind. Inform. 18 (5) (2022) 3501–3509, http://dx.doi.org/10.1109/TII.2021.3119038.

[11] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, J. Roselander, Towards federated learning at scale: System design, in: A. Talwalkar, V. Smith, M. Zaharia (Eds.), Proceedings of Machine Learning and Systems, Vol. 1, 2019, pp. 374–388, URL https://proceedings.mlsys.org/paper/2019/file/bd686fd640be98efaae0091fa301e613-Paper.pdf.

[12] C. van Berkel, Multi-core for mobile phones, in: 2009 Design, Automation & Test in Europe Conference & Exhibition, 2009, pp. 1260–1265, http://dx.doi.org/10.1109/DATE.2009.5090858.

[13] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, http://dx.doi.org/10.48550/ARXIV.1610.05492, arXiv, URL https://arxiv.org/abs/1610.05492.

[14] V. Smith, C.-K. Chiang, M. Sanjabi, A.S. Talwalkar, Federated multi-task learning, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, URL https://proceedings.neurips.cc/paper/2017/file/6211080fa89981f66b1a0c9d55c61d0f-Paper.pdf.

[15] H.B. McMahan, D. Ramage, K. Talwar, L. Zhang, Learning differentially private recurrent language models, 2017, http://dx.doi.org/10.48550/ARXIV.1710.06963, arXiv, URL https://arxiv.org/abs/1710.06963.

[16] Y. Wu, X. Wang, W. Susilo, G. Yang, Z.L. Jiang, S.-M. Yiu, H. Wang, Generic server-aided secure multi-party computation in cloud computing, Comput. Stand. Interfaces 79 (2022) 103552, http://dx.doi.org/10.1016/j.csi.2021.103552, URL https://www.sciencedirect.com/science/article/pii/S0920548921000477.

[17] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–11, http://dx.doi.org/10.1145/3338501.3357370.

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1175–1191, http://dx.doi.org/10.1145/3133956.3133982.

[19] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: S. Chiappa, R. Calandra (Eds.), Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 2938–2948, URL https://proceedings.mlr.press/v108/bagdasaryan20a.html.

[20] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, D. Papailiopoulos, Attack of the tails: Yes, you really can backdoor federated learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 16070–16084, URL https://proceedings.neurips.cc/paper/2020/file/b8ffa41d4e492f0fad2f13e29e1762eb-Paper.pdf.

[21] N. Truong, K. Sun, S. Wang, F. Guitton, Y. Guo, Privacy preservation in federated learning: An insightful survey from the GDPR perspective, Comput. Secur. 110 (2021) 102402, http://dx.doi.org/10.1016/j.cose.2021.102402, URL https://www.sciencedirect.com/science/article/pii/S0167404821002261.

[22] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640, http://dx.doi.org/10.1016/j.future.2020.10.007, URL https://www.sciencedirect.com/science/article/pii/S0167739X20329848.

[23] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, P.S. Yu, Privacy and robustness in federated learning: Attacks and defenses, 2020, http://dx.doi.org/10.48550/ARXIV.2012.06337, arXiv, URL https://arxiv.org/abs/2012.06337.

[24] L. Lyu, H. Yu, Q. Yang, Threats to federated learning: A survey, 2020, http://dx.doi.org/10.48550/ARXIV.2003.02133, arXiv, URL https://arxiv.org/abs/2003.02133.

[25] N. Bouacida, P. Mohapatra, Vulnerabilities in federated learning, IEEE Access 9 (2021) 63229–63249, http://dx.doi.org/10.1109/ACCESS.2021.3075203.

[26] V. Tolpegin, S. Truex, M.E. Gursoy, L. Liu, Data poisoning attacks against federated learning systems, in: L. Chen, N. Li, K. Liang, S. Schneider (Eds.), Computer Security, ESORICS 2020, Springer International Publishing, Cham, 2020, pp. 480–501.

[27] Z. Sun, P. Kairouz, A.T. Suresh, H.B. McMahan, Can you really backdoor federated learning? 2019, http://dx.doi.org/10.48550/ARXIV.1911.07963, arXiv, URL https://arxiv.org/abs/1911.07963.

[28] M.S. Jere, T. Farnan, F. Koushanfar, A taxonomy of attacks on federated learning, IEEE Secur. Priv. 19 (2021).

[29] R.C. Geyer, T. Klein, M. Nabi, Differentially private federated learning: A client level perspective, 2017, http://dx.doi.org/10.48550/ARXIV.1712.07557, arXiv, URL https://arxiv.org/abs/1712.07557.

[30] D. Jatain, V. Singh, N. Dahiya, A contemplative perspective on federated machine learning: Taxonomy, threats & vulnerability assessment and challenges, J. King Saud Univ. - Comput. Inform. Sci. (2021) http://dx.doi.org/10.1016/j.jksuci.2021.05.016, URL https://www.sciencedirect.com/science/article/pii/S1319157821001312.

[31] P.M. Mammen, Federated learning: Opportunities and challenges, 2021, http://dx.doi.org/10.48550/ARXIV.2101.05428, arXiv, URL https://arxiv.org/abs/2101.05428.

[32] X. Hei, X. Yin, Y. Wang, J. Ren, L. Zhu, A trusted feature aggregator federated learning for distributed malicious attack detection, Comput. Secur. 99 (2020) 102033, http://dx.doi.org/10.1016/j.cose.2020.102033, URL https://www.sciencedirect.com/science/article/pii/S0167404820303060.

[33] C. Zhang, J. Zhang, D. Chai, K. Chen, Aegis: A trusted, automatic and accurate verification framework for vertical federated learning, 2021, http://dx.doi.org/10.48550/ARXIV.2108.06958, arXiv, URL https://arxiv.org/abs/2108.06958.

[34] L. Song, C. Ma, P. Wu, Y. Zhang, PPD-DL: Privacy-preserving decentralized deep learning, in: X. Sun, Z. Pan, E. Bertino (Eds.), Artificial Intelligence and Security, Springer International Publishing, Cham, 2019, pp. 273–282.

[35] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn, M. Purcell, A. Rawat, T. Minh, N. Holohan, S. Chakraborty, S. Whitherspoon, D. Steuer, L. Wynter, H. Hassan, S. Laguna, M. Yurochkin, M. Agarwal, E. Chuba, A. Abay, IBM federated learning: An enterprise framework white paper V0.1, 2020, http://dx.doi.org/10.48550/ARXIV.2007.10987, arXiv, URL https://arxiv.org/abs/2007.10987.

[36] X. Ma, Y. Zhou, L. Wang, M. Miao, Privacy-preserving Byzantine-robust federated learning, Comput. Stand. Interfaces 80 (2022) 103561, http://dx.doi.org/10.1016/j.csi.2021.103561, URL https://www.sciencedirect.com/science/article/pii/S0920548921000568.

[37] C. Fung, C.J.M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, 2018, http://dx.doi.org/10.48550/ARXIV.1808.04866, arXiv, URL https://arxiv.org/abs/1808.04866.

[38] C. Fung, C.J.M. Yoon, I. Beschastnikh, The limitations of federated learning in sybil settings, in: 23rd International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2020, USENIX Association, San Sebastian, 2020, pp. 301–316, URL https://www.usenix.org/conference/raid2020/presentation/fung.

[39] H. Weng, J. Zhang, F. Xue, T. Wei, S. Ji, Z. Zong, Privacy leakage of real-world vertical federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2011.09290, arXiv, URL https://arxiv.org/abs/2011.09290.

[40] S. Li, Y. Cheng, W. Wang, Y. Liu, T. Chen, Learning to detect malicious clients for robust federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2002.00211, arXiv, URL https://arxiv.org/abs/2002.00211.

[41] C. Wu, X. Yang, S. Zhu, P. Mitra, Mitigating backdoor attacks in federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2011.01767, arXiv, URL https://arxiv.org/abs/2011.01767.

[42] H. Lee, J. Kim, S. Ahn, R. Hussain, S. Cho, J. Son, Digestive neural networks: A novel defense strategy against inference attacks in federated learning, Comput. Secur. 109 (2021) 102378, http://dx.doi.org/10.1016/j.cose.2021.102378, URL https://www.sciencedirect.com/science/article/pii/S0167404821002029.

[43] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, H. Jegou, White-box vs black-box: Bayes optimal strategies for membership inference, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 5558–5567, URL https://proceedings.mlr.press/v97/sablayrolles19a.html.

[44] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, in: 2019 IEEE Symposium on Security and Privacy, SP, 2019, pp. 739–753, http://dx.doi.org/10.1109/SP.2019.00065.

[45] G. Costa, F. Pinelli, S. Soderi, G. Tolomei, Covert channel attack to federated learning systems, 2021, arXiv preprint arXiv:2104.10561.

[46] J. Feng, Q.-Z. Cai, Z.-H. Zhou, Learning to confuse: Generating training time adversarial data with auto-encoder, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, URL https://proceedings.neurips.cc/paper/2019/file/1ce83e5d4135b07c0b82afffbe2b3436-Paper.pdf.

[47] J. Zhang, D. Wu, C. Liu, B. Chen, Defending poisoning attacks in federated learning via adversarial training method, in: G. Xu, K. Liang, C. Su (Eds.), Frontiers in Cyber Security, Springer Singapore, Singapore, 2020, pp. 83–94.

[48] K. Singhal, H. Sidahmed, Z. Garrett, S. Wu, J. Rush, S. Prakash, Federated reconstruction: Partially local federated learning, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 11220–11232, URL https://proceedings.neurips.cc/paper/2021/file/5d44a2b0d85aa1a4dd3f218be6422c66-Paper.pdf.

[49] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, 2012, http://dx.doi.org/10.48550/ARXIV.1206.6389, arXiv, URL https://arxiv.org/abs/1206.6389.

[50] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, J. Liu, Data poisoning attacks on federated machine learning, IEEE Internet Things J. (2021) 1, http://dx.doi.org/10.1109/JIOT.2021.3128646.

[51] A. Shafahi, W.R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! Targeted clean-label poisoning attacks on neural networks, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018, URL https://proceedings.neurips.cc/paper/2018/file/22722a343513ed45f14905eb07621686-Paper.pdf.

[52] T. Gu, B. Dolan-Gavitt, S. Garg, BadNets: Identifying vulnerabilities in the machine learning model supply chain, 2017, http://dx.doi.org/10.48550/ARXIV.1708.06733, arXiv, URL https://arxiv.org/abs/1708.06733.

[53] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 634–643, URL https://proceedings.mlr.press/v97/bhagoji19a.html.

[54] N. Rodríguez-Barroso, E. Martínez-Cámara, M.V. Luzón, F. Herrera, Dynamic defense against Byzantine poisoning attacks in federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2007.15030, arXiv, URL https://arxiv.org/abs/2007.15030.

[55] N. Bouacida, P. Mohapatra, Vulnerabilities in federated learning, IEEE Access 9 (2021) 63229–63249, http://dx.doi.org/10.1109/ACCESS.2021.3075203.

[56] W.R. Huang, J. Geiping, L. Fowl, G. Taylor, T. Goldstein, MetaPoison: Practical general-purpose clean-label data poisoning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 12080–12091, URL https://proceedings.neurips.cc/paper/2020/file/8ce6fc704072e351679ac97d4a985574-Paper.pdf.

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

[58] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, 2017, http://dx.doi.org/10.48550/ARXIV.1712.05526, arXiv, URL https://arxiv.org/abs/1712.05526.

[59] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE, 2019, pp. 374–380, http://dx.doi.org/10.1109/TrustCom/BigDataSE.2019.00057.

[60] J. Zhang, B. Chen, X. Cheng, H.T.T. Binh, S. Yu, PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems, IEEE Internet Things J. 8 (5) (2021) 3310–3322, http://dx.doi.org/10.1109/JIOT.2020.3023126.

[61] V. Shejwalkar, A. Houmansadr, P. Kairouz, D. Ramage, Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning, 2021, http://dx.doi.org/10.48550/ARXIV.2108.10241, arXiv, URL https://arxiv.org/abs/2108.10241.

[62] M. Fang, X. Cao, J. Jia, N. Gong, Local model poisoning attacks to Byzantine-Robust federated learning, in: 29th USENIX Security Symposium, USENIX Security 20, USENIX Association, 2020, pp. 1605–1622, URL https://www.usenix.org/conference/usenixsecurity20/presentation/fang.

[63] S. Zhang, L. Xiang, X. Yu, P. Chu, Y. Chen, C. Cen, L. Wang, Privacy-preserving federated learning on partitioned attributes, 2021, arXiv preprint arXiv:2104.14383.

[64] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Model poisoning attacks in federated learning, in: In Workshop on Security in Machine Learning (SecML), Collocated with the 32nd Conference on Neural Information Processing Systems, NeurIPS'18, 2018.

[65] J. So, B. Güler, A.S. Avestimehr, Byzantine-resilient secure federated learning, IEEE J. Sel. Areas Commun. 39 (7) (2021) 2168–2181, http://dx.doi.org/10.1109/JSAC.2020.3041404.

[66] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, D. Sánchez, A. Flanagan, K.E. Tan, Achieving security and privacy in federated learning systems: Survey, research challenges and future directions, Eng. Appl. Artif. Intell. 106 (2021) 104468, http://dx.doi.org/10.1016/j.engappai.2021.104468, URL https://www.sciencedirect.com/science/article/pii/S095219762100316X.

[67] Z. Chen, P. Tian, W. Liao, W. Yu, Towards multi-party targeted model poisoning attacks against federated learning systems, High-Confidence Comput. 1 (1) (2021) 100002, http://dx.doi.org/10.1016/j.hcc.2021.100002, URL https://www.sciencedirect.com/science/article/pii/S2667295221000039.

[68] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 19586–19597, URL https://proceedings.neurips.cc/paper/2020/file/e32cc80bf07915058ce90722ee17bb71-Paper.pdf.

[69] X. Zhou, M. Xu, Y. Wu, N. Zheng, Deep model poisoning attack on federated learning, Future Internet 13 (3) (2021) http://dx.doi.org/10.3390/fi13030073, URL https://www.mdpi.com/1999-5903/13/3/73.

[70] M.T. Hossain, S. Islam, S. Badsha, H. Shen, Desmp: Differential privacy-exploited stealthy model poisoning attacks in federated learning, 2021, http://dx.doi.org/10.48550/ARXIV.2109.09955, arXiv, URL https://arxiv.org/abs/2109.09955.

[71] G. Baruch, M. Baruch, Y. Goldberg, A little is enough: Circumventing defenses for distributed learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, URL https://proceedings.neurips.cc/paper/2019/file/ec1c59141046cd1866bbbcdfb6ae31d4-Paper.pdf.

[72] V. Shejwalkar, A. Houmansadr, Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning, Internet Soc. (2021) 18.

[73] H. Chang, V. Shejwalkar, R. Shokri, A. Houmansadr, Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer, 2019, http://dx.doi.org/10.48550/ARXIV.1912.11279, arXiv, URL https://arxiv.org/abs/1912.11279.

[74] S. Arora, E. Hazan, S. Kale, The multiplicative weights update method: A meta-algorithm and applications, Theory Comput. 8 (6) (2012) 121–164, http://dx.doi.org/10.4086/toc.2012.v008a006, URL https://theoryofcomputing.org/articles/v008a006.

[75] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, J. Han, Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 1187–1198, http://dx.doi.org/10.1145/2588555.2610509.

[76] J. So, B. Güler, A.S. Avestimehr, Byzantine-resilient secure federated learning, IEEE J. Sel. Areas Commun. 39 (7) (2021) 2168–2181, http://dx.doi.org/10.1109/JSAC.2020.3041404.

[77] Z. Zhang, J. Li, S. Yu, C. Makaya, Safelearning: Enable backdoor detectability in federated learning with secure aggregation, 2021, http://dx.doi.org/10.48550/ARXIV.2102.02402, arXiv, URL https://arxiv.org/abs/2102.02402.

[78] C. Xie, K. Huang, P.-Y. Chen, B. Li, DBA: Distributed backdoor attacks against federated learning, in: International Conference on Learning Representations, 2019.

[79] A. Huang, Dynamic backdoor attacks against federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2011.07429, arXiv, URL https://arxiv.org/abs/2011.07429.

[80] C. Zhao, Y. Wen, S. Li, F. Liu, D. Meng, FederatedReverse: A detection and defense method against backdoor attacks in federated learning, in: Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, in: IH&amp;MMSec '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 51–62, http://dx.doi.org/10.1145/3437880.3460403.

[81] C. Wu, X. Yang, S. Zhu, P. Mitra, Mitigating backdoor attacks in federated learning, 2020, http://dx.doi.org/10.48550/ARXIV.2011.01767, arXiv, URL https://arxiv.org/abs/2011.01767.

[82] S. Truex, L. Liu, M.E. Gursoy, L. Yu, W. Wei, Towards demystifying membership inference attacks, 2018, http://dx.doi.org/10.48550/ARXIV.1807.09173, arXiv, URL https://arxiv.org/abs/1807.09173.

[83] L. Melis, C. Song, E. De Cristofaro, V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, in: 2019 IEEE Symposium on Security and Privacy, SP, 2019, pp. 691–706, http://dx.doi.org/10.1109/SP.2019.00029.

[84] L. Bottou, Stochastic gradient descent tricks, in: G. Montavon, G.B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade, second ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 421–436, http://dx.doi.org/10.1007/978-3-642-35289-8_25.

[85] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, in: 2019 IEEE Symposium on Security and Privacy, SP, 2019, pp. 739–753, http://dx.doi.org/10.1109/SP.2019.00065.

[86] J. Sun, Y. Yao, W. Gao, J. Xie, C. Wang, Defending against reconstruction attack in vertical federated learning, 2021, arXiv preprint arXiv:2107.09898.

[87] Y. Li, Y. Li, H. Xu, S. Ren, An adaptive communication-efficient federated learning to resist gradient-based reconstruction attacks, Secur. Commun. Netw. 2021 (2021).

[88] Y. Wu, X. Wang, W. Susilo, G. Yang, Z.L. Jiang, J. Li, X. Liu, Mixed-protocol multi-party computation framework towards complex computation tasks with malicious security, Comput. Stand. Interfaces 80 (2022) 103570, http://dx.doi.org/10.1016/j.csi.2021.103570, URL https://www.sciencedirect.com/science/article/pii/S0920548921000659.

[89] H. Fang, Q. Qian, Privacy preserving machine learning with homomorphic encryption and federated learning, Future Internet 13 (4) (2021) http://dx.doi.org/10.3390/fi13040094, URL https://www.mdpi.com/1999-5903/13/4/94.

[90] H. Ku, W. Susilo, Y. Zhang, W. Liu, M. Zhang, Privacy-preserving federated learning in medical diagnosis with homomorphic re-encryption, Comput. Stand. Interfaces 80 (2022) 103583, http://dx.doi.org/10.1016/j.csi.2021.103583, URL https://www.sciencedirect.com/science/article/pii/S0920548921000787.

[91] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the GAN: Information leakage from collaborative deep learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 603–618, http://dx.doi.org/10.1145/3133956.3134012.

[92] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 2512–2520, http://dx.doi.org/10.1109/INFOCOM.2019.8737416.

[93] G. Han, T. Zhang, Y. Zhang, G. Xu, J. Sun, J. Cao, Verifiable and privacy preserving federated learning without fully trusted centers, J. Ambient Intell. Humaniz. Comput. (2021) 1–11.

[94] Y. Sun, N.S.T. Chong, H. Ochiai, Information stealing in federated learning systems based on generative adversarial networks, in: 2021 IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2021, pp. 2749–2754, http://dx.doi.org/10.1109/SMC52423.2021.9658652.

[95] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, URL https://proceedings.neurips.cc/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf.

[96] B. Zhao, K.R. Mopuri, H. Bilen, iDLG: Improved deep leakage from gradients, 2020, http://dx.doi.org/10.48550/ARXIV.2001.02610, arXiv, URL https://arxiv.org/abs/2001.02610.

[97] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller, Inverting gradients - How easy is it to break privacy in federated learning? in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 16937–16947, URL https://proceedings.neurips.cc/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf.

[98] J.Q. Lim, C.S. Chan, From gradient leakage to adversarial attacks in federated learning, in: 2021 IEEE International Conference on Image Processing, ICIP, 2021, pp. 3602–3606, http://dx.doi.org/10.1109/ICIP42928.2021.9506589.

[99] J. Sun, A. Li, B. Wang, H. Yang, H. Li, Y. Chen, Provable defense against privacy leakage in federated learning from representation perspective, 2020, http://dx.doi.org/10.48550/ARXIV.2012.06043, arXiv, URL https://arxiv.org/abs/2012.06043.

[100] Z. He, T. Zhang, R.B. Lee, Model inversion attacks against collaborative inference, in: Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 148–162, http://dx.doi.org/10.1145/3359789.3359824.

[101] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1322–1333, http://dx.doi.org/10.1145/2810103.2813677.

[102] M.P.M. Parisot, B. Pejo, D. Spagnuelo, Property inference attacks on convolutional neural networks: Influence and implications of target model's complexity, 2021, http://dx.doi.org/10.48550/ARXIV.2104.13061, arXiv, URL https://arxiv.org/abs/2104.13061.

[103] M. Xu, X. Li, Subject property inference attack in collaborative learning, in: 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics, Vol. 1, IHMSC, 2020, pp. 227–231, http://dx.doi.org/10.1109/IHMSC49165.2020.00057.

[104] M. Shen, H. Wang, B. Zhang, L. Zhu, K. Xu, Q. Li, X. Du, Exploiting unintended property leakage in blockchain-assisted federated learning for intelligent edge computing, IEEE Internet Things J. 8 (4) (2021) 2265–2275, http://dx.doi.org/10.1109/JIOT.2020.3028110.

[105] R. Gupta, D. Reebadiya, S. Tanwar, 6G-enabled edge intelligence for ultra - reliable low latency applications: Vision and mission, Comput. Stand. Interfaces 77 (2021) 103521, http://dx.doi.org/10.1016/j.csi.2021.103521, URL https://www.sciencedirect.com/science/article/pii/S0920548921000167.

[106] J. Jia, N.Z. Gong, AttriGuard: A practical defense against attribute inference attacks via adversarial machine learning, in: 27th USENIX Security Symposium, USENIX Security 18, USENIX Association, Baltimore, MD, 2018, pp. 513–529, URL https://www.usenix.org/conference/usenixsecurity18/presentation/jia-jinyuan.

[107] ML-Doctor: Holistic risk assessment of inference attacks against machine learning models, in: 31st USENIX Security Symposium, USENIX Security 22, USENIX Association, Boston, MA, 2022, URL https://www.usenix.org/conference/usenixsecurity22/presentation/liu-yugeng.

[108] S. Mehnaz, N. Li, E. Bertino, Black-box model inversion attribute inference attacks on classification models, 2020, http://dx.doi.org/10.48550/ARXIV.2012.03404, arXiv, URL https://arxiv.org/abs/2012.03404.

[109] K. Wei, J. Li, M. Ding, C. Ma, H.H. Yang, F. Farokhi, S. Jin, T.Q.S. Quek, H.V. Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3454–3469, http://dx.doi.org/10.1109/TIFS.2020.2988575.

[110] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, J. Li, A training-integrity privacy-preserving federated learning scheme with trusted execution environment, Inform. Sci. 522 (2020) 69–79, http://dx.doi.org/10.1016/j.ins.2020.02.037, URL https://www.sciencedirect.com/science/article/pii/S0020025520301201.

[111] Z. Iqbal, H. Chan, Concepts, key challenges and open problems of federated learning, Int. J. Eng. 34 (7) (2021) 1667–1683.

[112] C. Fang, Y. Guo, Y. Hu, B. Ma, L. Feng, A. Yin, Privacy-preserving and communication-efficient federated learning in Internet of Things, Comput. Secur. 103 (2021) 102199, http://dx.doi.org/10.1016/j.cose.2021.102199, URL https://www.sciencedirect.com/science/article/pii/S0167404821000237.

[113] H. Teymourlouei, V.E. Harris, Effectiveness of real-time network monitoring for identifying hidden vulnerabilities inside a system, in: 2020 International Conference on Computational Science and Computational Intelligence, CSCI, 2020, pp. 43–48, http://dx.doi.org/10.1109/CSCI51800.2020.00014.

[114] M. Vieira, N. Antunes, H. Madeira, Using web security scanners to detect vulnerabilities in web services, in: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, 2009, pp. 566–571, http://dx.doi.org/10.1109/DSN.2009.5270294.