POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Adversarial Attacks Against Federated Learning Systems: A Review

Tesi di Laurea Magistrale in
Computer Science and Engineering
- Ingegneria Informatica

Author: **Marco Tagliafierro**

Student ID: 969645
Advisor: Prof. Michele Carminati
Academic Year: 2021-22

Abstract

In the last few years interest in Federated Learning systems has increased dramatically: they have been proven to be effective and efficient in all those scenarios where a safe distributed training mechanism is needed, allowing to overcome the privacy and performance limitations that standard techniques face. The growing adoption of this kind of system has also led to the development of many attacks against it to achieve different adversarial objectives, ranging from backdooring the learned model to leaking the private information owned by the participants of the training process.

This thesis' objective is to study the current state-of-the-art of such techniques, providing a complete overview and categorization of the most effective ones; this could be beneficial not only to further improve them, overcoming their limitations, but also to help Federated Learning systems' designers to develop more robust and secure architectures.

To do this, several papers discussing adversarial attacks have been deeply analyzed and a Systematization of Knowledge - SoK - has been created using a number of parameters to provide an easy way to compare each technique against each other. Moreover, some possible future developments of the considered works have been proposed, taking into consideration their main weaknesses and the less analyzed scenarios. In conclusion, a comparison with similar surveys has been created to highlight how this thesis can provide a more comprehensive overview of Federated Learning attacks.

Keywords: Federated Learning, adversarial, attacks, SoK

Abstract in lingua italiana

Negli ultimi anni l'interesse per i sistemi di Federated Learning è aumentato significativamente: questi si sono dimostrati particolarmente efficaci in tutti quei contesti in cui è necessario un meccanismo di training distribuito sicuro, che permetta di affrontare le limitazioni in termini di privacy e performance che le tecniche standard non permettono di superare. La crescente adozione di questa tipologia di sistemi ha portato allo sviluppo di numerosi attacchi con lo scopo di raggiungere diversi obiettivi avversariali, che vanno dalla creazione di backdoor nel modello prodotto alla divulgazione di informazioni private possedute dai partecipanti del processo di training.

L'obiettivo di questa tesi è quello di studiare l'attuale stato dell'arte di tali tecniche, fornendo una completa panoramica e categorizzazione delle migliori; questo può risultare utile non solo per lo studio di eventuali migliorie degli attacchi considerati ma anche per fornire un supporto ai progettisti nel creare sistemi di Federated Learning più robusti e sicuri.

Per fare ciò, sono stati analizzati numerosi paper riguardanti attacchi avversariali e una Systematization of Knowledge - SoK - è stata creata tenendo in considerazione vari parametri con lo scopo di fornire un modo rapido e intuitivo di comparare le diverse tecniche studiate. Sono stati inoltre riportati diversi possibili sviluppi futuri dei lavori considerati, con lo scopo di superare le loro principali debolezze e aumentare il numero di applicazioni negli scenari meno studiati. Infine, una comparazione con studi simili a quello proposto è stata realizzata per evidenziare come questa tesi sia in grado di fornire una overview più completa sugli attacchi ai sistemi di Federated Learning.

Parole chiave: Federated Learning, attacchi, avversariali, SoK

Contents

Introduction

Federated Learning is a novel branch of artificial intelligence that overcomes the limitations that traditional centralized machine learning approaches face in distributed settings by enabling multiple devices, such as smartphones, personal computers or IoT apparatus, to collectively train a model without needing to share their private information. In Federated Learning systems the training phase changes drastically: it is pushed back to the devices owning the used datasets, which will contribute in an iterative process to the generation of the global model by only exchanging small bits of data, either in the form of gradients or parameters, not containing any private information.

The lack of sample sharing with a central entity or other clients is crucial in guaranteeing privacy and ownership of the datasets belonging to each participant. Moreover, this aspect helps to satisfy the requirements of the latest data policies, like the European General Data Protection Regulations - GDPR - or the Cybersecurity Law of the People's Republic of China, which pose new challenges in the data elaboration field that are difficult to overcome with standard centralized machine learning techniques.

It is also important to note that conducting the training phase on huge heterogeneous datasets, that would otherwise not be easily accessible, makes it possible to produce models which can achieve higher performances than the ones trained following classical centralized paradigms. Several variations to the original Federated Learning concept have been and are still being developed to adapt it to the various scenarios where its characteristics are needed: while this type of distributed learning is commonly adopted in smartphones and other personal devices thanks to its privacy features - for example, think of the way GBoard, Google's mobile keyboard, uses it to improve its word prediction capabilities [12] -, some variants are also used by organizations in contexts where there is the need to train a model combining their datasets without disclosing them.

Their distributed nature exposes Federated Learning systems to a completely new set of security threats compared to traditional machine learning settings: the increased complexity of these paradigms creates several new attack surfaces that can be exploited by internal and external adversarial agents; malicious parties may now leverage new approaches to tamper with the learning process and its privacy guarantees. While many defense and prevention techniques have been developed to secure Federated Learning, several novel attacks exploiting different weaknesses are being studied and presented. This thesis' main objective is to create a comprehensive overview of the state-of-the-art of adversarial attacks against Federated Learning systems, classifying the most relevant works to understand the concepts behind them and to compare them against each other.

Approach overview

What follows is an overview of the approach that has been adopted to realize this thesis.

- Different papers analyzing the Federated Learning paradigm have been studied to understand the main concepts behind it.
- Other existing surveys on attacks against Federated Learning systems have been considered to understand which are the main attack surfaces and techniques currently known and used; they have also been used to gather an initial set of papers to analyze.
- Once the attacks linked to the analyzed surveys have been studied, I proceeded to search for other correlated papers.
- The collected material has then been filtered to keep only the most relevant works and thoroughly analyzed to understand the current state-of-the-art.
- A Systematization of Knowledge - SoK - table has been created to categorize all the considered attacks.

Results of the proposed analyses

The analyses carried out in this thesis highlights which are the most relevant and effective adversarial attack types against Federated Learning systems:

- Free-rider attacks
- Poisoning attacks, divided into data and model poisoning attacks
- Inference attacks, divided into features, labels, membership and properties inference attacks

Many of the considered attacks are applicable even in realistic scenarios where robust defenses or aggregation mechanisms are deployed, taking advantage of a variety of techniques, ranging from optimization algorithms to Generative Adversarial Networks, to achieve their adversarial objectives. Some common weaknesses have also been pointed out, like for example the need for prior knowledge about the data distribution of the clients' private datasets or the need for the adversarial devices to be selected at each round of the training process: it is fundamental to understand how these limitations may affect the applicability of the studied techniques to evaluate in which cases they can and cannot be applied.

Moreover, the proposed Systematization of Knowledge highlights which are the most covered scenarios and how each considered attack compares against similar ones: it is possible to note how most of the analyzed papers study poisoning and inference attacks, focusing on HFL scenarios and client adversarial devices. It is also highlighted how it is usually required to have at least access to the data belonging to the controlled devices and to be able to participate during the training process of the Federated Learning system.

Thanks to the comparison between this thesis and other similar surveys, it has been underlined how many of them aren't providing an intuitive way of comparing each attack technique against each other; it should also be noted how most of these papers don't consider free-rider attacks, don't give a complete overview of the analyzed techniques and/or don't highlight their weak points, providing only a partial overview of the current state-of-the-art of adversarial attacks against Federated Learning systems.

Main contributions

The main contributions of this thesis are the following:

- Review of the current state-of-the-art of adversarial attacks against Federated Learning systems, taking into account all the major system and attack types.
- Categorization of the studied works using a complete yet synthetic approach to understand how the analyzed attacks are implemented and provide a way of comparing them.
- Proposal of possible future development directions with respect to the analyzed works.
- Creation of a brief analysis and comparison against similar surveys to show what are the main advantages of this thesis.

1 | Approach

What follows is the approach that has been adopted to collect, analyze and classify all the relevant studies related to adversarial attacks against Federated Learning systems. Most of the considered papers have been found using either Google Scholar or ACM Digital Library.

- Given an already good understanding of the theory behind machine learning, several papers analyzing the Federated Learning paradigm have been studied in order to understand how it works, what problems it solves and how it is currently implemented and used in the existing systems.
- From this starting point, other existing surveys on adversarial attacks against Federated Learning systems have been studied to understand the main attack surfaces and techniques currently known and used. Moreover, these surveys have been used to gather an initial set of papers to be analyzed.
- Once the attacks linked to the considered surveys have been studied, I proceeded to search for pertinent works by looking at the ones in their citations lists: this process has been repeated for each newly considered paper until I wasn't able to find any other interesting study regarding Federated Learning attacks.
- After creating a first categorization of the found attacks, further research has been done to look for less-represented attack categories and to ensure that every relevant work has been included. This allowed obtaining papers that were not related to the ones considered in the initial phases; their citation lists have been analyzed as previously discussed.
- The collected material has then been filtered by removing all the studies which weren't sufficiently documented, didn't include enough details on how the attack was tested or did consider scenarios that were too unrealistic. I avoided considering the number of citations since I found it to be not representative of the actual quality of the papers.
- The remaining attacks have been grouped by category and thoroughly analyzed to

highlight their key concepts.

- Eventually, I created a Systematization of Knowledge - SoK - table comprising 32 different parameters to categorize and compare the studied attacks.

Overall, a total of 51 sources have been used to write this thesis, with 42 different papers regarding adversarial attacks of three different types:

- 3 regarding free-rider attacks
- 17 regarding poisoning attacks, describing:
    - 12 data poisoning techniques
    - 6 model poisoning techniques
- 22 regarding inference attacks, describing:
    - 5 features inference techniques
    - 5 labels inference techniques
    - 8 membership inference techniques
    - 5 properties inference techniques

In Table 1.1 are summarized the above values:

Table 1.1: Number of analyzed techniques for each attack type

Number of analyzed attacks	Free-rider attacks	Poisoning attacks		Inference attacks			
		data poisoning	model poisoning	feature	labels	membership	properties
Per sub-category	3	12	6	5	5	8	5
Per category	3	18		23			
Overall	44 - some papers discussed more than one attack type						

By looking at the number of analyzed attacks it is possible to understand how inference and poisoning ones are the most covered: given the current implementations of Federated Learning systems, they pose the biggest threats to the fundamental assumptions of this paradigm, allowing attackers to waste participants' resources, to modify the produced model or to steal private information that would otherwise be inaccessible.

While less studied and generally more difficult to apply, also free-rider techniques are worth being analyzed to have a complete understanding of how non-participating clients may be able to obtain the global model: this is an important threat to consider in all those scenarios where the produced model has a very high commercial or strategic value.

In Figure 1.1 is depicted a graph showing the number of publications related to the keywords "Federated Learning" and "Attacks" while in Figure 1.2 a timeline representing how many of the analyzed papers have been presented in each year is proposed. These graphs help to visualize how the interest in adversarial attacks against Federated Learning systems has increased over time.

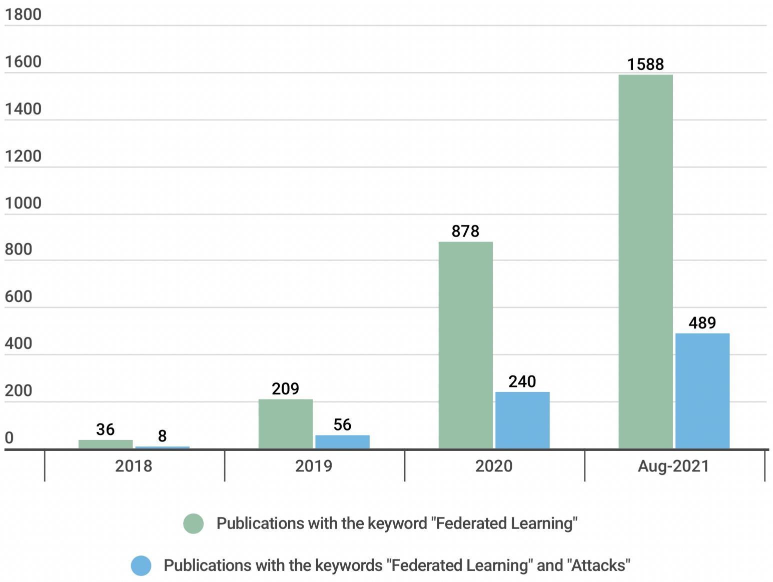

Figure 1.1: Total number of publications regarding Federated Learning and attacks against it - data taken from [29]

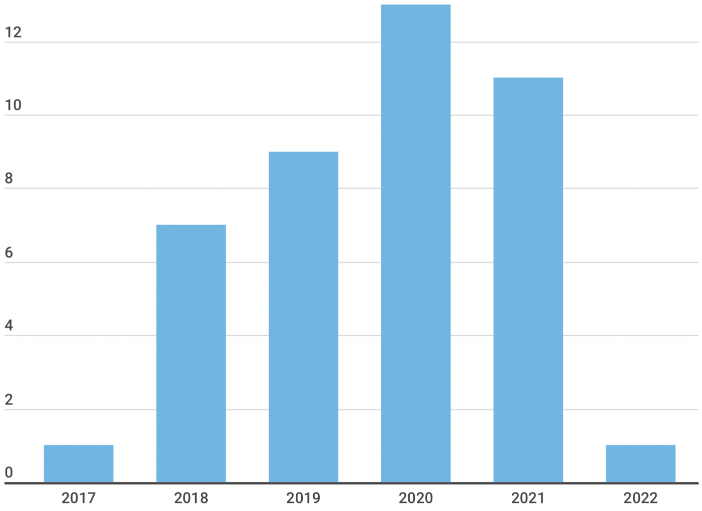

Figure 1.2: Number of analyzed papers per year

2 | Federated Learning Basics

While the final objective is similar, Federated Learning paradigm and standard centralized machine learning have a key difference that is to be found in the training phase: it is decentralized and delegated to the devices actually owning the data, which do not share any information with each other.
This characteristic creates systems that are far more structured and complex to manage than classical ones, involving coordination algorithms and usually requiring the presence of a central server that manages all the participants and aggregates their updates.

2.1. Phases of the Federated Learning process

The learning process of the most common Federated Learning systems, the ones comprising a central server that coordinates several client devices, is an iterative mechanism that is run until the global model converges to a desired level of performance.
It is possible to identify four main phases in this process:

- Data collection: as a preliminary step, each participant needs to collect the data that will be used in the training phase. Data may then be pre-processed and/or filtered, based on some rules agreed upon between participants, to optimize it for the training process, for example removing duplicates or reducing the noise present in each one of the samples.

- Model selection and initial training: a proper model needs to be chosen to solve the task we are interested in and training with a sample dataset takes place in the central server to set a starting point for the learning process; the produced model's parameters are then broadcasted to all the participants as a baseline for future updates.

- Local model training: at each training round every client receives the global model from the central server; then a randomly selected subset of participants will update the shared model using their private dataset and communicate their updates to the coordinator, either by sharing the computed models' parameters or their gradients.

The central server then proceeds to combine the received information using the chosen aggregation algorithm.

- Global model finalization and prediction phase: once the training process is finished the aggregated global model is sent to all the clients that took part in it; the model is then used locally by the participants to make predictions.

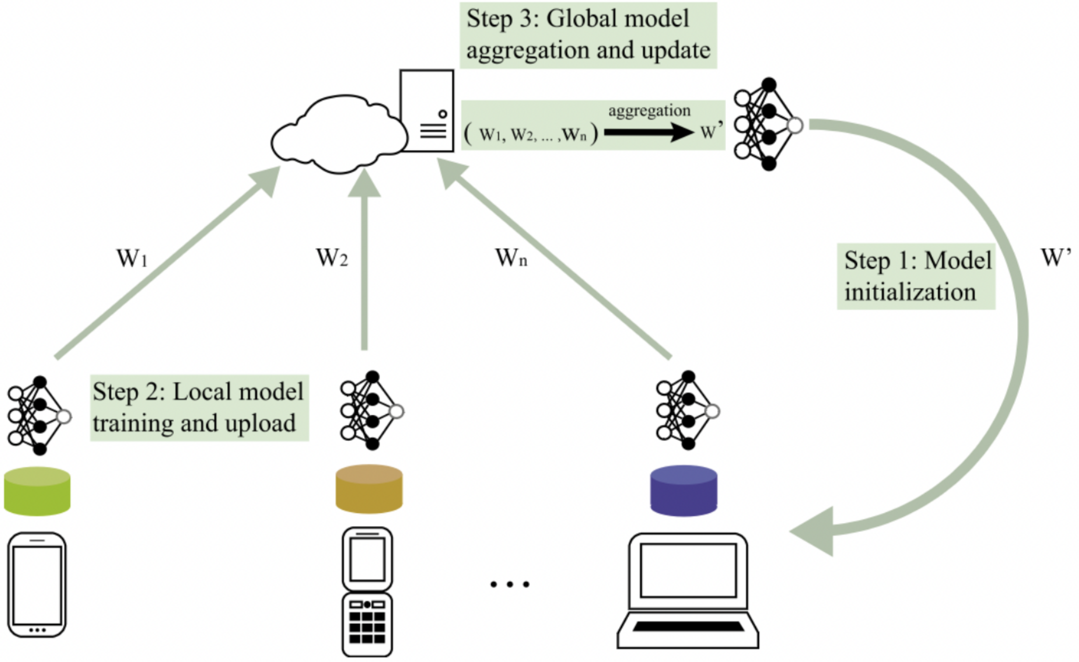
Step 3: Global model aggregation and update
aggregation
(W1, W2, ... ,Wn) → W'
W1
W2
Wn
Step 1: Model initialization
W'
Step 2: Local model training and upload
...

Figure 2.1: Graphical representation of the Federated Learning process' phases. Image taken from [44]

Those phases slightly change when the considered architecture is a fully decentralized one, in which a central server is not needed to make the learning possible and clients coordinate themselves using consensus protocols; anyhow, the main concepts behind the process remain similar.

2.2. Types of Federated Learning

Federated Learning systems can be classified in several ways depending on the considered aspect of the framework; the main classification that helps visualize how the process is carried out is based on the distribution of data samples, features and labels among the participants:

- Horizontal Federated Learning - HFL: clients' datasets share the same feature and label spaces while comprising different samples; this is the most common scenario in general applications of Federated Learning, deployed for example on phones or on IoT devices to train their machine learning models without compromising users' privacy.

- Vertical Federated Learning - VFL: clients' datasets share the same sample space but have different feature and label spaces; this scenario happens whenever two or more organizations own different features of the same samples (for example, different information about the same set of people) and they need to combine them, without disclosing any private information, to train a global model that will be useful to all the participants.
- Federated Transfer Learning - FTL: feature, label and sample spaces are not shared among clients; this is used to fine-tune pre-trained models in a distributed way with clients' private datasets when this allows for much better results with respect to standard techniques. It is important to note that in this case the initial model may be trained on similar datasets to the one owned by the participants but with a completely different problem to solve.

It is also worth mentioning a categorization that takes into account data availability and the number of participants in the learning process:

- Cross-Silo Federated Learning: client devices are typically a small number (a few hundred at most), they are indexed and should always be available to participate during the training process; these devices are usually servers with high data storage and computing capabilities (referred as *silos*) owned by organizations which may also choose to deploy custom Federated Learning algorithms to fit their specific needs and constraints, removing, for example, the necessity of a central authority. It is important to note that the training data may still be organized horizontally or vertically as previously presented.
- Cross-Device Federated Learning: clients' count can exceed the thousands of units and they are small devices with limited computing capabilities, such as smartphones, personal computers or other smart objects like IoT devices. In this setting, clients can only process small quantities of data and they may not be available for the entire training process due to unstable internet connection or other intrinsic aspects of their nature; this needs to be taken into account when deploying a Federated Learning process since these characteristics may affect how the training is carried out: for example, we may need Byzantine-robust aggregation mechanisms to cope

with clients' low dependability.

The last classification that should be taken into consideration is based on the logical organization of the devices taking part in the learning process:

- Centralized Federated Learning: the system is organized in a star topology where several decentralized clients are used to train the global model leveraging their private datasets, while a central entity manages and aggregates the received model's updates and eventually performs the final distribution to the participants. This architecture is mostly utilized in cross-device scenarios, that usually involve several heterogeneous devices.
- Decentralized Federated Learning: these architectures deploy a decentralized consensus protocol to aggregate the model's updates coming from every single device, which can only communicate with its neighbors. This approach has several advantages, mainly in cross-silo scenarios or in applications where the devices are highly trusted and can leverage low latency connections, since it removes both the central server, which constitutes a single point of failure, and the communication overhead, needed to interface all the clients with it.

2.3. Weak points and attack surfaces

The complexity of the Federated Learning framework and the number of different participants involved in the process introduces a lot of possible attack surfaces that can be exploited by adversarial parties to tamper with the model training and its delivery to the involved devices.
The following is a list of the most relevant ones exploited in the analyzed papers:

- Training data: being the training data not often subjected to validation by a single authority, an adversarial party controlling one or more clients or their data sources may try to modify the used samples to manipulate the training process into producing a modified global model or preventing its convergence.
- Participants: an adversary controlling one or more clients may try to modify the updates computed locally or craft them entirely to mimic legitimate ones, aiming to poison the global model or receive it without providing any meaningful contribution to the training process. Clients can also be exploited to infer private information owned by benevolent participants.
- Central server: Federated Learning architectures involving a single central coordi-

nator are subject to a single point of failure which could give an attacker complete control over the learning process. An adversarial party controlling the server may not only be able to tamper with the produced global model, modifying it, but could also target clients' updates trying to infer information about a specific participant's dataset.

- Communication between parties: the communications taking place during the Federated Learning process may be subject to eavesdropping or tampering of the exchanged information with the objective of overcoming possible defenses implemented by the participants.
- Aggregation algorithm: being the aggregation algorithm the central part of the Federated Learning framework, its vulnerabilities can be exploited by attackers to tamper with the entire learning process.

3 | Attacks against Federated Learning - Current state-of-the-art

The amount of attack surfaces exposed by the Federated Learning paradigm allows the development of various adversarial techniques with different objectives and peculiarities; to properly classify and compare them against each other a comprehensive list of parameters is needed.
What follows is a description of each one of them and of the possible values they can take.

- Federated Learning type: important to understand how the considered system is constituted; for example, in horizontal Federated Learning scenarios there are commonly several clients which can be targeted by an adversarial party usually aiming to backdoor the global model, to obtain it without contributing to the process or to gain some sort of information about the participants' datasets.
  On the other hand, in vertical Federated Learning scenarios the number of participants is way lower, even as low as two, and the attack's goal is usually to infer data owned by a specific client.
  - Horizontal Federated Learning - HFL: clients' datasets share the same feature and label spaces while having different sample spaces.
  - Vertical Federated Learning - VFL: clients' datasets share the same sample space but have different feature and label spaces.
  - Federated Transfer Learning - FTL: clients' datasets have different feature, label and sample spaces.
- Type of interaction: helpful to understand how the adversarial party interacts with the system and which capabilities are needed to carry out the attack.
  - Active: the adversary will modify the global model by injecting maliciously crafted updates to reach its goal or to amplify the effect of the attack.

  - Passive: the adversary will only analyze the shared updates (from the global model and/or the single clients, if it is capable of doing so) or the shared global model, without modifying it or interfering with the learning process.

- Attacker party: represents which participant of the Federated Learning system is partially or completely controlled by the attacker.

  - Client: the attacker is able to control one or more clients independently or to coordinate a set of them, orchestrating a more structured attack (sibyls attack).

  - Server: the attacker is able to control the central server and, therefore, can also interfere with every single client's learning process by targeting specific updates.

- Model knowledge: important to understand how much information the attacker has about the used model; it is particularly relevant in inference attack scenarios where the adversarial party can take advantage of any prior knowledge to make the attack more efficient.

  - White-box: the attacker has full knowledge about the model, including not only its structure but also its parameters and outputs.

  - Black-box: the attacker is only able to query the model, without having any information about its architecture or parameters, usually using predefined queries exposed by some kind of API.

  - Gray-box: similar to the black-box scenario, the attacker can access the model with predefined queries while also having partial knowledge of it.

- Data knowledge: helpful to understand if the attacker has any insights or access to the participants' datasets; for example, it may have the ability to sample data from them or to know their underlying data distribution.

  - None: the attacker knows nothing about the clients' datasets nor has access to them.

  - Partial knowledge: the attacker only has partial pieces of information about clients' datasets; for example, it can access some samples, understand the data distribution or some other characteristics that may turn useful for the attack's success.

  - Knowledge of compromised devices' datasets: the attacker can access the compromised devices' datasets without any limitation, understanding underlying

  characteristics of the used data that can be useful also to make assumptions about legitimate participants' data. This is usually the case whenever an adversarial party controls a client in its entirety.
  - Full knowledge: the attacker knows the data distribution and can access the samples of each client's dataset, even without necessarily having access or control of those devices. This is clearly an unrealistic scenario that gives the attacker a lot of power.

- Phase in which the attack takes place: represents the phase of the Federated Learning process in which the attack is executed.
  - Training time: the attack is carried out during the training phase of the global model; this also comprises all those scenarios where the attacker injects malicious samples into the clients' datasets before the actual training begins.
  - Inference time: the attack is carried out after the model's training ends; this kind of attack is usually studied in a more generic machine learning setting but is possible to find some applications even in the Federated Learning one.

- Number of interactions needed: helpful to understand how many interactions with the Federated Learning process the attacker needs to reach its goal. This is strictly related to how easy it could be for a training time poisoning attack to be successful since not all the clients are selected at each round to update the global model.
  - One-shot: the attacker only needs one interaction with the system to reach its goal, leading to a more effective attack which, on the other hand, may also be easier to detect since it usually involves a bigger perturbation of the global model.
  - Multiple interactions: the attacker needs more than one interaction with the system to reach its goal, facing problems like the degradation of the injected modifications or the possibility of not being selected for enough training rounds.

- Type of attack: represents the objective of the attack carried out by the adversary; further subclasses are proposed and better described in the following sections.
  - Free-rider attacks: the adversarial goal is to get the global model without actually participating in the training phase, either because it does not own any meaningful data or because it does not have enough computing power.
  - Model poisoning: the adversarial goal is to poison the global model, either to prevent its convergence or to modify its behavior during the inference phase.

    For example, a backdoor can be inserted in the model to make it misclassify some samples when they contain some predefined characteristics, called triggers, or belong to a given class.

  - Inference: the adversarial goal is to infer some kind of information about the participants' datasets, without actually having access to them nor having much information related to them to begin with.

- The last parameters resume the datasets used to test the proposed attacks and if they are publicly available or not; this information is important to understand if the results shown in the considered papers can be compared against each other or not and if they are replicable. Moreover, i.i.d. or non-i.i.d. assumptions on the considered data are noted.

3.1. Free-rider attacks

Free-rider attackers can be defined as adversarial individuals whose goal is to benefit from the Federated Learning process without having any useful data nor enough computational resources to be able to contribute to it, stealing the trained global model which may have high commercial or intellectual value. This clearly compromises the fairness of the learning mechanism which uses the global model as a reward or incentive for the participants to share their resources during the training process.

In this kind of scenario, the attacker wants to avoid any possible interference with the learning procedure to not be detected as an outlier and, most importantly, to obtain the best possible global model produced by legitimate participants.

3.1.1. Plain free-rider attacks

This is the simplest of the analyzed techniques where malicious clients return the same model parameters received during the training process without altering them in any way, as discussed in [8]; while not modifying these values ensures that the global model converges to the best possible one, it is quite easy to detect devices implementing this kind of attack by verifying that they are not contributing to the current global model in any meaningful way (i.e., the difference between subsequent updates is zero).

3.1.2. Random weights attacks

Attackers implementing this technique, proposed in [17], return arbitrarily generated updates by copying the received global model ones and replacing some of their components with random values; the main challenge in this scenario is to craft realistic parameters that mimic real ones, without having access to a legitimate dataset or any other participant's updates. This technique has been shown to be effective against autoencoder detection, although it can be detected with DAGMM - Deep Autoencoding Gaussian Mixture Model - method, whether each client has a similar local data distribution or not.

Another similar approach is described in [8] and consists in adding Gaussian white noise to the received global updates, tuning it to adopt a noise structure similar to one of the fair clients and to be also time-varying to produce more credible updates; this approach has been tested to prove it can make the global model converge but no data about its effectiveness against any defense has been provided.

3.1.3. Delta weights attacks

Delta-weights and advanced delta-weights attacks, proposed in [17], are the most advanced techniques regarding free-rider scenarios where honest clients' behavior is reproduced using the last two global model updates, either by simply exploiting their plain difference or by adding to it a Gaussian noise term. While these methods are resilient against DAGMM defense, they are not effective against the proposed STD-DAGMM - Standard Deviation Deep Autoencoding Gaussian Mixture Model - one.

A similar technique that also implements a decay factor, represented by the l_2 norm of the previous two updates with respect to the ones considered in the difference, to simulate the local updates' decadence to zero as the learning process converges, is discussed in [51]. Moreover, also in this case an advanced version of the attack with the addition of a Gaussian noise term is proposed; effectiveness against DAGMM is proven but no other defense technique has been tested.

3.1.4. Free-rider attacks SoK

In Table 3.1 the categorization of the considered free-rider attacks is proposed; it highlights how all of them share the same main features, which are expected given the setting and the objective they are studied for: every technique is developed considering a passive scenario where the adversary controls only client devices with no access to the data used by legitimate participants. Moreover, all these attacks are carried out at training time. It is also worth noting how all these techniques take into consideration horizontal Federated Learning scenarios, where it may be simpler for a non-contributing device to go undetected by a central aggregator server deploying a secure aggregation mechanism.

Table 3.1: Free-rider attacks

	FL type		Type of interaction		Attacker		Model knowledge		Data knowledge				Phase			Datasets			
	VFL	**HFL**	**active**	**passive**	**client**	**server**	**white box**	**black box**	**none**	**partial**	**comp. devices**	**full**	**training time**	**inference time**	**free-rider**	**dataset(s)**	**public**	**private**	**data distribution**
[8]		•		•	•			•	•				•		•	MNIST, CIFAR-10M, Shakespeare	•		i.i.d. / non-i.i.d.
[17]		•		•	•			•	•				•		•	MNIST	•		i.i.d. / non-i.i.d.
[51]		•		•	•			•	•				•		•	MNIST, Fashion-MNIST	•		i.i.d. / non-i.i.d.

3.2. Poisoning attacks

Poisoning attacks are meant to interfere with the Federated Learning process by trying to modify the learned global model; this could be done in a targeted way, trying to leave the overall performance of the learned model unaffected while introducing a predefined perturbation that pursues a given goal, or randomly (untargeted), aiming at reducing the general global model's accuracy or directly preventing its convergence wasting other participants' resources.

Usually, this kind of attack is carried out by the clients and, intuitively, the proportion between compromised and uncompromised participants plays a key role in the attack's effectiveness due to how the clients are randomly selected to participate in the training process of Federated Learning systems.

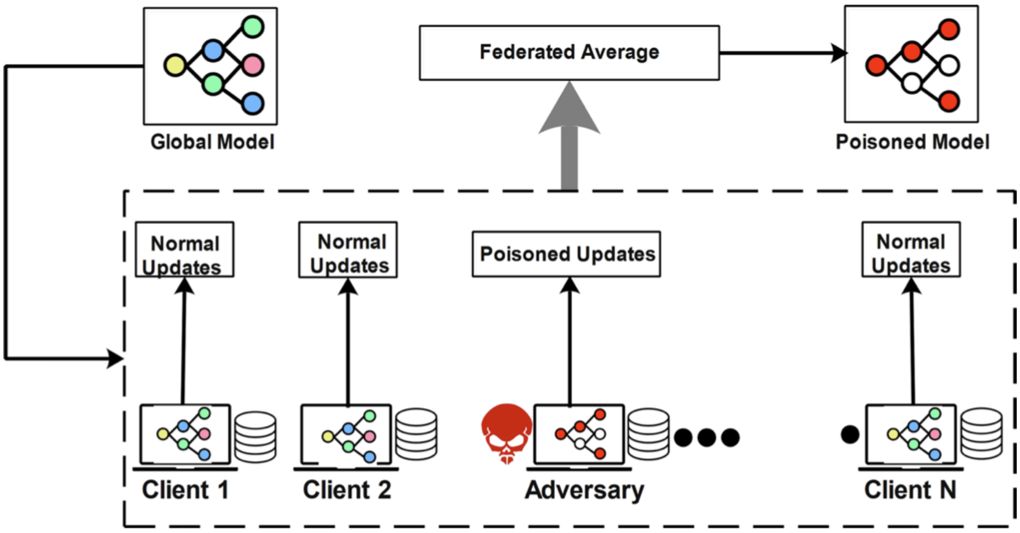

Figure 3.1: Graphical representation of poisoning attacks. Image taken from [29]

3.2.1. Data poisoning attacks

Data poisoning attacks try to effectively alter the dataset used by one or more clients during the training phase to be able to modify the global model or prevent the learning process to converge [10]; this can be done by either directly altering the data samples owned by the participants or by controlling their data sources.

They can be classified into two main categories:

- Dirty-label attacks: the adversary can directly modify the data and is also able to change the samples' labels.

- Clean-label attacks: the adversary can only partially modify the training data, poisoning the samples by adding noise or other patterns, but cannot change any label. This kind of technique can be adopted in scenarios where, for example, a dataset validation process takes place and therefore labels cannot be modified without it being noticed.

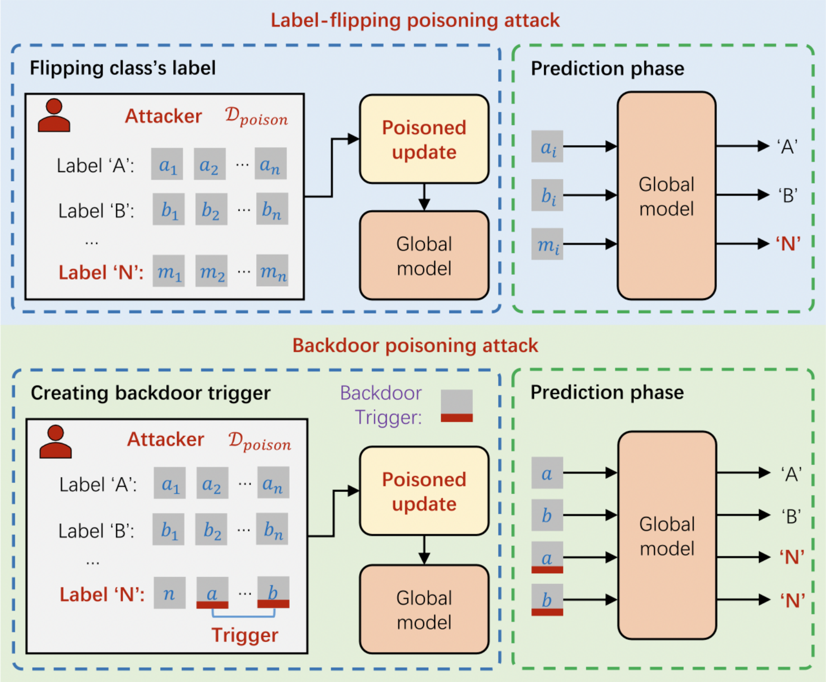
Label-flipping poisoning attack
Flipping class's label
Attacker $\mathcal{D}_{poison}$
Label 'A': a_1 a_2 ··· a_n
Label 'B': b_1 b_2 ··· b_n
...
Label 'N': m_1 m_2 ··· m_n
Poisoned update
Global model
Prediction phase
a_i
b_i
m_i
Global model
'A'
'B'
'N'
Backdoor poisoning attack
Creating backdoor trigger
Backdoor Trigger:
Attacker $\mathcal{D}_{poison}$
Label 'A': a_1 a_2 ··· a_n
Label 'B': b_1 b_2 ··· b_n
...
Label 'N': n a ··· b
Trigger
Poisoned update
Global model
Prediction phase
a
b
a
b
Global model
'A'
'B'
'N'
'N'

Figure 3.2: Graphical representation of clean and dirty-label poisoning attacks. Image taken from [46]

Most of the analyzed attacks belonging to this category are implemented in a horizontal Federated Learning setting and can be classified as label-flipping attacks (dirty-label scenario), where the training sets samples' labels are changed from their true value to another one to reach the attack's goal.

Simply exchanging the labels of a source class into ones of another target class, as proposed in [35], can effectively lead the global model to misclassify the samples belonging to the source class. In the analyzed paper it is highlighted how this kind of attack is not only easy to perform and energy-efficient, but also does not require the knowledge of the global data's distribution among participants, the architecture of the model or any other characteristic of the overall system. A defense named PCA is also proposed to protect against it and is shown to be effective.

A similar approach is considered in [34], where besides the possibility for an adversary to train and inject into the learning process a backdoored model that misclassifies a chosen category of samples, it is also shown how a boosting technique can be implemented to optimize the outcome of the attack; this is done by trying to perform model replacement in order to be able to amplify, as much as possible, the effect that the poisoned updates have on the global model during every single interaction with the attacker.

This paper also discusses how to norm-bound the generated updates to avoid them being marked as malicious using norm thresholding techniques; it is then shown how weak differential privacy is effective in reducing the impact of such attacks by adding a certain quantity of Gaussian noise to clients' updates, partially disrupting the effect of the attack.

Label flipping can be also used to perform what is called an edge-case backdoor attack, as shown in [37], where the adversary focuses on the input data points that are rare and/or underrepresented in fair participants' training data. While this technique can achieve better results than traditional data poisoning and is also proven to be resilient against defenses like NDC and RFA, it can be effectively prevented by adopting KRUM and Multi-KRUM aggregation mechanisms.

A similar concept to the ones mentioned above has been explored in [27]: the poisoning attack is in this case implemented by injecting maliciously crafted data samples into the training dataset instead of flipping the labels of already existing ones; this technique has been proposed considering the IoT setting, where data is not ready beforehand and is collected from local devices by a gateway that will also train the local model. Therefore, it may be interesting to consider this approach in all those scenarios where the used data is generated from external sources during the training process.

The focus of the described technique is to avoid the injected traffic to be detected as malevolent and avoid the poisoned global model from deviating too much from what would be its optimal performance level. Moreover, this is done by only gaining control of the devices generating the used samples while leaving the ones actually training the local models unaffected.

GAN networks can also be deployed to implement data poisoning attacks generating malicious data, as studied in [45] and [46]: the adversary initially acts like a benign participant and trains a generative adversarial network - composed of two separate models, a generator and a discriminator - to generate samples that mimic legitimate ones using as discriminator the shared models, basically trying to reproduce benign clients' private datasets; after this initial phase, the wrong labels are attached to the crafted data and the generated gradients are shared with the central server to poison the global model.

In order for the attack to be successful, the crafted updates need to be amplified using a scaling factor that will allow it to survive the averaging phase implemented by the central aggregator. For both papers, the attacks have been shown to be able to successfully affect the model reaching a good accuracy on the poisoning task. Some defense techniques have been cited but no experimental result on their effectiveness is proposed.

Clean-label data poisoning attack approaches usually allow the backdooring of the global model by inserting triggers into the training samples; these are patterns embedded into the data that will activate a backdoor in the model when present.
In [40] it is shown how a trigger can be decomposed into local patterns distributed to multiple adversarial parties: each one of them will train its local model independently, without knowing the global trigger, and this allows the attack to be undetected by robust aggregation mechanisms or clustering-based anomaly detection techniques. It is also shown how it is more effective in pursuing the adversarial goal with respect to standard backdooring attacks.
Clean-label techniques can also be used to create a covert channel to utilize the Federated Learning system as a stealth communication infrastructure to transmit single bits of data, as shown in [6]; in this case, the effect of the model's poisoning is not visible to the benign participants and, moreover, the overall performance level of the model is not affected.
The key concept behind this attack consists in training the adversarial clients' local models with malicious samples that are able to induce a perturbation in the global model which can be tested by other malevolent participants; by using it to make predictions on the same set of samples and interpreting the outcome, it is possible to deduce whether the transmitted bit is either a 0 or a 1. It should be noted how the process requires a calibration phase during which the receiver observes the global model updates and computes the channel parameters - like the number of training rounds to transmit a single bit or the crafted samples to be used - that are needed to implement the communication channel and are required to be shared with the sender (hard coding them before deploying the sender or transmitting them through a secondary channel). It is also discussed how to implement multiple parallel communication channels, although this may cause interferences and poor performances in general.

3.2.2. Model poisoning attacks

Model poisoning attacks aim to modify the local model's updates produced by one or more clients, either controlling them singularly or coordinating them, before sending them to the central server to induce a predefined effect on the generated global model.
While this technique is more complex to implement compared to data poisoning ones and requires deeper access to the learning process and the participating systems in general, it gives the attacker more control over the adversarial objective and allows for better results,

especially in Byzantine-robust Federated Learning systems.
The main challenges faced by the adversarial parties are related to the updates' crafting process to prevent them from being detected as malevolent and rejected by a robust aggregation mechanism.

One of the possible objectives of a model poisoning attack is increasing the global model's error rate and slowing down its convergence (untargeted attack): in [7] it is studied how to craft local updates on the compromised clients to deviate the aggregated global model towards the inverse of the update direction it would normally follow without any adversarial interaction.
The updates' creation is formulated as an optimization problem to be solved at each iteration of the learning process where the objective is maximizing the deviation obtained from the standard model; the attack is shown to be successful in both full and partial knowledge scenarios - with the only difference being the knowledge of the local models and datasets of other clients - against Krum, Trimmed-Mean and Median based aggregation techniques, with the last two shown to be more robust against the attack. In any case, it is important to note that the full knowledge setting may have limited applicability in real scenarios.
Explicit boosting may be used also in this scenario and in similar ones, like targeted misclassification, to be able to influence the global model as much as possible in each round where the adversary is selected. As shown in [3], this is done by multiplying the crafted weights' updates by a given factor to increase the perturbation of the global model, allowing the poisoning effect to better survive the averaging phase done by the coordinator.
It is also analyzed how to avoid the updates being detected as malevolent by the central server: it is proposed to add two terms to the objective function of the considered optimization problem to make the generated weights as close to real ones as possible; one will take into account the accuracy on the validation data, which can be accounted for using the training data loss, and the other will consider some weights statistics to limit the distance between crafted and real ones (l_2 norm is used as an example). This should be enough in real applications to mislead the aggregator mechanism into classifying crafted weights as legitimate. The proposed technique is again proven to be effective and well-performing in systems using Krum and Median based aggregation algorithms.
Assuming that the parameters produced by all the participants involved in the learning process are i.i.d. and therefore expressible by a normal distribution, which may be a strong assumption to fulfill in real scenarios, it is possible to obtain a range in which the values of the weights can be crafted to fool the system into classifying them as benign. As proposed in [2], this is done by taking into consideration all the values between the mean

of all the updates and the ones of the honest clients that are pushing in the direction the adversary needs to pursue its goal.
This concept is crucial to hide Byzantine workers' updates within the variance of the benign ones, while still being able to reach the attack's objective which may be backdooring the model or preventing its convergence; an optimization process similar to the one adopted by the above attacks is still involved to produce the desired updates.
Similar considerations are made in [32], where the adversary computes a benign reference weight vector using some updates it observed and, then, crafts a malicious perturbation in the opposite direction; particular attention is given to selecting the most appropriate perturbation vector for the given Federated Learning setting to maximize the attack's impact allowing it to go undetected from robust aggregation mechanisms. Different techniques to compute the vector are discussed (inverse unit vector, inverse standard deviation and inverse sign) and is also discussed how to tackle the most used aggregation mechanisms and how to deal with the case of the adversarial not knowing which algorithm is used by the central server. The attacks are shown to achieve good performances in the conducted tests, even if it should be considered that knowing the used aggregation mechanism in a Federated Learning system may not be trivial. A dimensionality reduction defense using random sampling followed by outliers removal is proposed to defend against the detailed technique.

An interesting approach that avoids compromising the global model's performance while causing target misclassification is analyzed in [49]; since only small sets of neurons are activated and used during the model's training phase, it is proposed to inject adversarial neurons, crafted to accomplish the attacker's objective, into the redundant space of the neural network: this allows to leave the other useful neurons, used for the primary task, untouched avoiding the degradation of the model's performance. An optimization process is still involved, composed of the main task and the adversarial one that has the goal of understanding which are the unused neural paths that can be exploited for the attack.
Both single-shot and multiple-shot versions of the attack are tested and shown to be effective and more persistent than normal attacks. It is important to note that the malicious client is assumed to be chosen at each round, which may be an unrealistic scenario considering normal Federated Learning systems.

3.2.3. Poisoning attacks SoK

In Table 3.2 the categorization of the considered poisoning attacks is presented; it helps to visualize how all the analyzed techniques are meant to be applied in a horizontal Federated Learning scenario where the adversary controls one or more clients and deploys an active attack during the learning process, which is expected given that the final objective is to modify the global model. Most of the studied attacks are applied in a white-box scenario, requiring deeper access to the model trained by the adversarial clients; the majority of them also only requires access to the data belonging to the controlled devices and needs more than one training round to reach the final objective, which is found to be one of the most common limitations given how the training process works in HFL systems.

It is important to note that there are still some data-poisoning attacks that can be carried out in a black-box scenario; moreover, some considered techniques are able to affect the global model with just one interaction, creating more powerful attacks which on the other hand face and higher probability of being detected and prevented by the central aggregator.

In conclusion, it can be highlighted how most of the included attacks are targeted with only a few model-poisoning ones aiming at reducing the global model's accuracy or preventing its convergence.

Table 3.2: Poisoning attacks

	FL type		Type of interaction		Attacker		Model knowledge		Data knowledge				Phase		Number of interactions							Datasets			
	VFL	**HFL**	**active**	**passive**	**client**	**server**	**white box**	**black box**	**none**	**partial**	**comp. devices**	**full**	**training time**	**inference time**	**one shot**	**multiple**	**targeted**	**untargeted**	**dirty label**	**clean label**	**optimization**	**dataset(s)**	**public**	**private**	**data distribution**
[7]		•	•		•		•				•		•			•		•			•	MNIST, Fashion-MNIST, CH-MNIST, Breast Cancer Wisconsin Dataset	•		Non-i.i.d.
[3]		•	•		•		•				•		•			•	•				•	Fashion-MNIST, UCI Adult Census	•		i.i.d.
[2]		•	•		•		•				•		•			•	•	•			•	MNIST, CIFAR-10, CIFAR-100	•		i.i.d.
[32]		•	•		•		•					•	•			•		•			•	MNIST, CIFAR-10, Purchase, FEMNIST			i.i.d.
[49]		•	•		•		•				•		•		•	•	•				•	MNIST, CIFAR-100	•		Non-i.i.d.
[1]		•	•		•		•				•		•		•		•			•		CIFAR-10, Reddit	•		Non-i.i.d.
[37]		•	•		•		•	•			•		•			•	•		•		•	CIFAR10, ImageNet, EMNIST, Reddit, Sentiment140	•		
[34]		•	•		•		•	•				•	•		•		•		•			EMNIST	•		Non-i.i.d.
[42]		•	•		•		•				•		•			•	•			•		TensorFlow Fed., FATE	•		
[35]		•	•		•		•				•		•			•	•		•			CIFAR-10, Fashion-MNIST	•		i.i.d.
[27]		•	•		•		•				•		•			•	•		•			DIoT-Benign, DIoT-Attack, UNSW-Benign		•	
[10]		•	•		•			•			•		•			•	•		•			MNIST, VGGFace2, KDDCup, Amazon	•		i.i.d. / non-i.i.d.
[45]		•	•		•		•				•		•			•	•			•		MNIST, AT&T	•		
[46]		•	•		•		•				•		•		•		•			•		MNIST, Fashion-MNIST, CIFAR-10	•		Non-i.i.d.
[40]		•	•		•		•				•		•			•	•			•		LOAN, MNIST, CIFAR-10, Tiny-imagenet	•		Non-i.i.d.
[6]		•	•		•			•			•		•			•	•			•		MNIST	•		
[23]		•	•		•		•					•	•			•	•		•						

3.3. Inference attacks

Inference attacks' main goal is to compromise the privacy characteristics and guarantees of the Federated Learning paradigm, enabling the attacker to reconstruct some information or even some samples from clients' private datasets that would otherwise be inaccessible to third parties. This is made possible thanks to the analyses of the updates shared by the participants during the training phase of the Federated Learning process or by querying the final shared model.

The attacks falling in this category are worth being analyzed under both HFL and VFL settings: based on the case we are in, the attacker may leverage different peculiarities of the training process to reach its goal, either by controlling one or more clients or the central server.

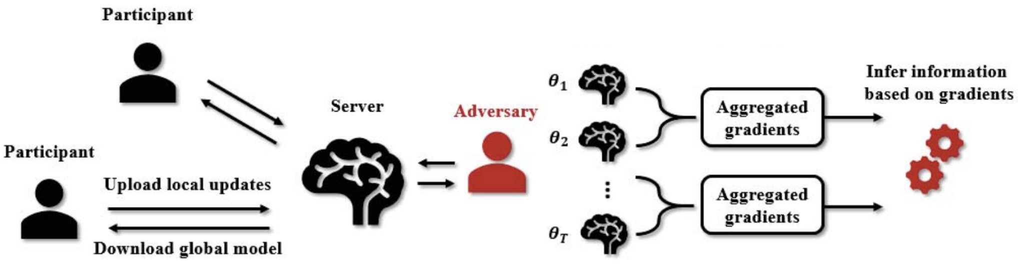

Figure 3.3: Graphical representation of inference attacks leveraging the shared gradients. Image taken from [24]

3.3.1. Features inference attacks

Feature inference attacks aim at reconstructing part of the datasets of the participants in the Federated Learning process, generating samples that are likely to have been used by the clients to train their local models; while this, in most cases, will not recover the exact same samples used, it may be useful to infer some particular information or general characteristics about them.

Gradient inversion can be used to uncover private users' data from the parameters' gradients: in [11] it is shown how an honest-but-curious server may be able to use a numerical reconstruction method to implement a multi-image recovery starting from the gradients received from the clients. The described technique consists in implementing an optimization process that maximizes the similarity between the received gradients and the ones generated by the considered possible inputs, iteratively creating more realistic data.

In this study the analyzed gradients are the result of local training on multiple images;

the server only needs to know the number of participating samples, even though this can be inferred by running the reconstruction process over a range of candidate numbers, choosing the one that leads to the smallest error. While with this technique most of the recovered images will be non-recognizable, a few samples may be recovered with enough precision to reveal important information about the used datasets.

A completely different approach is discussed in [13] where the adversary pretends to be an honest participant in the Federated Learning process while also deploying a generative adversarial network - GAN - to generate samples of given classes using the shared models as the discriminators of the network; given that the attacker needs to know the data labels of other participants, it will also influence their training phase by sharing specially-crafted gradients to trick them into leaking more information about their local samples.

This attack is shown to be effective against differential privacy and other obfuscation algorithms, being able to recover better and clearer samples than model inversion techniques that tend to be able to output only prototypical examples of the real data.

A similar attack, where the adversary controls the server instead of the clients, is analyzed in [39]: also in this case the attacker deploys a GAN to recover the data samples and it will target a single specific client; the model updates coming from the victim are used to train the GAN into generating more specific samples.

With respect to similar approaches, in this case, the attacker also aims at compromising the client-level privacy represented by specific properties, identifying each client, by deploying a multitask discriminator; this is able to not only recover the data used during the training phase but also to associate it with a specific participant of the learning process.

It is important to note that the server needs to have samples representative of the targeted client's dataset, which can be recovered starting from a testing set - usually provided to the central server - and the updates received from the participant.

Moreover, this attack is both proposed in a passive setting, where the attacker does not interfere with the learning procedure, and in an active one, where the server isolates the victim sending it a model which will not be shared with anyone else; in both cases, the attack does not interfere with the global training procedure and it is shown to produce more accurate results compared to model inversion techniques. No defenses are analyzed.

While the above-mentioned attacks are all related to the horizontal Federated Learning settings, some techniques have also been studied to tackle data privacy in the vertical one, where multiple parties share the same sample space but have different label and feature spaces.

Two different attacks have been analyzed in [20] to perform feature inference starting from a single prediction of the produced model: an equality-solving attack is proposed to tackle logistic regression models while a path restriction one is detailed to deal with decision trees. While the former is based on the resolution of a system of equations, the latter sees the adversarial party restricting the possible prediction paths in the tree model based on its own data's predicted classes.

To cope with more complex models, like neural networks or random forests, an attack based on multiple predictions has also been proposed. By relying on a set of predictions, the generator model can be used to minimize the loss between these values and the ones corresponding to the generated samples. It is worth noting that this attack does not need the adversary to have any background knowledge of the target's data distribution or any intermediate information disclosed during the computation of the ground-truth predictions. Several defenses against these attacks have been discussed; it is highlighted how hiding the generated model, preventing the adversary from having access to it in plaintext, could mitigate this attack as well as the verification of the prediction output of the model would make it possible for legitimate participants to understand if it could leak some information or not. On the other hand, differential privacy is shown to be ineffective.

3.3.2. Labels inference attacks

Label inference attacks' goal is to generate both the samples and the associated labels used during the training phase of the Federated Learning process.

Deep Leakage from Gradients - DLG - is an approach to this kind of attack, analyzed in [50], where the adversary controls the central server, randomly generates a sample-label couple and then computes the derived gradient on the used model; deploying an optimization algorithm, the dummy sample-label pair can be iteratively tweaked to generate a gradient as close as possible to the real one.

It is worth noting that this technique does not rely on any generative network. Moreover, the paper proposes two defenses that are shown to be effective against it: gradient perturbation and gradient compression. The former adds noise to the gradients, while the latter prunes all the gradients with small enough magnitudes.

An improvement over DLG, named iDLG, has been studied in [48]: an additional step has been added to the algorithm above in order to produce better samples, allowing it to first recover the ground-truth labels from the shared gradients and then to optimize a random sample, as seen in the base version of the attack. This is proven not only to be able to extract the exact label for every sample, which is a problem DLG faces, but also to significantly reduce the mean square error on the tested datasets compared to the previous attack.

The adversary can take advantage of a GAN also in this kind of attack, as shown in [30], where the attacker deploys two separate networks: a generative adversarial network to generate images using the shared model as a discriminator and a simple fully-connected layer to produce the labels associated to the considered samples. The generated couples are then used to compute the gradients given the current global model and the whole architecture is trained to minimize the distance between the fake gradients and the shared ones. Based on the tests run by the authors of the paper, this technique can achieve better results when compared to DLG and is also proven to be at least partially resilient to the addition of noise to gradients shared by each client.

Taking into consideration specifically the vertical Federated Learning scenario, a two phases attack is discussed in [15]. In the first phase, the inputs of the used model's first fully-connected layers are recovered and then, in the second step, they are used as a regularizer to improve the results of the optimization process, where random sample-label couples are iteratively improved to reduce the reconstruction error. While this attack can be defeated by clients uploading fake gradients as a defense mechanism, it is shown to be more effective and provide better overall results compared to DLG and iDLG techniques overcoming the batch limitation problem affecting other attacks.
Moreover, a gradient inversion technique adopting homomorphic encryption is discussed in [19]: the adversarial party sets up an internal model aiming at guessing the labels by training it to reduce the distance between the gradients computed from the generated labels and the ones received during the Federated Learning process. This technique is shown to be very effective even if the used gradients are batch-averaged, which is commonly associated with a greater security level; a defense technique, where the true labels are transformed into fake ones using a so-called confusional autoencoder - CoAE -, is proposed and proven to be effective against the studied attack.

3.3.3. Membership inference attacks

Membership inference attacks try to infer partial knowledge about clients' private data involved in the Federated Learning process, giving the adversarial the ability to tell whether one or more samples were used during the training phase.
This category of attacks can take advantage of generative adversarial networks, as described in [47], creating fake samples that have the same distribution as the ones in the legitimate participants' training datasets by using the aggregated model as a discriminator; this data is used to enrich the one the attacker has access to in order to train a binary classification model. This is then used to infer the membership status of the samples based on the label predicted by the global model.

Thanks to the augmentation of the used data's diversity, this attack is shown to be very effective on the tested datasets and to keep a good accuracy even when the global model is not overfitted - a characteristic that would make the attack easier.

A slightly richer technique is discussed in [14] where not only the adversary aims to understand if a given sample is part of one of the participants' datasets, but will also try to infer the specific client it belongs to. This is done by controlling the central server and comparing the loss of every participant's shared model on the sample of interest, assuming that the smaller the loss, the higher the probability the sample took part in that specific model's training.

This attack has been proven to be more effective when the training datasets of the participating clients are heterogeneous; it is worth noting that it can be successfully prevented using differential privacy techniques.

Similar concepts are also expressed in [36], where membership inference in the machine learning as a service scenario is discussed: the Federated Learning case is cited as the most vulnerable one since the adversary is not limited to querying the model but can also access the learning process as an insider.

A white-box approach is proposed in [26]: the gradients from different layers of the target model are processed separately and the extracted information is combined, allowing the attacker to compute the membership status of the analyzed data samples through an attack model, composed by feature extraction components followed by an encoder one.

Furthermore, an adversary can take advantage of a sequential learning setting, as described in [28], where each client trains the model on a small subset of its training dataset and then passes it to the next participant without the need for a central server; thanks to the sequential learning setting, an attack model can be used to discriminate if a sample belongs to the training set of a targeted client or not. It is trained using several shadow models replicating a realistic behavior.

The assumption behind this concept is that similar models should behave similarly when the considered data is similar. Therefore, the targeted model can be simulated using other ones trained on datasets that are close to the ones used by legitimate clients; this clearly requires the attacker to have access to realistic datasets. A possible way to defend against this attack consists of adding random noise to the used samples or randomizing the order of nodes in each training cycle, to reduce the amount of information that the adversarial party can obtain.

3.3.4. Properties inference attacks

Properties inference attacks are designed to infer some private information or patterns about the datasets belonging to the clients involved in the training process.
In a black-box scenario, this kind of attack is possible following the technique described in [4], where the adversary can interact with the global model using predefined queries and submitting also some data samples to be used during the learning process. The basic idea is to poison the model in such a way that its behavior depends on the average of the target property; doing so makes the adversary able to infer the property of interest by simply querying the model. This procedure has been developed considering training algorithms that will output Bayes optimal classifiers, which is an assumption that in practice may not always hold.
On the other hand, taking into consideration a white-box scenario, the attack analyzed in [41] aims at recovering some properties of the used data with a passive technique: it uses the intermediate output generated by the global model on the local datasets during each iteration of the Federated Learning process to infer the correlation between the used data and the embedding derived by the intermediate output using a meta-classifier model.
Three different properties inference attacks, built using supervised classification tasks, are discussed in [38] with the goal of demonstrating that this kind of attack can be carried out by adversaries that can only control one legal participant of the Federated Learning process. It is worth noting that the described techniques do not require access to gradient updates from individual clients but need some prior knowledge about the training process, for example, the average number of labels owned by each participant.
The first one is a class sniffing attack, which allows the adversary to infer if a particular class of training data has been used during the learning process. Then a quantity inference attack is proposed to judge how many clients own data with a particular label. The last one, denominated *whole determination*, allows the malicious participant to understand the composition proportion of the labels present in the dataset used to train the global model.
Considering the vertical Federated Learning scenario, in [9] it is explained how one of the parties of a learning process implementing model splitting can exploit its local model to infer the privately owned labels of another participant; this is possible thanks to the fact that the updates shared by the global aggregator help the local models to learn a good feature representation with respect to the labels and can therefore be used in the attack by adding an inference head and tuning it in a semi-supervised manner. It is also discussed how to implement an active attack where the adversary boosts its model's learning rate, tricking the server into relying more on its local model and indirectly gaining

more information about the labels. Defenses like noisy gradients and gradient compression are shown to be effective but, applied with the strength necessary to prevent the attack, they will also degrade the global model's performance.

3.3.5. Inference attacks SoK

In Table 3.1 the categorization of the analyzed inference attacks is presented; it highlights how they are more heterogeneous compared to the other types of attacks: the only common features that can be underlined are that most of them are carried out during the training phase and that in many cases it is required to have at least a partial knowledge about the data owned by the participating devices.

Both the vertical and the horizontal Federated Learning scenarios are equally considered and most of the proposed techniques operate passively, with only a minority of them including an active phase to produce better results by making the model leak more information. It may also be underlined how features and label inference attacks mainly operate in a black-box scenario and at a server level, while membership and properties inference ones mostly require white-box access to the model and are carried out by malicious clients.

Table 3.3: Inference attacks

	FL type		Type of interaction		Attacker		Model knowledge			Data knowledge			Phase						Datasets			
	VFL	**HFL**	**active**	**passive**	**client**	**server**	**white box**	**black box**	**none**	**partial**	**comp. devices**	**full**	**training time**	**inference time**	**features**	**labels**	**membership**	**properties**	**dataset(s)**	**public**	**private**	**data distribution**
[11]		•		•		•		•	•				•		•				CIFAR-10	•		
[13]		•	•		•		•				•		•		•				CIFAR-10	•		
[20]	•			•	•			•			•			•	•				bank mark. dataset, credit card dataset, drive diagnosis dataset, news pop. dataset		•	
[39]		•	•	•		•		•	•				•		•				MNIST, AT&T	•		Non-i.i.d.
[43]		•		•	•		•				•		•		•				Penn Treebank, WikiText-2, Enwik8	•		
[15]	•			•		•		•		•			•			•			CIFAR-10, Yale human faces	•		
[19]	•			•	•			•			•		•			•			NUS-WIDE MNIST, CIFAR-10, CIFAR-100	•		
[50]	•	•		•		•		•	•				•			•			MNIST, CIFAR-100, SVHN, LFW	•		
[30]	•	•		•		•		•	•				•			•			MNIST, CIFAR-100, LFW	•		
[48]	•	•		•		•		•	•				•			•			MNIST, CIFAR-100, LFW	•		
[24]		•	•	•	•			•		•			•				•	•	LFW, FaceScrub, PIPA, Yelp-health, Yelp-author, FourSquare, CLiPS SI	•		
[14]		•		•		•		•		•			•				•		Synthetic, Location, Purchase, CHMNIST, MNIST, CIFAR-10	•		Non-i.i.d
[47]		•		•	•		•				•		•				•		MNIST	•		
[36]		•		•	•		•				•			•			•		Adult, MNIST, CIFAR-10, Purchase-[10, 20, 50, 100]	•		
[26]		•	•	•	•	•	•				•		•				•		CIFAR-100, Purchase100, Texas100	•		
[5]		•		•	•		•				•		•				•		MNIST, CIFAR-10	•		
[28]		•		•	•		•			•			•				•		Purchases	•		
[16]	•			•	•		•				•		•				•		Criteo, Avazu, ISIC	•		
[4]		•	•		•			•			•		•					•	Cnesus, Enron	•		
[9]	•		•	•	•		•				•		•					•	CIFAR-10, CIFAR-100, CINIC-10, Yahoo Answers, Criteo, BHI	•		
[38]		•	•		•		•				•		•					•	MNIST, CIFAR-10, Fer2013, HAM10000	•		
[41]		•	•	•	•		•				•		•					•	CelabA, LFW	•		

4 | Discussion on future possible directions of work

The analysis of the current state-of-the-art of adversarial attacks against Federated Learning systems highlights some possible promising directions that may be worth considering, either to improve the current adversarial techniques or to create more secure systems. What follows is a list of research paths that have not been explored in detail in the analyzed papers.

- Most of the analyzed attacks take into consideration the horizontal Federated Learning scenario, while the vertical and transfer ones are not considered as much. In particular, I think that attacks regarding the VFL setting can be further developed considering its peculiarities: for example, they may take advantage of the fact that the participants are much more likely to be selected at each training round compared to normal HFL scenarios, removing the need of deploying boosting techniques that can make the adversary easier to detect during the aggregation phase.
  Moreover, it might be interesting to consider a scenario where one of the participants in the learning process tries to implement a free-rider attack; given VFL systems' structure and functioning, this may be a very difficult attack to carry out but, at the same time, it may pose a serious threat for all those environments where the created model has a high economical or strategical value.

- Decentralized Federated Learning attacks have not been studied as much as the ones for systems that include a central server, probably due to their lower adoption rate; it may be useful to analyze if the techniques proposed to tackle with the most common Federated Learning settings can be applied or adapted to work with fully decentralized systems, taking advantage of their specific characteristics.
  This could be an interesting subject to delve into since this kind of paradigm can be adopted by organizations that may prefer not to rely on a single central aggregator, either because they need to handle very sensitive data or they want to leverage low-latency communications to maintain a certain level of performance during the

training phase. Tampering with this kind of process may produce huge damages to such organizations.

- Free-rider attacks are less discussed even though they pose a serious threat to Federated Learning systems: if implemented correctly, they allow an adversarial party to take possession of the produced global model which may have been trained using valuable datasets or computing resources.
  Also, the analyzed works do not discuss the possible benefits of a generative adversarial network approach: for instance, an attacker could deploy a GAN to generate samples, following the principles exposed in other works, that can be used to craft realistic updates. From my intuition, this would reduce the possibility of being flagged as a non-contributing client while also avoiding the degradation of the global model's performances and, therefore, it could be an interesting path to explore.

- Most of the studied inference attacks carried out at a server level are much more effective and powerful than the ones taking into consideration a client adversarial device. Since being in control of the central aggregator may be a very difficult assumption to fulfill in real scenarios, where these parties are usually controlled by organizations that may put in place robust defenses to create secure systems, it may be worth exploring more powerful techniques deployed at a client level that may find more practical applications.

5 | Comparison with other surveys

To understand the main advantages of this thesis, Table 5.1 provides a comparison with similar works presented over the last few years, highlighting their main characteristics and strengths. It is composed as follows:

- Brief description: a brief description of the analyzed survey, defining its main features and structure.
- Introduction to FL: takes as possible values *Brief*, *Partial* and *Complete* based on how complete the introduction to the Federated Learning framework and the concepts needed to understand the rest of the survey is.
- Complete overview: checked if a complete overview of the possible attacks' categories is given and some attacks are considered for each one of them.
- Comparison: checked if different attacks are compared to each other using a table or other methods to intuitively show their main differences.
- Weak points: checked if the attacks' weak points are discussed.
- Datasets: checked if the datasets used in the analyzed works are mentioned.

It is possible to note how all the analyzed surveys lack at least one of the features described above, leading to an incomplete overview of the state-of-the-art of adversarial attacks against Federated Learning systems; moreover, many papers do not discuss free-rider techniques, that represent an important threat in certain scenarios, and do not include suggestions for future developments, which may be useful to summarize the weaknesses found in the analyzed attacks and guide possible new studies.

Overall the biggest advantage of this thesis is the presence of a comparison table and the complete analysis of the included works: these characteristics allow not only to understand which are the main existing techniques and how they are implemented but also to compare them against each other considering also the datasets used during their development.

Table 5.1: Comparison against other similar surveys

Surveys	Brief description	Introduction to FL	Description of analysed works			
			Complete overview	Comparison	Weak points	Datasets
[31] - Survey on Federated Learning Threats: Concepts, Taxonomy on Attacks and Defenses, Experimental Study and Challenges	Adversarial techniques are analysed based on the following criteria: attack time, objective of the attack, poisoned part of the FL scheme, frequency of the attack (one-shot or multiple), feature inference, membership inference, property inference	Brief				•
[21] - Threats to Federated Learning: a Survey	Adversarial techniques are divided into poisoning and inference ones; data poisoning, model poisoning, class representatives inference, membership inference, properties inference and labels inference categories are considered.	Brief				
[29] - Federated Learning Attack Surface: Taxonomy, Cyber Defenses, Challenges and Future Directions	Adversarial techniques are divided into poisoning, inference, free-rider and CIA triad attacks. Each category is then divided in more detailed ones	Complete	•	•		•
[18] - Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives	Adversarial techniques are divided based on the phase in which they are carried out (data auditing, training and prediction phase); different subcategories are then analyzed for each one	Complete		•	•	
[22] - Privacy and Robustness in Federated Learning: Attacks and Defenses	Focus is put on privacy and poisoning attacks: class representative, membership, properties and labels inference attacks are considered. Untargeted and targeted poisoning attacks are also considered but no further distinction is provided.	Partial	•		•	
[25] - A Survey on Security and Privacy of Federated Learning	Adversarial techniques are divided into data poisoning, model poisoning, data modification, membership inference, data leakage, GAN inference, backdoor and GAN based attacks. More focus is put on defense techniques.	Complete				
[33] - A Detailed Survey on Federated Learning Attacks and Defenses	Adversarial techniques are divided into data poisoning, model poisoning and inference attacks. A lot of focus is put on generic attacks for ML systems and defenses against them	Complete				
This thesis		**Complete**	•	•	•	•

6 | Conclusions and future works

Given the increasing interest in Federated Learning frameworks and their adoption in contexts where hundreds of thousands of devices are involved, it's crucial for the designers of such systems to understand what are the most common security threats and attack techniques they should consider to provide the highest possible level of security and dependability. This thesis wants to be a starting point for such considerations, making easily accessible the current state-of-the-art of adversarial attacks in the Federated Learning scenario.

As a possible future development of my work, it may be worth refining the attacks' analyses proposed here, testing each one of the considered techniques against each other on the same datasets to better understand their limitations, verify the reported performance levels and compare them more effectively.

Bibliography

[1] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020.

[2] G. Baruch, M. Baruch, and Y. Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.

[4] M. Chase, E. Ghosh, and S. Mahloujifar. Property inference from poisoning. *arXiv preprint arXiv:2101.11073*, 2021.

[5] J. Chen, J. Zhang, Y. Zhao, H. Han, K. Zhu, and B. Chen. Beyond model-level membership privacy leakage: an adversarial approach in federated learning. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9. IEEE, 2020.

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

[7] M. Fang, X. Cao, J. Jia, and N. Z. Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1623–1640, 2020.

[8] Y. Fraboni, R. Vidal, and M. Lorenzi. Free-rider attacks on model aggregation in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1846–1854. PMLR, 2021.

[9] C. Fu, X. Zhang, S. Ji, J. Chen, J. Wu, S. Guo, J. Zhou, A. X. Liu, and T. Wang. Label inference attacks against vertical federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1397–1414, 2022.

[10] C. Fung, C. J. Yoon, and I. Beschastnikh. The limitations of federated learning in sybil settings. In *RAID*, pages 301–316, 2020.

[11] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[12] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[13] B. Hitaj, G. Ateniese, and 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*, pages 603–618, 2017.

[14] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang. Source inference attacks in federated learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1102–1107. IEEE, 2021.

[15] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen. Cafe: Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems*, 34:994–1006, 2021.

[16] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang. Label leakage and protection in two-party split learning. *arXiv preprint arXiv:2102.08504*, 2021.

[17] J. Lin, M. Du, and J. Liu. Free-riders in federated learning: Attacks and defenses. *arXiv preprint arXiv:1911.12560*, 2019.

[18] P. Liu, X. Xu, and W. Wang. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity*, 5(1):1–19, 2022.

[19] Y. Liu, T. Zou, Y. Kang, W. Liu, Y. He, Z. Yi, and Q. Yang. Batch label inference and replacement attacks in black-boxed vertical federated learning. *arXiv e-prints*, pages arXiv–2112, 2021.

[20] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi. Feature inference attack on model predictions in vertical federated learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 181–192. IEEE, 2021.

[21] L. Lyu, H. Yu, and Q. Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[22] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and S. Y. Philip. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 2022.

[23] S. Mahloujifar, M. Mahmoody, and A. Mohammed. Universal multi-party poisoning attacks. In *International Conference on Machine Learning*, pages 4274–4283. PMLR, 2019.

[24] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[25] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[26] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pages 1–15, 2018.

[27] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi. Poisoning attacks on federated learning-based iot intrusion detection system. In *Proc. Workshop Decentralized IoT Syst. Secur.(DISS)*, pages 1–7, 2020.

[28] A. Pustozerova and R. Mayer. Information leaks in federated learning. In *Proceedings of the Network and Distributed System Security Symposium*, volume 10, 2020.

[29] A. Qammar, J. Ding, and H. Ning. Federated learning attack surface: taxonomy, cyber defences, challenges, and future directions. *Artificial Intelligence Review*, pages 1–38, 2022.

[30] H. Ren, J. Deng, and X. Xie. Grnn: generative regression neural network—a data leakage attack for federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–24, 2022.

[31] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173, 2023.

[32] V. Shejwalkar and A. Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021.

[33] H. S. Sikandar, H. Waheed, S. Tahir, S. U. Malik, and W. Rafique. A detailed survey on federated learning attacks and defenses. *Electronics*, 12(2):260, 2023.

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

[35] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pages 480–501. Springer, 2020.

[36] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 14(6):2073–2089, 2019.

[37] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

[38] L. Wang, S. Xu, X. Wang, and Q. Zhu. Eavesdrop the composition proportion of training labels in federated learning. *arXiv preprint arXiv:1910.06044*, 2019.

[39] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE conference on computer communications*, pages 2512–2520. IEEE, 2019.

[40] C. Xie, K. Huang, P.-Y. Chen, and B. Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2020.

[41] M. Xu and X. Li. Subject property inference attack in collaborative learning. In *2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 1, pages 227–231. IEEE, 2020.

[42] X. Xu, J. Wu, M. Yang, T. Luo, X. Duan, W. Li, Y. Wu, and B. Wu. Information leakage by model weights on federated learning. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, pages 31–36, 2020.

[43] X. Yuan, X. Ma, L. Zhang, Y. Fang, and D. Wu. Beyond class-level privacy leakage: Breaking record-level privacy in federated learning. *IEEE Internet of Things Journal*, 9(4):2555–2565, 2021.

[44] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[45] J. Zhang, J. Chen, D. Wu, B. Chen, and 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)*, pages 374–380. IEEE, 2019.

[46] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal*, 8(5):3310–3322, 2020.

[47] J. Zhang, J. Zhang, J. Chen, and S. Yu. Gan enhanced membership inference: A passive local attack in federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[48] B. Zhao, K. R. Mopuri, and H. Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[49] X. Zhou, M. Xu, Y. Wu, and N. Zheng. Deep model poisoning attack on federated learning. *Future Internet*, 13(3):73, 2021.

[50] L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[51] Z. Zhu, J. Shu, X. Zou, and X. Jia. Advanced free-rider attacks in federated learning. *1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting*, 2020.