Survey on Federated Learning Threats: concepts, taxonomy on attacks and defences, experimental study and challenges

Nuria Rodríguez-Barroso[a,*], Daniel Jiménez López[a], M. Victoria Luzón[b], Francisco Herrera[a], Eugenio Martínez-Cámara[a]

[a]*Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*

[b]*Department of Software Engineering, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*

Abstract

Federated learning is a machine learning paradigm that emerges as a solution to the privacy-preservation demands in artificial intelligence. As machine learning, federated learning is threatened by adversarial attacks against the integrity of the learning model and the privacy of data via a distributed approach to tackle local and global learning. This weak point is exacerbated by the inaccessibility of data in federated learning, which makes harder the protection against adversarial attacks and evidences the need to furtherance the research on defence methods to make federated learning a real solution for safeguarding data privacy. In this paper, we present an extensive review of the threats of federated learning, as well as as their corresponding countermeasures, attacks versus defences. This survey provides a taxonomy of adversarial attacks and a taxonomy of defence methods that depict a general picture of this vulnerability of federated learning and how to overcome it. Likewise, we expound guidelines for selecting the most adequate defence method according to the category of the adversarial attack. Besides, we carry out an extensive experimental study from which we draw further conclusions about the behaviour of attacks and defences and the guidelines for selecting the most adequate defence method according to the category of the adversarial attack. This study is finished leading to meditated learned lessons and challenges.

Keywords: Federated learning, adversarial attacks, privacy attacks, defences

*Corresponding author

Email addresses: rbnuria@ugr.es (Nuria Rodríguez-Barroso), dajilo@ugr.es (Daniel Jiménez López), luzon@ugr.es (M. Victoria Luzón), herrera@decsai.ugr.es (Francisco Herrera), emcamara@decsai.ugr.es (Eugenio Martínez-Cámara)

1. Introduction

Data-driven machine learning methods currently dominate artificial intelligence. This reliance on data allows us to stand out three artificial intelligence challenges. The former is the preservation of data privacy, since artificial intelligence methods process personal and sensitive data, such as health [1] and financial data [2]. Likewise, the growing interest in data privacy safeguarding is reflected in emerging legal frames such as the General Data Protection Regulation (GDPR) [3]. The second challenge is related to the increasing availability of data, which, on the one hand, is furthering the progress of artificial intelligence [4], and, on the other hand, it arises new challenges related to its storage and processing that are even exacerbated when data stemmed from distributed sources, as in IoT scenarios [5]. The latter challenge emerges from the need to distributively process data when it is not possible to transfer it to a central server, because of legal or regulatory restrictions, communication costs or other kind of technical limitations. Due to this distributed scenario, new difficulties appear linked to dissimilar data distributions from the same domain and the likely large size of data sources [6].

Federated learning (FL) is a machine learning paradigm proposed as a possible response to the three previous challenges, and especially for the demand of preserving data privacy, together with a distributed approach to tackle local and global learning [7]. FL aims at generating a collaboratively trained global learning model without sharing the data owned by the distributed data sources. Frequently, it requires a coordinator agent, which is in charge of managing the information exchange required to train the global learning model. In this way, the data is protected from unauthorised access, either by other data sources or the coordinator party.

Machine learning is vulnerable to adversarial attacks mainly focused on impairing the learning model or violating data privacy [8]. Likewise, FL is exposed to the same jeopardy, since it is an specific machine learning setting. Some of those attacks are grounded in the maliciously manipulation of the training data [9], which are inaccessible in FL and, then, we cannot rely on the use of data inspection techniques for detecting that altered data. Therefore, one of the weak points of FL is being exposed to adversarial attacks that may violate the integrity of the learning model or the privacy of data.

The evidence that adversarial attacks are a weak point of FL is built upon the fact of the large volume of publications centred on the identification of vulnerabilities in the form of adversarial attacks [10, 11, 12, 13], and on the corresponding large volume of defence proposals against to those attacks [14, 15, 16, 17]. This effervescent quantity of publications is the cause of the publication of several survey works on adversarial attacks that attain to review and summarise the latest papers related to this weak point. These surveys lack of an holistic view of FL and the review of the defences against adversarial attacks, because of the following reasons: (1) most of them are only focused on one kind of adversarial attacks, namely there are surveys reviewing attacks to the federated model [18, 19, 20] or privacy attacks [21, 22, 23], but any of them encompass both sort of attacks; (2) the vast majority does not include any experimental study [24, 25, 26, 27, 28], so it is

not possible to compare the strength of the attacks and the robustness of the defences in a common evaluation framework; and (3) by default they only focus on horizontal FL ignoring vertical and federated transfer learning.

Due to the mentioned facts, we propose a new survey on FL threats, and additionally we provide several taxonomies on adversarial attacks and defences, an experimental study and a final discussion about lessons learned and challenges. This survey differs from previous ones due to the following contributions:

1. To provide a general picture of the field of adversarial attacks and defences by considering the threats to the learning model and to the integrity of the privacy of data.
2. To review the threats and the defences of horizontal FL, vertical FL and federated transfer learning.
3. To define a taxonomies of adversarial attacks and their corresponding defensive countermeasures. These two taxonomies encompass the different categories of adversarial attacks and defences, which will shed light in this crucial field of making FL a robust learning paradigm.
4. To provide a guidelines for selecting the right defence category according to the threatening adversarial attack.
5. To compare in a common evaluation framework the strength of the most relevant adversarial attacks, and the defence capacity of the most prominent defence methods.
6. To expound some learning lessons stemmed from the literature review and the experimental study conducted.
7. To also expound their relations to the challenges in the field of adversarial attacks.

The rest of the paper is organised as follows: the following section introduces the propaedeutic concepts necessary for this survey to be illustrative. Section 3 presents the taxonomy of adversarial attacks in FL, while Section 4 expounds the taxonomy of defences against them. We conduct the experimental study in Section 5. In Section 6 we provide the guidelines for selecting the right defence category. Finally, we discuss the lessons learned and challenges in Section 7 and 8, and include some conclusions in Section 9.

2. Background concepts on Federated Learning threats

The concepts described throughout this paper require the knowledge of some propaedeutic concepts related to FL and its threats. Accordingly, we introduce FL and the categories of FL in Section 2.1, we formally define differential privacy (DP) in Section 2.2, since a considerable amount of defence methods are based on DP, and we detail the categorization of the attacks in terms of the threat model in Section 2.3.

2.1. Federated Learning

FL is a distributed machine learning paradigm with the aim of building a ML model without explicitly exchanging training data between parties [7]. It consists in a network of clients or data owners $\{C_1, \ldots, C_n\}$, who participate in two main processes:

1. *Model training phase:* each client exchange information without revealing any of their data to collaboratively train a ML model, $\mathcal{M}_f$, which may reside at one client or may be shared between a few clients.
2. *Inference phase:* clients collaboratively apply the jointly trained model, $\mathcal{M}_f$, to a new data instance.

Both processes can be either synchronous or asynchronous, depending on the data availability of the clients and the trained model.

It must be highlighted the fact that privacy is not the only motivation of this paradigm, there should be a fair value-distribution mechanism to share the profit gained by the collaboratively trained model, $\mathcal{M}_f$.

The distribution of characteristics of the data among clients in FL shapes the procedure to follow in the two main processes of FL, particularly we focus on the following distributions: (1) clients share the feature space but not the sample space, (2) clients share the sample space but not the feature space, and (3) clients share only a small overlap in feature space. These distributions allow us to present three categories of FL [7] in terms of the feature space (X), the label space (Y) and the sample ID space (I) as follows:

Horizontal Federated Learning (HFL). In this scenario, clients data share the feature and labels space, but differ in the sample space. Formally, we can define as:

$$X_i = X_j,\ Y_i = Y_j,\ I_i \neq I_j,\ \forall D_i, D_j,\ i \neq j$$

where the feature and labels space of the clients (i, j) is depicted by (X_i, Y_i) and (X_j, Y_j) and it is assumed to be the same, while the samples I_i and I_j are not the same. D_i and D_j depict the data of the clients i and j.

Vertical Federated Learning (VFL). In this scenario, clients share the sample space but neither the feature space nor the label space. Formally, we can define as follows:

$$X_i \neq X_j,\ Y_i \neq Y_j,\ I_i = I_j,\ \forall D_i, D_j,\ i \neq j$$

Federated Transfer Learning (FTL). This scenario is similar to the traditional transfer learning. The clients share neither the feature space, nor label space, nor the sample space. Formally, we can define as follows:

$$X_i \neq X_j,\ Y_i \neq Y_j,\ I_i \neq I_j,\ \forall D_i, D_j,\ i \neq j$$

Although the feature spaceand the label space are not the same, in FTL there is a certain overlap or similarity, since the aim is to transfer knowledge from one client to another securely. FTL was presented in [29] and it represents higher difficulty than HFL and VFL, since it implies the use of techniques that preserve the data privacy. We represent the different categories of FL in Figure 1.

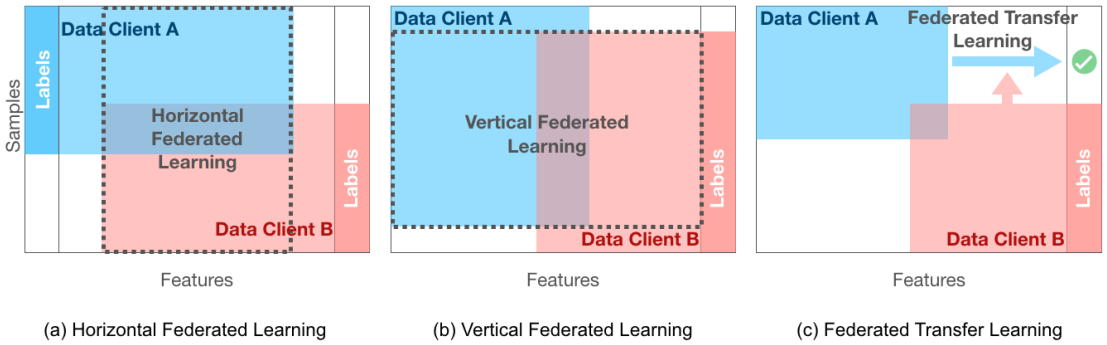

Figure 1: Representation of the different categories in FL. Source [7].

FL is a learning setting composed of a set of key elements. Since FL is a specific configuration of a machine learning environment, it shares with machine learning some of those key elements, such as the data and the learning model. Nonetheless, the particularities of FL make necessary additional key elements, such as clients and a learning coordinator that orchestrates the two main processes of FL. A detailed description of FL key elements focused on HFL is in [30], and here we describe the common ones to all the FL categories.

Data. It plays a central role in machine learning. In FL, data is distributed among the different clients according to two possibilities: (1) IID (Independent and Identically Distributed), when the data in each client is independent and identically distributed, as well as representative of the population data distribution; and (2) Non-IID (non Independent and Identically Distributed), when the data distribution in each client is not independent no identically distributed from the population data distribution. These data distributions are mainly relevant to HFL. In VFL and FTL categories, clients do not share neither the feature space nor the label space, and consequently the data distribution among clients is relegated to a second place.

In most HFL scenarios, each client only stores the data generated on the client itself, ensuring the non-IID property of the global data. Moreover, even if the IID scenario were present, it would not be known because of the data privacy properties of FL. Hence, the non-IID scenario is the best choice and it represents a real challenge.

Clients. Each client of a federated scenario plays a key role in a federated paradigm, as a data owner and as a part of the distributed scheme. Typical clients in FL could be servers, smartphones, IoT devices, connected vehicles, hospitals, banks or insurance companies. Privacy is not their only motivation, they also want to profit from the *model training phase*.

As a consequence, a reward mechanism is expected, such as owning the collaboratively trained model, $\mathcal{M}_f$, in HFL or the outputs of the *inference phase* in VFL and FTL.

Learning coordinator. The learning coordinator orchestrates the communication among the clients in the two main processes of FL. While it is not strictly necessary, when present, it also plays the role of a trusted authority. In VFL, the learning coordinator receives and combines partial updates from clients and shares the corresponding part of the combined update with each client in the *model training phase*. Moreover, in the *inference phase* it helps to perform the inference by combining the outputs of each client as the collaboratively trained model, $\mathcal{M}_f$, is split among them. In contrast to VFL, in HFL the learning coordinator is usually known as th federated server and it only participates in the *model training phase*: (1) receiving the trained parameters of the local models, (2) aggregating the trained parameters of each client model using federated aggregation operators and (3) updating every learning model with the aggregated parameters.. Moreover, the *inference phase* is not performed in a collaborative way as the collaboratively trained model, $\mathcal{M}_f$ is stored in each client and in the federated server.

2.2. Differential Privacy

DP allows retrieving information, rigorously bounding the harm caused to individuals whose sensitive data are stored in the database [31, 32]. Basically, it hides the presence of an individual in the database. To achieve this, DP adds random noise to the outputs. Such noise is calibrated to the magnitude of the largest contribution that can be made to the output by an individual. It is important to note that DP assumes that the adversary owns arbitrary external knowledge.

DP is the key property used to provide a certain level of privacy to any sensitive data access, in a way it is both, secure and measurable. It is secure because it has a theoretical background which supports it. It is measurable as every access to private data has a privacy cost either in terms of ϵ or in terms of (ϵ, δ).

This interpretation naturally leads to define the *distance between databases*: two databases x, y are said to be n-neighbouring if they differ by n entries. In particular, if the databases only differ in a single data element ($n = 1$), the databases are simply addressed as *neighbouring*.

Differential Privacy definition. A database access mechanism, $\mathcal{M}$, preserves ϵ-DP if for all neighbouring databases x, y and each possible output of $\mathcal{M}$, represented by $\mathcal{S}$, it holds that:

$$P[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon}\, P[\mathcal{M}(y) \in \mathcal{S}] \tag{1}$$

If, on the other hand, for $0 < \delta < 1$ it holds that:

$$P[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon}\, P[\mathcal{M}(y) \in \mathcal{S}] + \delta \tag{2}$$

then the mechanism possesses the property of (ϵ, δ)-DP, also known as approximate DP.

In other words, DP specifies a "privacy budget" given by ϵ and δ. The way in which it is spent is given by the concept of privacy loss. The privacy loss allows us to reinterpret both, ϵ and δ in a more intuitive way:

- ϵ limits the quantity of privacy loss permitted, that is, our privacy budget.
- δ is the probability of exceeding the privacy budget given by ϵ, so that we can ensure that with probability $1 - \delta$, the privacy loss will not be greater than ϵ.

DP has some interesting properties, which makes it even more appealing in a privacy context.

1. **DP is immune to post-processing**. if an algorithm protects an individual's privacy, then there is not any way in which privacy loss can be increased.
2. **DP can be used to protect the privacy of groups**. Let $\mathcal{M}$ be a ϵ-differentially private mechanism, then $\mathcal{M}$ is $K\epsilon$-differentially private for groups of size K.
3. **DP mechanisms can be composed multiple times and remain differentially private**. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be ϵ_1-differentially private mechanism and ϵ_2-differentially private mechanism, respectively. Then, their composition output given by the concatenation of the output of $\mathcal{M}_1$ and $\mathcal{M}_2$ over the same input is $\epsilon_1 + \epsilon_2$-differentially private

2.3. Threat Model

Threat models in machine learning are structured representation of information, which help to identify and define potential security issues. They can be defined in terms of the information available and the scope of action of the attacker. In this regard, we define the following set of mutually exclusive terms that allow us to define the FL threat model.

Insider vs. Outsider. One of the key elements of any distributed system is the communication between different parts. The communication is very vulnerable, since it can be compromised by agents from outside the learning system, which are known as outsider attackers. When the attack is carried out by one of the participants in the distributed system, either one or more clients, or the server, it is known as an insider attacker. Clearly, the scope of the two attacks is very different: insider attacks are more harmful and may be aimed at modifying the behaviour of the model or inferring valuable information from other clients, while those carried out by outsiders are usually aimed only at inferring information about the data or the resulting learning model. Outsider attacks mainly focus on sniffing information of the communication channels between the involved agents. They are either side-channel attacks, when the attacker gains information from the implementation of the FL scenario, or man-in-the-middle attacks, when the attacker intercepts the communication channel by disguising herself as the receiver part. Both attacks are related to the protocols used to establish communication and their implementation.

We focus on insider attacks, in which we highlight the following categorisations:

- **Byzantine attacks**. They consist in sending arbitrary updates to the server, so it compromises the performance of the global learning model.
- **Sybil attacks**. They consist of collaborative attacks, either by several attackers joining together or by simulating fictitious clients in order to be more disruptive.

Client vs. Server. Regarding insider attacks, in HFL it is natural to differentiate between two types of attacks, depending on whether they are carried out by a client or by a server. The main point of difference lies in the amount of information available. While the attacks carried out by clients only have information of one or several clients, the server holds information about the model architecture and the updates of the clients in each round of learning. Even, in cryptographic implementations of the federated communication among the federated server and the clients, the server owns more information than the clients, as it is the only one with enough knowledge to decipher the models.

Attacker knowledge. In centralised settings, the white-box attacker has full access to the target model, including the model architecture, the parameters and its internal state. In contrast, the black-box attacker does not have any access to the target model and additionally, she might have some additional information about the architecture of the target model or its training procedure. These two classifications of attacker knowledge are too general to represent every type of attacker, because there is no middle ground to consider attackers whose knowledge in the black box setup is too restricted, and in the white box setup is not enough constrained. To address this issue, a grey-box attacker was introduced in [33], which is a black-box attacker with some specific statistical knowledge not publicly available that concerns her victim. This description of attacker knowledge is tailored for a centralised learning setting, and as a consequence it does not fit other learning settings as the attack surface changes. In a FL system, white-box, grey-box or black-box attackers can be any node, either the clients or the server. Moreover, the exposed attack surface is greater than in centralised settings. Most attacks are related to the data owned by the clients and the communication among the federated server and the clients, therefore, we also require including the information available regarding the federated training process and to the client's private data. In order to address such requirements, we define the following classification of the attacker's knowledge suited for HFL and VFL:

In a standard HFL system, an attacker which owns a client has *client-side knowledge*:

- White-box access to the aggregated model.
- White-box access to the client's locally trained model.
- Access to the owned client's dataset.

If the attacker has access to local data of other clients or their labels, she has *extra client-side knowledge*.

An attacker which owns a federated server has *server-side knowledge*:

- White-box access to the aggregated model after each communication round.

- White-box access to trained models shared by the clients or, alternatively, access to their gradients.
- The identifiers of the clients aggregated in each communication round.
- The labels owned by each client and, optionally, the size of their dataset.

In a standard VFL system, an attacker which owns a client has *party-side knowledge*:

- White-box access to the parameters related to the features of the owned client.
- Access to the client's private dataset.
- The partial output of the parameters, when an inference is requested.

Additionally, if the attacker has access to information related to the features of the other clients, she has **extra party-side knowledge**.

An attacker which owns the learning coordinator in a VFL system has *third party-side knowledge*:

- The gradients shared by each client.
- The computed loss.
- The partial output of each client, when an inference is requested.

If only a subset of the specified knowledge is available to the attacker, then she has *partial* knowledge, and we specify the content of that subset of knowledge. Moreover, defences are expected to reduce the attacker knowledge, therefore in the presence of a defence an attacker is expected to have *partial* knowledge.

In both HFL and VFL systems, if the attacker only have access to the outputs of the federated model, she has *outsider-side knowledge*.

We highlight the fact that the categories stated are not mutually exclusive, that is, an attacker can own multiple types of knowledge at the same time. Realistic attack scenarios tend to require lesser attacker knowledge, while more complex and specific attacks require knowledge from multiple participants of a FL task.

Honest-but-curious vs. Malicious. A malicious (or active) attacker tries to interfere in the training process of the learning model with the aim of corrupting the target model, for example, damaging its performance or injecting a secondary task. On the contrary, an honest-but-curious (or passive) attacker does not interfere in the training process and follows the federated learning protocols, but try to obtain private information about other clients from the received information.

Collusion vs. No-collusion. The collusion threat lies in the fact that the attacker who controls more clients has more power in a distributed system. There are two collusion types: (1) server-participants, in which the attacker controls some benign participants and the server, and it aims to infer information about the rest of the clients; and (2) participant-participant, in which the attacker controls a fraction of the benign clients and aims to infer information about benign clients, the server or to harm the learning model.

3. Adversarial Attacks in Federated Learning: Taxonomies

Adversarial attacks represent one of the more challenging problems in FL, due to the large number of existing attacks and the difficulty of defending against them. Moreover, the distribute nature of FL makes it vulnerable to wide variety of adversarial attacks aiming at different objectives and using different ways to achieve these objectives. Due to this wide variety in the nature and target of attacks, it is difficult to establish a common taxonomy for all types of adversarial attacks. For this reason, we propose the first broadly classification by differentiating between:

- **Attacks to the federated model**, which aim at modifying its behaviour.
- **Privacy attacks**, whose purpose is to infer sensitive information from the learning process.

In Figure 2 we represent this first categorisation of the adversarial attacks in FL.

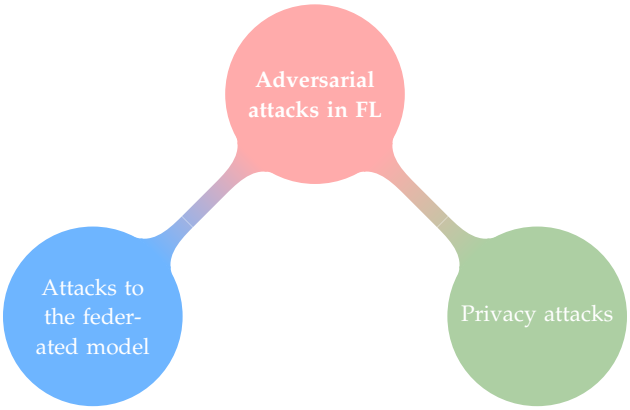

Figure 2: First, categorisation of the adversarial attacks in FL into two broad categories: attacks to the federated model and privacy attacks.

Once this initial classification into these two main categories of attacks has been established, we further examine each category by proposing a taxonomy based on different criteria and review the most relevant works on each topic. In Section 3.1 we focus on attacks to the federated model and the Section 3.2 is dedicated to the privacy attacks.

3.1. Adversarial attacks to the federated model

One of the main limitation of FL, and more specifically of the HFL, in terms of adversarial attacks, is that clients have the ability to harm the model by sending poisoned updates, while the server cannot inspect the training data stored on the clients. This fact makes the adversarial attacks to the federated model become one of the most significant challenges in FL.

In general, these attacks are carried out by clients and the white-box feature of these attacks correspond to the situation in which the attacker has client-side knowledge, either there are one or several adversarial clients (attackers). In some situations attackers are

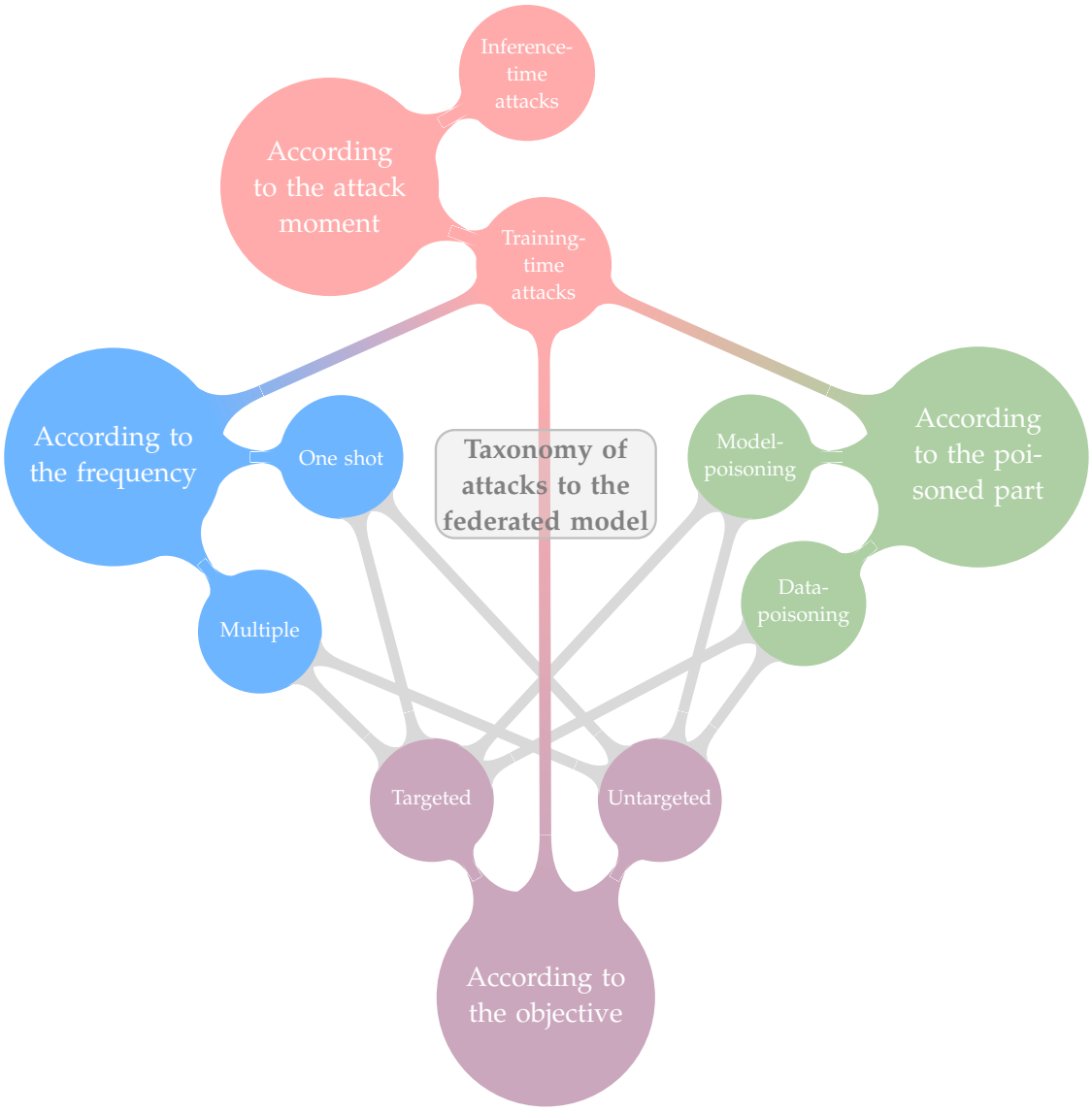

Figure 3: Representation of the attack taxonomies to the federated model according to the different criteria. The grey links represent the possibility of combination of both categories. For the sake of clarity, we don't show redundant connections between categories already connected with other links.

considered to have access to more white-box information, for example about the aggregation mechanism used on the server, which is not a realistic situation. We therefore highlight those attacks that only require information from the adversarial client.

Within this broad category, we propose a taxonomy that encompasses a range of attacks according to different criteria, which we depict in Figure 3. Thus, each type of attack in the literature belongs to four different categories, one for each criterion. From the main taxonomy, we additionally propose four more taxonomies linked to each criterion, namely: (1) the attack moment in Section 3.1.1, (2) the objective in Section 3.1.2, (3) the poisoned part of the FL scheme in Section 3.1.3 and (4) the frequency in Section 3.1.4.

3.1.1. Taxonomy according to the attack moment

We present the taxonomy according to the time at which the attack is carried out, which completely determines the ability of the attack to influence the federated model. We classify the following two types of attacks:

Training time attacks. The training time phase includes from data collection and data preparation to model training. These attacks are carried out during this phase, either continuously or as a single attack. They are the most common in the literature since they have the ability to modify the federated model that is still being trained [10, 34, 35] and to infer some information from training data [36] (see Section 3.2).

Inference time attacks. These attacks are carried out in the *inference phase* when the model has been trained. They are called evasion or exploratory attacks [25]. Generally, the objective is not to modify the trained model, but to produce wrong predictions or to collect information about the characteristics of the model.

3.1.2. Taxonomy according to the objective

The most widely used categorisation in the literature, which makes it the most significant criteria is based on the target of the attack. Although all the attacks in this section are gathered under the scope of modifying the model, the modifications can be quite diverse. We distinguish two broad groups depending on the target of the attack:

Targeted or backdoor attacks [37, 38, 10]. The main task is to inject a secondary or backdoor task into the model. In other words, a backdoor attack is successful as long as it succeeds in preserving its performance in the original task while injecting a second task. These attacks are very stealthy, since they generally do not affect the performance of the original task [39], which makes them hard to detect. Note that although they do not pose a danger to the FL main task, they do represent a danger to the integrity of the system, since the attacker takes advantage of the federated infrastructure to perform a certain backdoor action, representing a security breach. The nature of such attacks is broad, given the great variety of secondary tasks. We present a taxonomy based on different criteria, which is shown in Figure 5, with the following categories being the most frequent:

- *Input-instance-key strategies*. The objective is that the model labels specific input examples with a specific target label different from the original one. For example, in a face recognition system that allows access to a house, to identify five specific people from the input set, who originally did not have access (negative label as origin label) as people who can access (positive label as a target label). Some works which implement this kind of the attack are [18] where the authors analyse the impact of different attacks scenarios, [40] where the authors prove that you can really backdoor FL even using existing defences and [41] where the aim is to present the data-poisoning attacks.
- *Pattern-key strategies*. The objective is that the model associates a particular pattern in an input sample with a particular target label. For example, in the face recogni-

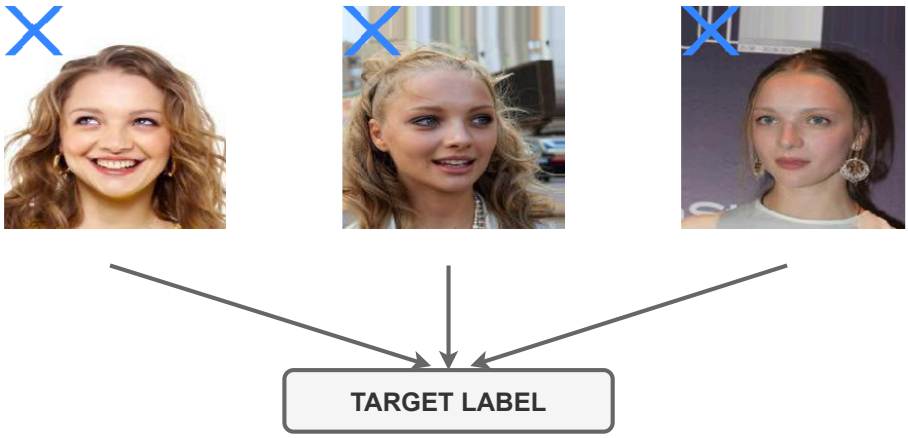

Figure 4: Representation of an attack using pattern-key strategy based on associate the blue cross with some prefixed target label.

tion system above, to allow access to any person wearing a polka-dot bow. In this way the system would identify the pattern "polka-dot bow" with the target label (positive label). In practice, a simple pattern of a cross or similar mark are chosen for association with a target label [38, 10]. In Figure 4, we depict an attack using the pattern-key strategy of associating the blue cross with the target label.

Additionally, these attacks can also be categorised according to different criteria about the injected pattern as shown in Figure 5.

Regarding the design of the pattern in [37] the authors introduce the following terminology with the aim of classifying pattern attacks. Although this classification is not usually specified in other FL work, it is common in ML, and we believe it would be useful to use this notation in FL attacks as well.:

- *Blended injection strategy.* This strategy generates backdoor instances by blending a benign input instance with the key pattern using a blend ratio. The pattern can be any image, for example cartoon images or randomly generated patterns. The main limitation is that this mechanism requires to modify the entire sample during both training and testing, which may not be feasible.
- *Accessory injection strategy.* This attack arises as a solution to the main limitation of the Blended injection strategy and proposes to generate backdoor images adding patterns to some regions of the original images. They are equivalent to wearing an accessory in real life.
- *Blended accessory injection strategy.* It takes advantage of both strategies by combining the accessory and the blended approach.

Regarding the number of patterns:

- *Single pattern attack.* It refers to when all adversarial clients inject the same

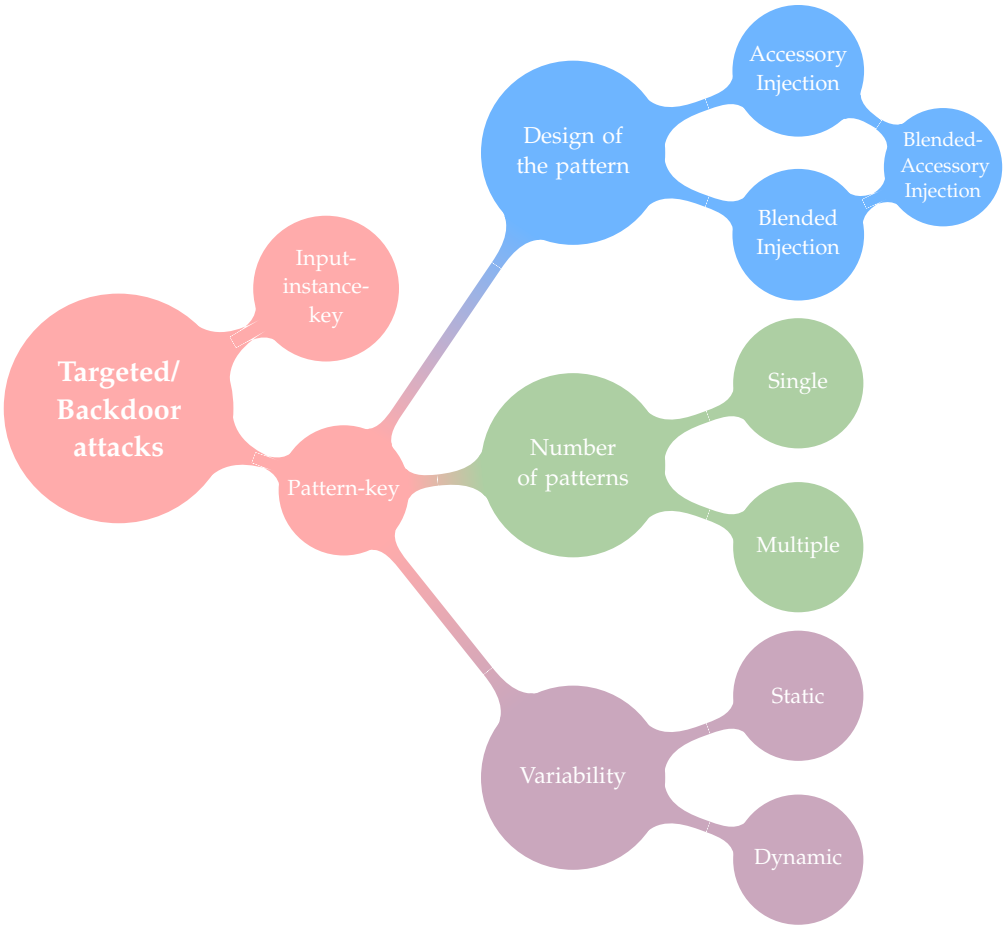

Figure 5: Representation of the taxonomy of backdoor attacks.

pattern into the model. They are usually more successful as they are a collective attack on the same target, but at the same time easier to identify on the server. This situation is the most common one and some works such as [10, 37] where the authors focus on presenting the vulnerabilities of FL to such attacks, or [15] where the aim is to propose a defence mechanism against them implement single pattern attacks.

– *Multi-backdoor attack* [10]. It is composed of several coordinated adversarial clients (sybils), where each of them injects a different pattern or part of a common pattern to the model [42]. On the contrary, they are more difficult to detect on the server because the distribution of the pattern across clients enhances the stealth. Though, it is more complicated for clients to inject backdoor tasks into the model, due to the diversity of secondary tasks.

Regarding the variability over time of the pattern:

– *Static attack*. When the pattern of the attack is maintained over time regardless of the frequency of the attack. This situation is the most common one and some

works cited before such as [10, 37, 15] implement static attacks.

- *Dynamic attack*. The pattern changes over time, which is a challenge both for the defences, as the pattern to be identified changes, and for the adversarial clients, as they have to continuously adapt to new secondary tasks increasing the computation required. Salem et al. [43] propose to use meta-learning in order to speed up the adaptation of clients to the new backdoor tasks, and design a "symbiosis network" in which the clients weight the update of the model weights with the global model, instead of completing replacement in order to maintain the performance on the backdoor tasks.

Some works question the strength of backdoor attacks, since the most naive approaches are mitigated by simple defences [38]. However, the potential of these attacks is shown in Wang et al. [40], where they demonstrate that poisoning samples belonging to the tails of the data distribution is enough to compromise the federated global model. In addition, Liu et al. [44] show that even attackers with no access to training labels can inject backdoor attacks in feature-partitioned collaborative learning. In conclusion, preliminary studies show that backdoor attacks are a real threat to FL, which further increases the interest in this research area.

Untargeted attacks *[45, 46].* As opposed to targeted attacks, the only goal of untargeted attacks is to impair the performance of the model on the original task. The most extreme scenario is known as *Byzantine attacks* [47, 48], in which adversarial clients share randomly generated model updates or train over randomly modified data, generating random model updates as well. Clearly, these attacks are inherently less stealthy than targeted attacks, and can be detected merely by analysing the performance of the local models updates on the server, although it is sometimes difficult to differentiate them from clients with very particular training data distributions.

It is worth mentioning the ***free-riders attacks***. It is common in FL systems for clients to be awarded rewards for participation, as they provide crucial and necessary information. These rewards may tempt some clients to pretend that they are participating in the local training process and send updates to their models. To this end, they generate their "model updates" randomly resulting in the same effect as Byzantine attacks [49].

3.1.3. Taxonomy according to the poisoned part of the FL scheme

Most training-time model attacks are based on poisoning client's information in order to corrupt the global learning model. Depending on which part of the client's information is poisoned, we differentiate between data-poisoning and model-poisoning attacks, and we refer both attacks as poisoning attacks. Figure 6 shows the taxonomy presented in the rest of the section. In the following, we detail each one of them:

Data-poisoning attacks *[50, 51].* The attacker is assumed to have access to the training data of one or more clients and to be able to modify it. Depending on the characteristics of the poisoning, we distinguish between the following attacks:

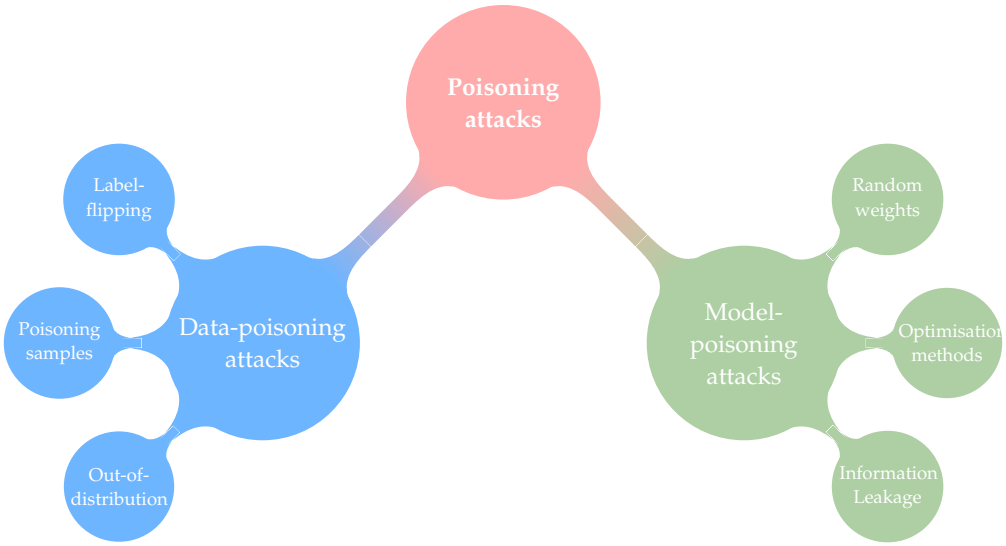

Figure 6: Representation of the taxonomy of backdoor attacks according to the poisoned part of the FL scheme.

- *Label-flipping attack* [52]. This attack consists of modifying the labels of a portion of the training data. It can be either targeted, by exchanging some specific labels [50], or untargeted [48], by random label shuffling.
- *Poisoning samples attack*. Unlike the previous one, this attack consists of modifying part of the training data samples. The poisoning can be of different types, such as including patterns in the samples and associate it with some target class, or normalizing the samples and adding uniform noise with the aim of impairing the performance of the model. In recent years, the use of Generative Adversarial Nets (GANs) [53] to generate these poisoned samples has become popular, to maximize the target of the attack on the one hand, and on the other hand, to maximize the disguise of the attack to overcome the possible defences of the server on the other [54]. A further clear example is the case of the attack proposed in [55], which consists in: (1) the attacker first behaves as a benign client and trains a GAN to mimic prototypical samples of other benign clients and, then, (2) the attacker generates the poisoned samples using these generated samples in order to compromise the global model by sending scaled poisoning updates as their local model updates.
- *Out-of-distribution attack*. This attack is similar to the poisoning samples attacks, although they differ in that the poisoned training samples are not modifications of the original ones, but samples from outside the input distribution [56]. It is possible to use either samples from another domain with the same characteristics or samples made of random noise.

One of the key factors for the success of a data poisoning attack is the proportion of adversarial clients, and the amount of data they poison. In [51], they experiment with different data-poisoning attacks and conclude that: (1) the attack success increases linearly with

the number of poisoned samples; (2) the increment of the number of attackers could improve the attack success without changing the total number of poisoned samples; and (3) the attack success increases faster with the number of poisoned samples than when there are more attackers involved.

The goal of most data-poisoning attacks is to impair the global model and thus the local models of all clients. However, it is also possible that the goal of the attackers is not to impair of the local models, but only a specific subset of them. In Sun et al. [41], they define a set of target nodes as those nodes (clients or server) to be compromised by the attack. According to this definition, we may differentiate between the following three types of data-poisoning attacks depending on the access level the attackers have to the target nodes:

- *Direct attack*. The attackers have access to target nodes, so they inject poisoning samples directly on them.
- *Indirect attack*. The attackers have no access to target nodes, so they have to employ further mechanisms such training themselves (in case they are clients) on the poisoned samples to poison the global model, which will then shared with the target clients.
- *Hybrid attack*. When the attackers combine both previous attacks.

In the vast majority of the attacks in the literature, the attackers are supposed to have access to the target nodes, so the most common attacks are direct attacks.

Model-poisoning attacks. These attacks consist of directly poisoning the model updates sent by the clients to the server. Although data-poisoning attacks naturally lead to model-poisoning attacks, in this section we focus only on those attacks that directly modify the local update weights. Depending on how these model weights are generated, we distinguish between:

- *Random weights generation*. These attacks are based on generating the model weights as a vector of randomly generated values of the same dimension as the model weights received from the server. Two specific examples are: (1) the *random weights attack* [19], in which an interval [-R,R] is inferred from the global learning model and the weights randomly generated in that interval; and (2) the *Gaussian attack* [11], a white-box attack, which chooses as model weights a sample from the Gaussian distribution resulting of the other clients' model updates. By construction, the random weights attacks are more harmful while being easier to detect, so depending on the scenario it would be more dangerous one or the other.
- *Optimization methods*. They consist of maximizing performance in the backdoor task, while minimizing the differences of the poisoned model with respect to the shared model by the server in the last round, thus maximizing effectiveness and stealth. This challenge is approached as a multi-objective optimization problem [57]. This

methodology is quite versatile and can be used to attack in special situations. For example, it is widely used to attack specific defences by introducing new criteria to be optimized [11] in order to overcome defences discarding conditions specific to each defence. In addition, in [57] they also prove that regularization techniques decrease the impact of the training data in the resulting model. For that reason, they propose to train adversarial clients without any regularization mechanism in order to increase the impact of the poisoned samples. This kind of attack is probably the most efficient approach to perform targeted attacks on the model.

- *Information leakage*. A particular use case of model-poisoning attacks in FL is information leakage, where the objective is not to compromise the global model, but the communication among the attackers through a secure protocol [58]. It consists in the fact that certain clients are coordinated in such a way that they know common rules and by modifying small parts of the model weights they can communicate. In [58] is proposed to adjust the training data strategically so that the weight of a particular dimension in the global model will show a pattern known by the rest of the malicious clients. Along very similar lines, Costa et al. [59] put forward a novel attacker model aiming at turning FL systems into covert channels to implement a stealth communication infrastructure by means of modifying certain bits of the models.

In FL, with the assumption that the proportion of adversarial clients is significantly lower than that of benign ones, the effect of the attack is expected to be dissipated in the aggregation. Therefore, *model-replacement* techniques [39, 38, 10] are used, which consist of weighting the contribution of adversarial clients using boosting techniques in order to replace the aggregated model with its local updates. Formally, if we consider the update of the global model in the learning round t is computed as follows in Equation 3:

$$G^t = G^{t-1} + \frac{\eta}{n} \sum_{i=1}^{n} (L_i^t - G^{t-1}), \tag{3}$$

where G^t is the aggregated model at the learning round t, L_i^t the model update of the client i at the learning round t, n the number of clients participating in the aggregation and η the server's learning round.

In this context, we consider the local model update of the adversarial client trained on the poisoned training data as follows in Equation 4:

$$\hat{L}_{adv}^t = \beta(L_{adv}^t - G^{t-1}), \tag{4}$$

where $\beta = \frac{n}{\eta}$ is the boost factor. After that, replacing Equation 4 in Equation 3 we have[1]

[1]We assume that the adversarial client is client 1.

$$G^t = G^{t-1} + \frac{\eta}{n}\frac{n}{\eta}(L^t_{adv} - G^{t-1}) + \frac{\eta}{n}\sum_{i=2}^{n}(L^t_i - G^{t-1}). \quad (5)$$

According to the definition of FL [60], eventually the FL model will converge to a solution, so we can assume that $L^t_i - G^{t-1} \approx 0$ for benign clients. Hence, we rewrite Equation 5 as follows

$$G^t \approx G^{t-1} + \frac{\eta}{n}\frac{n}{\eta}(L^t_{adv} - G^{t-1}) = L^t_{adv}, \quad (6)$$

which replaces the global model with the model updates of the adversarial clients. If there is more than one adversarial client, the boosting factor is divided among all of them.

Boosting techniques depends on knowing the number of clients participating in the aggregation, which is a much more restrictive client-side knowledge condition. In practice, clients estimate this value by making several tests with different values and analysing the model updates returned by the server. However, in the vast majority of the experimental works it is assumed the worst situation in which the adversarial clients know the number of clients of each aggregation for a better behaviour of the attack and a fair comparison between the proposed defences [10].

3.1.4. Taxonomy according to the frequency

As training-time phase is maintained over long periods of time, training-time attacks can be carried out at any time of the training and on one or several occasions [10]. We differentiate between the following two categories:

- *One-shot attack.* The attack is carried out in a single moment of the training, in a specific learning round. In Bagdasaryan et al. [10] the authors experiment with backdoor attacks at different stages of convergence and conclude that converged model attacks are more effective over several learning rounds, since the learning model does not vary and the secondary task remains injected into the global model.
- *Multiple or adaptive attack.* The attacks are carried out continuously during the training process, either during all the learning rounds or a portion of them. They are more elaborate as the attackers have to become part of the aggregation in several rounds, but this kind of attack can be more effective and stealthy [61].

3.2. Privacy attacks

Privacy attacks are designed to disclose information about the participants of a machine learning task. Not only they pose a threat to the privacy of the data used to train the machine learning models, they also pose a privacy risk to those people who agreed to share their private data. FL was thought of as a privacy preserving distributed machine learning paradigm, however the learning process exposes a broad attack surface. While the private data never leaves their owner, the exchanged models are prone to memorization of the private training dataset. In this section, we present a wide taxonomy which

aims to ease the understanding the diversity of privacy attacks. It is designed around the objective of the privacy attacker, a summary of it is shown in Figure 7.

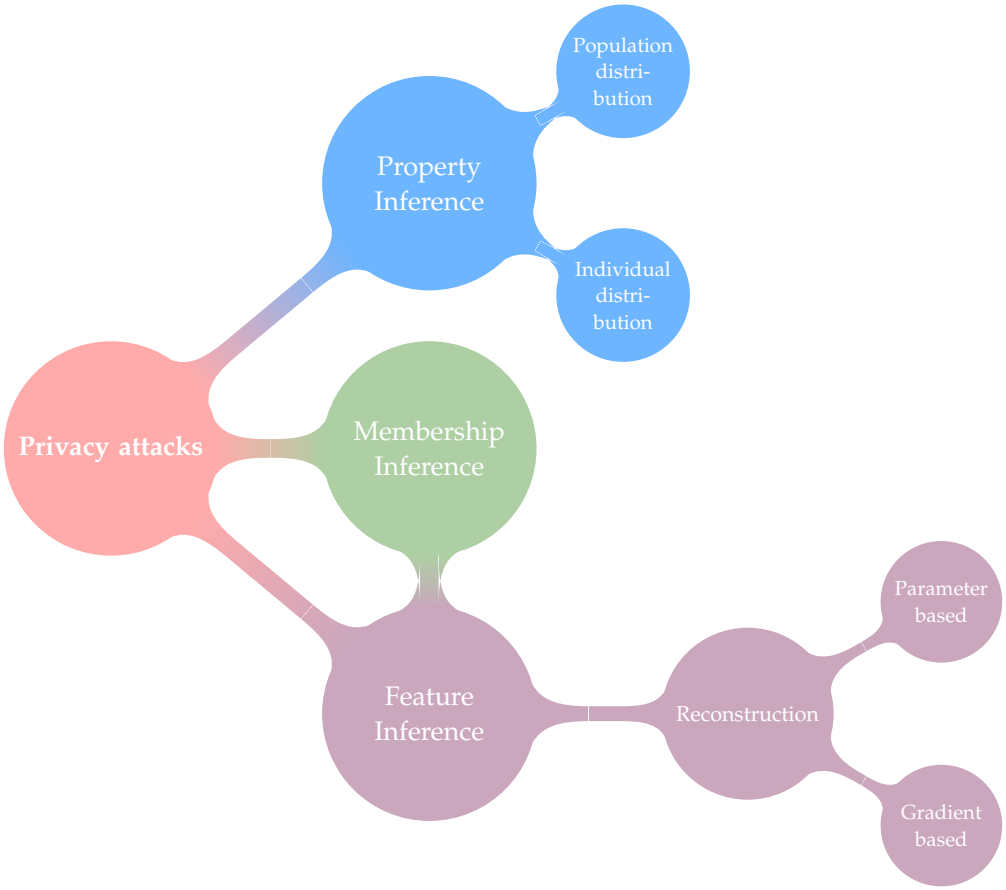

Figure 7: Representation of the taxonomy of privacy attacks in terms of the objective of the privacy attacker.

3.2.1. Feature inference attacks

Also known as *Reconstruction attacks* when referring only to HFL. The aim of these attacks is recovering the dataset of a client who participates in a FL task. Usually the recovered data are images or plain text. An example of the capabilities of such attacks can be seen in Figure 8. Particularly, in VFL the extracted data are the private features owned by the parties.

Accounting only for HFL, we can partition the Feature inference attacks according to the federated clients attack surface, that is, the information exchanged between the clients and the federated server:

- *Gradient based*: selected clients share their gradients with the federated server in the communication rounds, that is, a federated SGD based training procedure. Therefore, the attack surface is the clients' gradients. To our knowledge, Zhu and Han [12] are the first ones to exploit this setting. Their proposed passive attack is able to recover images and text owned by the target client. The attacker requires partial

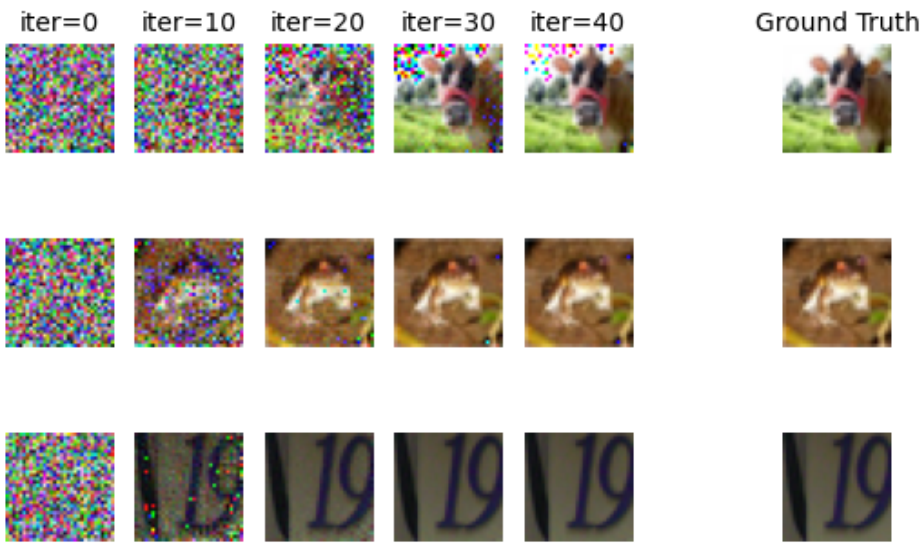

Figure 8: Gradient based Feature inference attack from Zhu and Han [12] applied to CIFAR10, CIFAR100 and SVHN datasets.

client-side knowledge, that is, accessing the gradients shared by the attacked client. However, their attack depends on its initialization and has stability issues. Zhao et al. [62] fixes the initialization and stability problems, but the attacker requires the batch size of the clients to be 1. With the same attacker knowledge, Li et al. [63] propose a framework to measure the effectiveness of passive Feature inference attacks on logistic regression models, whose inputs are binary. Geiping et al. [64] and Ren et al. [65] propose different approaches to solve the initialization and stability problems of [12] and their attacks can handle batches of up to 100 and 256 elements, respectively. With the same attacker knowledge, Wei et al. [66] propose an extensive study to measure the capabilities of passive reconstruction attacks focused on recovering images. They also propose a new attack which combines the attacks proposed in [12, 62]. To our knowledge, Jin et al. [67] are the first ones to extend and improve the attack proposed by Zhu and Han [12] to a VFL setting, having the attacker third party-side knowledge. In such setting, the attacker can handle batches of up to 160 elements. When it comes to their HFL setting, the attacker requires server-side knowledge. Their proposed attack seems to be slightly better than the one proposed by Geiping et al. [64], but further experimentation is required to confirm their superiority. The same can be applied to Ren et al. [65], whose comparison with others than Zhu and Han [12] remains undone.

- *Parameter based*: selected clients share their local model parameters with the federated server in the communication rounds. Therefore, the attack surface is the clients' parameters. Focused on reconstructing training images, Hitaj et al. [68] presents a GAN-based active attack, where the key to train the GAN is using the global model as discriminator. The attacker requires client-side knowledge as well as extra client-

side knowledge. The latter gathers the assumption that the target client and the attacker share a label, so that the inference can occur on a non-shared target label. We highlight that the attacker tricks the target client to release more information about the target label, by mislabelling the generated samples of the non-shared target label as the shared label. In the same line, Wang et al. [69] changes the attacker knowledge to server-side knowledge and changes the GAN architecture to a proposed multi-task GAN. To further improve the effectiveness of their attack, the active attacker isolates the target client, so it does not receive global model updates.

Steeping out of GAN-based attacks, Yuan et al. [70] focuses on reconstructing text from natural language processing tasks, particularly, language modelling tasks. The passive attacker is an observer of the federated train procedure, then she requires access to the global model at each communication round and one of the following: (1) to know whether the target client is selected for the communication round or (2) to inject a record into the target client's training data. That is, she requires partial server-side knowledge and optionally partial client-side knowledge. Their proposed attacks rely on the correlation between the privacy exposure and the clients selected in each federated aggregation step.

The popularity of deep learning models in HFL cannot be denied, however in VFL a wider range of machine learning models benefit from this setting. Luo et al. [71] designed passive attacks for decision tree, logistic regression, random forest and neural network models. The attacker requires from the target client the feature names, types and their value range, that is, partial party-side knowledge, in addition to outsider-side knowledge. In two clients VFL setting, focusing on logistic regression and XGBoost models with party-side knowledge Weng et al. [72] propose a passive attacker that can reconstruct the features from the other client. Although, the logistic regression attack also requires partial third party-side knowledge to gather some coefficients.

3.2.2. Membership inference attacks

The main objective of these attacks is to determine whether the provided data was used to train the victim model given a client's model and some data. In federated settings, they are commonly carried out in the *model training phase*. Truex et al. [33] study the application of Membership inference attacks to both non-federated and HFL settings. In the HFL setting, their passive attack, inspired by Shokri et al. [73], considers two different attacker knowledge: (1) where the attacker owns client-side knowledge and (2) where the attacker owns outsider-side knowledge. Shokri et al. [73] show that the first form of knowledge is more effective than the second one. Nasr et al. [36] propose an attack with active and passive versions, each one with two options for attacker knowledge. The attack can have either client-side knowledge or server-side knowledge, where the latter is the most powerful one. Their attack consist in training a meta-classifier on the hidden layers output, the gradients, and outputs of the target client model. Such meta-classifier is a neural network with a custom architecture suited for each part of the internal state of the victim model. In the federated setting, the attack is not as effective as in the centralised

scenario, so two techniques are introduced to boost the effectiveness of the attack. The first one is known as Gradient Ascent. It consists in nullifying the effect of the gradient descent on the instances used to test the attack. As a result, it broadens the difference between the data points used to train the victim model and the data points not used to train the victim model. The second one is known as Client Isolation. The objective of this technique is overfitting the victim model by not sharing with the victim client the global learning model, that is, isolating the victim client from any update. Overfitting makes the victim model retain more information about its training dataset.

As data is a scarce resource, these attacks can be boosted by means of Feature inference attacks to improve the data availability [74, 75, 76]. Zhang et al. [75] is a great example of using a GAN architecture for data augmentation to boost the effectiveness of the passive attacker with client-side knowledge from Nasr et al. [36]. Increasing the attacker knowledge from client-side knowledge to client-side and server-side knowledge, and making the attacker active, Mao et al. [74] propose a similar use of a GAN with an attack inspired by the shadow models attack of Shokri et al. [73]. Chen et al. [76] reduces the attacker knowledge to client-side knowledge and extra client-side knowledge, that is, the labels owned by each client. In addition, the attacker is passive. However, they add a new restrictive assumption, clients do not share any label.

VFL is not free from Membership inference attacks. In a two-client VFL setting, Li et al. [77] proposes a passive attacker with party-side knowledge in a federated binary classification task.

3.2.3. Property inference attacks

This kind of attacks, which are also known as *attribute inference attacks*, aims at extracting whether a property of a client or a property of the population of participants in a FL task, which might be uncorrelated with the main task of the machine learning model, is present in the FL model. In other words, the aim is to infer some property of an individual or the population which is not expected to be shared. An example of inferring an uncorrelated property is the following: consider a machine learning model whose objective is to detect faces, then the objective of the attack is inferring whether there are training images with blue-eyed faces. As stated, we can categorise these attacks according to the target of the attacker:

- *Population distribution*: the attacker tries to infer the distribution of a feature in a population of federated clients. In a federated SGD environment, Wang et al. [78] proposes a set of passive attacks. In conjunction, they can be used to infer the proportion of each label in a communication round. This attacker requires client-side knowledge and partial server-side knowledge, that is, the approximate number of clients selected by the server in a single training round, the average number of labels owned by each participant and the probable number of data samples per label. In a general HFL setting, Zhang et al. [13] reduces the attacker knowledge to outsider-knowledge to perform a passive attack capable of inferring the distribution of a

sensitive attribute in the training population.

- *Individual distribution*: the attacker objective is to reveal whether a target client has a property which might not be related with the main FL task. Mo et al. [79] provide a formal framework to evaluate the property leakage of each layer of a deep learning model in a federated SGD environment. In the same federated environment, Melis et al. [80] develops both passive and active Property inference attacks, whose attacker requires only client-side knowledge. We highlight that the active attack is powered by multitask learning [81]. Sharing the same attacker knowledge, Xu and Li [82] switches the environment to a standard FL setting to propose an attack with passive and active versions. The active attack employs the CycleGAN [83] to reconstruct gradients with the target attribute. Chase et al. [84] propose a Property inference attack by means of a poisoning attack. The poisoning attack requires that the attacker can modify the dataset of the target client, that is, partial client-side knowledge. Additionally, it requires the attacker to have outsider-side knowledge. In a more exotic FL environment, blockchain assisted HFL, Shen et al. [85] propose an active attack with the requirement of server-side knowledge.

4. Defence methods against adversarial attacks: Taxonomy

At the same time that the diversity and complexity of adversarial attacks against FL is enlarging, new defences are emerging to mitigate their malicious effects. While adversarial attacks can be split into disjoint categories, the same is not true for their defences as some of them are effective for more than one type of attack category. Consequently, instead of grouping defences according to the attack defended, we categorise them into three main groups according to the federated scheme they are implemented in: the client, the server or the communication channel. Additionally, we specify for each type of defence the attacks it can defend. In this section, we propose a taxonomy for each of these three groups of defences and highlight the most representative proposals of the state-of-the-art, which is shown in Figure 9.

4.1. Server defences

The federated server is usually assumed to be reliable, because it is a controlled and accessible federated element by FL experts, in contrast to clients that are independent and inaccessible elements. Accordingly, most of defence mechanisms are implemented on the federated server. Within this type of defences, we present the following taxonomy. Note that some defences may combine characteristics of two categories of the taxonomy. In this taxonomy, we have classified the defences according to the category that we consider best represents them.

4.1.1. ***Robust aggregation operators***

The first and most common approach to defend against poisoning attacks to the federated model is to use estimators that are statistically more robust than the mean to outliers or extreme values. Some aggregation operators, such as FedAvg [86], are susceptible to

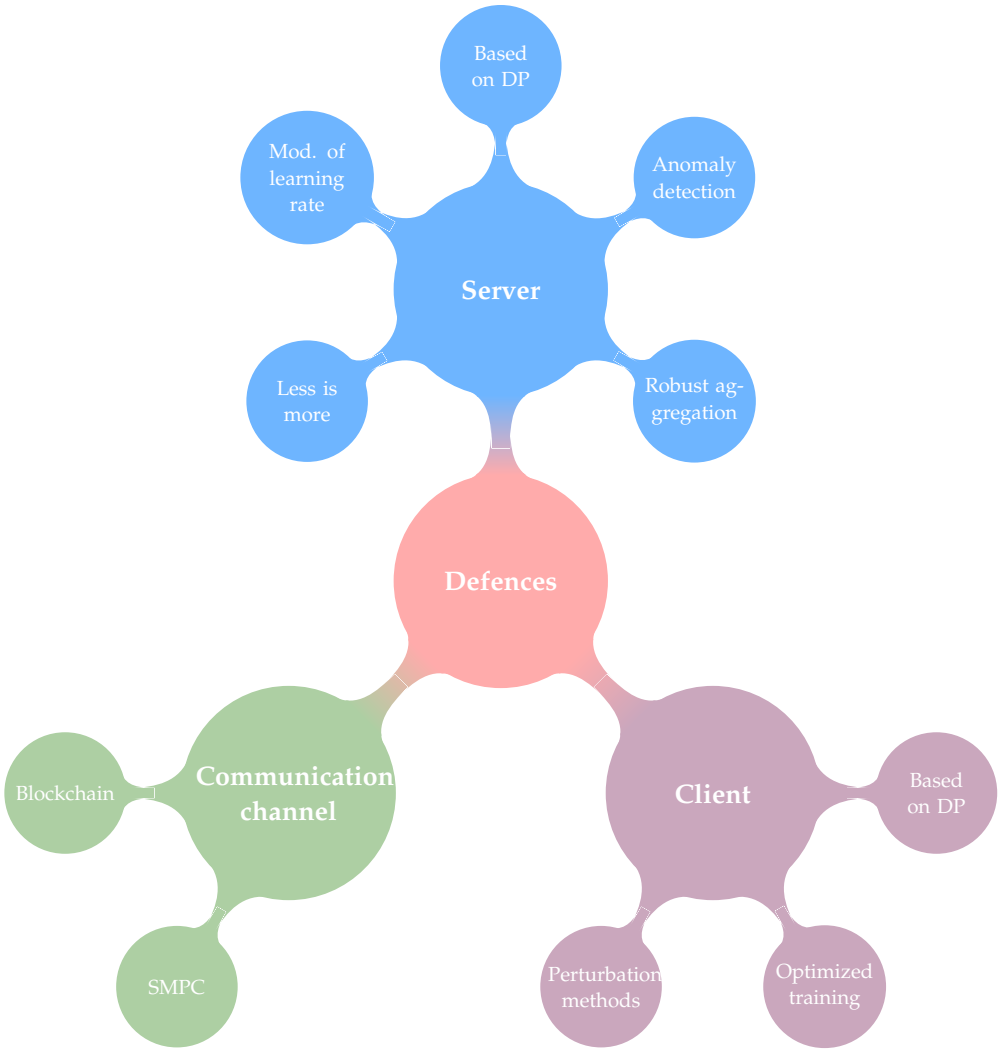

Figure 9: Representation of the taxonomy of defences against adversarial attacks.

outliers. For that reason, many aggregation operators based on more robust estimators have been proposed. We highlight the following ones:

- *Median* [87]: It is a robust-aggregation operator based on replacing the arithmetic mean by the median of the model updates, which choose the value that represents the centre of the distribution.
- *Trimmed-mean* [87]: It is a version of the arithmetic mean, consisting of filtering a fixed percentage k-% of extreme values both below and above the data distribution.
- *Geometric-mean* [88, 89]: It represents the central tendency or the typical value of the data distribution by using the product of their values. In other words, it chooses a reliable vector to represent the local model updates through majority voting.
- *Norm thresholding* [38]: It is a robust-aggregation operator, where the norm of the model updates is clipped to a fixed value, effectively limiting the contribution of

each individual update to the aggregated model.

- *Krum and Multikrum* [90]. This aggregation operator is designed ad-hoc to prevent attacks to the federated model, so it is based on filtering out the model updates of the clients which present and extreme behaviour. For that, it sorts the clients according to the geometric distances of their model updates distributions and chooses the one closest to the majority as the aggregated model. Multikrum incorporates a d parameter, which specifies the number of clients to be aggregated (the first d after being sorted) resulting in the aggregated model.
- *Bulyan* [91]. The authors design an federated aggregation operator to prevent poisoning attacks, combining the MultiKrum federated aggregation operator and the trimmed-mean. Hence, it sorts the clients according to their geometric distances, and according to a f parameter filters out the $2f$ clients of the tails of the sorted distribution of clients and aggregates the rest of them.
- *Adaptive Federated Averaging (AFA)* [92]. Proposal of a defence mechanism against Byzantine attacks based on the weighting of each client using Hidden Markov model by means of the cosine similarity to measure the quality of model updates during training. The authors report that it discards both poor and malicious clients, improving the computational and communication efficiency.
- *Residual-based Reweighting* [93]. They propose an improvement of the median-based aggregation operator combining repeated median regression with the reweighting scheme in Iteratively Reweighted Least Squares (IRLS) based on reweighting each parameter by its vertical distance (residual) to a robust regression line.
- *Sageflow* [14]. A defence based on staleness-aware grouping with entropy-based filtering and loss-weighted averaging, to handle both stragglers and adversaries simultaneously. They establish a theoretical bound to provide key insights into its convergence behaviour.
- *Game-theory approach* [94]. The authors design the aggregation process with a mixed-strategy game played between the server and each client, where the valid actions of each client are to send good or bad model updates while the server can accept or ignore them. They weight the contribution of each client by means of the probability of providing good updates, determined employing the Nash Equilibrium property [95]. The main limitation is that it works only on IID training data distributions, which is unusual for real-world federated data.

4.1.2. ***Anomaly detection***

These defence methods consist in identifying adversarial clients as anomalous data in the distribution of local model updates and remove them from the aggregation. For this purpose, multivariate or adaptations of univariate anomaly detection machine learning techniques are applied.

In Shen et al. [96], the authors propose *AUROR*, a defence mechanism against poisoning attacks in collaborative learning based on K-Means with $k = 2$, thus distinguishing between benign and suspicious clusters. Although it was a promising proposal, the main problem is that in the presence of a non-IID distribution of data between clients it could fail to identify clusters. In Andreina et al. [61], they experiment with different anomaly detection mechanisms and combine the results with adaptive clipping and noise. Along the same lines, in Sattler et al. [97] the authors propose to divide the model updates into clusters according to the cosine distance and Preuveneers et al. [98] proposed an incremental defence based on unsupervised deep learning anomaly detection system integrated in a blockchain process. In a similar vein, Hei et al. [99] proposed an alert filter identification module in the blockchain FL process. Also in a blockchain domain, HoldOut SGD is proposed in [100], which uses the holdout estimation technique in order to select the model updates that are likely to be adversarial ones. It consists in selecting two groups of clients: (1) the ones that use their private data to training in order to send their model updates and (2) a voting committee that use their private data as holdout data for selecting the best model update proposals using a voting scheme. This Graph-based anomaly detection has also been proposed in [51], where the authors propose Sniper, a defence mechanism built upon the graph whose vertices are the updates of the local models and the edges exists only if the two vertices are close enough. They finally identify benign local models by solving a maximum clique problem in this graph. Another example is Nguyen et al. [101], where the authors propose an anomaly based system based on a Gated Recurrent Unit (GRU) and test it on Internet of Things (IoT) specific databases. Along the lines of using deep learning to detect anomalies, Zhao et al. [102] employ GANs by using partial classes data to reconstruct the prototypical samples of client' training data for auditing the accuracy of each client's model.

The main problem with anomaly-based approaches is that the model updates are likely to be very high dimensional, coming from neural networks in most cases. In Tolpegin et al. [50], they propose to apply Principal Components Analysis (PCA) for dimensionality reduction before anomaly detection. In Li et al. [103] they also propose a spectral anomaly detection, which detects abnormal model updates based on their low-dimensional embeddings. The main idea is to embed both original and poisoned samples into a low-dimensional latent space and find these that differs significantly. Although these approaches reduces the problem to a low-dimensional problem, they have the limitation of losing information during the dimensionality reduction.

4.1.3. ***Based on Differential Privacy***

Even though privacy is a topic out of the scope of adversarial attacksto the federated model, DP has been proven to be a viable defence method against these attacks [104, 38]. However, it is also known that DP greatly deceives the performance of the model under circumstances of data imbalance [105, 106], which is expected to happen in most federated scenarios. Applying DP to the aggregation operator overcomes it to some extent. DP-FedAvg [107], also known as Central DP, is a differentially private aggregation

operator which stems from the FedAvg operator. It shares some ideas with the robust-aggregation operators, given that it removes extreme values by clipping the norm of the model updates, like the Norm thresholding operator, and then adds Gaussian noise calibrated to the clip. To provide guarantees of (ϵ, δ)-DP, the order of Gaussian noise required is high enough to reduce significantly the accuracy of the federated task. In Sun et al. [38], they introduce an alternative to Central DP aggregation operator, known as Weak DP, which shares the same aggregation procedure, but it does not guarantee (ϵ, δ)-DP nor any known privacy preserving property. It adds sufficient Gaussian noise to defeat the adversarial attack and preserve the accuracy of the federated task.

4.1.4. ***Modification of the learning rate***

One of the advantages of the server is that it sets the learning rate that controls the weighting between the previous version of the global model and the aggregate of the client model updates by means of

$$G^t = G^{t-1} + \eta \Delta(L_1^t, \ldots, L_n^t) \tag{7}$$

where G^t is the global model in the learning round t, η is the learning rate, Δ the aggregation operator and L_i^t the model update of the client i in the learning round t. It can also decompose η in a vector of learning rates, one per dimension. Thus, the server controls the participation in each dimension of the model updates. This decomposition approach has been used in the literature as a defence mechanism against adversarial attacks to the federated model.

Ozdayi et al. [15] propose *Robust Learning Rate* (RLR) as an improvement of *signSGD* [108]. It is a defence based on adjusting the server's learning rate η, per dimension, at each learning round according to the sign information of the clients model updates. For each dimension, they examine whether the clients agree on the direction of the model update using a predefined threshold. If the agreement is higher than required by the threshold, the learning rate is maintained, otherwise the sign of the learning rate is changed. It can also be combined with other defences, such as those based on DP.

4.1.5. ***Less is more***

Another defence approach in the literature against adversarial attacks to the federated model is based on the fact that original task knowledge will be located in most of the weights in the model, while the weights affected by poisoning attacks will be a small portion of them. Based on this assumption, a post-training defence is proposed in [109], which consists of pruning the resulting global model in order to protect it against attacks that may have taken place during training. Specifically, the authors design a federated pruning method to remove redundant neurons from the neural network and to adjust the outliers of the model. They propose two pruning approaches based on majority vote and ranking vote. The main limitation is that it is usually necessary to perform fine-tuning afterwards on a validation set to compensate for the loss of accuracy caused by pruning.

In [110], the authors highlight that previous works ignore the issue of unbalanced data or assume that the server owns this information. They focus on this issue and propose a practical weight-truncation-based preprocessing method, which achieves quite a balance between model performance and Byzantine robustness. The novel truncation process is based on an element-wise truncation in function of some pre-fixed parameters. Although the choice of parameters is a disadvantage, the authors propose procedures for selecting them.

4.2. Client defences

Server defences assume that the federated server is trusted as a data collector and aggreg ator. However, this assumption might be too strong, therefore there is a requirement for defences when the assumption of a trusted server is removed. In such situation, defences at client level must be deployed and as a consequence, at least a portion of the clients is supposed to be benign. In contrast to server-side protection which protects clients as a whole, client-side defences are thought to be strongest as they provide protection for each client individually.

*4.2.1. **Based on Differential Privacy***

Generally, these defences are designed to defend against server-side privacy attacks, although some may prevent clients from adversarial attacks. Local DP [107] based on the DP-SGD algorithm presented in Abadi et al. [111], is the main client-side defence based on DP. Subsequently authors have proposed improvements to Local DP in terms of DP relaxations, such as the f-DP [112]. Bu et al. [113] applies f-DP to a HFL setting, achieving a better privacy analysis than Abadi et al. [111], that is, it provides a tighter usage of the privacy budget. Its effectiveness against adversarial attacks have been studied [104], and in Bagdasaryan et al. [10] the reduction in performance of this technique has been related to the reduction of the effectiveness of the adversarial attack. Moreover, Cao et al. [114] designed a successful adversarial attack aimed at Local DP protocols for frequency estimation and heavy hitter identification. In order to stop the gradient leakage, that is, privacy attacks in federated SGD settings, Yadav et al. [115], Hao et al. [116] and Wei et al. [117] made the shared gradients differentially private to protect them. If instead of exchanging parameters or gradients in HFL, clients share predictions of unlabelled data, it is possible to apply DP to protect from privacy attacks. Such setting is known as Knowledge Transfer model [118], and it provides privacy with a great preservation of utility using voting based approaches [119, 120, 121].

Regarding defences against privacy attacks based on DP in VFL, Wang et al. [122] propose to perturbate the intermediate outputs shared between parties in the *model training phase* of a Generalized Linear Model. Additionally, such perturbation removes the requirement of a learning coordinator and the necessity of costly Homomorphic Encryption schemes, as they are already private. However, it is a field to be explored in more depth because, to our knowledge, it is the only publication inside it.

Bhowmick et al. [123] step out of the standard Local DP protocol, to relax it and provide

only defence against Feature inference attacks, that is, they assume that the attacker does not have any background data about her victim.

4.2.2. *Perturbation methods*

They are an alternative approach to provide defences against privacy attacks that are not based on DP. Its main aim is to introduce noise to the most vulnerable components of the federated model, such as shared model parameters or the local dataset of each client, to reduce the amount of information an attacker can extract. Zhu and Han [12] not only propose a Feature inference attack, they also propose some defences against it, such as gradient compression, which prunes gradients which are below a threshold magnitude. Lee et al. [124] perturb the local client data with a multitask-based neural network. It preprocesses the data to increase the distance with the original data while preserving useful features for the *model training phase*.

In the same line of multitask based defences, Fan et al. [125] perturb the local training by means of a special loss in conjunction with an additional hidden neural network. Sun et al. [16] perturb only the parameters related to fully connected layers as they build a reconstruction procedure that can effectively reconstruct data from such layers. Zhang and Wang [126] propose to use the technique known as Random Sketching [127] applied to shared client's parameters to defend against client-side privacy attacks. Trying to protect from the same type of client-side attacks, Yang et al. [128] add a kind of perturbation to the parameters that can be removed by the server, so attackers that intercepts them are not able to recover information.

4.2.3. *Optimised training*

The optimisation of the benign clients training may be one way to prevent the federated system from adversarial attacks. Chen et al. [129] propose to perform fine-tuning in benign clients in order to increase the impact of these clients in the aggregation. They decide which clients are benign ones by means of "matching networks", which consist of measuring the similarity between some inputs (the model updates) and a support set (the last global model). This way, they succeed in identifying allegedly benign clients and can conduct fine-tuning. In their experimental study, they succeed at filtering out backdoor tasks at the cost of reducing the performance of the original task.

One of the most recent works in this line presents the client-based defence named *White Blood Cell for Federated Learning* (FL-WBC) [17], which aims to mitigate model poisoning attacks that have already poisoned the global model. The author based the proposal on identifying the parameter space where long-lasting attacks effect on parameters resides and perturb that space during the local training of each client.

The most widespread training approach aimed at preventing adversarial attacks to the federated model is *adversarial training*. These defences consist in taking advantage of the robustness obtained from adversarial training in an FL setting. For example, in [130] the authors propose to use pivotal training, which enables a learning model to pivot on the

sensitive attributes with the aim of making the predictions independent of the sensitive attributes embedded in the training data.

4.3. Communication channel defences

These defences cover the space of secure implementations of FL. They enable multiple clients or parties to perform a global task, assuming the presence of some malicious actors that try to deter it. For our purposes, such actors can be embodied as the attackers that perform some adversarial attacks mentioned before. While the privacy of inputs of the global computing task is preserved, the output is revealed to some parties, if not all. Therefore, the privacy of the output is not assured, although some privacy attacks are stopped because the attacker loses access to the intermediate outputs of the global task such as the parameters or gradients shared by the clients. In other words, these defences are capable of reducing server-side knowledge to partial server-side knowledge, given that the server can only access the aggregated model or the aggregated gradients.

Secure Multi-Party Computation. Secure Multi-Party Computation (SMPC) protocols are tightly related to Secure FL (SFL) protocols [131]. Note that We refer to SFL protocols as FL protocols that attains the security in the simulation-based framework used to formalize the notion of security [132, 133, 134]. SMPC rely on Homomorphic Encryption (HE) as a key component to provide security. Consequently, HE can be regarded as the building blocks of any SMPC protocol. It provides multiple cryptographic primitives which allows for secure computations such as Secret Sharing [135], Zero Knowledge Proofs [136] and Garbled Circuits [137]. Most HE based protocols only support single key encryption, which might pose a risk if the key is compromised, that is, a single point of failure. This situation has been addressed in [138, 139], where the authors have developed SFL systems with multiple encryption keys.

VFL settings heavily rely on SMPC protocols to perform at the beginning of the training the private entity alignment. Additionally, when training and performing inference, partial updates and predictions are shared and the final update and prediction is computed by means of SMPC protocols.

The complexity of SMPC grows with the number of parties involved in the computation. This fact reduces the feasibility of SFL as the number of parties in a FL task can be huge [106]. As a consequence, the idea of full-fledged SMPC protocols that involve the entire federated training procedure are abandoned in favour of SMPC protocols that involve the communication steps in FL. As a remarkable example, a key step in HFL protocols, where SMPC protocols can ensure security and efficiency, is the aggregation step. Bonawitz et al. [140] defined an efficient and robust SMPC protocol for the aggregation procedure and, later on, studied its parameter selection [141]. Similar ideas and improvements have been explored by multiple authors [142, 143, 144].

To provide complete protection for both, adversarial and privacy attacks, some additional protection such as DP must be provided. SFL protocols which include DP as an additional security measure have been developed [145, 146, 147, 148]. In addition, secure

aggregation schemes have been improved in terms of privacy with the addition of DP mechanisms [149, 150, 151, 152] .

Blockchain based FL. In contrast to SFL protocols, Blockchain based FL enables a decentralized FL environment without single point of failure risks and improved scalability [153, 154]. However, this emerging approach inherits the already existing security issues of the blockchain: 51% attacks [155], forking attacks [156], double spending and reentrancy attacks on smart contract [157] amongst others. In addition, it requires a way to encourage users to join the federated tasks to compensate the storage and computational usage [158].

5. Experimental study

The aim of the experimental study is to analyse how attacks behave under certain circumstances and which defences are effective for which attacks, in a comparative way. For this purpose, we choose the highest-impact attack of each kind,[1] according to the previous taxonomies, and we set the same experimental framework for each attack and test the performance of the defences in this framework.

For each attack, we test the effectiveness of the defences in three different classification images datasets:

- EMNIST Digits (Extended MNIST [159])[2] [160]: it is an extension of the handwritten digits dataset, MNIST. It has approximately 400,000 samples, of which 344,307 are training samples and 58,646 are test samples.
- The Fashion MNIST[3] [161]: it contains a balanced set of the 10 different classes of images of clothes, containing 7,000 samples of each class. The dataset thus consists of 70,000 samples, of which 60,000 are training samples and 10,000 test samples.
- The CIFAR-10[4] dataset is a labelled subset of the 80 million tiny images dataset [162]. It consists of 60,000 32x32 colour images in 10 classes, with 6,000 images per class. The classes are: airplace, automobile, bird, cat, deer, dog, frog, horse, ship and truck. There are 50,000 training images and 10,000 test images, which correspond to 1,000 images of each class.

For EMNIST and Fashion-MNIST we employ a standard convolutional network used in Sun et al. [38] depicted in Figure 10: two convolutional layers with 3x3 kernel of 32 and 64 units followed by a 2x2 max pooling layer and a fully connected layer with 128 units with

[1] The implementation of the adversarial attacks considered in the experimental study is the provided by the authors in some cases, and the one developed by the authors of this paper thoroughly following the description of the attack on its corresponding paper.

[2] https://www.nist.gov/itl/products-and-services/emnist-dataset

[3] https://github.com/zalandoresearch/fashion-mnist

[4] https://www.cs.toronto.edu/~kriz/cifar.html

a dropout of 0.5. For the CIFAR-10 dataset, we employ a Transfer Learning approach using an EfficientNetB0 [163] model pretrained on ImageNet. A fully connected layer with 256 units is added to the pretrained model.

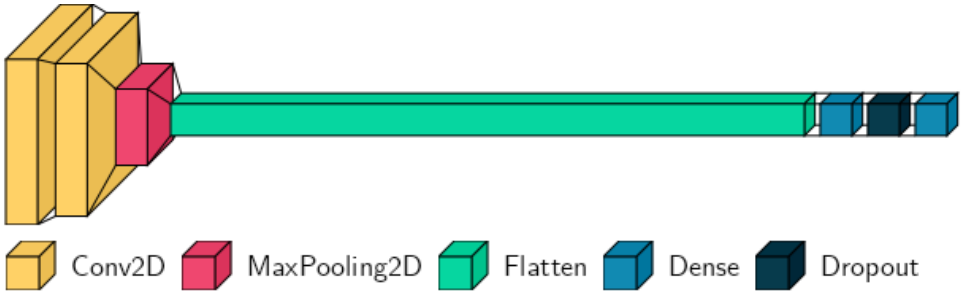

Figure 10: Convolutional network architecture used in the experimental study for processing the EMNIST and Fashion-MNIST datasets.

In the following sections, we analyse the results obtained in the adversarial attacks to the model in Section 5.1 and to the privacy model in Section 5.2.

5.1. Adversarial attacks to the federated model

Although the taxonomy of attacks on the model presented is broad, in this study we analyse those ones most used in the literature. We assume that all the attacks are performed at training time and are multiple and static attacks, that is, the same attack is repeated in each round of learning.

For the whole experimentation of adversarial attacks to the federated model, we consider the following federated distribution of the datasets:

- The federated version of the Digits dataset of EMNIST, *Digits FEMNIST*. The Digits dataset of the federated version of EMNIST, where each client corresponds to an original writer.
- In Fashion MNIST, we set the number of clients to 500 and distribute the training data among them following a non-i.i.d distribution caused by the fact that each client randomly knows a subset of the total number of labels in the set.
- In CIFAR-10, we set the number of clients to 100 and distribute the training data among them following a non-i.i.d distribution caused by the fact that each client randomly knows a subset of the total number of labels in the set.

For all the experiments carried out in this section, we use the accuracy as evaluation measure.

Among the taxonomies presented, the one based on the existence of an specific target objective is probably the most significant. We use this classification to divide this section into the following two subsections, corresponding to untargeted (see Section 5.1.1) and targeted attacks (see Section 5.1.2).

	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**
No attack	**0.965**	**0.965**	**0.962**	0.871	0.871	0.869	0.835	0.835	0.823
FedAvg	0.159	0.421	0.400	0.191	0.366	0.432	0.118	0.143	0.244
Trim.-mean	0.942	0.873	0.837	0.867	0.832	0.861	0.823	0.734	0.822
Median	0.931	0.916	0.909	0.867	0.847	0.858	0.828	0.809	0.828
Krum	0.891	0.870	0.863	0.726	0.719	0.747	0.747	0.761	0.769
MultiKrum (5)	0.913	0.927	0.918	0.840	0.843	0.825	0.816	0.823	0.811
MultiKrum (20)	**0.956**	**0.957**	0.950	**0.872**	**0.872**	0.868	0.843	**0.847**	0.851
Bulyan (f=1)	0.952	0.781	0.580	0.868	0.783	0.787	0.826	0.659	0.645
Bulyan (f=5)	0.936	0.942	**0.951**	0.861	0.865	**0.872**	**0.849**	0.845	**0.854**

Table 1: Mean results for the *label-flipping Byzantine data-poisoning attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

5.1.1. Experimental study of untargeted attacks

Within this kind of attacks, we differentiate between: (1) those attacks that modify clients' training data, producing an alteration of the models (data-poisoning attacks) and (2) those that directly modify the weights of the learning models (model-poisoning attacks). In order to provide a variety of experimentation, we choose the following attacks:

- Data-poisoning attacks: Random label-flipping attack and Out-of-distribution attack (see Section 3.1.3). Clearly, to make these attacks effective, we combine them with model-replacement techniques.
- Model-poisoning attacks: Random weights (see Section 3.1.3), which we also combine with model-replacement.

Regarding the ratio of adversarial clients, we considered different distributions in order to analyse the influence on both the performance of the attack and the defences. In particular, we name x-out-of-n the situation where x of the n clients participating in the aggregation are adversarial ones.

We chose as defences those that have been shown to be state of the art in the literature. In particular, we use the following ones (see Section 4.1):

- Median and Trimmed-mean [87].
- Krum and Multi-Krum [90] with different values for the parameter d, which detail the number of client selected. We consider $d = 5$ and $d = 20$.
- Bulyan [91] different values for the parameter f, which determines the tails of the distribution to be filtered. We consider $f = 1$ and $f = 2$.

In Tables 1, 2 and 3 we show the results of assessing the different defences in label-flipping, out-of-distribution data-poisoning attacks and random weights model-poisoning attack, respectively. In the following, we analyse the behaviour of both attacks and defences in each situation from different effectiveness and behaviour of the defences.

	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**
No attack	**0.965**	**0.965**	**0.962**	0.871	0.871	0.869	0.835	0.835	0.823
FedAvg	0.409	0.440	0.435	0.204	0.366	0.465	0.146	0.192	0.341
Trim.-mean	0.945	0.860	0.853	0.865	0.834	0.831	0.820	0.744	0.740
Median	0.934	0.920	0.914	0.866	0.846	0.845	0.822	0.801	0.807
Krum	0.869	0.866	0.862	0.736	0.706	0.728	0.720	0.731	0.740
MultiKrum (5)	0.916	0.933	0.919	0.849	0.843	0.834	0.830	0.819	0.802
MultiKrum (20)	**0.954**	**0.954**	**0.950**	**0.874**	**0.871**	0.873	**0.860**	**0.851**	**0.852**
Bulyan (f=1)	0.950	0.787	0.581	0.870	0.760	0.693	0.831	0.686	0.555
Bulyan (f=5)	0.935	0.938	**0.950**	0.871	0.865	**0.875**	0.844	0.849	0.848

Table 2: Mean results for the *out-of-distribution Byzantine data-poisoning attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**
No attack	**0.965**	**0.965**	**0.962**	0.871	0.871	0.869	0.835	0.835	0.823
FedAvg	0.099	0.099	0.100	0.100	0.101	0.099	0.099	0.099	0.100
Trim.-mean	0.953	0.103	0.099	0.875	0.100	0.099	0.860	0.099	0.099
Median	0.936	0.935	0.934	0.865	0.861	0.855	0.849	0.866	0.864
Krum	0.831	0.865	0.854	0.715	0.745	0.734	0.718	0.716	0.799
MultiKrum (5)	0.932	0.922	0.919	0.834	0.834	0.827	0.816	0.811	0.816
MultiKrum (20)	**0.956**	**0.957**	0.951	**0.876**	**0.875**	0.867	0.848	**0.848**	**0.853**
Bulyan (f=1)	0.959	0.099	0.099	0.099	0.100	0.099	0.852	0.099	0.099
Bulyan (f=5)	0.937	0.937	**0.951**	0.874	0.869	**0.874**	**0.850**	0.841	0.851

Table 3: Mean results for the *random weights Byzantine model-poisoning attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

Effectiveness of the attack. If we compare the effectiveness of the attack in function of the type of attack, we conclude that the most damaging attack is the random weights attack. In fact, this attack manages to totally confuse the federated model, to the extent that it behaves as a most frequent label classification model. If we focus on the data-poisoning attacks, we get that the label-flipping attack is sightly more effective than the out-of-distribution attack. This is probably because the label-flipping attack learns miss-labelled samples from within the distribution, while the out-of-distribution attack, theoretically, only adds error to samples from outside the distribution.

Regarding the ratio of adversarial clients participating in each aggregation, we found that there are significant differences, being the most effective one carried out by a single adversarial client (1-out-of-30). While this may seem contradictory, there is an explanation. When the attack is carried out by several clients, the boosting factor is divided among these adversarial clients. This divides the strength of the attack among all the adversarial clients, which thus weak the power of the attack, whereas when carried out by a single client, all the boosting is reflected in a single attacker, making it more effective.

Behaviour of the defences. As a general rule, the defences that best mitigate the effect of the attacks are Multikrum (20) and Bulyan (f=5), with MultiKrum (20) standing out slightly.

As we have shown, although Bulyan is presented as an improvement of MultiKrum in combination with trimmed-mean, if the pre-selected clients are benign clients, this truncation is not necessary and even superfluous. On the other hand, the more basic defences such as median and trimmed-mean show good enough behaviour in some experiments, even outperforming MultiKrum and Bulyan with some parameters.

This superiority of the most basic defences over MultiKrum and Bulyan with specific parameters values evidences the high dependence of these defences on the values of the input parameters. This behaviour matches with the assertion of the authors of MultiKrum and Bulyan, they are the most robust defences with the optimal value of the input parameters. This dependency on the values of the input parameters represents an obstacle for the use of this defences, since the value of some parameters is difficult to know, like the number of adversarial clients. A clear example of this problem is Bulyan (f=1) in the random weights Byzantine model-poisoning attack, whose results are comparable to using no defence at all by filtering out too few adversarial clients.

To conclude, untargeted attacks are highly effective, especially those based on model-poisoning, which achieve random behaviour in the federated model. The defences proposed in the literature perform reasonably well, substantially improving the effect of the attacks, even the simplest ones. However, none of them manage to completely dissipate the attack, and the best-performing ones are highly dependent on configuration parameters, so there is still room for improvement in designing defences against Byzantine attacks.

5.1.2. Experimental study of targeted attacks

In order to make a sufficiently broad experimental study, in this section we consider backdoor attacks from the two main groups presented: (1) Input-instance-key strategies and (2) pattern-key strategies. With respect to attacks implementing input-instance-key strategies, we perform a single attack where the target samples correspond to some samples belonging to the adversarial clients for each dataset and associate them with a specific target label. However, with respect to the pattern-key attacks, we choose for each dataset a different static, single and accessory injection pattern.

We chose the state of the art against Backdoor attacks as baselines. In particular, we use the following ones (see Section 5.1.1):

- Median and Trimmed-mean [87].
- Norm-clipping [38].
- Weak Differential Privacy (Weak DP) Sun et al. [38].
- Robust Learning Rate (RLR) Ozdayi et al. [15].

For these defences based on clipping and noise addition, we use M and σ to specify both the clip factor and the noise added, respectively. For the experiments, we choose the values recommended by the authors.

Study of Input-instance-key attacks. In Table 4 we show the results obtained after testing the input-instance-key attack and the different defences. For the implementation, we randomly select some samples of the adversarial clients and associate them with the target label "0". We evaluate the effectiveness of the attack, showing both the original and backdoor performances. We measure the original performance using the mean accuracy in the original test dataset and the backdoor performance by means of the mean accuracy in the set of selected samples for the attack.

			Federated MNIST		**Fashion MNIST**		**CIFAR-10**	
	M	σ	**Original**	**Backdoor**	**Original**	**Backdoor**	**Original**	**Backdoor**
No attack	0	0	**0.965**	-	0.871	-	0.835	-
FedAvg	0	0	0.866	0.823	0.804	0.944	0.612	0.903
Median	0	0	0.944	0.030	**0.875**	0.032	0.861	0.140
Trim.-mean	0	0	0.952	0.025	0.872	0.016	**0.863**	0.133
NormClip	3	0	**0.960**	0.876	0.863	0.144	0.843	0.115
Weak DP	3	2.5e-3	0.937	0.157	0.843	0.119	0.823	0.093
RLR	0.5/0.5/1	1e-4	0.954	**0.012**	0.863	**0.002**	0.853	**0.014**

Table 4: Mean results for the input-instance backdoor attack in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

If we first analyse the effectiveness of the attack (see row of *FedAvg* and *Backdoor* columns) we find the attack is relatively effective, with the result in Fashion MNIST standing out, and always being higher than 0.82 of accuracy. However, if we focus on the stealthiness we note that this type of attack lacks this valuable quality, even affecting the performance on the original task (see row of *FedAvg* and *Original* columns) in 22 points of accuracy (CIFAR-10).

Regarding the performance of the defences, we find that every one of the defences leads to a substantial improvement, both increasing the original task accuracy and reducing the backdoor task accuracy. In addition, we would like to highlight the good performance of the simpler defences, such as trimmed-mean, which achieves very competitive results. If we analyse the state-of-the-art defences (Weak DP and RLR), we found the results to be appropriate, but perhaps a mite disappointing on a complexity-performance trade-off compared to the other defences. Moreover, there are likely to be other M and σ parameters that optimize the results of these defences, but there are not known in advance, which is the main weakness of such parameter-dependent defences.

To conclude, input-instance-key backdoor attacks are considerably powerful, performing better in the backdoor task than in the original one, but being too eye-catching and detrimental to the original task. Moreover, although the defences mitigate the effect of the attack, none of them completely dissipate it, so there is still plenty of scope for further research.

Study of the pattern-key attacks. The Table 5 shows the results obtained after testing the different pattern-key attacks with the considered defences. We implement the attacks by

randomly selecting the adversarial clients and poisoning some of their samples with different patterns. In particular, we use the following patterns of different levels of difficulty according to the number of pixels: (1) one single black pixel for Federated MNIST, (2) a red cross of length 4 for Fashion MNIST (8-pixel pattern) and (3) a white pixel in each of the corners of the image (4-pixel pattern) for CIFAR-10. We evaluate both the effectiveness and the stealthiness of the attack. We measure the stealthiness of the attack by means of the mean accuracy obtained in the original task (Original). We also evaluate the effectiveness of the attack in terms of two additional tests: (1) Backdoor, which contains the poisoned samples of the adversarial clients and (2) Test, which represents the test of the backdoor task and is composed of test samples poisoned following the specific pattern. Therefore, an attack will be more effective the higher performance it obtains in both the original and the backdoor task, while a defence will be better if it manages to maintain the performance in the original task while decreasing as much as possible the performance in the backdoor task.

			Federated MNIST			**Fashion MNIST**			**CIFAR-10**		
	M	σ	**Original**	**Backdoor**	**Test**	**Original**	**Backdoor**	**Test**	**Original**	**Backdoor**	**Test**
No attack	0	0	0.965	-	-	0.871	-	-	0.835	-	-
FedAvg	0	0	0.974	1.0	1.0	0.843	0.999	0.944	0.413	1.0	0.99
Median	0	0	0.954	0.009	0.015	**0.873**	0.067	0.053	0.854	0.193	0.183
Trim.-mean	0	0	0.966	0.011	0.014	0.872	0.052	0.065	0.853	0.194	0.170
NormClip	1	0	**0.968**	0.055	0.053	0.843	0.143	0.164	0.834	0.143	0.131
Weak DP	1	2.5e-3	0.935	0.093	0.0175	0.869	0.053	0.074	**0.859**	0.144	0.170
RLR	1	1e-4	0.962	**0.008**	**0.008**	0.870	**0.020**	**0.031**	0.856	**0.073**	**0.061**

Table 5: Mean results for the *pattern-key backdoor attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

Regarding the effectiveness of the attack without any defence (see row of the *FedAvg* and *Backdoor* and *Test* columns), it reaches a performance of 100% or close to it of accuracy, which shows it harmfulness. However, if we analyse the stealthiness of the attack (see row of the *FedAvg* and *Original* columns), the conclusions depend on the dataset. While in Federated MNIST and Fashion MNIST the performance in the original task is maintained or even improved, the performance in the original task in CIFAR-10 is reduced by up to half.

Regarding the behaviour of the defences, we also obtain a substantial improvement with respect to the scenario without any defence with all of them. As in the untargeted attacks, the simplest defences obtain competitive results, even outperforming the most complex defences in some situations. In general, deciding which defence is superior is not a trivial task. Since it is a matter of achieving the best trade-off between performance in the original task and dissipation of the backdoor attack. For example, RLR achieves in Federated EMNIST the best defence against the attack, but it is more detrimental to performance on the original task. However, in general, we can affirm that it is the best performing defence, standing out particularly in CIFAR-10.

To conclude, pattern-key backdoor attacks are highly threatening attacks, as they achieve almost 100% success in the backdoor task, without, in most cases, harming the performance of the original task. Defences manage to dissipate the effect of the attack in the backdoor task, but in most cases impair performance in the original task. Therefore, in this case, the key is to find the trade-off between mitigating the attack and not harming the performance of the model.

5.2. Privacy attacks

Even tough there is a wide range of privacy attacks, in this section we study those which meet the following requirements:

1. The attack is performed while the federated model is being trained. As a consequence, most defences are aimed to make the training secure from privacy attacks. Alternatively, the defences mask or perturb the shared information to make it less vulnerable.

2. The description of the attack and its setup in its publication is enough to implement it or an implementation which matches the publication is publicly available. The same applies for defences.

The found privacy attacks that matched our requirements allowed us to divide this section into the following two subsections, corresponding to Membership inference attacks (see Section 5.2.1) and Feature inference attacks (see Section 5.2.2), restricted to HFL scenarios.

5.2.1. Experimental study of Membership inference attacks

We choose to implement the federated white-box Membership inference attack from Nasr et al. [36] using the source code publicly available for the white-box centralized setting[1] as there is no public implementation of the federated version. Both clients and server can be the attacker. On the one hand, when the attacker is the client, her objective is to infer the membership of data points belonging to other clients. On the other hand, when the attacker is the server every client is attacked individually, thus the objective is to infer the membership of data points for each client. We mainly focus on their server side attack or global attacker as it is the most powerful, that is, it poses the highest treat to privacy.

We make our federated scenarios the same as the ones proposed in Nasr et al. [36], which represents a small population of clients with big amounts of sensitive data such as banks or hospitals, willing to jointly train a privacy preserving deep learning model. As each client owns great quantities of data, some records can be duplicated among them, that is, the dataset owned by each client is sampled uniformly with replacement from the following datasets: EMNIST, Fashion MNIST and CIFAR-10. Consequently, each of them is divided between 4 clients and each client owns a sample of half the size of the entire dataset, sampled with replacement.

[1] https://github.com/privacytrustlab/ml_privacy_meter

Each federated task is run for 300 rounds where each client shares her local model after each local training epoch, the attacker observes the rounds: 50, 100, 200, 250 and 300. The attack is trained for 100 epochs and the model with best testing accuracy is selected. The attacker training dataset is made of 4000 random samples belonging to each attacked client, 4000 random samples which do not belong to any client and each one is labelled according to its membership to the attacked client. For all the experiments, the batch size is set to 32. We highlight that this federated setup is taken from Nasr et al. [36].

We report the averaged accuracy and AUC of the global attacker in the described settings in Table 6. Note that, the membership inference attack is performed by a binary classifier, therefore the choice of the classification threshold is key to separate between member and non member instances. An attacker with background knowledge may have the ability of selecting a classification threshold that maximizes the separation between member and non members, leading to a greater privacy leakage [164]. While the authors of the attack focus on reporting the accuracy, we have found in our experiments that the AUC metric better shows the capabilities of the attacker, due to the fact that AUC is independent of the classification threshold used to perform the inference. This decision is also driven by the fact that a single classification threshold only represents a possible attacker, therefore we need a way of evaluating every possible attacker, including those with great amounts of background knowledge. We can observe that in our experiments the attack is barely effective, as both accuracy and AUC are close to 0.5. We also highlight that the Gradient Ascent technique does not bring significant performance improvements, probably because it is hard to calibrate. While in the MNIST dataset we see that the attack is not successful, in the other the membership of some instances is revealed, so there is a privacy leak, although it is very small.

We also report the success of the attack with the state-of-the-art defence Local DP in Table 6. The privacy budget in each client of the Local DP is $\epsilon = 3, \delta = 10^{-5}$, which is considered in the literature to be a high privacy budget. We employ the AutoDP framework[1] to calibrate the differentially private Gaussian noise to the privacy budget using Renyi DP [165]. We can observe that this defence is quite successful as it avoids leaking any membership information, thus making the attack classifier behaves randomly.

In Table 7, we can see the accuracy of the federated task with the attack. As noted before, the Gradient Ascent technique degrades the accuracy of the federated task, mainly due to the fact that some of the instances which were ascent belong to the federated test set. While this is true for the MNIST and Fashion MNIST datasets, it is not true for the CIFAR-10 dataset. It might be because of the transfer learning approach used for this dataset being more resilient to gradient direction changes. As expected, DP based defences reduce the accuracy of the federated task. The smallest reduction of federated task accuracy is achieved with the CIFAR-10 dataset, which confirms that the transfer learning approach is more resilient to gradient changes, moreover the Gradient Ascent technique

[1] https://github.com/yuxiangw/autodp

	Without Local DP defence				With Local DP defence			
	Client Isolation		Client Isolation + Gradient Ascent		Client Isolation		Client Isolation + Gradient Ascent	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
MNIST	0.501	0.502	0.489	0.502	0.500	0.497	0.496	0.500
Fashion MNIST	0.513	0.546	0.511	0.516	0.500	0.499	0.497	0.500
CIFAR-10	0.540	0.551	0.500	0.528	0.500	0.500	0.500	0.500

Table 6: Accuracy and AUC of the global federated attack from Nasr et al. [36] with and without Local DP defence.

does not change significantly the accuracy when applied.

	Without Local DP defence		With Local DP defence	
	Client Isolation	Client Isolation + Gradient Ascent	Client Isolation	Client Isolation + Gradient Ascent
MNIST	0.990	0.100	0.672	0.100
Fashion MNIST	0.910	0.100	0.579	0.100
CIFAR-10	0.862	0.862	0.686	0.668

Table 7: Federated task accuracy while the global federated attack from Nasr et al. [36] is performed with and without Local DP defence.

In this experimental study, we have explored the performance of a Membership inference attack on a federated setting of few clients with big amounts of data. We have found that the success of the attack is small, even though the membership of some instances was revealed. The DP based defence stopped these leakages of privacy, at the cost of a considerable reduction of the federated task accuracy. Additionally, we have found that using a transfer learning approach might reduce the impact of DP in the federated task accuracy while also being resilient to the Gradient Ascent technique which have drastically reduced the federated task accuracy with the other datasets and deep learning approaches.

5.2.2. Experimental study of Feature inference attacks

We study multiple gradient based Feature inference attacks, which use stolen gradients from the federated training procedure, particularly we focus on the attacks described in Zhu and Han [12], Geiping et al. [64], Wei et al. [66]. In order to do it, we use the code provided with each publication, which is publicly available.[1,2,3]

The federated scenario which fits these attacks is the following: clients with very little data, such as IoT devices or smartphones, which run a federated task where they share gradients from small batches. We study under which circumstances we can reconstruct

[1] https://github.com/mit-han-lab/dlg
[2] https://github.com/JonasGeiping/invertinggradients
[3] https://github.com/git-disl/CPL_attack

images from gradients. Our study focus on three aspects to evaluate the success of these attacks:

- **The success rate**. The approximate probability of convergence of each attack. The majority of the attacks studied in this section are known to have stability issues, that is, their convergence greatly depends on the initialization seed used to bootstrap them. For Wei et al. [66] and Zhu and Han [12], we choose as initialization a geometric pattern which improves both convergence rate and speed. It consists in covering a small portion of the initialization space with a random image and duplicate it to fill the feature space. In our experiments, we choose $1/4$ of the feature space as in Wei et al. [66]. For the attack of Geiping et al. [64], we choose random initialization, as it does not seem to be affected by the choice of the initialization pattern.
- **The training stage of the local model at which the attack can succeed**. Most of the studied attacks consider an untrained model as they claim that the attack can run at any point of the training procedure, however this claim does not seem to have a lot of experimental support. As a consequence, we want to confirm such claims and find whether the stage of training of the local model is relevant to the success of the attack.
- **The success of the defences against Feature inference attacks**. We study the performance of two state-of-the-art defences: gradient compression and the addition of Gaussian noise. Which are known to thwart the effectiveness of the attack from Zhu and Jin [166], so we evaluate whether these defences are also applicable to the other attacks.

We begin our study analysing the success rate of each attack, as they are known to suffer from stability issues [66]. We run each attack with gradients from an untrained simple convolutional model *LeNet* [167] as in [66, 12] with a batch size of 1. Each attack is run until one of the following conditions is satisfied:

- Success condition: for the attacks [66, 12], we consider that the attack is successful if the Mean Square Error (MSE) with respect to the target image to reconstruct is smaller than 0.5 and the Multi-Scale Structural Similarity (SSIM) [168] is greater than 0.5. The purpose of these criterions is twofold, the former ensures that the reconstructed image is close enough to the target image and the latter ensures that the reconstructed image is perceptibly similar to the target image.
- Failure condition: if the maximum number of epochs set for the attack is reached without satisfying any of the conditions stated below, then we consider the attack is marked as a failure. In other words, the attack as failed to converge.

Additionally, we want to study whether the training stage at which the attack is performed is relevant. To do so, we run each attack at different moments of the local training process: before any training, after 1, 5 and 10 rounds of training. Each attack is going to

try to reconstruct an image that belongs to their training set, but it has not been used to train the model previously. We report the success rate of each attack across 25 runs, using the same end conditions specified before.

The experimental results of the study of **the success rate** and **the training stage of the local model at which the attack can succeed** are shown in Tables 8, 9 and 10. First, we highlight the results from Table 10 that show that the attack from Geiping et al. [64] is independent of the considered training stage of the local model. The same is not true for the results in Tables 8 and 9. In its first column of results, we can see that the attacks have almost no issues to converge when the local model is not trained, so we can conclude that if the appropriate initialization method is chosen, the attacks are almost 100% guaranteed to converge. If we observe the remaining columns of the Tables 8, 9, the results change considerably. The attack from Wei et al. [66] (Table 9) has slightly better convergence rates than the attack from Zhu and Han [12] (Table 8), both show a similar trend: the more trained is the model, the harder it is for the attacks to achieve success.

The complexity of the dataset has an important role in the success of the attacks from [66] and Zhu and Han [12]. Both EMNIST and Fashion-MNIST are considered easier datasets, as there are many works that achieve high training accuracy after few epochs of training [169, 170]. The same is not true for CIFAR-10, as more complex models are required to achieve a reasonable accuracy [171, 172]. EMNIST and Fashion-MNIST images are hard to reconstruct after 1 training epoch, that is, the gradients after 1 training epoch leak little information about the datasets. An example of such difficulties is shown in Figure 11. In contrast, in CIFAR-10 the training model takes longer to converge and the gradients leaks a lot of information, even after 10 epochs of training. The main reason that allow us to understand this behaviour is the fact that both attacks try to mimic the structure and content of the shared gradient (that is, minimizing the MSE between the shared gradient and the reconstructed image), so the more information is stored in the gradient, the easier is the reconstruction process. In other words, gradients that more significantly change the weights of the model make the reconstruction process easier. This is not true for the attack from [64], as its objective is to minimize the cosine similarity between gradient vectors.

Dataset	Before training	After 1 training epoch	After 5 training epochs	After 10 training epochs
EMNIST	1	0	0.04	0
Fashion-MNIST	1	0.28	0	0.08
CIFAR-10	0.96	0.80	0.60	0.68

Table 8: Success rate of 25 trials of reconstructing an image from a shared gradient of a local model with the attack from Zhu and Han [12]. We run the attack at different stages of training of the local model. *Before training* means that the local model has not been trained at all.

Dataset	Before training	After 1 training epoch	After 5 training epochs	After 10 training epochs
EMNIST	1	0	0	0
Fashion-MNIST	1	0.32	0.12	0.24
CIFAR-10	1	0.96	0.84	0.80

Table 9: Success rate of 25 trials of reconstructing an image from a shared gradient of a local model with the attack from Wei et al. [66]. We run the attack at different stages of training of the local model. *Before training* means that the local model has not been trained at all.

Figure 11: From left to right, reconstruction using the attack of Wei et al. [66] of an image with label 0 from Fashion-MNIST dataset which correspond to the t-shirt/top category, after 1, 5 and 10 epochs of local training.

Dataset	Before training	After 1 training epoch	After 5 training epochs	After 10 training epochs
EMNIST	1	1	1	1
Fashion-MNIST	1	1	1	1
CIFAR-10	1	1	1	1

Table 10: Success rate of 25 trials of reconstructing an image from a shared gradient of a local model with the attack from Geiping et al. [64]. We run the attack at different stages of training of the local model. *Before training* means that the local model has not been trained at all.

To end our study, we study the performance of two state-of-the-art defences:

- Gradient compression with 20% sparsity.
- The addition of Gaussian noise with variance of 10^{-2}.

We run each attack with defences 25 times with a batch size of 1, with the model untrained and report the success rate of each attack.

	Attack of Zhu and Han [12]		Attack of Wei et al. [66]		Attack of Geiping et al. [64]	
Dataset	Gaussian noise	Gradient compression	Gaussian noise	Gradient compression	Gaussian noise	Gradient compression
EMNIST	0	0	0	0	0	0.04
Fashion-MNIST	0	0	0	0	0	0.48
CIFAR-10	0	0	0	0	0	0.12

Table 11: Success rate of 25 trials of the reconstruction attacks from [12], [66] and Geiping et al. [64] with Gaussian noise and Gradient compression defences.

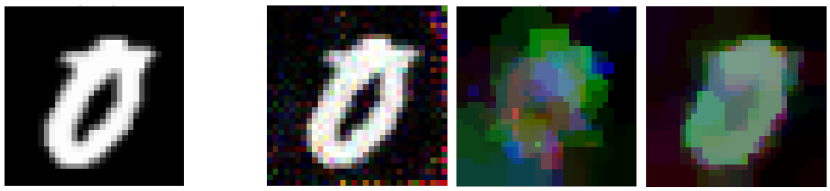

Figure 12: From left to right, reconstruction using the attack of Geiping et al. [64] of an image with label 0 from EMNIST dataset, without any defence, with Gaussian noise defence and with Gradient compresion defence.

In Table 11, we can observe the stunning performance of both defences as they completely stop the attacks of [12, 66] from achieving success. While the addition of Gaussian noise of this magnitude is known to reduce the performance of the task [12], the gradient compression defence can handle higher compression rates without significantly hurting performance [173]. When it comes to the attack of Geiping et al. [64], we find that the Gaussian noise defence is as effective as in the other attacks. This might be due to the differentially private properties of the Gaussian noise. However, the Gradient compression defence fails to completely stop the attack of Geiping et al. [64]. Specially for the Fashion-MNIST dataset, where almost half of the times the attack succeeded. An example of a reconstruction trial with and without defences is shown in Figure 12, it gives an hint of the behaviour of the attack. Gradient compression is the worst performing defence, probably due to the fact that compressing the gradient does not affect the task of minimizing the cosine similarity between the shared and the reconstructed image gradient.

In conclusion, the Feature inference attacks studied in this section pose a high risk to privacy, as there are many attacks that succeed at extracting private information from gradients. Luckily, there are defences that can thwart the success rate of the attacks and provide privacy without changing significantly the performance of the trained model. However, this is not true for all the analysed attacks, there is still room for improvement as the attack from Geiping et al. [64] seems to be able to escape them in some situations. Additionally, this threat is not only related to FedSGD scenarios, it is also related to federated scenarios where parameters are exchanged between clients.

6. Guidelines for the application of defences against adversarial attacks

Due to the large number of attacks identified, and the wide variety of defences proposed in the literature, it can be difficult to choose which type of defence is appropriate for each situation. Moreover, most defences are designed with the objective of defending against a particular adversarial attack. However, as a collateral benefit, they can prevent the success of other types of adversarial attacks.

In this section we provide some guidelines in terms of a summary of which categories of defences work to defend the identified categories of attacks, specifying the degree to which they do so.

In Table 12 we summarize which categories of defences are able to defend against attacks to the model and privacy attacks, respectively. For the sake of clarity, we represent the relationship between categories of attacks and categories of defences. Hence, when we affirm that a category of defence is able to defend against a category of attacks means that there are at least one defence belonging to that category who is able to defend against them.

In this line, we differentiate between:

- Complete defence ●: the defence category is able to stop the attacks of the attack category.
- Partial defence ●: the defence category is able to significantly reduce the performance of the attacks of the attack category but not stop it.
- No defence ●: the performance of the attacks of the attack category is not affected significantly by the defence category.
- Unknown defence ●: there is neither enough experimentation available nor theoretical support to assess the behaviour of the attacks of the attack category with the defence category.

		Attacks to the federated model		**Privacy attacks**		
		Untargeted	**Targeted**	**Property**	**Membership**	**Feature**
Server defences	**Mod. of learning rate**	●	●	●	●	●
	Robust agg.	●	●	●	●	●
	Anomaly detection	●	●	●	●	●
	Based on DP	●	●	●	●	●
	Less is more	●	●	●	●	●
Client defences	**Based on DP**	●	●	●	●	●
	Optimized training	●	●	●	●	●
	Perturbations methods	●	●	●	●	●
Comm. channel	**Blockchain**	●	●	●	●	●
	SMPC	●	●	●	●	●

Table 12: Summary of application of the defences to adversarial attacks, both attacks to the federated model and privacy attacks.

The summary of the state of the art provided in the Table 12 allows us to draw the following conclusions:

1. In general, defences based on DP, which are designed to defend against privacy attacks, partially defend against attacks to the model, specially those based on DP,

but not the other way around.

2. Broadly speaking, the defence against attacks to the model is more settled than the defences against privacy attacks. In particular, for property inference attacks, there is no defence considered as complete.

3. There is still a long way to go in designing defences to prevent attacks in FL and, crucially, to find a defence that prevents from all types of attacks at the same time.

7. Lessons learned

Based on the extensive research and analysis of the available works, we have built up the taxonomy proposed in this paper. However, what has been learned goes beyond this. To sum up, the lessons learned are:

1. The identification of vulnerabilities in the form of adversarial attacks and the proposal of defences against them in FL is a field of research in continuous development. The volume to date of scientific work covering these challenges is growing and is not likely to diminish in the coming years.

2. Attacks to the federated model are easier to defend against than the privacy-attacks. However, they have shown much greater effectiveness, with even the simplest attacks being detrimental to the model.

3. Privacy attacks require very peculiar settings to achieve a reasonable success, that is, most of the assumptions required to perform them are very hard to achieve in real FL scenarios. Such scenarios are usually bounded by the lack of the following resources: data, raw computing power and time.

4. Most defences against inference attacks, although designed for inference attacks, dissipate the performance of attacks against the federated model, but not the other way around. Therefore, the use of DP-inspired mechanisms will be crucial if we want to defend against generic category attacks.

5. The implementation of defences based on DP and based on perturbation methods require extensive fine-tuning in order to provide a nice trade-off between privacy and performance. Most of the defences require access to big computational resources or they will be too slow to apply. Therefore, such defences might not be suitable for FL settings with low power devices. Additionally, to our knowledge, there is not a consensus about how to measure the trade-off between privacy and performance.

To finish, as a fundamental lesson learned is that the field of adversarial attacks and defences in FL is a research area in steady development, which is not expected to change in the forthcoming years. There are still many vulnerabilities which need to be faced in order to ensure a truly secure and privacy-preserving learning environment.

8. Challenges of addressing federated learning threats

Regarding the previous lessons learnt, we identified the following challenges that the field will have to face up in the next years.

Defences vs. attacks. An identified trend is that for each defence proposed, it is possible to identify a vulnerability that can be turned into an adversarial attack, and vice versa. Therefore, one of the biggest challenges is to find both: (1) all vulnerabilities present in a FL scenario that an attacker could exploit, and (2) a defence effective enough to defend against any attack. For the time being, this seems a long way off, as the different perspectives from which both problems have been approached are ad hoc to the type of attack to be identified or defended. From our point of view, the study of defences is crucial, since the final goal is to achieve a secure, robust and private learning environment. Along this vein, the optimal defence will be the one that combines the best proposals in each of the categories, in such a way that it manages to defend against all types of attacks while maintaining performance in the original task. There are existing approaches that combine client's filtering with noise addition [15], although there is still a long way to go.

Trade-off in defences. In most defences, we find that it is difficult to strike a trade-off between preventing the model from attacking and not impairing performance in the original task. For example, in those based on DP, we find that in order to ensure data privacy, a large amount of noise has to be added, which significantly impairs the performance of the model [105]. Therefore one of the main challenges would be the development of more efficient DP methods, and the extension of DP to defences against all adversarial attacks. This situation also occurs in defences based on client filtering when more clients than necessary are filtered out, thus losing information in the aggregation process. In short, it is difficult to strike a trade-off between preventing an attack and not losing or poisoning the information received by clients.

Non-IID assumptions. The non-IID nature of the training data distributed among clients often makes it difficult to differentiate between adversarial clients and those who have had a very different from the rest, but still valuable, learning process. One common approach is to use anomaly detection algorithms suitable for non-IID distributions [174] or approaches which not rely on data distribution [109], however, there are still problems in differentiating between clients with a highly skewed distribution and adversarial clients in most cases.

Generalised FL. The vast majority of adversarial attacks have been identified for HFL. In particular, the adversarial attacks to the federated model. Although there is already existing work on privacy attacks in VFL [77], there is still a long way to go in identifying and analysing the vulnerabilities in terms of leakage of information of attacker possibilities in other categories of FL wich are becoming widely used such as VFL or FTL [175]. Therefore, we believe that in the coming years, work on identifying attacks for VFL and FTL and the research in defences against them will take centre stage.

Combination with other trends. While ensuring data privacy is one of the main goals of FL, there are other desirable features. For example, some of the most popular trends are Personalised FL [176] or fairness in FL [177]. We believe that, at the end of the day, data security and privacy must be a requirement in all other approaches. Therefore, several future works will address this issue as a cross-cutting objective while developing proposals for more concrete desirable features. For example, a method of personalised FL that is secure against adversarial attacks.

9. Conclusions

FL emerges as a solution to the computational costs and privacy-preserving demands of the most groundbreaking ML. However, this new learning paradigm brings new challenges, particularly in terms of adversarial attacks and defending against them. Hence, several proposals of new adversarial attacks or adaptations of centralised ones as well as defences ad hoc to these attacks have been proposed in the recent years.

We proposed several taxonomies according to different criteria that eases the knowledge of the wide field of FL threats. In addition, we conducted an extensive experimental study which leads us to propose a guidelines for the application of defences against adversarial attacks, and to highlight a set of lessons learned and challenges related to FL threats.

We conclude that the study of FL threats is an ongoing field of research, due to its importance in ensuring FL as a robust machine learning paradigm that safeguards data privacy. There are still several challenges to be faced and directions to be studied in order to identify additional threats (or vulnerabilities) of FL, as well as the appropriate mechanisms to make it a resilient and robust learning paradigm against those threats.

Acknowledgments

This research work is partially supported by the Trust-ReDaS (PID2020-119478GB-I00), the FedDAP (PID2020-116118GA-I00) and the EQC2018-005084-P projects from the Spanish Government, and a grant from the European Regional Development Fund (ERDF). Nuria Rodríguez Barroso and Eugenio Martínez Cámara were supported by the Spanish Government fellowship programmes Formación de Profesorado Universitario (FPU18/04475) and Juan de la Cierva Incorporación (IJC2018-036092-I) respectively.

References

[1] S. Al-Kuwari, Multiple Perspectives on Artificial Intelligence in Healthcare: Opportunities and Challenges, Springer International Publishing, 2021, pp. 65–77.

[2] F. Boissay, T. Ehlers, L. Gambacorta, H. S. Shin, The Palgrave Handbook of Technological Finance, Springer International Publishing, 2021, pp. 855–875.

[3] M. Goddard, The EU General Data Protection Regulation (GDPR): European regulation that has a global impact, International Journal of Market Research 59 (2017) 703–705.

[4] O. Gómez-Carmona, D. Casado-Mansilla, F. A. Kraemer, D. L. de Ipiña, J. García-Zubia, Exploring the computational cost of machine learning at the edge for human-centric internet of things, Future Generation Computer Systems 112 (2020) 670–683.

[5] J. Zhang, D. Tao, Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things, IEEE Internet of Things Journal 8 (2021) 7789–7817.

[6] C. Ma, J. Konečný, M. Jaggi, V. Smith, M. Jordan, P. Richtárik, M. Takáč, Distributed optimization with arbitrary local solvers, Optimization Methods and Software 32 (2017) 813–848.

[7] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, H. Yu, Federated Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, 2019.

[8] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, J. D. Tygar, Adversarial machine learning, Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (2011) 43–58.

[9] N. Dalvi, P. Domingos, Mausam, S. Sanghai, D. Verma, Adversarial classification, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, p. 99–108.

[10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, 2020, pp. 2938–2948.

[11] M. Fang, X. Cao, J. Jia, N. Z. Gong, Local model poisoning attacks to byzantine-robust federated learning, in: USENIX Security Symposium, 2020, pp. 1605–1622.

[12] L. Zhu, S. Han, Deep leakage from gradients, in: Federated learning, Springer, 2020, pp. 17–31.

[13] W. Zhang, S. Tople, O. Ohrimenko, Leakage of dataset properties in Multi-Party machine learning, in: 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2687–2704.

[14] J. Park, D.-J. Han, M. Choi, J. Moon, Sageflow: Robust federated learning against both stragglers and adversaries, in: Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS), 2021.

[15] M. S. Ozdayi, M. Kantarcioglu, Y. R. Gel, Defending against backdoors in federated learning with robust learning rate, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 9268–9276.

[16] J. Sun, A. Li, B. Wang, H. Yang, H. Li, Y. Chen, Soteria: Provable Defense Against Privacy Leakage in Federated Learning From Representation Perspective, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9311–9319.

[17] J. Sun, A. Li, L. DiValentin, A. Hassanzadeh, Y. Chen, H. Li, FL-WBC: enhancing robustness against model poisoning attacks in federated learning from a client perspective, in: Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS), 2021.

[18] A. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: Proceedings of the 36th International Conference on Machine Learning (ICML), volume 97, 2019, pp. 634–643.

[19] Y. Fraboni, R. Vidal, M. Lorenzi, Free-rider attacks on model aggregation in federated learning, in: Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130, 2021, pp. 1846–1854.

[20] V. Shejwalkar, A. Houmansadr, P. Kairouz, D. Ramage, Back to the drawing board: A critical evaluation of poisoning attacks on federated learning, in: 43rd IEEE Symposium on Security and Privacy, 2022.

[21] D. Enthoven, Z. Al-Ars, An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies, Studies in Computational Intelligence 965 (2021) 173–196.

[22] M. Asad, A. Moustafa, C. Yu, A critical evaluation of privacy and security threats in federated learning, Sensors (Switzerland) 20 (2020) 1–15.

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

[24] L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, P. S. Yu, Privacy and Robustness in Federated Learning: Attacks and Defenses, CoRR abs/2012.06337 (2020).

[25] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Machine Learning and Knowledge Discovery in Databases, 2013, pp. 387–402.

[26] L. Lyu, H. Yu, J. Zhao, Q. Yang, Threats to Federated Learning, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12500 LNCS (2020) 3–16.

[27] M. S. Jere, T. Farnan, F. Koushanfar, A Taxonomy of Attacks on Federated Learning, IEEE Security and Privacy 19 (2021) 20–28.

[28] N. Bouacida, P. Mohapatra, Vulnerabilities in Federated Learning, IEEE Access 9 (2021) 63229–63249.

[29] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology 10 (2019) 12:1–12:19.

[30] N. Rodríguez-Barroso, G. Stipcich, D. Jiménez-López, J. A. Ruiz-Millán, E. Martínez-Cámara, G. González-Seco, M. V. Luzón, M. Ángel Veganzones, F. Herrera, Federated learning and differential privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy, Information Fusion 64 (2020) 270 – 292.

[31] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Foundations and Trends® in Theoretical Computer Science 9 (2014) 211–407.

[32] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography, 2006, pp. 265–284.

[33] S. Truex, L. Liu, M. E. Gursoy, L. Yu, W. Wei, Demystifying membership inference attacks in machine learning as a service, IEEE Transactions on Services Computing (2019).

[34] C. Fung, C. J. M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, CoRR abs/1808.04866 (2018).

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

[36] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, IEEE Symposium on Security and Privacy (SP) (2019) 739–753.

[37] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, CoRR abs/1712.05526 (2017).

[38] Z. Sun, P. Kairouz, A. T. Suresh, H. B. McMahan, Can you really backdoor federated learning?, CoRR abs/1911.07963 (2019).

[39] A. N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: Proceedings of the 36th International Conference on Machine Learning (ICML), volume 97, 2019, pp. 634–643.

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

[41] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, J. Liu, Data poisoning attacks on federated machine learning, IEEE Internet of Things Journal PP (2021) 1–1.

[42] C. Xie, K. Huang, P.-Y. Chen, B. Li, Dba: Distributed backdoor attacks against federated learning, in: International Conference on Learning Representations, 2020.

[43] A. Salem, R. Wen, M. Backes, S. Ma, Y. Zhang, Dynamic backdoor attacks against deep neural networks, 2021.

[44] Y. Liu, Z. Yi, T. Chen, Backdoor attacks and defenses in feature-partitioned collaborative learning, CoRR abs/2007.03608 (2020).

[45] M. Fang, X. Cao, J. Jia, N. Z. Gong, Local model poisoning attacks to byzantine-robust federated learning, 29th USENIX Security Symposium (2020).

[46] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 119–129.

[47] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (1982) 382–401.

[48] S. Hu, J. Lu, W. Wan, L. Y. Zhang, Challenges and approaches for mitigating byzantine attacks in federated learning, CoRR abs/2112.14468 (2021).

[49] Y. Fraboni, R. Vidal, M. Lorenzi, Free-rider attacks on model aggregation in federated learning, in: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS), 2021, pp. 1846–1854.

[50] V. Tolpegin, S. Truex, M. Gursoy, L. Liu, Data Poisoning Attacks Against Federated Learning Systems, 2020, pp. 480–501.

[51] D. Cao, S. Chang, Z. Lin, G. Liu, D. Sun, Understanding distributed poisoning attack in federated learning, in: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), 2019, pp. 233–239.

[52] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, Y. Liu, Lomar: A local defense against poisoning attack on federated learning, IEEE Transactions on Dependable and Secure Computing (2021) 1–1.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, volume 27, 2014, pp. 2672–2680.

[54] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, S. Yu, Poisongan: Generative poisoning attacks against federated learning in edge computing systems, IEEE Internet of Things Journal 8 (2021) 3310–3322.

[55] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2019, pp. 374–380.

[56] S. Fort, J. Ren, B. Lakshminarayanan, Exploring the limits of out-of-distribution detection, CoRR abs/2106.03004 (2021).

[57] P. W. Koh, J. Steinhardt, P. Liang, Stronger data poisoning attacks break data sanitization defenses, CoRR abs/1811.00741 (2018).

[58] X. Xu, J. Wu, M. Yang, T. Luo, X. Duan, W. Li, Y. Wu, B. Wu, Information leakage by model weights on federated learning, in: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, Association for Computing Machinery, 2020, p. 31–36.

[59] G. Costa, F. Pinelli, S. Soderi, G. Tolomei, Covert channel attack to federated learning systems, CoRR abs/2104.10561 (2021).

[60] J. Konečný, H. McMahan, F. Yu, P. Richtarik, A. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, in: NIPS Workshop on Private Multi-Party Machine Learning, 2016.

[61] S. Andreina, G. A. Marson, H. Möllering, G. O. Karame, Baffle: Backdoor detection via feedback-based federated learning, in: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021, pp. 852–863.

[62] B. Zhao, K. R. Mopuri, H. Bilen, idlg: Improved deep leakage from gradients, CoRR abs/2001.02610 (2020).

[63] Z. Li, Z. Huang, C. Chen, C. Hong, Quantification of the leakage in federated learning, CoRR abs/1910.05467 (2019).

[64] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller, Inverting Gradients - How easy is it to break privacy in federated learning?, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 16937–16947.

[65] H. Ren, J. Deng, X. Xie, GRNN: Generative Regression Neural Network – A Data Leakage Attack for Federated Learning, CoRR abs/2105.00529 (2021).

[66] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, Y. Wu, A framework for evaluating client privacy leakages in federated learning, in: European Symposium on Research in Computer Security, 2020, pp. 545–566.

[67] X. Jin, R. Du, P.-Y. Chen, T. Chen, CAFE: Catastrophic Data Leakage in Vertical Federated Learning, CoRR 1412.6830 (2021).

[68] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the GAN: information leakage from collaborative deep learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 603–618.

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

[70] X. Yuan, X. Ma, L. Zhang, Y. Fang, D. Wu, Beyond Class-Level Privacy Leakage: Breaking Record-Level Privacy in Federated Learning, IEEE Internet Things J. 4662 (2021) 1–11.

[71] X. Luo, Y. Wu, X. Xiao, B. C. Ooi, Feature inference attack on model predictions in vertical federated learning, in: Proc. - Int. Conf. Data Eng., volume 2021-April, 2021, pp. 181–192.

[72] H. Weng, J. Zhang, F. Xue, T. Wei, S. Ji, Z. Zong, Privacy leakage of real-world vertical federated learning, CoRR abs/2011.09290 (2020).

[73] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 3–18.

[74] Y. Mao, X. Zhu, W. Zheng, D. Yuan, J. Ma, A Novel User Membership Leakage Attack in Collaborative Deep Learning, in: 2019 11th Int. Conf. Wirel. Commun. Signal Process. WCSP, 2019, pp. 1–6.

[75] J. J. Zhang, J. J. Zhang, J. Chen, S. Yu, GAN Enhanced Membership Inference: A Passive Local Attack in Federated Learning, in: IEEE Int. Conf. Commun., volume 2020-June, 2020, pp. 5–10.

[76] J. Chen, J. Zhang, Y. Zhao, H. Han, K. Zhu, 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), 2020, pp. 1–9.

[77] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, C. Wang, Label Leakage and Protection in Two-party Split Learning, CoRR abs/2102.08504 (2021).

[78] L. Wang, S. Xu, X. Wang, Q. Zhu, Eavesdrop the Composition Proportion of Training Labels in Federated Learning, CoRR abs/1910.06044 (2019).

[79] F. Mo, A. Borovykh, M. Malekzadeh, H. Haddadi, S. Demetriou, Layer-wise Characterization of Latent Information Leakage in Federated Learning, CoRR abs/2010.08762 (2020).

[80] L. Melis, C. Song, E. De Cristofaro, V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, IEEE Symposium on Security and Privacy (2019) 691–706.

[81] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Transactions on Knowledge and Data Engineering (2021) 1–1.

[82] M. Xu, X. Li, Subject Property Inference Attack in Collaborative Learning, in: Proc. - 2020 12th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2020, volume 1, 2020, pp. 227–231.

[83] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[84] M. Chase, E. Ghosh, S. Mahloujifar, Property Inference From Poisoning, CoRR abs/2101.11073 (2021).

[85] M. Shen, H. Wang, B. Zhang, L. Zhu, K. Xu, Q. Li, X. Du, Exploiting Unintended Property Leakage in Blockchain-Assisted Federated Learning for Intelligent Edge Computing, IEEE Internet Things J. 8 (2021) 2265–2275.

[86] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54, 2017, pp. 1273–1282.

[87] D. Yin, Y. Chen, R. Kannan, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: Proceedings of the 35th International Conference on Machine Learning (ICML), volume 80, 2018, pp. 5650–5659.

[88] Z. Wu, Q. Ling, T. Chen, G. B. Giannakis, Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks, IEEE Transactions on Signal Processing 68 (2020) 4583–4596.

[89] K. Pillutla, S. M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, CoRR abs/1912.13445 (2019).

[90] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, Advances in Neural Information Processing Systems 30 (2017) 119–129.

[91] E. M. El Mhamdi, R. Guerraoui, S. Rouault, The hidden vulnerability of distributed learning in Byzantium, in: Proceedings of the 35th International Conference on Machine Learning, volume 80, 2018, pp. 3521–3530.

[92] L. Muñoz-González, K. T. Co, E. C. Lupu, Byzantine-robust federated machine learning through adaptive model averaging, CoRR abs/1909.05125 (2019).

[93] S. Fu, C. Xie, B. Li, Q. Chen, Attack-resistant federated learning with residual-based reweighting, CoRR abs/1912.11464 (2021).

[94] E. Tahanian, M. Amouei, H. Fateh, M. Rezvani, A game-theoretic approach for robust federated learning, International Journal of Engineering 34 (2021) 832–842.

[95] J. F. Nash, Equilibrium points in n-person games, Proceedings of the National Academy of Sciences 36 (1950) 48–49.

[96] S. Shen, S. Tople, P. Saxena, Auror: defending against poisoning attacks in collaborative deep learning systems, in: Proceedings of the 32nd Annual Conference on Computer Security Applications, 2016, pp. 508–519.

[97] F. Sattler, K.-R. Müller, T. Wiegand, W. Samek, On the byzantine robustness of clustered federated learning, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 8861–8865.

[98] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, E. Ilie-Zudor, Chained anomaly detection models for federated learning: An intrusion detection case study, Applied Sciences 8 (2018).

[99] X. Hei, X. Yin, Y. Wang, J. Ren, L. Zhu, A trusted feature aggregator federated learning for distributed malicious attack detection, Computers & Security 99 (2020) 102033.

[100] S. Azulay, L. Raz, A. Globerson, T. Koren, Y. Afek, Holdout sgd: Byzantine tolerant federated learning, CoRR abs/2008.04612 (2020).

[101] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, A.-R. Sadeghi, Dïot: A federated self-learning anomaly detection system for iot, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 756–767.

[102] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, S. Yu, Pdgan: A novel poisoning defense method in federated learning using generative adversarial network, in: ICA3PP, 2019.

[103] S. Li, Y. Cheng, W. Wang, Y. Liu, T. Chen, Learning to detect malicious clients for robust federated learning, CoRR abs/2002.00211 (2020).

[104] M. Naseri, J. Hayes, E. De Cristofaro, Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy, CoRR abs/2009.03561 (2020).

[105] E. Bagdasaryan, O. Poursaeed, V. Shmatikov, Differential privacy has disparate impact on model accuracy, Advances in Neural Information Processing Systems 32 (2019) 15479–15488.

[106] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. B. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. Y. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, O. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. X. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, S. Zhao, Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (2021) 1–210.

[107] B. McMahan, D. Ramage, K. Talwar, L. Zhang, Learning differentially private recurrent language models, in: International Conference on Learning Representations (ICLR), 2018.

[108] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, A. Anandkumar, signSGD: Compressed optimisation for non-convex problems, in: Proceedings of the 35th International Conference on Machine Learning (ICML), volume 80, 2018, pp. 560–569.

[109] C. Wu, X. Yang, S. Zhu, P. Mitra, Mitigating backdoor attacks in federated learning, CoRR abs/2011.01767 (2020).

[110] A. Portnoy, Y. Tirosh, D. Hendler, Towards federated learning with byzantine-robust client weighting, CoRR abs/2004.04986 (2021).

[111] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318.

[112] Q. Zheng, S. Chen, Q. Long, W. Su, Federated f-differential privacy, in: International Conference on Artificial Intelligence and Statistics, 2021, pp. 2251–2259.

[113] Z. Bu, J. Dong, Q. Long, W. J. Su, Deep learning with gaussian differential privacy, Harvard data science review 2020 (2020).

[114] X. Cao, J. Jia, N. Z. Gong, Data poisoning attacks to local differential privacy protocols, in: 30th {USENIX} Security Symposium ({USENIX} Security 21), 2021, pp. 947–964.

[115] K. Yadav, B. B. Gupta, K. T. Chui, K. Psannis, Differential privacy approach to solve gradient leakage attack in a federated machine learning environment, in: International Conference on Computational Data and Social Networks, 2020, pp. 378–385.

[116] M. Hao, H. Li, G. Xu, S. Liu, H. Yang, Towards efficient and privacy-preserving federated deep learning, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6.

[117] W. Wei, L. Liu, Y. Wut, G. Su, A. Iyengar, Gradient-leakage resilient federated learning, in: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021, pp. 797–807.

[118] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, K. Talwar, Semi-supervised knowledge transfer for deep learning from private training data, CoRR abs/1610.05755 (2016).

[119] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, Ú. Erlingsson, Scalable private learning with PATE, CoRR abs/1802.08908 (2018).

[120] Y. Zhu, X. Yu, M. Chandraker, Y.-X. Wang, Private-knn: Practical differential privacy for computer vision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11854–11862.

[121] Y. Zhu, X. Yu, Y.-H. Tsai, F. Pittaluga, M. Faraki, Y.-X. Wang, et al., Voting-based Approaches For Differentially Private Federated Learning, CoRR abs/2010.04851 (2020).

[122] C. Wang, J. Liang, M. Huang, B. Bai, K. Bai, H. Li, Hybrid differentially private federated learning on vertically partitioned data, CoRR abs/2009.02763 (2020).

[123] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, R. Rogers, Protection against reconstruction and its applications in private federated learning, CoRR abs/1812.00984 (2018).

[124] H. Lee, J. Kim, S. Ahn, R. Hussain, S. Cho, J. Son, Digestive Neural Networks: A Novel Defense Strategy Against Inference Attacks in Federated Learning, Computers & Security (2021).

[125] L. Fan, K. W. Ng, C. Ju, T. Zhang, C. Liu, C. S. Chan, Q. Yang, Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks, in: Federated Learning, Springer, 2020, pp. 32–50.

[126] M. Zhang, S. Wang, Matrix Sketching for Secure Collaborative Machine Learning, in: International Conference on Machine Learning (ICML), 2021, pp. 12589–12599.

[127] D. P. Woodruff, Sketching as a Tool for Numerical Linear Algebra, 2014.

[128] X. Yang, Y. Feng, W. Fang, J. Shao, X. Tang, S.-T. Xia, R. Lu, An Accuracy-Lossless Perturbation Method for Defending Privacy Attacks in Federated Learning, CoRR abs/2002.09843 (2021).

[129] C.-L. Chen, L. Golubchik, M. Paolieri, Backdoor attacks on federated meta-learning, CoRR abs/2006.07026 (2020).

[130] J. Zhang, D. Wu, C. Liu, B. Chen, Defending poisoning attacks in federated learning via adversarial training method, in: Frontiers in Cyber Security, Springer Singapore, 2020, pp. 83–94.

[131] H. Zhu, On the relationship between (secure) multi-party computation and (secure) federated learning, CoRR abs/2008.02609 (2020).

[132] Y. Lindell, How to simulate it–a tutorial on the simulation proof technique, Tutorials on the Foundations of Cryptography (2017) 277–346.

[133] O. Goldreich, The Foundations of Cryptography - Volume 1, Basic Techniques., 2001.

[134] O. Goldreich, The Foundations of Cryptography - Volume 2, Basic Applications., 2004.

[135] A. Beimel, Secret-sharing schemes: A survey, in: International conference on coding and cryptology, 2011, pp. 11–46.

[136] O. Goldreich, Y. Oren, Definitions and properties of zero-knowledge proof systems, Journal of Cryptology 7 (1994) 1–32.

[137] M. Bellare, V. T. Hoang, P. Rogaway, Foundations of garbled circuits, in: Proceedings of the 2012 ACM conference on Computer and communications security, 2012, pp. 784–796.

[138] J. Ma, S.-A. Naas, S. Sigg, X. Lyu, Privacy-preserving Federated Learning based on Multi-key Homomorphic Encryption, CoRR abs/2104.06824 (2021).

[139] Z. L. Jiang, H. Guo, Y. Pan, Y. Liu, X. Wang, J. Zhang, Secure Neural Network in Federated Learning with Model Aggregation under Multiple Keys, in: 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2021, pp. 47–52.

[140] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191.

[141] K. Bonawitz, F. Salehi, J. Konečnỳ, B. McMahan, M. Gruteser, Federated learning with autotuned communication-efficient secure aggregation, in: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, 2019, pp. 1222–1226.

[142] D. Meng, H. Li, F. Zhu, X. Li, FedMONN: Meta Operation Neural Network for Secure Federated Aggregation, in: 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2020, pp. 579–584.

[143] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, K. Ramchandran, Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning, CoRR abs/2009.11248 (2020).

[144] T. Sandholm, S. Mukherjee, B. A. Huberman, SAFE: Secure Aggregation with Failover and Encryption, CoRR abs/2108.05475 (2021).

[145] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 1–11.

[146] M. Asad, A. Moustafa, T. Ito, Fedopt: towards communication efficiency and privacy preservation in federated learning, Applied Sciences 10 (2020) 2864.

[147] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, X. Zheng, Privacy-preserving federated learning framework based on chained secure multiparty computing, IEEE Internet of Things Journal 8 (2021) 6178–6186.

[148] N. K. Le, Y. Liu, Q. M. Nguyen, Q. Liu, F. Liu, Q. Cai, S. Hirche, Fedxgboost: Privacy-preserving xgboost for federated learning, CoRR abs/2106.10662 (2021).

[149] Y. Li, T.-H. Chang, C.-Y. Chi, Secure federated averaging algorithm with differential privacy, in: 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), 2020, pp. 1–6.

[150] C. Sabater, A. Bellet, J. Ramon, Distributed differentially private averaging with improved utility and robustness to malicious parties, CoRR abs/2006.07218 (2020).

[151] B. Ghazi, R. Pagh, A. Velingker, Scalable and differentially private distributed aggregation in the shuffled model, CoRR abs/1906.08320 (2019).

[152] P. Kairouz, Z. Liu, T. Steinke, The distributed discrete gaussian mechanism for federated learning with secure aggregation, CoRR abs/2102.06387 (2021).

[153] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, W. Luo, Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive, IEEE Transactions on Dependable and Secure Computing (2019).

[154] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, H. V. Poor, Federated learning meets blockchain in edge computing: Opportunities and challenges, IEEE Internet of Things Journal (2021).

[155] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Generation Computer Systems 107 (2020) 841–853.

[156] S. Wang, C. Wang, Q. Hu, Corking by forking: Vulnerability analysis of blockchain, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, 2019, pp. 829–837.

[157] S. Zhang, J.-H. Lee, Mitigations on sybil-based double-spend attacks in bitcoin, IEEE Consumer Electronics Magazine (2020).

[158] R. Qin, Y. Yuan, S. Wang, F.-Y. Wang, Economic issues in bitcoin mining and blockchain research, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 268–273.

[159] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, volume 86, 1998, pp. 2278–2324.

[160] G. Cohen, S. Afshar, J. Tapson, A. van Schaik, Emnist: Extending mnist to handwritten letters, in: International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2921–2926.

[161] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, CoRR abs/1708.07747 (2017).

[162] A. Torralba, R. Fergus, W. T. Freeman, 80 million tiny images: A large data set for nonparametric object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1958–1970.

[163] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning (ICML), 2019, pp. 6105–6114.

[164] S. Yeom, I. Giacomelli, M. Fredrikson, S. Jha, Privacy risk in machine learning: Analyzing the connection to overfitting, in: 2018 IEEE 31st Computer Security Foundations Symposium (CSF), 2018, pp. 268–282.

[165] Y.-X. Wang, B. Balle, S. P. Kasiviswanathan, Subsampled rényi differential privacy and analytical moments accountant, in: The 22nd International Conference on Artificial Intelligence and Statistics, 2019, pp. 1226–1235.

[166] H. Zhu, Y. Jin, Multi-objective evolutionary federated learning, IEEE Transactions on Neural Networks and Learning Systems 31 (2020) 1310–1322.

[167] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural computation 1 (1989) 541–551.

[168] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, volume 2, 2003, pp. 1398–1402.

[169] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, R. Fergus, Regularization of neural networks using dropconnect, in: Proceedings of the 30th International Conference on Machine Learning(ICML - 13), 2013, pp. 1058–1066.

[170] A. Novikov, D. Podoprikhin, A. Osokin, D. P. Vetrov, Tensorizing neural networks, in: Advances in Neural Information Processing Systems, volume 28, 2015, pp. 442–450.

[171] M. Tan, Q. Le, Efficientnetv2: Smaller models and faster training, in: Proceedings of the 38th International Conference on Machine Learning (ICML), volume 139, 2021, pp. 10096–10106.

[172] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. J'egou, M. Douze, Levit: a vision transformer in convnet's clothing for faster inference, CoRR abs/2104.01136 (2021).

[173] Y. Lin, S. Han, H. Mao, Y. Wang, W. J. Dally, Deep gradient compression: Reducing the communication bandwidth for distributed training, CoRR abs/1712.01887 (2017).

[174] G. Pang, L. Cao, L. Chen, Homophily outlier detection in non-iid categorical data, Data Mining and Knowledge Discovery (2021) 1–62.

[175] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2019).

[176] A. Z. Tan, H. Yu, L. Cui, Q. Yang, Towards personalized federated learning, CoRR abs/2103.00710 (2021).

[177] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, S. Avestimehr, Fairfed: Enabling group fairness in federated learning, in: Workshop on New Frontiers in Federated Learning: Privacy, Fairness, Robustness, Personalization and Data Ownership (NeurIPS 2021), 2021.