applied sciences

MDPI

Article

Privacy and Security in Federated Learning: A Survey

Rémi Gosselin [1,†], Loïc Vieu [1,†], Faiza Loukil [2,*] and Alexandre Benoit [2,*]

[1] Polytech Annecy-Chambéry, Savoie Mont Blanc University, F-74944 Annecy, France
[2] LISTIC, Savoie Mont Blanc University, F-74944 Annecy, France
* Correspondence: faiza.loukil@univ-smb.fr (F.L.); alexandre.benoit@univ-smb.fr (A.B.)
† These authors contributed equally to this work.

Abstract: In recent years, privacy concerns have become a serious issue for companies wishing to protect economic models and comply with end-user expectations. In the same vein, some countries now impose, by law, constraints on data use and protection. Such context thus encourages machine learning to evolve from a centralized data and computation approach to decentralized approaches. Specifically, Federated Learning (FL) has been recently developed as a solution to improve privacy, relying on local data to train local models, which collaborate to update a global model that improves generalization behaviors. However, by definition, no computer system is entirely safe. Security issues, such as data poisoning and adversarial attack, can introduce bias in the model predictions. In addition, it has recently been shown that the reconstruction of private raw data is still possible. This paper presents a comprehensive study concerning various privacy and security issues related to federated learning. Then, we identify the state-of-the-art approaches that aim to counteract these problems. Findings from our study confirm that the current major security threats are poisoning, backdoor, and Generative Adversarial Network (GAN)-based attacks, while inference-based attacks are the most critical to the privacy of FL. Finally, we identify ongoing research directions on the topic. This paper could be used as a reference to promote cybersecurity-related research on designing FL-based solutions for alleviating future challenges.

Keywords: survey; federated learning; deep learning; machine learning; distributed learning; privacy; security; blockchain; deep learning security and privacy threats

Citation: Gosselin, R.; Vieu, L.; Loukil, F.; Benoit, A. Privacy and Security in Federated Learning: A Survey. *Appl. Sci.* **2022**, *12*, 9901. https://doi.org/10.3390/app12199901

Academic Editors: George Drosatos and Gianluca Lax

Received: 30 August 2022
Accepted: 27 September 2022
Published: 1 October 2022

1. Introduction

Machine Learning (ML) approaches can be considered as classical methods to address complex problems when the underlying physical model is not perfectly known. They are also a hot topic when considering a high semantic level analysis of complex data, such as object recognition on images and anomaly detection on time series. It is now possible to learn complex non-linear models directly from large quantities of data and deploy them for a variety of domains, including sensitive ones such as autonomous driving and medical data analysis. Numerous domains indeed generate ever-increasing quantities that thus allow for the application of ML methods. As an illustration, connected edge devices being integrated into most domains are expected to increase their number of collected data by more than 75% by 2025 [1], encouraged by new wireless technologies, namely 5G [2].

As a counterpart, such a successful model optimization requires an extended processing power to process large quantities of training data to improve robustness and generalization behaviors. As illustrated in Figure 1a, traditional machine learning approaches generally rely on centralized systems that gather both computing resources and the entirety of the data. However, this strategy raises confidentiality issues when transferring, storing, and processing data. In addition, it implies high communication costs that may forbid its use in a variety of sensitive application domains.

Decentralized learning is an alternative approach that aims to optimize models locally to reduce communication costs and preserve privacy. This strategy is challenging, since

the end-user expects at least similar performances and generalization behaviors as those that originate from a centralized approach while dealing with smaller and potentially biased local data collections, i.e., non-IID (Identically and Independently Distributed) data. Then, collaborative approaches are developed to introduce communication between local learner agents to look for a robust, general model. More specifically, the recently introduced Federated Learning (FL) [3] has been subject to a growing interest to address complex problems while never sharing raw data between collaborative agents, only exchanging model parameters. As illustrated in Figure 1b,c, a variety of system infrastructures are possible. Agents can communicate in a peer-to-peer (P2P) fashion or with a centralized parameter server. Further, hierarchical structures are possible in order to consider different model aggregation scales [4]. With such an approach, sensitive data are strictly kept on the client's device, at the edge, thus initiating privacy. In addition, communication costs can be significantly reduced, since model parameters are significantly smaller than the raw data. This, however, expects edge devices to be powerful enough to conduct local learning. Nevertheless, the quantity of data is smaller than with the centralized approach, and thus allows for cheaper hardware. In conclusion, federated learning is an attractive solution for multiple application domains and technologies, from medical applications to the Internet of Things, and is subject to intensive research.

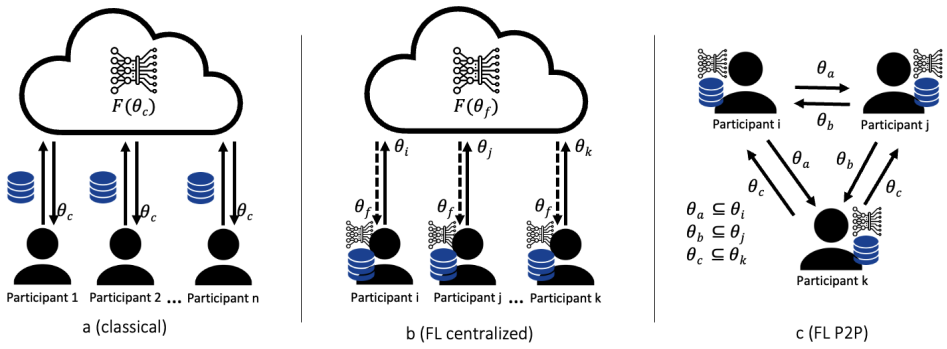

Figure 1. Comparison between classical centralized ML approach and centralized or P2P FL.

Nevertheless, despite being more privacy friendly than the centralized approach, FL is still subject to attacks that may impact the learned model's relevance, as well as the data integrity. A typical example is poisoning attacks [5,6], which aim to distort the model by sending corrupted training data. This attack type is facilitated by the centralized federated learning topology, since the parameter server cannot directly verify the data. User data integrity can also be compromised by relying on generative adversarial networks (GANs)-based attacks [7] to reconstruct data from local node models. Thus, despite the improvements compared to the centralized approaches, data security and privacy with federated learning are still burning issues that must be resolved.

The current state-of-the-art research surveys and reviews in the field provide remarkable work from various perspectives of FL, while focusing only on categorizing existing FL models [8], summarizing implementation details in the vertical FL approach [9] and open issues in FL [10]. While research already exists on the FL topic, the examination of FL security and privacy has not been sufficiently addressed in the literature. This work discusses the privacy and security risks of FL in terms of FL formal definitions, attacks, defense techniques, and future challenges, in an attempt to help the community and newcomers by providing in-depth information and knowledge of FL security and privacy.

Thus, the aim of this paper is to conduct a comprehensive study that highlights open security and privacy issues related to federated learning by answering several research questions. Figure 2 depicts the proposed taxonomy of security and privacy studies in FL. Following this structure, the remainder of this paper is then organized as follows. Section 2 presents the FL approach, the main infrastructure topology, and the related most common aggregation techniques. Section 3 provides an overview of the security flaws inherent to FL

and the defensive techniques proposed in the state of the art. Section 4 provides an outlook on the various privacy breaches and how advisers can minimize them. Section 5 discusses current issues and future trends. Finally, Section 6 concludes this survey paper.

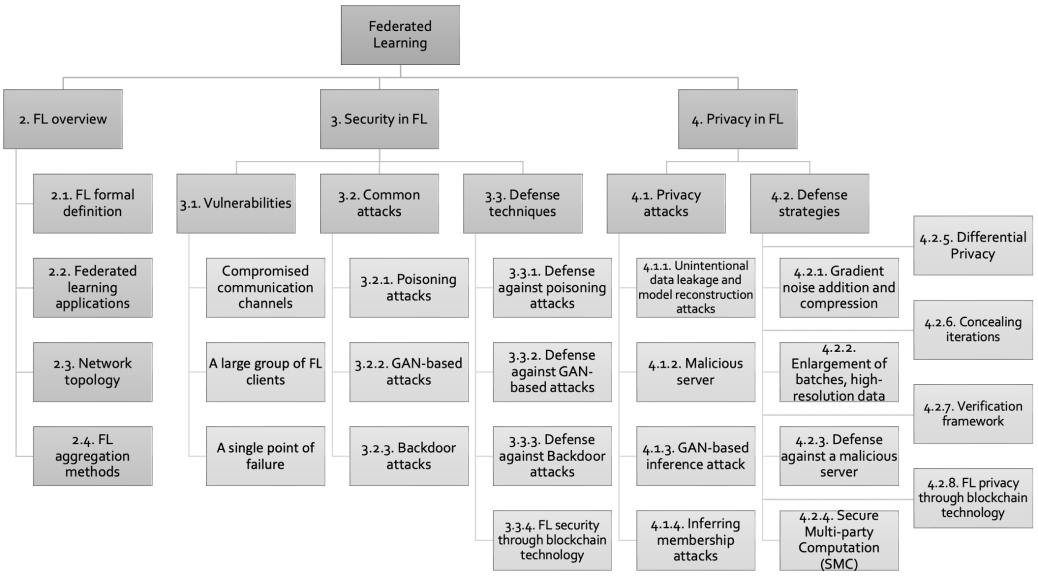

Figure 2. Taxonomy of security and privacy studies in FL. We report the sections of the paper that discuss each of the mentioned items.

2. An Overview of Federated Learning

In order to present the main security and privacy threats related to federated learning, this section first presents the main principles of FL compared to the traditional centralized learning approaches. Figure 1 illustrates the discussed approaches. Then, some real-world scenarios illustrate the FL relevance. Finally, the main features, topology, and model aggregation methods are presented.

2.1. From Centralized to Federated Learning

In a traditional centralized approach, the aim is to learn a single model $f(\theta_C)$ from a large data collection built from the aggregation of data coming from a variety of sources. Despite its impressive success, several issues must be highlighted. Indeed, when dealing with very large data collections and complex models, server-level data and communication costs must be reduced to efficiently distribute the data across computing devices [11]. Moreover, advanced machine learning techniques require large computing resources, which induce costs that may not be sustainable depending on the application economic model and may create dependencies on such computation power in the long term.

To overcome traditional machine learning weaknesses, federated learning was introduced in 2016 by McMahan et al. [3] as a learning technique that allows users to collectively reap the benefits of shared models trained from their data, without the need to centrally store them. Federated learning aims at identifying a general model $f(\theta_{FL})$ by aggregating local ones $f(\theta_i)$ trained by a set of participating clients that keep their data, but occasionally share their parameters. Indeed, FL introduces a new paradigm that reduces communication costs and pushes forward data privacy approaches in several application domains [12].

2.2. Federated Learning Applications

The first large-scale demonstration of FL was Google predictive keyboard (Gboard) on Android smartphones. The smartphone's processing power is used to train a local model from the typing data of the device owner. By occasionally communicating local models to a Google central server that performs aggregation, the general word prediction model improves and is distributed to all the users. This approach allows integration of

both global user behaviors, and the specialization of a local model [13]. Compared to the centralized approach, personal typing data should have been sent to the cloud and would have led to privacy issues. Further, this use case is interesting in terms of understanding the limits of the classic ML approach. Indeed, a smartphone keyboard has to fit the user's language. This results in a very personal ML model; however, to be efficient and benefit from the richness of the language, this model should be trained at a larger scale, taking advantage of other people making use of similar languages. However, in a privacy concern, it is impossible to collect the user's typing data. FL then enables taking advantage of both the global knowledge and the specialization of a local model [13]. Similarly, healthcare is another typical case study. The COVID-19 crisis illustrated the use of FL in predicting the future oxygen needs of patients for the EXAM model. In [14], data collected from over 20 institutions around the world were used to train a global FL model. Although no raw data were communicated to a central server, FL allowed for the optimization of a general model relying on a large amount of data.

2.3. Network Topology

The studied state-of-the-art papers mainly report two typical FL communication schemes that impact the network topology, which we summarize as follows.

2.3.1. Centralized FL

This approach is the standard one. The central server is the cornerstone of the architecture. It manages clients, centralizes their local models, and updates the global model. FL optimization is an iterative process wherein each iteration improves the global ML model. It consists of three main steps:

- Model initialization: FL is based on an initial model generally prepared on the server side. Initial weights can rely on a pretraining step. The model parameters are then distributed to the participating clients and will be updated along the next steps in accordance with clients' feedback.
- Local model training: a selection of participating clients is defined. Each of them receives the global model and fine-tuning parameters, which rely on their local data for a set of training epochs. Then, the locally updated model weights are sent to the central server to update the global model.
- Aggregation: the central server collects the participating clients' updated models. Then, an aggregation of their parameters yields an updated general model. This step is critical and should integrate several factors, including client confidence and participation frequency, in order to reduce bias.

Steps 2 and 3 constitute a single round that is repeated until a stop condition is reached.

2.3.2. Peer-to-Peer FL

In a fully decentralized approach, there is no longer a central server that acts as an initiator, coordinator, and model aggregator. Communication with the central server is replaced by peer-to-peer communication as shown in Figure 1c. In this type of topology, network agents learn personalized models. Communication with other members with a common goal is essential in order to increase the quality of their model. The gossip communication protocol is one of the most widely used and efficient protocols today [15,16].

2.4. FL Aggregation Methods

The aggregation of the local models should result in an improved and more general model. Some of the main state-of-the-art methods are presented below:

- The first proposal was the Federated Averaging Algorithm (FedAvg), introduced in [3]. Considered the default approach for centralized FL, the central server will generate the global model by averaging all the participating client models. This approach can be considered as gradient descent on the server side. Extensions have been proposed to adapt efficient optimization strategies, such as Adam and Adagrad, to this context [17].

- FedProx [18] was proposed as a generalization of the FedAvg method. It takes into account the variable amount of work to be performed by each client. This also depends on global data distributions across clients, local computational resources, energy, and communication limits. It has been shown that FedProx results in better averaging accuracy than FedAvg in heterogeneous settings.
- The Federated Matched Averaged (FedMa) method [19] aims to update the global model via layer-wise matching and aggregation of inner model components, namely neurons. This approach only works on neural networks due to its specificities. Moreover, it only works on simple neural networks such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM)-based models. This method presents good results with heterogeneous learners and better performances than FedAvg and FedProx within a few training iterations.

3. Security in Federated Learning

Clients participating in a federated learning model can be numerous. This opens doors to a variety of attacks on the client, server, and communication sides. Thus, the development of models via this technology must follow and take into account the main concepts of information security confidentiality, integrity, and availability.

Current studies that explore vulnerabilities and provide existing defensive techniques for security attacks of FL are very limited. Thus, we define the following research questions on the security aspect of the FL:

- RQ1: What are the major vulnerabilities in the FL domain?
- RQ2: What security threats and attacks do FL models face?
- RQ3: What are the defensive techniques against security attacks in the FL ecosystem?

In the rest of this section, we answer each research question based on the studied publications in the FL domain.

3.1. Vulnerabilities in the Ecosystem

A vulnerability can be defined as a weakness in a system that provides an opportunity for curious/malicious attackers to gain unauthorized access [20]. Scanning all sources of vulnerabilities and tightening defenses are thus mandatory steps needed to build up a federated learning-based system while ensuring the security and privacy of the data. Based on the studied publications, we answer the RQ1 question below. Vulnerability sources are actually similar to those of distributed applications. We categorize them as follows:

- Compromised communication channels: an insecure communication channel is an open vulnerability in the FL process that could be addressed using a cryptography public key, which keeps message content secure and safe throughout the communication.
- A large group of FL clients: The number of clients participating in the model is large, so the general model is likely to receive models from Byzantine nodes. This type of safety threat is called Data Poisoning [21].
- A single point of failure: The central parameter server, which is at the heart of the network, must be robust and secure to prevent intrusions. Then, its vulnerabilities must be checked to ensure that they are not exploited. Furthermore, the security updates and all the security recommendations must be followed to limit the risk of intrusions [22].

3.2. Common Attacks in FL

A cyber-attack consists of any action aiming at undermining a computer element: network, software, or data with a financial, political, or security purpose [23]. The number of attacks against a deep learning model is high, such that we focus on the three main ones in terms of impact on the system, frequency, and relevance concerning federated learning. Interested readers can find more details in [24]. These attacks are derivations of the so-called Poisoning Attacks we mentioned previously. Next, we provide the answer to the RQ2 question by introducing some specific and challenging approaches.

3.2.1. Poisoning Attacks

A Poisoning Attack is one of the most common techniques used in the FL. This type of attack can be of a different nature, but the principle remains essentially the same. A malicious client will send rigged data to affect the global model [22,25]. Such an attack can be conducted by two means:

- Data poisoning: The aim is to incorporate malicious data points to create bias in the global model. To be undetectable, attackers slightly modify the model parameters they send but repeat this operation over several training iterations or rounds. This makes it difficult to determine whether a local parameter set is poisoned or not. With several model updates of such a corrupted client, the global model becomes biased enough to degrade its task performance [26].
- Model poisoning: This type of poisoning is more direct; it seeks to manipulate the global model without going through the insertion of malicious data points. It is generally permitted thanks to an intrusion at the server level.

3.2.2. Generative Adversarial Network (GAN) Based Attacks

Poisoning attacks have evolved and new methodologies for creating poisoned models have been proposed. One of them is called PoisonGAN [27]. This attack uses a generative adversarial network (GAN) to generate realistic datasets that are controlled at will by the attacker. A GAN is optimized on a given client side, relying on model updates with the aim of manipulating the parameters of the global model, as shown in Figure 3. Attacks using GANs have several advantages. Indeed the attacker does not need to have a dataset before making the attack [6]. Such kind of attack is actually facilitated by the federated learning infrastructure since the attacker has access to the local model, which would be more difficult with centralized learning.

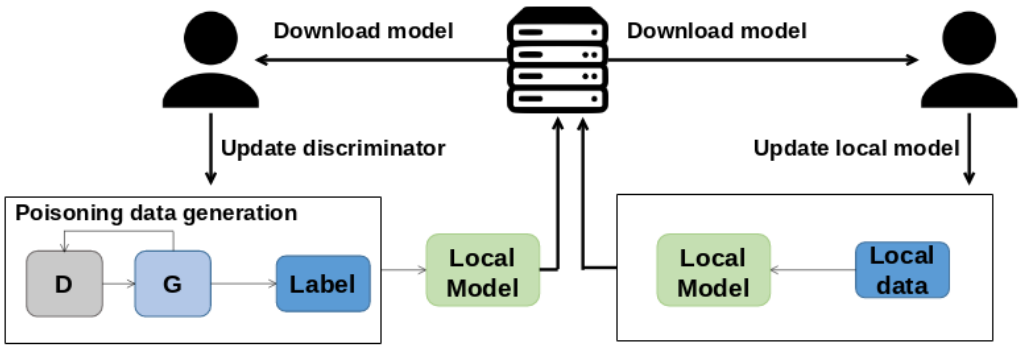

Figure 3. Overview of GAN-based poisoning attack in FL.

Finally, such GAN-based attacks are the most difficult to detect since the generated data are realistic. Its application can yield disastrous consequences for the model's accuracy by introducing strong and controlled bias.

3.2.3. Backdoor Attacks

Bagdasaryan et al. [22] show that it is possible to create a backdoor by poisoning the models. This type of attack does not aim to reduce the global model accuracy, but introduces a very specific bias focused on certain labels. A classic example of a backdoor in FL is the image detection models that we find in [27], where the image classifier is distorted to assign a specific label chosen by the attacker.

3.3. Techniques for Defending against Attacks in FL

As a general rule, good security practices for information systems and networks such as encryption of communications must be put in place. No computer system is impenetrable; however, protecting against the various known methods can greatly reduce

the number of attacks and their impact. There are two main defense approaches: (i) the proactive one, which is upstream of the attack, and which looks for ways to detect and blocking the latter, and (ii) the reactive one, which is set up when the attack has already taken place and aims to mitigate the consequences and make the patch. With the emergence of federated learning and its specific attacks, dedicated protections are proposed. To answer the RQ3 question, strategies against the most common vulnerabilities and attacks presented in Sections 3.1 and 3.2) are explained below.

3.3.1. Defense against Poisoning Attacks

The proposed methods are mainly reactive ones which continuously monitor client behaviors. Rodríguez-Barroso et al. [28] have proposed a method to screen out malicious clients. This method works by using artificial intelligence that detects model changes or nonconforming data distributions. Another method following the same principle but applied to an attack by several malicious users is proposed in [29]. This is called sniper. Another defense against poisoning attacks is proposed in [21]. This technique consists of checking the performance of the global model at each new model update.

3.3.2. Defense against GAN-Based Attacks

Those specific poisoning attacks require dedicated approaches, such as advanced Byzantine actor detection algorithms [30]. Additionally, defensive techniques are enabled via heterogeneous federated learning via model distillation, which is detailed in [31]. However, defense techniques against this type of attack are still poorly developed and documented.

3.3.3. Defense against Backdoor Attacks

The main approach consists of minimizing the size of the model to reduce its complexity and capacity while potentially improving its accuracy. This technique is called pruning [32]. Since the resulting model is less expressive, backdoor attacks are more complex to carry out. Such a method also introduces some beneficial side effects. The reduced number of parameters indeed reduces communication costs and reduces message interception probability.

One can highlight the fact that the absence of a central server avoids the risk of an attack at the heart of the system. Thus, peer-to-peer federated learning infrastructure could be an interesting solution. However, this would reduce global monitoring capacities and delegate a fraction of this task to each client. Peer-to-peer approaches then introduce additional constraints on the potentially limited capacity edge nodes, thus limiting its application.

3.3.4. FL Security through the Blockchain Technology

As defined by [33], blockchain technology can be represented as a distributed database—a ledger that can be consulted by everyone and everything. Each new element in the database is verified and forgery-proofed [34].

Technologies such as blockchain can be used to secure federated learning and introduce device and model trust. This has been proven in [35,36]. Blockchain technology would act at two levels. The first would be to encourage users to participate in the development of the global model by rewarding contributors for their involvement. The second would be to save the evolution of the parameters in the blockchain. This can then increase user confidence and reduce the risk of poisoning attacks. One of the most cited models following this approach is called blockFL [36]. It proposes a way to trust devices in which a blockchain network replaces the central server of a classical centralized FL infrastructure and conducts local model validation, but model aggregation is performed on the client side. In further detail, each client sends its updates to an associated miner in the network. Miners are in charge of exchanging and verifying all the model updates. For a given operation, a miner runs a Proof of Work (PoW), which aims to generate a new block where the model

updates are stored. Finally, the generated block stores the aggregated local model updates that can be downloaded by the other members of the network. Then, each client can compute the global model locally. This strategy makes poisoning attacks more difficult by certifying model updates. As a counterpart, communication and client computation costs are increased. Additionally, in order to keep attackers from tampering with local models, Kalapaaking et al. [37] proposed a blockchain-based Federated Learning framework with an Intel Software Guard Extension (SGX)-based Trusted Execution Environment to securely aggregate local models in Industrial Internet of Things. Indeed, each blockchain node could (i) host an SGX-enabled processor to securely perform the FL-based aggregation tasks, (ii) verify the authenticity of the aggregated model, and (iii) add it to the distributed ledger for tamper-proof storage and integrity verification.

4. Privacy in Federated Learning

Some of the most frequent concerns about traditional ML include privacy weaknesses. In the industrial area, companies invest to protect their intellectual property. However, traditional ML and, more specifically, deep learning model optimization often go against those privacy requirements. It is indeed necessary to store potentially large quantities of data close to the large processing power to train, validate, and test models. Therefore, such data collections, when sensitive, must be communicated and centralized at a high-security level to prevent any data leakage or attacks. However, threats remain. The traditional ML approach thus has topology threats, which require alternatives to be found. As an answer to these issues, federated learning promotes a new topology to limit the data transfers and, consequently, the data footprint. However, some privacy issues are already identified and must be combated in this new setting.

Current studies exploring privacy attacks in FL and providing existing privacy-preserving defense techniques are very limited. Thus, we define the following research questions on the privacy aspect of the FL:

- RQ4: What are the privacy threats and attacks in the FL ecosystem?
- RQ5: What are the privacy-preserving techniques that tackle each type of the identified attacks in RQ1'?
- RQ6: What new technology could enhance the general privacy-preserving feature of FL?

In the rest of this section, we answer each research question based on the studied publications in the FL domain.

4.1. Privacy Attacks in FL

Federated learning [3] introduces the assumption that it is safer for user data privacy thanks to its topology. Each client shares its model updates instead of its dataset, preventing user data interception while communicating with the server or peer [38]. However, several studies have shone light on many attacks that still compromise data privacy. This section provides the answer to the RQ4 question by presenting several popular attacks, allowing user data to be inferred or the local model to be reconstructed.

4.1.1. Unintentional Data Leakage and Model Reconstruction Attacks

Despite the non-communication of data, Nasr et al. [39] have shown that data could be inferred by considering only model weights. Indeed, an honest-but-curious adversary eavesdrops communications of clients and the server, allowing for the reconstruction of the client model. Ref. [40] defines an honest-but-curious client as a client, which aims to store and process data on its own. Unlike other attacks such as data poisoning, the client may not interfere with the collaborative learning algorithm.

Several studies demonstrated the possibility of reconstructing local data collections by inverting the model gradients sent by the clients to the server [40,41]. Nonetheless, this attack has some limitations. It performs well on models that have been trained on small batches of data or data points with poor diversity. It is not suitable when multiple

stochastic gradient updates have been performed locally before communicating with the server. Other recent studies [42] go further by reconstructing the local client model without interfering in the training process. This attack, called the model reconstruction attack, copes with the previous limits, allowing for high-quality local model reconstruction.

4.1.2. Malicious Server

In a centralized federated learning setting, the central server is one of the most critical parts of the architecture. If the server is malicious or is accessed by unauthorized persons, then all local models sent by clients can be intercepted and studied in order to reconstruct a part of the original client data. The malicious server can then analyze the model shared by the clients in a discontinuous way (passive mode) or follow the chronological evolution of the model shared by the victim (active mode) [43].

4.1.3. GAN-Based Inference Attack

This kind of attack has already been experienced in [40,44–46] and may be passive or active. The passive attack aims to analyze the user inputs at the server level, and the active one works on sending the global update to only an isolated client.

Experiments in [47,48] demonstrated that with only some of the computed gradients of a neural network, training data could be inferred. With only 3.89% of the gradients, Aono et al. [47] have been able to reconstruct an image close enough to the original one to infer the information. Further, Zhao et al. [48] were able to reconstruct nearly 100% of the original data by inference. Their Deep Leaked Gradients algorithm was the first algorithm capable of rebuilding a pixel-wise image and token-wise matching texts. These works prove that is it possible to infer data from only leaked gradients using an honest-but-curious client with only a few iterations or a small number of gradients.

4.1.4. Inferring Membership Attacks

The purpose of the Inferring Membership is to determine if data have already been seen by the model. This attack can be carried out in two ways: actively or passively. During a passive attack, the user will only observe the updates of the global model [49]. For the active attack, the adversary participates in the creation of the model, which allows him to recover more information. In this type of attack, the goal is to follow the evolution and the behavior of the global model. This type of attack exploits the stochastic gradient descent (SGD) algorithm to extract information about training sets. According to [39], during training, the SGD aims to make the gradient of the loss leaning zero for information extraction.

4.2. Privacy-Preserving Defense Strategies

To mitigate the aforementioned attacks and answer the RQ5 question, the state of the art already reports some defensive strategies, which are summarized in the following subsections.

4.2.1. Gradient Noise Addition and Compression

Despite the performance of the Deep Leakage from Gradients (DLG) algorithm to infer data from gradients, Zhu et al. [7] demonstrated that in FL, and more generally in ML, the addition of noise to the gradients makes the inference more difficult. The authors show that from a variance larger than 10^{-2}, the accuracy drops significantly and leads to an inability to infer the original data. Another solution proposed in [50] is the compression of the model gradients. This approach sparsifies gradients, which impacts the sensitivity of algorithms such as DLG. The authors of [50] show that the gradients can be compressed by 300× before affecting the accuracy of the model. Conversely, the tolerance of the DLG algorithm is around 20% of sparsity, and the compression makes it ineffective.

4.2.2. Enlargement of Batches, High-Resolution Data

Despite the good performances of the DLG algorithm, the technique is not yet generalizable enough to present a significant threat to private data. One of the limits of the DLG algorithm is related to large batch size and image resolution. In their study, Zhu et al. [7] used a batch size of up to eight images and an image resolution of up to 64×64, which illustrates that some FL configurations rely on small datasets. In another larger-scale setting still addressable by FL, more data and processing power would allow for large batch training and/or high-resolution data, such as the data used for classical centralized machine learning. In such a context, DLG would be out of its operating conditions and would not allow for private data inference.

4.2.3. Defense against a Malicious Server

Due to privacy concerns and critical communication bottlenecks, it can become impractical to send the FL updated models to a centralized server. Thus, a recent study [51] proposed an optimized solution for user assignment and resource allocation over hierarchical FL architecture for IoT heterogeneous systems. According to the study's results, the proposed approach could significantly accelerate FL training and reduce communication overhead by providing 75–85% reduction in the communication rounds between edge nodes and the centralized server, for the same model accuracy.

4.2.4. Secure Multi-Party Computation (SMC)

Initially, the secure multi-party computation aims to jointly compute a function on different parties over their personal data. One of the benefits of this approach is the possibility of keeping the inputs private thanks to cryptography. According to [47], SMC is currently used in FL but in a different version, which only needs to encrypt the parameters instead of the large volume of data inputs. Although this approach prevents leaks from a malicious central server, the encryption is expensive to use and may have an impact on a larger scale. Thus, the main cost of this solution is efficiency loss due to encryption.

4.2.5. Differential Privacy

Differential Privacy (DP) is a technique widely used to preserve privacy in industry and academic domains. The main concept of DP is to preserve privacy by adding noise to sensitive personal attributes. In FL, DP is introduced to add noise to clients' uploaded parameters in order to avoid inverse data retrievals. For instance, the DPFedAvgGAN framework [52] uses DP to make GAN-based attacks inefficient in training data inference of other users for FL-specific environments. Additionally, Ghazi et al. [53] improve privacy guarantees of the FL model by combining the shuffling technique with DP and masking user data with an invisibility cloak algorithm. However, such solutions bring uncertainty into the uploaded parameters and may harm the training performance.

4.2.6. Concealing Iterations

Xu et al. [54] have suggested iterations concealing as a method of avoiding privacy threats. In typical FL, the client model iterations are visible to multiple actors of the system, such as the central server and the clients chosen to participate in each round. Thus, a malicious server may be able to infer data through iterations thanks to GAN-based attacks. By concealing iterations, each client should make the learning phase inside a Trusted Execution Environment (TEE). A TEE provides the ability to run code on a remote trusty machine, regardless of the trust placed in the administrator. Indeed, a TEE limits the abilities of any party, including the administrator, to interact with the machine. This specificity results in three properties that make TEE trusty:

- Confidentiality: The running code and its execution state are not shared outside of the machine and cannot be accessed.
- Integrity: Because of access limitations, the running code cannot be poisoned.

- Attestation: The running virtual machine can prove to an external entity what code is executing and the inputs that were provided.

Thus, thanks to TEE, the model iterations can be concealed, and the model parameters may be encrypted inside the TEE before being shared. Nonetheless, this approach might not be used across clients, especially with devices such as smartphones. In fact, TEE may not be powerful enough for training and may involve too many computational and memory costs. These weaknesses may have a significant impact on devices' autonomy, performance, and user experience. For these reasons, in its current state, TEE cannot yet be used on limited-capacity edge devices.

4.2.7. Verification Framework

VerifyNet [54] is a FL framework that provides the possibility for clients to verify the central server results to ensure its reliability. To preserve privacy, it provides a double masking protocol, which makes it difficult for attackers to infer training data. Moreover, VerifyNet is robust enough to handle clients' dropouts due to the user's battery or hardware issues. However, this framework imposes more communication with the clients, which involves a decrease in performance and additional costs.

4.2.8. FL Privacy through the Blockchain Technology

As discussed in Section 3.3.4, and to provide an answer to the RQ6 question, we argue that blockchain technology brings a new FL network topology. Above the security benefits, it can also provide privacy benefits. Indeed, the decentralized network allows each participant to access all transactions made on the network. This transparency makes the network more reliable and increases trust in the learning process. Thus, instead of sharing gradients to a centralized server, the model parameters and clients' updates are stored on the blockchain. In order to introduce privacy in blockchain, the authors of [55] presented a specific design adapted to IoT that relies on differential privacy. Experiments showed that robustness against GAN and curious server-based attacks is increased while maintaining significant task performance levels. Moreover, the blockchain allows for client updates audits in order to maintain model reliability and cope with malicious participants. Thus, blockchain-based models such as BlockFL [36] provide a layer of security that guarantees the model parameters' integrity. Nonetheless, such an approach does not prevent privacy attacks, and it must be enhanced with privacy-oriented methods such as differential privacy.

5. Open Issues and Future Trends

Federated learning is a recent and continuously evolving approach in machine learning that can already benefit many economic, medical, and sociological domains. Industrial sectors and, more specifically, the so-called Industry 4.0 that introduces the data processing paradigm to improve the manufacturing process can push the implications of federated learning [56] further by integrating contextual factors, such as decentralized production models. FL thus appears as an interesting general solution that effectively associates global and local machine learning with privacy by design. However, despite overcoming some classical machine learning issues, some remaining ones and new challenges must be faced in this context. This paper therefore provides an overview of recent trends in the specific problems related to privacy and security. Nevertheless, as reported in recent publications such as [10], other critical challenges are related to system architecture and learning bias. As an opening discussion, we thus elaborate on the following subsections, connections between security, privacy, and bias, and identify research directions.

5.1. Security, Privacy, and Bias Mitigation Dependency Management for Trustfully and Fair Federated Learning

Machine learning model bias is a general problem encountered by machine learning methods. It is critical, since it actually impacts model decisions and can yield discrimination of data populations that have a direct impact on our life, including sexism and racism. More

specifically, learning bias can in fact be aggravated by Federated Learning. As reported in recent studies, such as [10], bias is indeed mostly related to the non-IID nature of the data distributed across federated participants. Among other bias sources, participants' involvement frequency in the learning process is a significant factor. Bias mitigation techniques have long been proposed in classical machine learning [57], but they must be adapted to the specific context of Federated Learning [58]. Indeed, we would like to point out in this discussion that bias mitigation, privacy, and security, in the context of federated learning, are highly associated and thus should not be considered separately. For instance, the search for bias in the data, as well as in the model predictions, generally relies on global knowledge that conflicts with client data privacy. Additionally, Poisoning Attacks typically introduce bias, such that their detection allows for security flaws to be counterfeit. Those connections allow us to identify some promising research directions that should, in our opinion, increase federated learning trust and fairness.

5.2. Research Directions

Recent contributions such as [59] propose innovative approaches that consist of detecting communities of clients that share similar behaviors. The aim is next to build up a general model that integrates all those populations in an equal way and thus reduces bias. Such an approach could also help detect specific outliers such as poisoning sources, thus increasing security. Such a direction is indeed promising; however, the identification of those communities may lead to new issues related to user privacy. In addition, the semantics and ethics behind the identified clusters should also be clearly reported. Addressing those challenges is, therefore, of real interest.

A complementary research direction relates to partial federated learning recently introduced in [60]. Such an approach considers the fact that clients may need to share only a subset of their knowledge and still improve each other. As for real applications related to healthcare, autonomous driving, and home automation, clients may have different sensors but try to solve the same problem or try to solve different tasks with the same sensors. A given model may thus be considered as a set of cascading functions, some being local and private while others are shared. Such an approach has many potential advantages. It better corresponds to real-life applications and could also reduce global costs while improving generalization behaviors for a variety of complementary problems. In addition, it provides means to increase security and privacy by sharing only a fraction of the knowledge. We believe that this research direction is promising, and suggest associating this with bias detection and mitigation research.

Finally, new models of artificial intelligence approaches that specialize in blockchain technology seem relevant, sharing similar ideas. For instance, Swarm Learning (SL) [61] takes the main principle of FL, where the data are not shared with a central server. However, the computation of the global model is deferred to the customers. In each training cycle, the operation falls to a node that the others have elected. SL implementations have been applied to the medical field [61,62]. In such a sensitive context, it has been shown that SL can push the BlockFL approach further by using smart contracts to protect the model from misleading participants and keep data and learning safe [63]. Nevertheless, some security issues related to computation and leader node selection still remain, but show interest in blockchain-based methods. This direction, which maintains connections with federated learning while providing an innovative communication paradigm, is therefore of interest for further research.

6. Conclusions

Despite the recent improvements and the growing interest in the federated learning field, it is still in its infancy. Recent studies demonstrated that federated learning is a serious alternative to traditional machine learning methods. While existing surveys have already studied FL from various perspectives, sufficient progress has not been made concerning understanding FL for its security and privacy risks. This paper intends to fill

this gap by presenting a comprehensive study of privacy and security issues related to federated learning. Thus, we introduced an overview of the FL applications, topology, and aggregation methods. Then, we discussed the existing FL-based studies in terms of security and privacy defensive techniques that aim to counteract FL vulnerabilities. Finally, FL open issues and future trends are identified to be addressed in further studies. Based on the findings from the study, we conclude that FL fixed several privacy and security issues of classical ML. Thus, it opens up new possibilities, especially in areas in which data are sensitive, such as the medical area or personal user data. Already used by some companies, such as Google, its uses are bound to develop and become more democratic. However, FL is confronted with privacy and security issues inherited from the distributed architecture and the traditional ML. As a consequence, several research works try to go further and integrate the FL principles into other architectures, such as blockchain architectures, to skirt threats.

Author Contributions: Conceptualization, R.G. and L.V.; methodology, F.L. and A.B.; formal analysis, F.L. and A.B.; investigation, R.G. and L.V.; resources, R.G. and L.V.; writing—original draft preparation, R.G. and L.V.; writing—review and editing, F.L. and A.B.; visualization, R.G. and L.V.; supervision, F.L. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Al Hayajneh, A.; Bhuiyan, M.Z.A.; McAndrew, I. Improving internet of things (IoT) security with software-defined networking (SDN). *Computers* **2020**, *9*, 8. [CrossRef]
2. Wang, J.; Lim, M.K.; Wang, C.; Tseng, M.L. The evolution of the Internet of Things (IoT) over the past 20 years. *Comput. Ind. Eng.* **2021**, *155*, 107174. [CrossRef]
3. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
4. Shayan, M.; Fung, C.; Yoon, C.J.; Beschastnikh, I. Biscotti: A blockchain system for private and secure federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1513–1525. [CrossRef]
5. Biggio, B.; Nelson, B.; Laskov, P. Poisoning Attacks against Support Vector Machines. *arXiv* **2012**, arXiv:1206.6389.
6. Zhang, J.; Chen, J.; Wu, D.; Chen, B.; Yu, S. Poisoning attack in federated learning using generative adversarial nets. In Proceedings of the 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), Rotorua, New Zealand, 5–8 August 2019; pp. 374–380.
7. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 14774–14784.
8. Li, Q.; Wen, Z.; He, B. Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *arXiv* **2019**, arXiv:1907.09693.
9. Yang, Q.; Liu, Y.; Cheng, Y.; Kang, Y.; Chen, T.; Yu, H. Federated learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2019**, *13*, 1–207.
10. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
11. Tang, Z.; Shi, S.; Chu, X.; Wang, W.; Li, B. Communication-Efficient Distributed Deep Learning: A Comprehensive Survey. *arXiv* **2020**, arXiv:2003.06307.
12. Sun, Z.; Strang, K.D.; Pambel, F. Privacy and security in the big data paradigm. *J. Comput. Inf. Syst.* **2020**, *60*, 146–155. [CrossRef]
13. McMahan, B.; Ramag, D. Federated Learning: Collaborative Machine Learning without Centralized Training Data. 2017. Available online: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html (accessed on 29 September 2022).
14. Dayan, I.; Roth, H.R.; Zhong, A.; Harouni, A.; Gentili, A.; Abidin, A.Z.; Liu, A.; Costa, A.B.; Wood, B.J.; Tsai, C.S.; et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nat. Med.* **2021**, *27*, 1735–1743. [CrossRef] [PubMed]
15. Hu, C.; Jiang, J.; Wang, Z. Decentralized Federated Learning: A Segmented Gossip Approach. *arXiv* **2019**, arXiv:1908.07782.
16. Vanhaesebrouck, P.; Bellet, A.; Tommasi, M. Decentralized collaborative learning of personalized models over networks. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 509–517.

17. Reddi, S.J.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; McMahan, H.B. Adaptive Federated Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Conference, 2020.
18. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
19. Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; Khazaeni, Y. Federated learning with matched averaging. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Conference, 2020.
20. OWSAP. OWSAP Defination for Vulnerability. 2018. Available online: https://owasp.org/www-community/vulnerabilities/ (accessed on 29 September 2022).
21. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 634–643.
22. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Palermo, Sicily, Italy, 26–28 August 2020; pp. 2938–2948.
23. Hathaway, O.A.; Crootof, R.; Levitz, P.; Nix, H. The law of cyber-attack. *Calif. Law Rev.* **2012**, *100*, 817.
24. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [CrossRef]
25. Lyu, L.; Yu, H.; Yang, Q. Threats to federated learning: A survey. *arXiv* **2020**, arXiv:2003.02133.
26. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data poisoning attacks against federated learning systems. In *Proceedings of the European Symposium on Research in Computer Security, Guildford, UK, 14–18 September 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 480–501.
27. Zhang, J.; Chen, B.; Cheng, X.; Binh, H.T.T.; Yu, S. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.* **2020**, *8*, 3310–3322. [CrossRef]
28. Rodríguez-Barroso, N.; Martínez-Cámara, E.; Luzón, M.; González Seco, G.; Ángel Veganzones, M.; Herrera, F. Dynamic Federated Learning Model for Identifying Adversarial Clients. *arXiv* **2020**, arXiv:2007.15030.
29. Cao, D.; Chang, S.; Lin, Z.; Liu, G.; Sun, D. Understanding distributed poisoning attack in federated learning. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; pp. 233–239.
30. Hayes, J.; Ohrimenko, O. Contamination attacks and mitigation in multi-party machine learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 6604–6616.
31. Li, D.; Wang, J. FedMD: Heterogenous Federated Learning via Model Distillation. *arXiv* **2019**, arXiv:1910.03581.
32. Liu, K.; Dolan-Gavitt, B.; Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses, Crete, Greece, 10–12 September 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 273–294.
33. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260. [CrossRef]
34. Crosby, M.; Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain technology: Beyond bitcoin. *Appl. Innov.* **2016**, *2*, 71.
35. Majeed, U.; Hong, C.S. FLchain: Federated learning via MEC-enabled blockchain network. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–4.
36. Kim, H.; Park, J.; Bennis, M.; Kim, S.L. Blockchained on-device federated learning. *IEEE Commun. Lett.* **2019**, *24*, 1279–1283. [CrossRef]
37. Kalapaaking, A.P.; Khalil, I.; Rahman, M.S.; Atiquzzaman, M.; Yi, X.; Almashor, M. Blockchain-based Federated Learning with Secure Aggregation in Trusted Execution Environment for Internet-of-Things. *IEEE Trans. Ind. Inform.* **2022**. [CrossRef]
38. Truong, N.; Sun, K.; Wang, S.; Guitton, F.; Guo, Y. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Comput. Secur.* **2021**, *110*, 102402. [CrossRef]
39. Nasr, M.; Shokri, R.; Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 739–753.
40. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706.
41. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting gradients-how easy is it to break privacy in federated learning? *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16937–16947.
42. Xu, C.; Neglia, G. What else is leaked when eavesdropping Federated Learning? In Proceedings of the CCS workshop Privacy Preserving Machine Learning, Virtual Event, USA, 2021.
43. Song, M.; Wang, Z.; Zhang, Z.; Song, Y.; Wang, Q.; Ren, J.; Qi, H. Analyzing user-level privacy attack against federated learning. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2430–2444. [CrossRef]
44. Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; Qi, H. Beyond inferring class representatives: User-level privacy leakage from federated learning. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2512–2520.
45. Truong, N.; Sun, K.; Wang, S.; Guitton, F.; Guo, Y. Privacy preservation in federated learning: Insights from the gdpr perspective. *arXiv* **2020**, arXiv:2011.05411.

46. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep models under the GAN: Information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 603–618.
47. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1333–1345.
48. Zhao, B.; Reddy Mopuri, K.; Bilen, H. iDLG: Improved Deep Leakage from Gradients. *arXiv* **2020**, arXiv:2001.02610.
49. Zari, O.; Xu, C.; Neglia, G. Efficient passive membership inference attack in federated learning. *arXiv* **2021**, arXiv:2111.00430.
50. Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, W.J. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. *arXiv* **2017**, arXiv:1712.01887.
51. Abdellatif, A.A.; Mhaisen, N.; Mohamed, A.; Erbad, A.; Guizani, M.; Dawy, Z.; Nasreddine, W. Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. *Future Gener. Comput. Syst.* **2022**, *128*, 406–419. [CrossRef]
52. Augenstein, S.; McMahan, H.B.; Ramage, D.; Ramaswamy, S.; Kairouz, P.; Chen, M.; Mathews, R.; y Arcas, B.A. Generative Models for Effective ML on Private, Decentralized Datasets. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
53. Ghazi, B.; Pagh, R.; Velingker, A. Scalable and Differentially Private Distributed Aggregation in the Shuffled Model. *arXiv* **2019**, arXiv:1906.08320.
54. Xu, G.; Li, H.; Liu, S.; Yang, K.; Lin, X. Verifynet: Secure and verifiable federated learning. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 911–926. [CrossRef]
55. Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D.; Li, Z.; Lyu, L.; Liu, Y. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet Things J.* **2020**, *8*, 1817–1829. [CrossRef]
56. Hao, M.; Li, H.; Luo, X.; Xu, G.; Yang, H.; Liu, S. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Trans. Ind. Inform.* **2019**, *16*, 6532–6542. [CrossRef]
57. Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
58. Ferraguig, L.; Djebrouni, Y.; Bouchenak, S.; Marangozova, V. Survey of Bias Mitigation in Federated Learning. In Proceedings of the Conférence Francophone D'informatique en Parallélisme, Architecture et Système, Lyon, France, July 2021.
59. Briggs, C.; Fan, Z.; Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–9.
60. Singhal, K.; Sidahmed, H.; Garrett, Z.; Wu, S.; Rush, J.; Prakash, S. Federated Reconstruction: Partially Local Federated Learning. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual Conference*; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 11220–11232.
61. Saldanha, O.L.; Quirke, P.; West, N.P.; James, J.A.; Loughrey, M.B.; Grabsch, H.I.; Salto-Tellez, M.; Alwers, E.; Cifci, D.; Ghaffari Laleh, N.; et al. Swarm learning for decentralized artificial intelligence in cancer histopathology. *Nat. Med.* **2022**, *28*, 1232–1239. [CrossRef]
62. Becker, M. Swarm learning for decentralized healthcare. *Der Hautarzt* **2022**, *73*, 323–325. [CrossRef]
63. Warnat-Herresthal, S.; Schultze, H.; Shastry, K.L.; Manamohan, S.; Mukherjee, S.; Garg, V.; Sarveswara, R.; Händler, K.; Pickkers, P.; Aziz, N.A.; et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature* **2021**, *594*, 265–270. [CrossRef] [PubMed]