Journal of Intelligent Information Systems (2023) 61:923–949
https://doi.org/10.1007/s10844-023-00797-x

REVIEW

A review of federated learning: taxonomy, privacy and future directions

Hashan Ratnayake[1] · **Lin Chen**[1] · **Xiaofeng Ding**[1]

Received: 26 February 2023 / Revised: 8 May 2023 / Accepted: 9 May 2023 /
Published online: 4 July 2023

Abstract
The data generated and stored in mobile devices owned by individuals as well as in various organizations contains a large amount of valuable and important information that can be used to improve service quality, user experience, and satisfaction. However, due to privacy concerns, many entities are reluctant to share their data with others, and this is a major barrier to developing comprehensive models that can provide accurate predictions. Federated learning is a state-of-the-art distributed machine learning approach where multiple clients are allowed to collaboratively train a model while keeping their private training data locally. Although federated learning seems to be a viable solution for jointly training a machine learning model without compromising privacy, sensitive privacy information may still be leaked through shared model parameters and query results. Over the past six years, the researchers have extensively studied privacy protection enhancements of federated learning, and they have revealed that general privacy protection mechanisms can be adopted to mitigate privacy issues of federated learning. However, protecting privacy through federated learning while maintaining data utility is still an open issue. This article provides an overview of federated learning while discussing privacy leakages, possible defense mechanisms, and future research directions of privacy-preserved federated learning.

Keywords Anonymization · Cryptography · Federated learning · Perturbation · Privacy

1 Introduction

As a result of the evolution of information and communication technology, people today are accustomed to using mobile phones, tablets, or any other portable devices with high

✉ Lin Chen
chenl@hust.edu.cn

Hashan Ratnayake
I202022009@hust.edu.cn

Xiaofeng Ding
xfding@hust.edu.cn

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

processing power, significant storage capacity, and powerful sensors, not only for communication purposes but also for primary calculations as well as data storage and access (Asad et al., 2021; McMahan et al., 2017). In addition, many banks, supermarkets, hospitals, and other service providers have adapted to maintain customer databases with their various interactions. Moreover, millions of embedded sensors such as city cameras, car navigation systems, and some wearable devices generate large amounts of data every day (McMahan et al., 2017). The data generated and stored by any of the above contains an abundance of important and beneficial information that can be used to improve user experience, comfort, and satisfaction. In practice, it is necessary to go through a process of data mining and analysis to identify hidden patterns and information in the data. However, the privacy-sensitive nature of this data and its sheer volume have been identified as major barriers to obtaining valuable information (Asad et al., 2021; Hu et al., 2021). There are a number of different approaches to training a machine learning model while preserving privacy, and federated learning (McMahan et al., 2017) is widely recognized as a suitable solution.

Federated learning is an emerging new research topic in machine learning with important real-world applications and many research challenges, and it provides the promise to learn from distributed data sources while preserving data privacy. The past six years have seen rapid growth in federated learning, and many researchers are engaged in improving the efficiency, effectiveness, and privacy of federated learning based on different settings. Li et al. (2020) discussed the characteristics and challenges of federated learning and Mothukuri et al. (2021) surveyed the security and privacy of federated learning. Meanwhile, Hu et al. (2021) reviewed the algorithms and general issues of federated learning while discussing the architectures and classification. In addition, Yang et al. (2019a) discussed architectures and applications of federated learning while briefing existing works in the field. Moreover, Liu et al. (2022) conducted a survey of possible threats, attacks, and defenses at different stages of federated learning. However, given the short history of the topic and its rapid development, there is still lacking an upright and comprehensive up-to-date survey paper on federated learning. Therefore, in this article, we present a review of federated learning by providing a brief introduction, taxonomy from various aspects, privacy issues and possible defense mechanisms, simulation frameworks, real-world applications, and future directions.

The remainder of this article is organized as follows. Section 2 provides an overview of federated learning that would be desirable in understanding the matters discussed in the following sections. Some of the key general privacy protection mechanisms are discussed in Section 3. In Section 4, we will review the privacy leakages in federated learning and discuss possible defense mechanisms for those privacy issues. Then, an outline of frameworks for implementing and simulating federated learning is provided in Section 5. Section 6 provides a brief discussion of the applications of federated learning and finally, we point out several future research directions on privacy protection in federated learning and provide a concluding remark.

2 An overview of federated learning

2.1 Introduction to federated learning

Federated learning is a distributed machine learning framework in which multiple data owners train a machine learning model collaboratively without disclosing their private data to each other (Hu et al., 2021). The term federated learning was first introduced by McMahan et al.

(2017), and the explosive growth of big data and laws and regulations introduced to protect data privacy have largely led to its rapid evolution over the past six years (Yin et al., 2021). The General Data Protection Regulation (GDPR) (Voigt & von dem Bussche, 2017) in European Union, Personal Data Protection Act (PDPA) (Chik, 2013) in Singapore, Cybersecurity Law (Aimin et al., 2018) in China, the California Consumer Privacy Act (CCPA) (Pardau, 2018), and the California Privacy Rights Act (CPRA) (Goldman, 2021) in the United States and the Health Insurance Portability and Accountability Act of 1996 (HIPAA) (Kulynych & Korn, 2003) are some of such laws and regulations introduced to protect data privacy. The initial federated learning framework consisted of data-owning clients and a centralized server dedicated to coordination and aggregation purposes (McMahan et al., 2017). However, certain clients perform the functions of the central server in a special federated learning setting (Roy et al., 2019; Yang et al., 2019c). Hitaj et al. (2017) named federated learning as collaborative learning because of the jointly training nature of the framework. The ultimate goal of federated learning is to collectively train a model for a specific application while preserving privacy, and the performance of the collaboratively learned model should be a good approximation of the model built with the traditional approach by pooling data from all data owners (Kairouz et al., 2021). The workflow of federated learning consists of several steps where the model engineer first needs to clearly identify the problem to be solved, create a prototype of the model, and test the learning hyperparameters using a sample data set. Next, if there is any requisite, the clients are instrumented to store the necessary training data locally and the server sends the initial model to each client device. Thereafter, each data-owning client trains its own local model using local data and then sends the model parameters to the central server, which updates the global model by combining the parameters from all data owners. After aggregating all model parameters, the server returns the new global model to each device for the next training round. Once a good model is obtained based on the evaluation, the final step is the model deployment process. Figure 1 illustrates the workflow of federated learning.

2.2 Advantages of federated learning

Federated learning has several advantages over centralized and distributed machine learning. One of the main advantages of federated learning is that it can better protect the security and privacy of user data (Hu et al., 2021). Federated learning can provide better data privacy protection than traditional and distributed machine learning methods because the model learning process is performed locally by each client with its own data and only model parameters such as model weights and gradients are uploaded to the central server for aggregation. Efficient use of hardware is another advantage of federated learning (McMahan et al., 2017). In most practical cases, the model update process occurs when the devices are idle. For instance, language model training on mobile phones takes place when the phone is connected to the charger or not in excessive use. In addition, the distributed nature of the model updating process provides a solution to hardware requirements with a larger storage capacity and higher processing power. Real-time continuous learning is an outstanding benefit of federated learning. As models are constantly being improved using local client data, the predictions using models take high accuracy. Furthermore, federated learning facilitates the efficient development of a machine learning model with diverse data of massively distributed data owners.

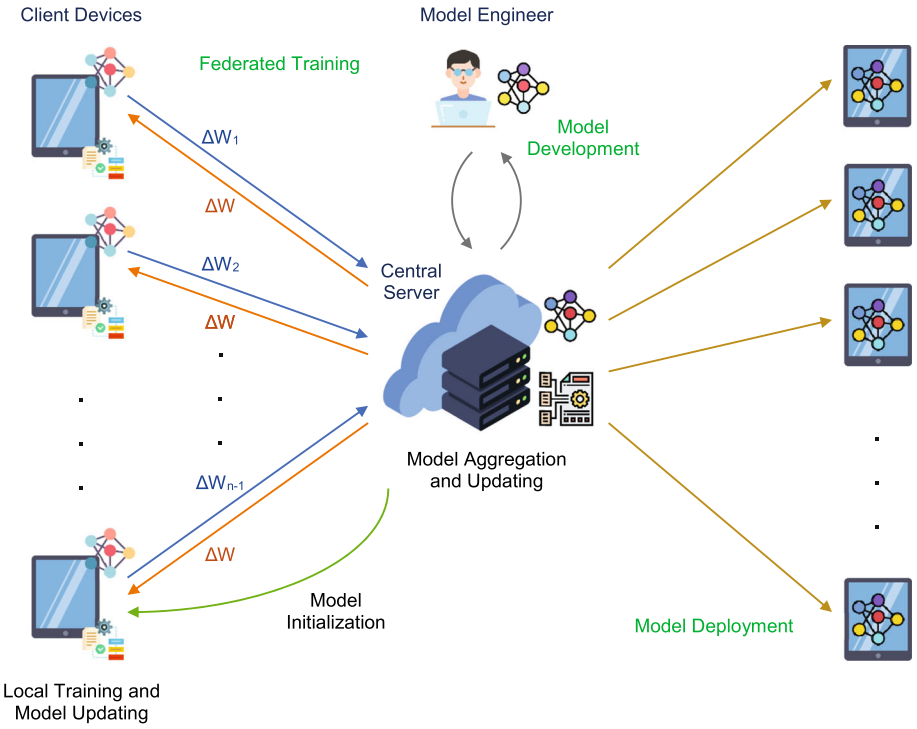

Fig. 1 Workflow of federated learning

2.3 Challenges in federated learning

2.3.1 Non-independent and identically distributed data

Non-Independent and Identically Distributed (Non-IID) data is a significant challenge in federated learning. Model updating with Independent and Identically Distributed (IID) data is one of the basic assumptions of federated learning. However, in many practical cases, the above assumption is rarely satisfied as it is difficult to find IID data (Xue et al., 2021). In such a case, the training complexity of the federated learning model can be greatly increased (Huang et al., 2022). Furthermore, many researchers (Huang et al., 2022; Zhao et al., 2018) have recognized that Non-IID data negatively affect the accuracy of federated learning. Moreover, Zhao et al. (2018) have experimentally shown that heterogeneous client data slow down convergence in addition to reducing test accuracy. When McMahan et al. (2017) first introduced the federated learning concept and optimized model averaging algorithm FedAvg, researchers observed that the accuracy of federated learning on Non-IID data was slightly lower than on IID data although the proposed approach is robust to the Non-IID data. In order to improve the test accuracy of federated learning with Non-IID data, Zhao et al. (2018) suggested sharing a small portion of the global data across all end devices participating in the training process. According to the authors, since the globally shared data is a separate data set from the clients' data, this mechanism does not affect the privacy of the participants. Although this approach allows for the creation of more accurate models, the lack of such a public data set is one of the most critical shortcomings of this mechanism. Huang et al. (2022)

proposed the FedFa algorithm by introducing a double-momentum mechanism that can cope with Non-IID data, which takes into account the influence of historical gradient information of both client and server to improve the convergence of the algorithm, and observed that FedFa expedites the convergence speed for Non-IID data compared to the baseline algorithm FedAvg. Furthermore, the authors observed an increase in fairness and accuracy by using information quantity about the accuracy and frequency of participation in the training to assign an appropriate weight to the client during server aggregation, however, the theoretical analysis was left as a future work.

2.3.2 Imbalanced data distribution

As some clients produce significantly more data for model learning and some clients produce less, the data amount on different clients may vary. This imbalanced nature of data amount among different clients is another challenge of federated learning. According to Huang et al. (2022), imbalanced data creates the problem of statistical heterogeneity and it will be the reason for increasing the training complexity of the federated learning model. Duan et al. (2020) proposed a self-balancing federated learning framework to deal with imbalanced data in mobile systems. There, the researchers performed Z-score-based data augmentation to smooth out global imbalance before training the model. Moreover, they performed a mediator-based multi-client rescheduling in order to achieve a partial equilibrium. During experiments, researchers observed that this new mechanism achieves better accuracy while significantly reducing communication traffic than FedAvg. Meanwhile, Yang et al. (2021) proposed a client selection scheme to mitigate the effect of class imbalance using reinforcement learning. Here, the researchers revealed the distribution of classes according to the updated gradients without using any raw data from the client devices considering the privacy of the users and observed an improvement in the convergence performance of the global model in the experiments.

2.3.3 Massively distributed nature

Generally, the number of clients participating in federated learning is larger than the average number of samples per client. This is another challenge in federated learning and is known as the massively distributed nature (McMahan et al., 2017). This problem is more pronounced in cross-device federated learning, where billions of clients participate in model training than in cross-silo federated learning, where there are a few participating clients. Kamp et al. (2021) proposed a novel approach to overcome this problem by interleaving the model aggregation and permutation steps, where local models are redistributed across clients through the server during the permutation step to allow each local model to train local data sets with each client while preserving data privacy. The authors further strengthened the privacy of this setting by combining it with differential privacy mechanisms and introducing random permutations. Compared to standard federated learning approaches, this mechanism significantly improved model quality in massively distributed instances.

2.3.4 Limited communication

Limited communication is also a widely discussed challenge in federated learning. Clients of federated learning can be IoT devices such as smartphones, cameras, and smart sensor devices, or large entities such as hospitals and banks. Such clients participating in model learning are often offline or on slow or expensive connections (Huang et al., 2022). In FedAvg (McMahan

et al., 2017), the researchers sought to solve the problem of existing network bandwidth by significantly reducing communication rounds relative to the synchronized stochastic gradient descent. Basically, this was achieved by adding more computation to each client by iterating the local updates several times before the model aggregation step. In addition, allowing each client to perform a more complex calculation between each communication round rather than performing a simple calculation such as a gradient calculation is another mechanism they adopted in reducing communication rounds. However, clients with relatively fast processing power are a key requirement in implementing such an arrangement, otherwise, the training process will be significantly delayed by clients with low processing power. The HybridAlpha framework (Xu et al., 2019) was able to reduce the communication costs as well as solve the problem of dynamically dropping or adding participants while ensuring model performance and privacy similar to other approaches.

Some researchers discussed the above challenges under the systems and statistical heterogeneity. The problem of imbalanced and Non-IID data was primarily linked to statistical heterogeneity, while the hardware variability, network connectivity issues, and dropping out of devices were associated with the system heterogeneity (Huang et al., 2022). Addressing the problem of heterogeneity is essential in order to increase the model accuracy (Hu et al., 2021).

2.3.5 Risk of privacy leakage

Apart from the aforementioned challenges, the risk of privacy leakage is often a major concern in federated learning. Although federated learning has taken steps to protect data generated on each device by sharing model parameters such as weights or gradients information instead of raw data, several studies (Luo et al., 2021; Wang et al., 2019) have shown that federated learning does not always ensure adequate privacy protection. The main reason for this is that malicious adversaries can reveal the privacy of local training data sets even with a small fraction of the original gradients (Phong et al., 2018). Recent studies (Park & Lim, 2022; Truex et al., 2019) have focused on enhancing the privacy of federated learning by applying various mechanisms such as differential privacy, homomorphic encryption, and secure multiparty computation. However, these approaches often provide privacy with reduced model performance or system efficiency, and therefore, balancing these trade-offs is a significant challenge in achieving privacy in federated learning. A detailed discussion of privacy leakages in federated learning and possible defense mechanisms is provided in Section 4 of this article.

2.4 Categorization of federated learning

2.4.1 Categorization based on the number of clients

Federated learning can be divided into two main clusters based on the number of clients participating in the model training. They are cross-device federated learning and cross-silo federated learning (Huang et al., 2022; Liang et al., 2021). In cross-device federated learning, there are hundreds of, thousands of, millions, or even billions of devices carrying very sensitive information from different people or entities participating in model training. Basically, these clients are a very large number of mobile or IoT devices (Liu et al., 2022). However, only a fraction of clients is available at any given time for the training process. On the other hand, cross-silo federated learning trains a model on siloed data. In cross-silo federated

learning, the participating clients are not individual devices, rather they are hospitals, banks, schools, government institutions, etc. (Liang et al., 2021). These institutions won't be able to share their data with each other and a central server. Compared to the cross-device setup, the stability of the cross-silo setup is better, as each party in the cross-silo setup can continuously engage in the model training throughout the entire process, while there is a high probability of clients dropping outs in the cross-device setup (Liang et al., 2021). Different characteristics of cross-device federated learning and cross-silo federated learning are compared in Table 1.

2.4.2 Categorization based on data distribution

In addition to classifying federated learning based on the number of clients participating in the collaborative training process, federated learning can be categorized into three different categories based on how data is distributed across multiple participating clients. They are horizontal federated learning, vertical federated learning, and federated transfer learning (Cheng et al., 2020; Hu et al., 2021; Yang et al., 2019a).

Horizontal federated learning refers to the scenario where a different set of clients have the same set of features, and is most applicable when there is a large overlap in the feature space between data sets (Mugunthan et al., 2021; Saha & Ahmad, 2021). This setup assumes that each client handles different data samples while sharing the same set of features (Xia et al., 2021; Zhu et al., 2021). The terms instance-based federated learning (Zhu et al., 2021), sample-partitioned federated learning, and example-partitioned federated learning (Cheng et al., 2020), have been used synonymously with horizontal federated learning in various studies. The general architecture of horizontal federated learning consists of a set of data owners with overlapping features and a centralized server (Huang et al., 2022; McMahan et al., 2017). However, horizontal federated learning systems can be implemented without a centralized server for orchestration purposes (Roy et al., 2019). Depending on the existence and non-existence of a centralized server, the communication architecture of horizontal federated learning may vary. Client-server architecture, also known as centralized federated learning, uses a centralized server to orchestrate the entire training process. The existence of honest clients and an honest-but-curious server is an underlying assumption of this architecture. Peer-to-peer architecture is another communication architecture used in horizontal federated learning. This architecture is also known as decentralized federated learning, as there is no central server and any client is selected as the server in each training round. If the clients are organized into a circular chain and select clients sequentially as the server in each training round until the termination condition is met, it is called cyclic transfer. Conversely,

Table 1 Categorization of federated learning based on the number of clients

Characteristic	Cross-device federated learning	Cross-silo federated learning
Participants in model training.	A very large number of mobile or IoT devices	Different organizations such as hospitals and banks
Typical number of clients.	Up to 10^{10}	Around 2-100
Client availability in model training.	Often only a fraction of the clients are available	All clients are almost always available
Client reliability.	Extremely unreliable	Relatively high reliability
Data partition axis.	Partition is fixed and horizontal	Partition is fixed and could be horizontal or vertical

selecting a client randomly with equal probability as the server and sending its model information to another randomly selected client to update the model, and continuing this process until the termination condition is met is called a random transfer.

In contrast to horizontal federated learning, vertical federated learning allows multiple clients to possess data of the same set of individuals but each client has a unique set of features (Mugunthan et al., 2021). Vertical federated learning is more suitable for scenarios where there is a large overlap in the sample space of the data set (Saha & Ahmad, 2021) and is also known as feature-partitioned (Cheng et al., 2020) or feature-based (Yang et al., 2019a) federated learning, as operations are carried across different columns in which data is partitioned by features. In practice, vertical federated learning is widely used in cross-silo setting (Liang et al., 2021). Although it has been assumed that the label information belongs to a single data holder in the studies of early stages, later studies considered the possibility of the existence of multiple label owners (Mugunthan et al., 2021; Zhu et al., 2021). Vertical federated learning focuses on two main functions, namely encrypted entity alignment, the encryption-based user alignment process to confirm standard users of different participating clients, and encrypted model training, which securely trains the machine learning model with data from common entities. The typical architecture of vertical federated learning consists of parties with overlapping data samples and a collaborator. In some cases, the collaborator is a third party (Hardy et al., 2017) while most of the time the collaborator is a participating entity that owns the labels (Yang et al., 2019c). The architecture with a third-party coordinator assumes that clients are honest-but-curious about each other and that there is an honest third-party coordinator. However, it is very difficult to find an authorized third party that other parties can trust and it also increases the risk of data leakage (Yang et al., 2019c). Therefore, researchers have focused on introducing a communication architecture that does not have a third-party coordinator, and the tasks to be performed by the third-party coordinator are often performed by the client that owns the label data. In this architecture, all clients are assumed to be honest-but-curious about each other. Compared to horizontal federated learning, vertical federated learning is more likely to occur in real-world applications and generally limits communication efficiency and privacy (Xia et al., 2021).

The above two approaches are only applicable in the context of data with common features or common samples under the federation, and in reality, such a set of common entities may be limited, resulting in the federation being less attractive (Liu et al., 2020). Federated transfer learning is a viable solution in such a scenario, which allows sharing of knowledge without compromising user privacy and enables complementary knowledge exchange among the parties. This mechanism allows the target-domain party to build more flexible and powerful models by leveraging rich labels from the source-domain party (Sharma et al., 2019). A federated transfer learning system is usually limited to two parties and its protocols are mostly similar to those of vertical federated learning (Yang et al., 2019a). Besides, the performance of federated transfer learning depends on how related the domains are, therefore in general it is more suitable for organizations in similar industries. Meanwhile, federated transfer learning requires additional security protocols to maintain privacy as training consists of collaborative computations using parameters related to data from both the source and target domains (Sharma et al., 2019). In particular, during the training phase of federated transfer learning, details of intermediate results should be shared extensively between the two parties of the federation. Therefore, both parties must encrypt their gradients before exchanging them, and use techniques such as random masking to prevent the parties from revealing each other's information at any stage in the process (Jing et al., 2019). This high computational overhead of the federated transfer learning security protocol has made it impractical in many real-world applications.

Figure 2 illustrates the distribution of data in the feature space and sample space relevant to horizontal federated learning, vertical federated learning, and federated transfer learning.

3 General privacy protection mechanisms

Privacy is one of the essential properties of federated learning, and a review of its privacy enhancements reveals that general privacy protection mechanisms can be adopted to protect the privacy of federated learning (Hu et al., 2019; Vaidya et al., 2013). Many privacy-preserving mechanisms are designed considering the two main metrics of privacy protection, which are data utility and loss of privacy of a data set (Nasr et al., 2019). General privacy-preserving techniques in federated learning can be classified into three major categories as cryptographic techniques, perturbation techniques, and anonymization techniques.

3.1 Cryptographic techniques

The purpose of most cryptographic techniques is to protect the input information throughout the information flow until it gets to the output. Homomorphic encryption, secure multi-party computation, and secret sharing can be introduced as the most commonly used cryptographic techniques to enhance the privacy of federated learning.

3.1.1 Homomorphic encryption

Homomorphic encryption, a public-key encryption scheme, allows computation on encrypted data throughout the information flow (Fang & Qian, 2021; Zheng et al., 2019). The concept of homomorphic encryption was first proposed by Rivest et al. (1978) for applications used in the banking sector. The basic idea of homomorphic encryption is that after encrypting plaintext to ciphertext, the result of performing certain operations on the ciphertext of the ciphertext space is the same as the result of the encryption operations on the plaintext of the plaintext space (Park & Lim, 2022). This process can be further explained as follows.

$$E(m_1) \otimes E(m_2) = E(m_1 \otimes m_2) \tag{1}$$

where,

$\forall m_1, m_2 \in M$; M denotes a set of plaintext and $\otimes$ is the operator fulfilling the criterion.

As an example, if the operator is addition, then the algorithm satisfies additive homomorphism and if the operator is multiplication, it satisfies multiplicative homomorphism. Depending on the supporting operators, homomorphic encryption methods can be categorized as fully homomorphic encryption and partially homomorphic encryption. Fully homomorphic

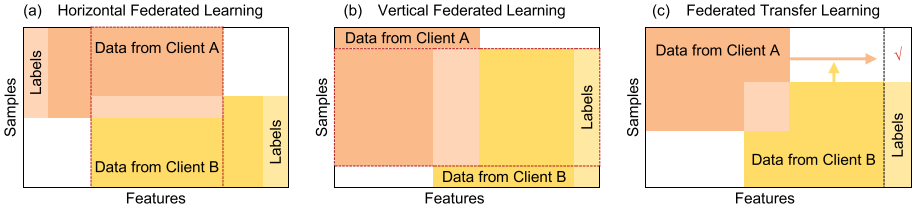

Fig. 2 Categorization of federated learning based on data distribution

encryption supports both additive and multiplicative operations while partially homomorphic encryption only supports either additive or multiplicative operations (Fang & Qian, 2021). The Paillier cryptosystem (Paillier, 1999) is an example of partially homomorphic encryption that satisfies additive homomorphism. A federated learning scheme with homomorphic encryption ensures that there is no performance loss in the model convergence since it does not alter the original data (Wu et al., 2021). Although fully homomorphic encryption provides stronger encryption than partially homomorphic encryption, the computation costs of fully homomorphic encryption are greater than those of partially homomorphic encryption.

3.1.2 Secure multi-party computation

Secure multi-party computation is another cryptographic framework that is compatible with the general setup of federated learning to enhance privacy protection (Goldreich, 1998; Zheng et al., 2019). This cryptographic technique enables distributed participants to collaboratively calculate an objective function without revealing their data to other users throughout the information flow (Cramer et al., 2000). The accuracy of the calculated function and the non-disclosure of any private information of a specific participant to any other participants are two major requirements to satisfy secure multi-party computation protocol. Sharemind (Bogdanov et al., 2008) is a state-of-the-art secure multi-party computation framework that can process data without compromising security or access to confidential values. High accuracy, the ability to eliminate the need for trusted third parties, and solve the problem of the trade-off between data utility and privacy are the main advantages of secure multi-party computation. However, the communication cost of this mechanism is high as it requires extensive message exchange due to the use of sophisticated protocols (Ling et al., 2007). Additionally, the computational overhead of secure multi-party computation is also significantly higher (Truex et al., 2019).

3.1.3 Secret sharing

Another cryptographic scheme, secret sharing, implies that a secret key consisting of n shares can be reconstructed only if a minimum number of t shares are combined (Bonawitz et al., 2017). This minimum number of t shares is called the threshold and any set of shares less than the threshold will not give any additional information about the secured secret. In secret sharing scheme, there are two roles as dealers and players. Generally, a dealer distributes the shares of a secret among n players in such a way that allocates one share to each player. Shamir's Secret Sharing (Shamir, 1979) is a well-known key distribution algorithm that divides a secret key into parts called shares. Using the threshold secret sharing scheme can solve the problem of users dropping out during the training. However, the secret sharing method is vulnerable to dishonest dealers or malicious players (Yin et al., 2021). Sharma et al. (2019) proposed a secure and efficient solution to protect privacy against malicious players in practical collaboration training under a data federation using secret sharing methods with secure multi-party computation.

3.2 Perturbation techniques

The main idea of perturbation techniques is to add noise to the original data or to map data instances to some representational data structure without adding noise so that the statistical information computed from the perturbed data is statistically indistinguishable from the

original data (George & Zoran, 2015). The most commonly used perturbation techniques in federated learning are differential privacy, additive perturbation, and multiplicative perturbation.

3.2.1 Differential privacy

Differential privacy was originally designed to ensure the privacy of individuals who contributed with their personal data to build a statistical database (Dwork, 2006; Yang et al., 2019b). This means that for an adversary who observes the output of a statistical database in which a differentially private mechanism is applied, the ability to detect the presence or absence of any person in the database is negligible (Alaggan et al., 2017). According to Dwork (2006), the formal definition of differential privacy is as follows.

A randomized function K gives ε-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(K)$.

$$Pr[K(D_1) \in S] \leq e^{\varepsilon} \times Pr[K(D_2) \in S] \tag{2}$$

To achieve differential privacy, noise is added to the output of the algorithm and this noise is proportional to the sensitivity of the output, where sensitivity measures the maximum difference in output due to the inclusion of a single data instance. This noise is controlled by epsilon (ε) and this phenomenon is called privacy budget (Arachchige et al., 2020). The Laplacian and Gaussian mechanisms are two popular mechanisms for achieving differential privacy. Many studies have shown that differential privacy can provide a strong privacy guarantee, even if some accuracy loss can be detected (Truex et al., 2019; Wei et al., 2020).

There are two types of differential privacy techniques known as global differential privacy and local differential privacy. Global differential privacy uses a trusted central party for the purpose of applying carefully calibrated random noise to the actual values returned for a particular query, so this setup is also known as the trusted curator model (Arachchige et al., 2020). In contrast to global differential privacy, local differential privacy does not require a trusted third party and each participant applies differential privacy locally before the central authority has access to the data (Arachchige et al., 2020; Xu et al., 2019). Therefore, local differential privacy is also known as the untrusted curator model. However, compared to global differential privacy, local differential privacy adds too much noise to the model parameter data from each node, leading to poor performance of the resulting model (Truex et al., 2019). In addition, the differential privacy guarantee in local differential privacy becomes meaningless for models with a smaller number of parameters.

3.2.2 Additive perturbation

Additive perturbation protects the privacy of original data by adding random noise from any distribution, such as uniform or Gaussian distributions (Chamikara et al., 2018; Kargupta et al., 2003). This is a type of unidimensional input perturbation. Data reconstruction algorithms that are based on concepts such as principal component analysis can be used to attack against additive perturbation. Although this is a simple way to protect the statistical properties of a data set, this method can degrade data utility (Chamikara et al., 2018).

3.2.3 Multiplicative perturbation

In contrast to additive perturbation, multiplicative perturbation multiplies original data by noise from any distribution such as uniform or Gaussian distributions to protect the privacy of the original data (Liu et al., 2005). Multiplicative perturbation is more effective than additive perturbation in preserving privacy because it is more difficult to reconstruct the original data from the perturbed data in multiplicative perturbation. However, data protected through multiplicative perturbation methods can be vulnerable to known sample attacks and Independent Component Analysis (ICA) attacks (Chamikara et al., 2018; Liu et al., 2005).

3.3 Anonymization techniques

Anonymization methods have been developed to preserve the privacy of structured data by removing identifiable information while maintaining the utility of the data set. The general purpose of anonymization methods is to release data about individuals without revealing sensitive information about them (Sweeney, 2002). The basic concept of anonymization is to hide both explicit identifiers as well as quasi-identifiers which is a set of attributes that can be linked to other public databases for identifying individual records (Ding et al., 2019; Raymond et al., 2009). After anonymization, the quasi-identifiers of a set of records are generalized to different groups in such a way that the record of an individual person cannot be re-identified. k-anonymity, l-diversity, and t-closeness are well-known three anonymization techniques in practice.

3.3.1 k-anonymity

A table is said to satisfy k-anonymity if each record in the table cannot be distinguished from at least $k - 1$ other records with respect to every set of quasi-identifiers (Sweeney, 2002). This means that for each combination of quasi-identifiers in the k-anonymous table, there will be at least k records that share the same values. Generalization and suppression techniques can be used to provide k-anonymity to a table of data (Samarati & Sweeney, 1998). Since k-anonymization was primarily developed for static data sets, data aging is a problem when working with dynamic data (Ugur & Osman, 2020). Although k-anonymity protects the privacy of data while maintaining the utility of published data, an attacker may still find the values of sensitive attributes when there is little diversity among the sensitive attributes. In addition, k-anonymity does not guarantee privacy against attackers with background knowledge of published data.

3.3.2 l-diversity

To address the shortcomings of k-anonymity in terms of background knowledge and homogeneity attacks, Machanavajjhala et al. (2007) introduced another anonymization-based technique called l-diversity. l-diversity requires that each equivalence class have at least l well-represented values for each sensitive attribute in such a way that adds diversity within a group. However, l-diversity is limited in its assumption of adversarial knowledge and may be vulnerable to attribute linkage attacks.

3.3.3 t-closeness

To provide privacy beyond k-anonymity and l-diversity, Li et al. (2007) proposed an anonymization-based technique called t-closeness. The key idea behind the t-closeness is to set the distribution of a sensitive attribute in any equivalence class close to the distribution of the attribute in the overall table. Although this mechanism effectively limits the amount of individual-specific information an observer can learn, it reduces the utility of the data.

Table 2 presents the positives and negatives of general privacy protection mechanisms.

Table 2 The positives and negatives of general privacy protection mechanisms

Mechanism	Positives	Negatives
Homomorphic Encryption. Fang and Qian (2021), Park and Lim (2022), Rivest et al. (1978)	Ensures that there is no performance loss in the model convergence.	Computation costs are significant.
Secure Multi-Party Computation. Cramer et al. (2000), Goldreich (1998), Truex et al. (2019)	High accuracy of computations.	Communication costs are significant.
	Ability to eliminate the need for trusted third parties.	Computational overhead is significant.
	Solves the problem of the trade-off between data utility and data privacy.	
Secret Sharing. Bonawitz et al. (2017), Shamir (1979)	Solves the problem of user dropout during the training.	Vulnerable to dishonest dealers or malicious players.
Differential Privacy. Dwork (2006), Wei et al. (2020)	Provides an acceptable accuracy loss along with a strong privacy guarantee.	Model utility reduces with the assurance of stronger privacy.
Additive Perturbation. Chamikara et al. (2018), Kargupta et al. (2003)	A simple way to protect the statistical properties of a data set.	Degrades data utility.
Multiplicative Perturbation. Chamikara et al. (2018), Liu et al. (2005)	Ensures a higher privacy.	Vulnerable to known sample attacks.
k-anonymity. Samarati and Sweeney (1998), Sweeney (2002)	Protects data privacy while maintaining the utility of published data.	Sensitive attributes are revealed when there is little diversity among the sensitive attributes.
		Does not guarantee privacy against attackers with background knowledge of published data.
l-diversity. Machanavajjhala et al. (2007)	Provides protection against background knowledge and homogeneity attacks.	May be vulnerable to attribute linkage attacks.
t-closeness. Li et al. (2007)	Effectively limits the amount of individual-specific information an observer can learn.	Reduces data utility.

4 Privacy leakages in federated learning and possible defense mechanisms

The concept of federated learning seems to inherently guarantee better privacy for sensitive user data as it facilitates the joint training of a global model by exchanging only model parameters instead of raw data. However, several recent studies have shown that privacy can be compromised in federated learning because malicious adversaries can reveal sensitive information from model parameters obtained during the training stage as well as finally released model (Luo et al., 2021; Wang et al., 2019). A privacy breach can happen in any phase of the life-cycle of federated learning and internal as well as external parties can harm privacy. In addition, attacks on federated learning can be active attacks that modify the protocol to retrieve sensitive information as well as passive attacks that learn from the system without making any changes to the system.

There is a high probability of encountering privacy leakages during the training phase of federated learning and vulnerabilities in system configurations as well as poor quality of client data can lead to these privacy breaches (Liu et al., 2022). Computation of local parameters by the clients, the transmission of intermediate updates between the clients and the aggregating server, and global model aggregation by the server are key functions of the model training phase. Therefore, during the training phase, all updated information such as local model parameters, aggregated model parameters, and the final model is exposed to malicious adversaries. As a result, different kinds of privacy breaching attacks can arise during each function in the model training phase. The malicious adversaries in the training phase can be internal parties such as participating clients and the central server as well as externals such as eavesdroppers (Truex et al., 2019). Since the participating clients or the central server have direct access to the intermediate model updates during the training phase, if they want, they can infringe on the privacy of others without much effort. Moreover, the eavesdroppers can acquire the intermediate model updates by intercepting the communication between the participating clients and the server. Figure 3 shows the internal and external attackers identified during the training phase of centralized federated learning. The privacy leakages in predicting phase of federated learning are generally based on the converged global model released to end users (Liu et al., 2022). Privacy risks are relatively low during the predicting phase as most end users have only access to query results provided by the platform (Shokri et al., 2017). Membership inference attacks, model inversion attacks, property inference attacks, and model extraction attacks are common privacy attacks found in federated learning.

4.1 Membership inference attacks

The purpose of membership inference attacks is to determine whether a specific data sample was used to train the target model (Lu et al., 2020; Nasr et al., 2019). Membership inference attacks can be either active or passive attacks, and active attacks are more powerful because of the ability to modify model parameters. Meanwhile, passive attacks can be further classified as passive black-box attacks and passive white-box attacks. In a passive black-box attack, the adversary can only access query results while in a passive white-box attack, the adversary can access the intermediate training updates, model parameters, and query results (Lu et al., 2020). Therefore, passive black-box attacks have a limited privacy risk while passive white-box attacks have significant privacy risks (Shokri et al., 2017). In general, an honest-but-curious server or honest client does not launch active inference attacks, and a malicious server or adversarial client may be prone to launch active inference attacks (Liu et al., 2022; Nasr et al.,

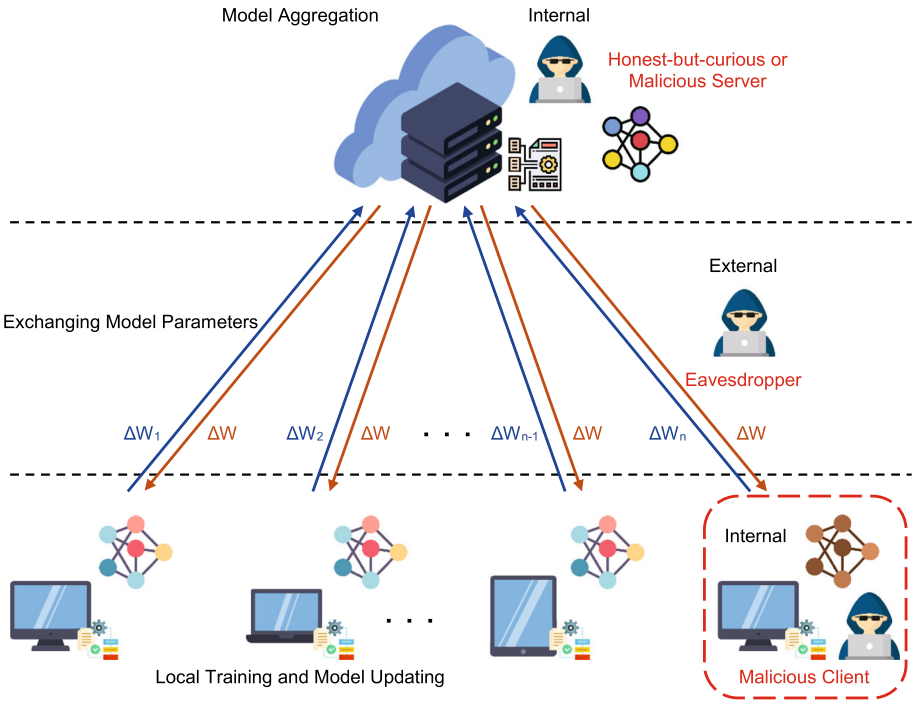

Fig. 3 Internal and external attackers during the training phase of centralized federated learning

2019). In the study by Nasr et al. (2019), the researchers found that even well-generalized models are significantly susceptible to white-box membership inference attacks. Furthermore, in the same study, the researchers pointed out that adversarial participants in the federated learning setting can successfully launch active membership inference attacks against other participants. Moreover, they revealed that a central parameter server can launch membership inference attacks more successfully than a local participant. Shokri et al. (2017) implemented and evaluated membership inference attacks against machine learning models when having black-box access to a model. Here the researchers adapted to the shadow training technique, which trains an attack model to distinguish whether the output of the target model was based on the samples included in the training data set. Additionally, in the same study, the researchers emphasized that overfitting is a common reason for a model to be vulnerable to membership inference and models with more output classes are more likely to be subject to membership inference attacks.

Many researchers (Carlini et al., 2022; Shokri et al., 2017; Zhong et al., 2022) proposed differential privacy as a viable solution to protect against membership inference attacks. Moreover, Ganju et al. (2018) as well as Melis et al. (2019) stated that membership inference attacks can be successfully suppressed by applying record-level differential privacy. Since overfitting is recognized as a major cause of membership inference attacks, training models adhering to standard techniques to reduce overfitting using regularization mechanisms such as weight decay and train-time augmentations is an ordinary solution to protect against membership inference attacks (Carlini et al., 2022; Zhong et al., 2022). Furthermore, Jia et al. (2019) revealed that adversarial examples can be used as a defense mechanism to protect against membership inference attacks. Unfortunately, many of these solutions degrade model

performance and have been shown to be vulnerable to more advanced types of attacks (Carlini et al., 2022).

4.2 Model inversion attacks

The purpose of the model inversion attacks is to infer the characteristics of each class so that making it possible to construct representatives of the respective classes (Melis et al., 2019). Basically, in the model inversion attacks, the output of the model is used to infer certain characteristics of the input for the model. Model inversion attacks are often carried out when an attacker can join the training process or obtain the predictive values of the target classifier. In other words, model inversion attacks can be carried out when the attacker has either white-box or black-box access to the model. Model inversion attacks can be further classified as class representative inference attacks and data reconstruction attacks. The purpose of class representative inference attacks is to generate representative samples of target labels that behave similarly to the actual data instances in the training data sets in order to study sensitive information about the training data sets (Wang et al., 2019). Generally, the input reconstructed using the model inversion attack is not an actual member of the training data set and is a sample representing the average of the features of a specified class when model inversion is conducted to infer the class representative (Shokri et al., 2017). On the other hand, the purpose of data reconstruction attacks is to reconstruct the original data samples used in the model training and the corresponding labels (Bhowmick et al., 2018; Nasr et al., 2019). Since the central server maintains the intermediate updates of all local clients during the training phase of federated learning, a malicious server can easily launch white-box data reconstruction attacks (Liu et al., 2022). Sannai (2018) presented a reconstruction theorem for the training samples based on the loss functions of deep neural networks. Additionally, Fredrikson et al. (2015) introduced model inversion attacks that could predict the sensitive features used as inputs for decision tree models and attacks that can recover images using black-box access to a facial recognition model. Here, the researchers used model inversion to reconstruct an image of the victim's face from the label produced by the model as well as learn the identity of the corresponding individual from an image containing a blurred-out face. Generative Adversarial Networks (GAN) can generate samples that appear to come from the training set by placing a generative deep neural network against a discriminative deep neural network (Hitaj et al., 2017). Here the GAN does not need access to the actual training set and it only relies on the information stored in the discriminative model. However, all data samples for a class need to be similar in order for the extracted class representatives by GAN to be similar to the training data (Melis et al., 2019). In addition, here the adversary must have prior knowledge about the data labels of the victim.

Fredrikson et al. (2015) demonstrated a defense mechanism against model inversion attacks by degrading the quality or precision of the gradient information retrievable from the model. Furthermore, the researchers emphasized that although this could be achieved in the black-box setting, there is no clear way to achieve this in the white-box setting while preserving the model's utility. Differential privacy mechanisms can be applied as a viable solution to model inversion attacks when the privacy budget (ε) is carefully chosen and the privacy guaranteed by differential privacy mechanisms increases for small ε while the protection decreases as ε increases (Fredrikson et al., 2014). Bhowmick et al. (2018) revealed that local differential privacy can be used to successfully defend against data reconstruction attacks. Additionally, Ganju et al. (2018) stated that differentially-private mechanisms can

provide better protection against model inversion attacks, as the attack focuses on the privacy of individual records in the data set.

4.3 Property inference attacks

Property inference attacks which are also known as distribution inference attacks are intended to infer property information related to training data sets (Melis et al., 2019). These attacks are based on the idea that machine learning models trained on similar data sets using similar training methods will represent similar functions (Ganju et al., 2018). In other words, the property inference attacks utilize the similarities in the parameters of models to predict whether some property can be exposed from a target model. Mainly, these inferred properties may not be relevant to the core training task and the goal of the adversary is to identify patterns in the target model to reveal any properties that the model owner is not desired to release (Melis et al., 2019). For example, when the main task is to train a model for identifying the nationality of individuals, the property inference attack may intend to infer the gender distribution in the training data set. The property inference attacks can be further categorized as passive and active property inference attacks. The active adversaries can deceive the federated learning model to better separate data with and without target attributes, thereby stealing more information while the passive adversaries can only observe model updates and train a binary attribute classifier of target property to perform inferences (Liu et al., 2022). In order to perform property inference attacks, the adversary needs the auxiliary training data that is correctly labeled with the property that wants to make the inference. In addition, for active property inference attacks, the auxiliary data point should also be labeled for the main task. Ateniese et al. (2015) tested property inference attacks against models based on Support Vector Machines (SVMs) and Hidden Markov Models (HMMs). There, the researchers created a set of shadow classifiers and trained them for a task similar to the target classifier with a data set similar to that of the target classifier, but have an idea of whether a specific property exists or not. Furthermore, they emphasized that differential privacy mechanisms that preserve record-level privacy are incapable of providing effective coverage against property inference attacks. Meanwhile, Ganju et al. (2018) focused on making inferences about sensitive global properties of a training data set of fully connected neural networks.

Ma et al. (2021) proposed a defense mechanism for property inference attacks during model training through collaborative meta-learning architecture to learn the common knowledge over all participants and utilize the natural advantage of meta-learning to hide the sensitive property data. Meanwhile, Stock et al. (2022) presented an approach called property unlearning based on adversarial training and training data preprocessing techniques to protect against white-box property inference attacks without any changes to the target model. Applying node multiplicative transformations, adding noise by flipping the labels of some training samples, and encoding a significant amount of information about the training set are some possible defenses against property inference attacks (Ganju et al., 2018).

4.4 Model extraction attacks

An adversary launches model extraction attacks for the purpose of extracting the parameters of a model trained on private data (Tramèr et al., 2016). Here, an adversary aims to construct a model which closely approximates or matches the functionality of the target model (Shokri et al., 2017; Wu et al., 2022). Launching a model extraction attack is a very complex task

because in most cases the attacker has no prior knowledge of the model and has only black-box access to the target model. In the study conducted by Tramèr et al. (2016), the researchers tested how to successfully launch model extraction attacks against various machine learning models such as decision trees, logistic regressions, support vector machines, and deep neural networks. Meanwhile, Wu et al. (2022) developed a series of model extraction attacks by generating legitimate-looking queries as the normal nodes among the target graph and then utilizing the query responses and accessible structure knowledge to reconstruct the model.

Tramèr et al. (2016) revealed that rounding confidence scores to a certain fixed precision is a possible defense against model extraction attacks, but it is not applicable to some models such as regression trees. In defending against model extraction attacks, Chen et al. (2022) proposed a meta-learning-based disruption detection algorithm to learn the fundamental differences between the distributions of disrupted and undisrupted query results. Although researchers have suggested applying differential privacy directly to model parameters as a defense mechanism against model extraction attacks, further study is needed to determine the required privacy budget for a strong privacy guarantee (Tramèr et al., 2016).

Table 3 presents the common privacy inference attacks on federated learning along with the purposes of those attacks and possible defense mechanisms.

Table 3 Privacy leakages in federated learning

Attack	Purpose of the Attack	Defense Mechanisms
Membership Inference Attacks. Lu et al. (2020), Shokri et al. (2017)	Determining whether a specific data sample was used to train the target model.	Applying record-level differential privacy.
		Reducing overfitting in model training.
		Using adversarial examples to mislead the attacker.
Model Inversion Attacks. Fredrikson et al. (2014), Fredrikson et al. (2015)	Inferring the characteristics of each class so that making it possible to construct representatives of the respective classes.	Applying local differential privacy.
Property Inference Attacks. Ateniese et al. (2015), Lu et al. (2020)	Inferring the property information related to training data sets.	Applying node multiplicative transformations.
		Adding noise by flipping the labels of some training samples.
		Encoding a significant amount of information about the training set.
Model Extraction Attacks. Tramèr et al. (2016), Wu et al. (2022)	Extracting the parameters of a model trained on private data.	Rounding confidence scores to a certain fixed precision.
		Application of differential privacy.

5 Frameworks for implementing and simulating federated learning

As a result of the growing demand and popularity of federated learning, the research and industrial community has focused on introducing a number of tools and frameworks for implementing and simulating federated learning systems. These tools and frameworks have different levels of capability to handle common issues of federated learning, such as heterogeneity, security, and privacy. In general, these frameworks can be categorized as simulation-oriented libraries and production-oriented libraries (He et al., 2020). TensorFlow Federated (Bonawitz et al., 2020), PySyft (Ryffel et al., 2018), FedML (He et al., 2020) and LEAF (Caldas et al., 2018) are examples of some simulation-oriented libraries, whereas production-oriented libraries include FATE (Liu et al., 2021b), PaddleFL (PaddlePaddle, 2020), IBM federated learning (Ludwig et al., 2020), and Flower (Beutel et al., 2020). The continued development of these tools and frameworks is highly affected to accelerate the progress of federated learning research.

TensorFlow Federated TensorFlow Federated (Bonawitz et al., 2020) is an open-source framework for machine learning that facilitates open research and experimentation on federated learning. TensorFlow Federated allows developers to experiment with large-scale simulation capabilities of existing federated learning algorithms on their models and data, as well as new algorithms. The interfaces of TensorFlow Federated consist of two main layers, federated learning API and Federated Core API. Basically, federated learning API allows the developers to apply Federated training and evaluation to their existing TensorFlow models, while the Federated Core API facilitates to experiment on novel federated algorithms by combining TensorFlow with distributed communication operators within a strongly-typed functional programming environment. In addition, it provides starting points on federated learning for beginners and complete examples of many types of research for researchers.

PySyft PySyft (Ryffel et al., 2018) is a reliable and secure, general framework for privacy-preserving deep learning. This Python library is built over PyTorch (Imambi et al., 2021) and the design of the framework relies on chains of tensors that are exchanged between local and remote workers. In addition, PySyft cares about data ownership and secure processing. Therefore, this framework facilitates the implementation of complex privacy-preserving constructs such as federated learning, secure multiparty computation, and differential privacy.

FedML FedML (He et al., 2020) is an open research-oriented library that facilitates the development of federated learning algorithms. This library supports a variety of algorithms that cover horizontal and vertical federated learning as well as split learning. In addition, FedML supports performance comparison of different federated learning algorithms. Supportability for on-device training for edge devices, distributed computing and single-machine simulation is another specialty of this open research library.

LEAF LEAF (Caldas et al., 2018) is a modular benchmarking framework for learning in Federated settings. This framework consists of three modules which are a suite of open-source datasets, a set of reference implementations, and an array of statistical and systems metrics. LEAF allows researchers and practitioners to experiment with federated learning, meta-learning, and multi-task learning under more realistic assumptions than before.

FATE Federated AI Technology Enabler (FATE) (Liu et al., 2021b), initiated by the AI department of WeBank and now part of the Linux Foundation, is the world's first open-source federated learning platform. The main objective of the FATE project was to develop and promote advanced AI technologies that protect data security and user privacy. FATE implements

secure computation protocols based on homomorphic encryption and secure multi-party computation. This industrial-grade secure computing framework facilitates a series of federated learning architectures and secure computation algorithms based on logistic regression, tree-based algorithms, artificial neural networks, and many more machine learning algorithms.

PaddleFL PaddleFL (PaddlePaddle, 2020) is an open-source federated learning framework based on the PaddlePaddle (Ma et al., 2019) and it facilitates the simulation and comparison of various federated learning algorithms. PaddleFL provides several federated learning strategies that can be used in applications such as computer vision, natural language processing, and recommendation. This framework supports horizontal federated learning as well as vertical federated learning architectures. Data Parallel, which allows distributed data holders to conduct their federated learning tasks based on common horizontal federated learning strategies such as FedAvg and federated learning with MPC, which is implemented based on secure multi-party computation to enable secure training and prediction are two main components in PaddleFL.

IBM Federated Learning IBM federated learning (Ludwig et al., 2020) framework has developed with the objectives of addressing the challenges in federated learning such as data heterogeneity, the robustness of the federation process, security and privacy inference prevention, as well as facilitating the integration of federated learning into effective machine learning activities in the enterprise environments. This Python-based framework is independent of a particular machine learning library or machine learning model and can be used for deep neural networks as well as traditional approaches such as decision trees or support vector machines.

Flower Flower (Beutel et al., 2020) is an extensive open-source federated learning framework that facilitates large-scale experiments about scalable federated learning workloads on heterogeneous edge devices. Secure aggregation implementation of the Flower meets the five goals of usability, flexibility, compatibility, reliability, and efficiency. Most importantly, Flower enables federated learning evaluation on real devices.

6 Applications of federated learning

Recently, various fields are paying more attention to the privacy of sensitive user information, and the applications of federated learning are widespread because its data privacy protection capability provides the necessary assurance for the development of intelligent systems. Healthcare, banking and finance, smart cities, and recommendation systems can be introduced as some of the current mainstream federated learning application areas.

Healthcare Federated learning addresses many challenges in medical applications, such as the lack of publicly available multicenter diverse data sets, as well as privacy and confidentiality issues. In cases where federated learning is applied to healthcare applications, it may be a cross-silo setup where each hospital represents the individual client while a data center of a government agency plays the role of the server. Meanwhile, it can be a cross-device arrangement where an individual with a wearable healthcare device is considered as a client and a data center of a healthcare organization is considered as the server. In a federated learning setting, Ma et al. (2022) proposed an assisted diagnosis model that would allow physicians to provide cancer patients with personalized nutrition plans and treatment options for patients and prolong patient life. Arikumar et al. (2022) presented a computationally efficient health

monitoring system based on federated learning that detects and monitors patient activity and movement through wearable sensor devices.

Banking and Finance The banking and finance sector uses federated learning approaches to better understand customers' investment potential and credit rating scores. These applications can be horizontal federated learning settings where several banks collaboratively train depositor credit forecasting models, as well as vertical federated learning setups where banks interact with various other financial institutions to train models for forecasting loan repayment ability. Currently, WeBank has deployed a vertical federated learning-based risk management application for small and micro-enterprise loans where a model is trained using data from an invoicing agency and a bank (Cheng et al., 2020). This application has been implemented with the FATE platform and it has been observed that the model built with federated learning performs significantly better than the model built using only data from the bank.

Smart Cities The concept of smart cities has become a reality through the joint efforts of government agencies, private enterprises and individuals, however, the requirement for data privacy limits data exchange at various levels and is a major barrier to the further development of smart cities. Fortunately, the emergence of federated learning has made it easier to effectively address the problem of data isolation islands and facilitate the integration of data at all levels of the city to provide better smart city applications. Targeting the problems of data islands and data leakage in smart city applications, Liu et al. (2021a) proposed a lightweight and reliable sharing mechanism by integrating blockchain and federated learning while introducing a node selection algorithm to improve the quality of federated learning. Meanwhile, Kuang and Chen (2023) introduced a federated learning approach to smart cities with privacy guarantees using encryption and differential privacy. Here, researchers were able to achieve effective data communication and high accuracy by applying an edge-asynchronous communication framework.

Recommendation Systems Recommendation systems have become an emerging tool in the marketing field to inspire users' next choice or purchase behavior. Generally, recommendation systems collect and analyze a large amount of user data, including previous user behavior, when recommending goods or services, but privacy issues remain an obstacle to the efficient implementation of recommendation systems. In such a situation federated learning can be applied as a viable solution and Du et al. (2021) proposed a user-level distributed matrix factorization framework for federated learning in recommendation systems. Additionally, here, the authors have enhanced the privacy of their framework with Homomorphic Encryption and randomized response. Meanwhile, Vyas et al. (2023) proposed a federated learning-based driver recommendation system for the next trip taking into account the driver's stress state and previous driving behavior.

7 Future research directions

Although there have been extensive studies on the privacy of federated learning, there are still several open problems in this field of research. Research efforts are possible to implement novel methods to address the trade-off between privacy and utility, as well as to introduce modifications to existing approaches to enhance privacy. Some suggestions for future research are as follows.

Client-Based Multi-Level Privacy in Horizontal Federated Learning Different clients participating in federated learning can demand different levels of privacy. For instance, in a cross-silo federated learning setting, data with one client may be more sensitive than data with another client, and clients with highly sensitive data can expect a higher level of privacy. In such a case, a suggestion can be made to implement a mechanism that has different levels of privacy protection for different clients. This approach may cause to improve the accuracy of the final model as well as training and predicting efficiency.

Privacy-Preserving vertical federated learning for Instances where Labels are Distributed between Clients Most of the existing research studies on vertical federated learning have considered situations where only one participating client holds the complete set of labels relevant to all data samples. However, there are many practical situations where labels are distributed among several clients while other features have vertically partitioned among different clients. Collaboratively training a machine learning model in such a situation is a complex task. Therefore, a new mechanism can be proposed to address this problem efficiently and effectively.

Hierarchical Setting to Preserve Privacy in Federated Learning Many existing federated learning approaches consist of a single coordinator who aggregates the model parameters sent by participating clients. This setup is susceptible to some privacy attacks, and communication between clients and the aggregator can also be quite inefficient. A system with several aggregators at different levels of a hierarchy can be proposed as a solution. Through careful selection of clients and a well-designed communication protocol, the trust between participating entities can be enhanced while reducing the communication overhead. Additionally, focus can be placed on improving accuracy and efficiency over existing mechanisms.

Data Anonymization Approach to Privacy Protection in Federated Learning Most federated learning related research has applied differential privacy to improve the privacy in the proposed solution. But perturbation-based techniques are more computationally complex and it may cause a reduction in the prediction accuracy of the trained model. It can be found that a very limited number of research have focused on the application of anonymization techniques in federated learning. Hence, future directions are open for applying data anonymization techniques to build an efficient solution for privacy in federated learning.

Hybrid Approach to Enhancing Privacy in Federated Learning Based on the review of various privacy-preserving mechanisms, it can be identified that different privacy protection techniques have their own strengths while having some weaknesses. Better data utilization while minimizing privacy breaches is possible by meaningfully applying different privacy protection mechanisms at different stages of federated learning. Therefore, future studies can be directed toward implementing novel effective privacy protection solutions by combining different general privacy protection mechanisms.

8 Conclusion

Federated learning allows distributed clients to collaboratively train a shared global model while retaining all the training data locally and this revolutionized traditional machine learning approaches by guaranteeing a higher level of privacy for client data. However, several recent studies have shown that there is still a risk of sensitive private data being leaked during the model training and predicting stages. Not only malicious clients and servers but also

eavesdroppers and end users of the trained model can infer sensitive information of data owners using intermediate model updates, final model, and query results obtained. Membership inference attacks, model inversion attacks, property inference attacks, and model extraction attacks are the most common privacy attacks faced by federated learning. Although general privacy protection mechanisms such as cryptographic techniques, perturbation techniques, and anonymization techniques can be applied to minimize privacy breaches in federated learning, it can still detect a trade-off between privacy and utility. Future research studies can focus on implementing mechanisms to improve the privacy of federated learning while providing better data utility.

Author Contributions Literature search and analysis: Hashan Ratnayake, Lin Chen; Writing - original draft preparation: Hashan Ratnayake; Writing - review and editing: Xiaofeng Ding, Lin Chen; Supervision: Xiaofeng Ding.

Funding No funding was received to assist with the preparation of this manuscript.

Declarations

Consent for publication We consent this paper to be published in the Journal of Intelligent Information Systems.

Competing interests The authors declare no competing interests.

References

Aimin, Q., Guosong, S., & Wentong, Z. (2018). Assessing China's Cybersecurity Law. *Computer Law & Security Review, 34*(6), 1342–1354. https://doi.org/10.1016/j.clsr.2018.08.007

Alaggan, M., Gambs, S., & Kermarrec, A.-M. (2017). Heterogeneous differential privacy. *The Journal of Privacy and Confidentiality, 7*(2), 1–27. https://doi.org/10.29012/jpc.v7i2.652

Arachchige, P. C. M., Bertók, P., Khalil, I., et al. (2020). Local differential privacy for deep learning. *IEEE Internet Things Journal, 7*(7), 5827–5842. https://doi.org/10.1109/JIOT.2019.2952146

Arikumar, K. S., Prathiba, S. B., Alazab, M., et al. (2022). FL-PMI: Federated learning-based person movement identification through wearable devices in smart healthcare systems. *Sensors, 22*(4), 1377. https://doi.org/10.3390/s22041377

Asad, M., Moustafa, A., & Ito, T. (2021). Federated learning versus classical machine learning: A convergence comparison (p. 9). arXiv preprint arXiv:2107.10976. https://doi.org/10.48550/arXiv.2107.10976

Ateniese, G., Mancini, L. V., Spognardi, A., et al. (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks, 10*(3), 137–150. https://doi.org/10.1504/IJSN.2015.071829

Beutel, D. J., Topal, T., Mathur, A., et al. (2020). Flower: A friendly federated learning research framework (p. 15). arXiv preprint arXiv:2007.14390. https://doi.org/10.48550/arXiv.2007.14390

Bhowmick, A., Duchi, J., Freudiger, J., et al. (2018). Protection against reconstruction and its applications in private federated learning (p. 45). arXiv preprint arXiv:1812.00984. https://doi.org/10.48550/arXiv.1812.00984

Bogdanov, D., Laur, S., & Willemson, J. (2008). Sharemind: A framework for fast privacy-preserving computations. *Computer Security - ESORICS 2008* (pp. 192–206). https://doi.org/10.1007/978-3-540-88313-5_13

Bonawitz, K., Eichner, H., Grieskamp, W., et al. (2020). *TensorFlow Federated: Machine learning on decentralized data*. Retrieved from April 10, 2023 from https://www.tensorflow.org/federated

Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2017). Practical secure aggregation for privacy-preserving machine learning. *ACM Conf. Comput. Commun.* (pp. 1175–1191). https://doi.org/10.1145/3133956.3133982

Caldas, S., Duddu, S. M. K., Wu, P., et al. (2018). LEAF: A benchmark for federated settings (p. 9). arXiv preprint arXiv:1812.01097. https://doi.org/10.48550/arXiv.1812.01097

Carlini, N., Chien, S., Nasr, M., et al. (2022). Membership inference attacks from first principles. *2022 IEEE Secur. Priv.* (pp. 1897–1914). https://doi.org/10.1109/SP46214.2022.9833649

Chamikara, M. A. P., Bertók, P., Liu, D., et al. (2018). Efficient data perturbation for privacy preserving and accurate data stream mining. *Pervasive and Mobile Computing, 48*, 1–19. https://doi.org/10.1016/j.pmcj.2018.05.003

Chen, Y., Guan, R., Gong, X., et al. (2022). D-DAE: Defense-penetrating model extraction attacks. *2023 IEEE Secur. Priv.* (pp. 432–449).

Cheng, Y., Liu, Y., Chen, T., et al. (2020). Federated learning for privacy-preserving AI. *Communications of the ACM, 63*(12), 33–36. https://doi.org/10.1145/3387107

Chik, W. B. (2013). The Singapore Personal Data Protection Act and an assessment of future trends in data privacy reform. *Computer Law & Security Review, 29*(5), 554–575. https://doi.org/10.1016/j.clsr.2013.07.010

Cramer, R., Damgård, I., & Maurer, U. (2000). General secure multi-party computation from any linear secret-sharing scheme. *Advances in Cryptology - EUROCRYPT 2000* (pp. 316–334). https://doi.org/10.1007/3-540-45539-6_22

Ding, X., Zhang, F., & Jin, H. (2019). Data anonymization for big crowdsourcing data. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 1–6). https://doi.org/10.1109/INFOCOMWKSHPS47286.2019.9093748

Du, Y., Zhou, D., Xie, Y., et al. (2021). Federated matrix factorization for privacy-preserving recommender systems. *Applied Soft Computing, 111*, 107700. https://doi.org/10.1016/j.asoc.2021.107700

Duan, M., Liu, D., Chen, X., et al. (2020). Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems, 32*(1), 59–71. https://doi.org/10.1109/TPDS.2020.3009406

Dwork, C. (2006). Differential privacy. *Automata, Languages and Programming* (pp. 1–12). https://doi.org/10.1007/11787006_1

Fang, H., & Qian, Q. (2021). Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet, 13*(4), 94. https://doi.org/10.3390/fi13040094

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *22nd ACM Conf. Comput. Commun.* (pp. 1322–1333). https://doi.org/10.1145/2810103.2813677

Fredrikson, M., Lantz, E., Jha, S., et al. (2014). Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. *23rd USENIX Security* (pp. 17–32).

Ganju, K., Wang, Q., Yang, W., et al. (2018). Property inference attacks on fully connected neural networks using permutation invariant representations. *ACM Conf. Comput. Commun.* (pp. 619–633). https://doi.org/10.1145/3243734.3243834

George, M., & Zoran, O. (2015). A distributed decision support algorithm that preserves personal privacy. *Journal of Intelligent Information Systems,* 107–132. https://doi.org/10.1007/s10844-014-0331-6

Goldman, E. (2021). An introduction to California's Consumer Privacy Laws (CCPA and CPRA). *Santa Clara Univ. Legal Studies Research Paper* (p. 9). https://doi.org/10.2139/ssrn.3896176

Goldreich, O. (1998). Secure multi-party computation. *Manuscript. Preliminary version, 78*, 110.

Hardy, S., Henecka, W., Ivey-Law, H., et al. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption (pp. 60). arXiv preprint arXiv:1711.10677. https://doi.org/10.48550/arXiv.1711.10677

He, C., Li, S., So, J., et al. (2020). FedML: A research library and benchmark for federated machine learning (p. 18). arXiv preprint arXiv:2007.13518. https://doi.org/10.48550/arXiv.2007.13518

Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep models under the GAN: information leakage from collaborative deep learning. *ACM Conf. Comput. Commun. Secur.*, 603–618. https://doi.org/10.1145/3133956.3134012

Hu, K., Li, Y., Xia, M., et al. (2021). Federated learning: A distributed shared machine learning method. *Complexity, 2021*, 20. https://doi.org/10.1155/2021/8261663

Hu, Y., Niu, D., Yang, J., et al. (2019). FDML: A collaborative machine learning framework for distributed features. *25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* (pp. 2232–2240). https://doi.org/10.1145/3292500.3330765

Huang, W., Li, T., Wang, D., et al. (2022). Fairness and accuracy in horizontal federated learning. *Information Sciences, 589*, 170–185. https://doi.org/10.1016/j.ins.2021.12.102

Imambi, S., Prakash, K. B., & Kanagachidambaresan, G. R. (2021). Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications* (pp. 87–104). https://doi.org/10.1007/978-3-030-57077-4_10

Jia, J., Salem, A., Backes, M., et al. (2019). MemGuard: Defending against black-box membership inference attacks via adversarial examples. *2019 ACM SIGSAC Conference on Computer and Communications Security*, (pp. 259–274). https://doi.org/10.1145/3319535.3363201

Jing, Q., Wang, W., Zhang, J., et al. (2019). Quantifying the performance of federated transfer learning (p. 7). arXiv preprint arXiv:1912.12795. https://doi.org/10.48550/arXiv.1912.12795

Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning, 14*(1–2), 1–210. https://doi.org/10.1561/2200000083

Kamp, M., Fischer, J., & Vreeken, J. (2021). Federated learning from small datasets (p. 13). arXiv preprint arXiv:2110.03469. https://doi.org/10.48550/arXiv.2110.03469

Kargupta, H., Datta, S., Wang, Q., et al. (2003). On the privacy preserving properties of random data perturbation techniques. *Third IEEE International Conference on Data Mining* (pp. 99–106). https://doi.org/10.1109/ICDM.2003.1250908

Kuang, Z., & Chen, C. (2023). Research on smart city data encryption and communication efficiency improvement under federated learning framework. *Egyptian Informatics Journal, 24*(2), 217–227. https://doi.org/10.1016/j.eij.2023.02.005

Kulynych, J., & Korn, D. (2003). The new HIPAA (Health Insurance Portability and Accountability Act of 1996) Medical Privacy Rule: Help or hindrance for clinical research? *Circulation, 108*(8), 912–914. https://doi.org/10.1161/01.CIR.0000080642.35380.50

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. *2007 IEEE 23rd Int. Conf. Data Eng.* (pp. 106–115). https://doi.org/10.1109/ICDE.2007.367856

Li, T., Sahu, A. K., Talwalkar, A., et al. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine, 37*(3), 50–60. https://doi.org/10.1109/MSP.2020.2975749

Liang, X., Liu, Y., Luo, J., et al. (2021). Self-supervised cross-silo federated neural architecture search (p. 12). arXiv preprint arXiv:2101.11896. https://doi.org/10.48550/arXiv.2101.11896

Ling, Q., Yingjiu, L., & Xintao, W. (2007). Preserving privacy in association rule mining with bloom filters. *Journal of Intelligent Information Systems,* 253–278. https://doi.org/10.1007/s10844-006-0018-8

Liu, C., Guo, S., Guo, S., et al. (2021). LTSM: Lightweight and trusted sharing mechanism of IoT data in smart city. *IEEE Internet of Things Journal, 9*(7), 5080–5093. https://doi.org/10.1109/JIOT.2021.3110097

Liu, K., Kargupta, H., & Ryan, J. (2005). Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering, 18*(1), 92–106. https://doi.org/10.1109/TKDE.2006.14

Liu, P., Xu, X., & Wang, W. (2022). Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity, 5*(1), 1–19. https://doi.org/10.1186/s42400-021-00105-6

Liu, Y., Fan, T., Chen, T., et al. (2021). FATE: An industrial grade platform for collaborative learning with data protection. *Journal of Machine Learning Research, 22*(1), 10320–10325.

Liu, Y., Kang, Y., Xing, C., et al. (2020). A secure federated transfer learning framework. *IEEE Intelligent Systems, 35*(4), 70–82. https://doi.org/10.1109/MIS.2020.2988525

Lu, H., Liu, C., He, T., et al. (2020). Sharing models or coresets: A study based on membership inference attack (p. 8). arXiv preprint arXiv:2007.02977. https://doi.org/10.48550/arXiv.2007.02977

Ludwig, H., Baracaldo, N., Thomas, G., et al. (2020). IBM Federated Learning: An enterprise framework white paper v0. 1 (p. 17). arXiv preprint arXiv:2007.10987. https://doi.org/10.48550/arXiv.2007.10987

Luo, X., Wu, Y., Xiao, X., et al. (2021). Feature inference attack on model predictions in vertical federated learning. *2021 IEEE 37th Int. Conf. Data Eng.* (pp. 181–192). https://doi.org/10.1109/ICDE51399.2021.00023

Ma, X., Li, B., Jiang, Q., et al. (2021). NOSnoop: An effective collaborative meta-learning scheme against property inference attack. *IEEE Internet of Things Journal, 9*(9), 6778–6789. https://doi.org/10.1109/JIOT.2021.3112737

Ma, Y., Yu, D., Wu, T., et al. (2019). PaddlePaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Computing*, *1*(1), 105–115. https://doi.org/10.11871/jfdc.issn.2096.742X.2019.01.011

Ma, Z., Zhang, M., Liu, J., et al. (2022). An assisted diagnosis model for cancer patients based on federated learning. *Frontiers in Oncology,* 713. https://doi.org/10.3389/fonc.2022.860532

Machanavajjhala, A., Kifer, D., Gehrke, J., et al. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, *1*(1), 3–es. https://doi.org/10.1145/1217299.1217302

McMahan, B., Moore, E., Ramage, D., et al. (2017). Communication-efficient learning of deep networks from decentralized data. *20th International Conference on Artificial Intelligence and Statistics* (pp. 1273–1282).

Melis, L., Song, C., De Cristofaro, E., et al. (2019). Exploiting unintended feature leakage in collaborative learning. *2019 IEEE Secur. Priv.* (pp. 691–706). https://doi.org/10.1109/SP.2019.00029

Mothukuri, V., Parizi, R. M., Pouriyeh, S., et al. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems, 115*, 619–640. https://doi.org/10.1109/SP.2019.00029

Mugunthan, V., Goyal, P., & Kagal, L. (2021). Multi-VFL: A vertical federated learning system for multiple data and label owners (p. 5). arXiv preprint arXiv:2106.05468. https://doi.org/10.48550/arXiv.2106.05468

Nasr, M., Shokri, R., & Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Secur. Priv.* (pp. 739–753). https://doi.org/10.1109/SP.2019.00065

PaddlePaddle (2020). *PaddlePaddle/PaddleFL: Federated Deep Learning in PaddlePaddle*. Retrieved April 10, 2023 from https://github.com/PaddlePaddle/PaddleFL

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. *International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 223–238). https://doi.org/10.1007/3-540-48910-X_16

Pardau, S. L. (2018). The California Consumer Privacy Act: Towards a European-style privacy regime in the United States. *Journal of Technology Law & Policy, 23*, 68.

Park, J., & Lim, H. (2022). Privacy-preserving federated learning using homomorphic encryption. *Applied Sciences, 12*(2), 734. https://doi.org/10.3390/app12020734

Phong, L. T., Aono, Y., Hayashi, T., et al. (2018). Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security, 13*(5), 1333–1345. https://doi.org/10.1109/TIFS.2017.2787987

Raymond, W., Jiuyong, L., Ada, F., et al. (2009). (α, k)-anonymous data publishing. *Journal of Intelligent Information Systems,* 209–234. https://doi.org/10.1007/s10844-008-0075-2

Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of Secure Computation, 4*(11), 169–180.

Roy, A. G., Siddiqui, S., Pölsterl, S., et al. (2019). BrainTorrent: A peer-to-peer environment for decentralized federated learning (p 9). arXiv preprint arXiv:1905.06731. https://doi.org/10.48550/arXiv.1905.06731

Ryffel, T., Trask, A., Dahl, M., et al. (2018). A generic framework for privacy preserving deep learning (p. 5). arXiv preprint arXiv:1811.04017. https://doi.org/10.48550/arXiv.1811.04017

Saha, S., & Ahmad, T. (2021). Federated transfer learning: Concept and applications. *Intelligenza Artificiale, 15*(1), 35–44. https://doi.org/10.3233/IA-200075

Samarati, P. & Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04 (p. 19).

Sannai, A. (2018). Reconstruction of training samples from loss functions (p. 11). arXiv preprint arXiv:1805.07337. https://doi.org/10.48550/arXiv.1805.07337

Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612–613. https://doi.org/10.1145/359168.359176

Sharma, S., Xing, C., Liu, Y., et al. (2019). Secure and efficient federated transfer learning. *2019 IEEE Int. Conf. Big Data* (pp. 2569–2576). https://doi.org/10.1109/BigData47090.2019.9006280

Shokri, R., Stronati, M., Song, C., et al. (2017). Membership inference attacks against machine learning models. *2017 IEEE Secur. Priv.* (pp. 3–18). https://doi.org/10.1109/SP.2017.41

Stock, J., Wettlaufer, J., Demmler, D., et al. (2022). Property unlearning: A defense strategy against property inference attacks (p. 16). arXiv preprint arXiv:2205.08821. https://doi.org/10.48550/arXiv.2205.08821

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10*(05), 557–570. https://doi.org/10.1142/S0218488502001648

Tramèr, F., Zhang, F., Juels, A., et al. (2016). Stealing machine learning models via prediction APIs. *25th USENIX Security* (pp. 601–618).

Truex, S., Baracaldo, N., Anwar, A., et al. (2019). A hybrid approach to privacy-preserving federated learning. *12th ACM AISec* (pp. 1–11). https://doi.org/10.1145/3338501.3357370

Ugur, S., & Osman, A. (2020). A utility based approach for data stream anonymization. *Journal of Intelligent Information Systems,* 605–631. https://doi.org/10.1007/s10844-019-00577-6

Vaidya, J., Shafiq, B., Fan, W., et al. (2013). A random decision tree framework for privacy-preserving data mining. *IEEE Transactions on Dependable and Secure Computing, 11*(5), 399–411. https://doi.org/10.1109/TDSC.2013.43

Voigt, P., & von dem Bussche, A. (2017). *Rights of Data Subjects*. Cham: Springer International Publishing.

Vyas, J., Bhumika, Das, D., et al. (2023). Federated learning based driver recommendation for next generation transportation system. *Expert Systems with Applications* (pp. 119951). https://doi.org/10.1016/j.eswa.2023.119951

Wang, Z., Song, M., Zhang, Z., et al. (2019). Beyond inferring class representatives: User-level privacy leakage from federated learning. *2019-IEEE Conf. Comput. Commun.* (pp. 2512–2520). https://doi.org/10.1109/INFOCOM.2019.8737416

Wei, K., Li, J., Ding, M., et al. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15*, 3454–3469. https://doi.org/10.1109/TIFS.2020.2988575

Wu, B., Yang, X., Pan, S., et al. (2022). Model extraction attacks on graph neural networks: Taxonomy and realisation. *ACM Conf. Comput. Commun.* (pp. 337–350). https://doi.org/10.1145/3488932.3497753

Wu, C., Wu, F., Cao, Y., et al. (2021). FedGNN: Federated graph neural network for privacy-preserving recommendation (p. 9). arXiv preprint arXiv:2102.04925. https://doi.org/10.48550/arXiv.2102.04925

Xia, W., Li, Y., Zhang, L., et al. (2021). A vertical federated learning framework for horizontally partitioned labels (p. 10). arXiv preprint arXiv:2106.10056. https://doi.org/10.48550/arXiv.2106.10056

Xu, R., Baracaldo, N., Zhou, Y., et al. (2019). HybridAlpha: An efficient approach for privacy-preserving federated learning. *12th ACM AISec* (pp. 13–23). https://doi.org/10.1145/3338501.3357371

Xue, Y., Niu, C., Zheng, Z., et al. (2021). Toward understanding the influence of individual clients in federated learning. *AAAI Conference on Artificial Intelligence, 35*(12), 10560–10567.

Yang, M., Wang, X., Zhu, H., et al. (2021). Federated learning with class imbalance reduction. *2021 29th European Signal Processing Conference (EUSIPCO)* (pp. 2174–2178).

Yang, Q., Liu, Y., Chen, T., et al. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology, 10*(2), 1–19. https://doi.org/10.1145/3298981

Yang, Q., Liu, Y., Cheng, Y., et al. (2019). *Federated Learning*. Switzerland: Springer Cham.

Yang, S., Ren, B., Zhou, X., et al. (2019c). Parallel distributed logistic regression for vertical federated learning without third-party coordinator (p. 6). arXiv preprint arXiv:1911.09824. https://doi.org/10.48550/arXiv.1911.09824

Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR), 54*(6), 1–36. https://doi.org/10.1145/3460427

Zhao, Y., Li, M., Lai, L., et al. (2018). Federated learning with non-iid data p. 12. arXiv preprint arXiv:1806.00582. https://doi.org/10.48550/arXiv.1806.00582

Zheng, W., Popa, R. A., Gonzalez, J. E., et al. (2019). Helen: Maliciously secure coopetitive learning for linear models. *2019 IEEE Secur. Priv.* (pp. 724–738). https://doi.org/10.1109/SP.2019.00045

Zhong, D., Sun, H., Xu, J., et al. (2022). Understanding disparate effects of membership inference attacks and their countermeasures. *2022 ACM on Asia Conference on Computer and Communications Security* (pp. 959–974). https://doi.org/10.1145/3488932.3501279

Zhu, H., Wang, R., Jin, Y., et al. (2021). PIVODL: Privacy-preserving vertical federated learning over distributed labels. *IEEE Transactions on Artificial Intelligence,* 1–13. https://doi.org/10.1109/TAI.2021.3139055