2024

Federated Learning: Overview, Strategies, Applications, Tools and Future Directions

Betul Yurdem
Izmir Bakircay University

Murat Kuzlu
Old Dominion University

Mehmet Kemal Gullu
Izmir Bakircay University

Maliha Tabassum
Old Dominion University, mtaba006@odu.edu

Original Publication Citation

Yurdem, B., Kuzlu, M., Gullu, M. K., Catak, F. O., & Tabassum, M. (2024). Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon*, *10*(19), 1-24, Article e38137. https://doi.org/10.1016/j.heliyon.2024.e38137

Heliyon 10 (2024) e38137

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Review article

Federated learning: Overview, strategies, applications, tools and future directions

Betul Yurdem [a], Murat Kuzlu [b], Mehmet Kemal Gullu [a], Ferhat Ozgur Catak [c,*], Maliha Tabassum [b]

[a] Department of Electrical and Electronics Engineering, Izmir Bakircay University, Izmir, Turkey
[b] Batten College of Engineering and Technology, Old Dominion University, Norfolk, VA, USA
[c] Department of Electrical Engineering and Computer Science, University of Stavanger, Rogaland, Norway

ARTICLE INFO

Keywords:
Data privacy
Federated learning
Distributed machine learning

ABSTRACT

Federated learning (FL) is a distributed machine learning process, which allows multiple nodes to work together to train a shared model without exchanging raw data. It offers several key advantages, such as data privacy, security, efficiency, and scalability, by keeping data local and only exchanging model updates through the communication network. This review paper provides a comprehensive overview of federated learning, including its principles, strategies, applications, and tools along with opportunities, challenges, and future research directions. The findings of this paper emphasize that federated learning strategies can significantly help overcome privacy and confidentiality concerns, particularly for high-risk applications.

1. Introduction

In recent years, Artificial Intelligence (AI) has become more popular with advanced computing technologies and has been applied in almost all sectors. For instance, in the gaming industry, when AlphaGo [1] defeated the top human Go players, it demonstrated the huge potential of AI. More complex cutting-edge AI technology was seen in many sectors such as healthcare, finance, and many applications. With the success of AlphaGo, a new technology called Federated Learning (FL) came into AI, providing the solution to the problems faced earlier. Federated learning is a distributed machine learning (ML) technique that uses multiple servers to share model updates without exchanging raw data. This helps to overcome the sensitivity to data and privacy-protected technology [2].
FL has been proposed as an alternative approach under the coordination of a central server, and a global model is trained in the form of a federation of devices that participate [3–5]. However, in terms of data science, there are some drawbacks.
First, AI and ML models require many datasets to train them successfully. In recent years, datasets have grown larger, and models have become more complex. Training ML models require optimization and distribution of model parameters on multiple machines [4]. Especially with medical data, the main problem arises when a model is trained with a huge dataset. It becomes more critical when working with medical data as the information becomes highly sensitive. There is great concern among the public, media, and government about recent news of leaks. For example, Facebook's recent breach of privacy caused several protests about data privacy [6]. In response, the European Union, on 25 May 2018, enforced the General Data Protection Regulation (GDPR) [7] to strengthen

* Corresponding author.
E-mail addresses: betul.yurdem@bakircay.edu.tr (B. Yurdem), mkuzlu@odu.edu (M. Kuzlu), kemal.gullu@bakircay.edu.tr (M.K. Gullu), f.ozgur.catak@uis.no (F.O. Catak), mtaba006@odu.edu (M. Tabassum).

https://doi.org/10.1016/j.heliyon.2024.e38137
Received 20 March 2024; Received in revised form 16 September 2024; Accepted 18 September 2024

Available online 20 September 2024

Table 1
Total number of selected articles relevant to the topic.

Keywords	Total number
"Federated learning"	3768
"Application" + "Federated learning"	3531
"Framework" + "Federated learning"	3304
"Privacy" + "Federated learning"	2956
"Strategies" + "Federated learning"	2919
"Security" + "Federated learning"	2543
"Tool" + "Federated learning"	2343
"Overview" + "Federated learning"	1918
"Aggregation" + "Federated learning"	1865
"Attack" + "Federated learning"	1832
"Heterogeneity" + "Federated learning"	1203
"Blockchain" + "Federated learning"	1254

laws to protect data security and privacy. European Parliament also regulates that users have complete power over their data [8]. People are also paying more attention to their data and where these data are used these days [9,10]. Thus, prior consent is required to use such data.

Second, if the number of datasets is limited, there is a restriction to the development of the training model. That is, the relation between the number of datasets and the model training is proportional. In a dataset, there is the possibility of the presence of all kinds of scenarios. Such as datasets that help detect a chronic wound to make it possible to diagnose the disease before any severe measures. However, obtaining this kind of data is difficult due to its sensitivity and high regulation [11].

Third, when dealing with healthcare data, a person's data becomes highly sensitive. This type of data contains private information. Moreover, while a large dataset is needed to train an AI model, the limited number of datasets hinders the development of ML models [12]. Due to its sensitivity and different types of regulations, obtaining this kind of data is difficult [13].

For example, the initial version of AlphaGo relied on 160,000 sets of human chess data. This had the ability to defeat beginner-level players. On the contrary, AlphaZero used human-generated chess data that has been shown to be capable of defeating professional players [14]. FL-based training approaches for a neural network that includes image classification [15] or natural language process [16] are promising along with new methods developed daily, and several new mobile applications are being developed. For example, the prediction of Google keyboard words was improved using FL by Hard et al. [17].

1.1. Research methodology

The research process for this review paper focused on a systematic way to provide an extensive survey of the overview, strategies, applications, tools, and future directions of Federated Learning. For this, publications and preprints consisting of review articles, research articles, book chapters, and conference papers on academic databases consisting of Web of Science, Elsevier, IEEE Xplore, Scopus, Springer, and Google Scholar were examined.

During the research studies, the main keyword used in searches was "federated learning". Additionally, keywords involved in data privacy and security, data heterogeneity, model aggregation, framework, and application topics were used in the review. Since it is a timely and wide topic, there may be unobserved publications on this subject with the possibility that similar terms may be overlooked. In this research, since 2016 was the first year of proposing the FL concept [4], the publications after this date were examined. Table 1 shows the number of selected articles found with the given main and additional keywords, the selection conditions of which were previously specified.

As can be seen from the table, although it is a new field of study, there are many studies in almost every sub-branch of this subject, and in this paper, these branches are explained. Overall, this paper is organized as follows. Section 2 provides an overview of Federated Learning, divided into two subsections covering the characteristics and categorization of FL. Section 3 provides various FL strategies. The security concerns associated with FL are addressed in Section 4. Section 5 presents various applications of FL. Section 6 examines some of the FL frameworks and tools. In the last section, opportunities, challenges, and future directions in FL are explained.

2. Overview of federated learning

Federated learning aims to build global AI models for multiple parties with diverse interests. Instead of collecting and combining data from different locations and gathering them in a central location, FL processes data from the place where it ensures the security of sensitive data and models.

Federated learning has undergone significant development since its introduction. Initially, researchers proposed distributed stochastic gradient descent (SGD) as a way to train deep neural networks on multiple devices. However, this approach requires that each device transmit its local model to a central server after every iteration, raising privacy and communication issues. In response, Google researchers proposed the FL approach in 2016 [18], which allowed model training to occur on the devices themselves without requiring the transmission of raw data to a central server. This approach has been extended along with various federated learning architectures and algorithms. An extension is the use of differential privacy introduced by Google researchers [19], which

is a mathematical framework to quantify and guarantee privacy in the AI/ML concept. Random noise is added to the model updates generated by participating devices in differential privacy. This approach has also been extended by using heterogeneous data [6], where devices may have different data distributions. In addition, federated learning has been extended to other domains, such as federated optimization [20] and federated reinforcement learning (FRL) [21]. In federated optimization, multiple entities cooperate to optimize a shared objective function, while agents learn to act in a decentralized environment without central coordination in FRL. These extensions will significantly contribute to the potential applications and open new research directions in federated learning. The future of federated learning promises additional developments, including the exploration of federated learning with non-IID data, the integration of federated learning with blockchain technology, and the development of more robust and scalable federated learning algorithms.

In recent years, federated learning has become more mature with its characteristics and categorizations depending on how the data is distributed, stored, and interrelated. The characteristics and categorization of FL are discussed in the following subsections.

2.1. Characteristics of federated learning

FL is a field of distributed machine learning, which puts a great deal of effort into data privacy. The latest studies also pay attention to distributed systems that preserve privacy. It connects multiple nodes located in different locations but is connected via a communication network. The network is under the control of the central server, and each node undertakes different parts of the same task to complete it. The main characteristics of FL are discussed below.

- **Universal cross-organization scenario:** FL was originally proposed by Google as a distributed ML technology. This allows participants to build a global model while keeping their underlying data local. The initial concept of FL is expanded to encompass all decentralized ML techniques that preserve privacy [6].
- **Massively non-identically independent distribution:** In FL, data is massively widespread among a huge number of edge notes or devices. The data available at each node might be less than the total number of nodes.
- **Decentralization:** It is not completely used in a technical sense. However, there is no definitive center. Each client is not determined to be the center, but they influence the central model. Parameter servers are used as a central server that is dominant enough to distribute the data. It works as a resource to obtain efficient collaboration [22].
- **Equality of status for each node:** In FL, all parties receive the same facility. This means that who has the largest number of data has the largest mass.

2.2. Categorization of federated learning

Federated Learning can be categorized in different ways based on data distribution. In this paper, the two most popular categorizations are explained, i.e., vertical federated learning and horizontal federated learning.

- **Vertical Federated Learning:** It is suitable in cases where data is partitioned in the verticle direction in accordance with feature dimension. Vertical federated learning (VFL) is where data features are split among multiple parties. It is the concept that collaborates to train a model on a dataset. Here, all parties have homogeneous data. That means that they overlap partially on sample ID but are different in feature space.
  For example, a medical organization intends to work to identify and predict diseases related to diabetes. According to research, people with high blood pressure and obesity are likely to develop Type-2 diabetes [23]. Therefore, analyzing the weight and age of patients can give rough dimensions along with their medical history. However, a young patient who consumes more sugar and lacks physical activity but is not obese or has high blood pressure is more prone to suffer from diabetes. This diagnosis is not predicted and is personalized due to lack of information. With the development of FL, some companies that work with smartphone application datasets, such as step counters or dietary structures, can cooperate with each other. This can be done without sharing the raw data transmissions as the figure shows. Normally, taking out similar entities that possess different characteristics to get joint training is a path taken by scholars.
  Another example is that patient data can be present in different healthcare organizations. As the identity of the patient is secret, those two different data cannot be merged to train a model. In order to address this issue, it is necessary to train a machine learning model through a collaborative effort that will be conducted within the appropriate premises. Unlike horizontal FL, vertical FL poses additional challenges in terms of entity resolution [24]. Unlike the straightforward approach of aggregating all datasets on a common server, this method is not effective in vertical FL. An example of how vertical FL works is given in Fig. 1. It can be inferred that there is still more potential for enhancing vertical FL to be utilized in a more intricate ML approach.
- **Horizontal Federated Learning:** In the case of horizontal FL, similarity exists in the data features that are spread across various nodes. Meanwhile, those data are different in sample space. The existing FL algorithms are primarily aimed at various applications of IoT devices or smart devices [2]. In these scenarios, FL is classified as horizontal-federated learning. Here, data in the sample space may differ drastically but have similar features. Furthermore, a hierarchical heterogeneous horizontal FL frame was introduced to meet the criteria for limited labeled entities [25]. Adapting a heterogeneous domain can solve the issue of label shortage adopted multiple times using each participant as the target domain. In real applications, such as healthcare, data collection is inseparable from a huge number of work. It is almost impossible for each hospital to build a data pool to share data.

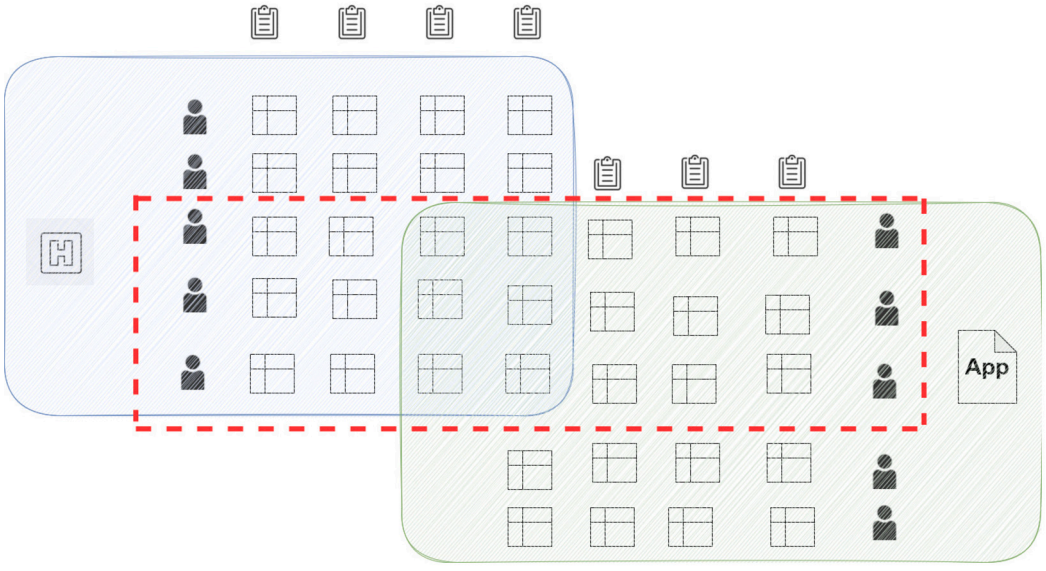

Fig. 1. An example of vertical FL-based application [2].

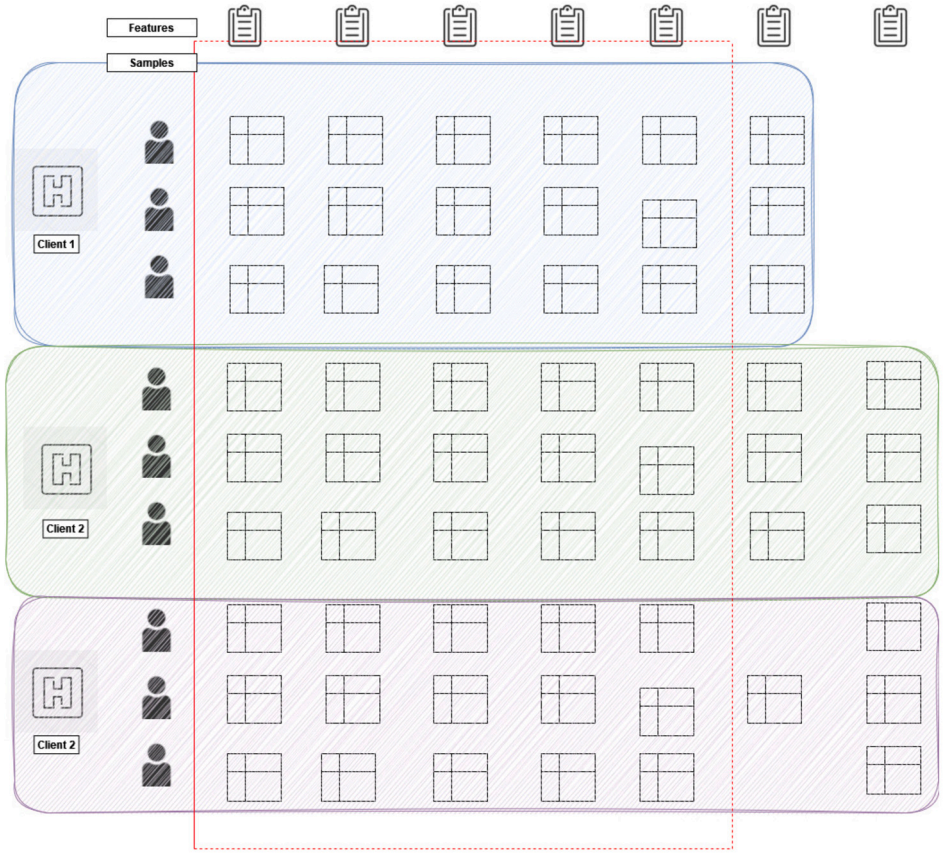

Fig. 2. An example of horizontal FL-based application [2].

Therefore, in horizontal FL, multiple parties train a model with similar datasets from different sources [26]. An example of this is illustrated in Fig. 2.

Table 2
Comparison of categorization of federated learning [27].

Criterions	VFL	HFL
Features	Different	Common
Samples	Common	Different
Training	Locally on client devices	
Shared with Server	Local model updates	
Get from server	Global model updates	
Data sharing with global server	No	
Privacy and security	Data is not shared with the global server	
Scalability	Low	High
Computing power	Bounded with the device capacities of clients	
Communication resources	Consumes less than HFL	Consumes more than VFL
Pros	Huge and detailed datasets	Lower communication cost than centralized ML
Cons	Works with a small number of clients	Data heterogeneity

In FL models, one model can have multiple characteristics discussed in the previous section. The different types vary in their needs. The categorization is not mutually exclusive. In Table 2, the comparison of these two models is shown according to their feature and sample diversity, training point, data sharing, privacy and security, scalability, computation features, and communication resources.

3. Federated learning strategies

In FL, multiple parties can collaborate without sharing raw data. Each party shares model updates with a central server. The data go through aggregation by the central server and are sent to local sources. In this way, the model is trained while benefiting all parties by providing privacy.
Distributing the data across different devices is more convenient than gathering all the data in one location or device. To achieve this, a variety of federated learning strategies have been developed [28]. Selected ones from the commonly used ones are explained in the next subsections.

3.1. Algorithms

3.1.1. Federated averaging

Federated Averaging, also known as FedAvg, is the most popular FL strategy. In FedAvg, a local model gets trained by each client with their own data. Allocating parameters to a large number of clients, recycling, training, and updating them are done by the central server while not sharing the client's information. The whole communication is done by the central client. The model parameters across all parties are averaged by aggregation. After that, each client receives the global model, and this process is then continued until the model reaches convergence. This process is explained in Appendix A - Algorithm 1. This model can provide the privacy promised by FL models. The key factor behind this is to minimize communication overhead. In the ML model, this is a promising approach. There are some significant advantages of FedAvg over other centralized ML models:

- It provides data privacy. Data is stored in the local model while the model updates are shared. In this case, a central server is not used to share data.
- Model updates are only shared between participants. It results in a significant amount of communication overhead, which makes it environmentally friendly.
- It is scalable to large-scale FL scenarios with a large number of participants. The participants have the freedom to join or leave the network at any time.

3.1.2. FedAdam

FedAdam is the reformulation of the Adam optimizer for FL. In this approach, training data is distributed among devices. The moving average of the model update follows the momentum decay rule. This technique adaptively updates the model's weights by utilizing moving averages and adjusting the learning rate for each weight. The exponential moving average of the first moment of the gradient is computed. The weights are updated with a learning rate and adapted based on the estimated variance and mean gradient. The primary distinction between FedAdam and the standard Adam algorithm is the use of a federated average to calculate the gradient across all devices. The model can also run well with the new data while preserving the old data.

3.1.3. FedYogi

In the case of FL, this strategy lets users train a model where the identity of the clients is not shared with the central server. This process avoids the collection of sensitive data that can breach privacy. Mostly, this is achieved by learning a single global model. This model is for all users, even if they are using different data distributions. For instance, users of a device prefer different suggestions while training a model. The personalization of this heterogeneity is the motivation behind each user's global model. In the case of privacy, learning the full model is prohibited for privacy reasons. That is why Federated Optimization is adapted with Yogi.

Table 3
Comparison of FL strategies.

FL Strategy	Utility
FedAvg	Simple and common
FedAdam	Effective optimization
FedYogi	More stable learning
FedAdagrad	Automatic learning rates
FedProx	Better for non-IID data
Scaffold	More stable convergence

3.1.4. FedAdagrad

FedAdagrad is the short form of Federated Adaptive Gradients, which is an extension of the gradient descent optimization algorithm. The gradients seen for the variable, i.e., partial derivatives, are optimized and observed throughout the search. Regarding all the model parameters, the algorithm is based on adapting the learning rate and first-order information with some properties of second-order methods and annealing. With the growth of the dynamic rate, it is inversely proportional to the magnitude of the gradient. The relationship between large gradients and smaller learning rates is proportional. The scale of gradience varies in each layer by several orders. Moreover, the denominator of the scaling coefficient has a similar effect as annealing, which gradually decreases the learning rate over time. The Appendix A - Algorithm 2 describes all FedAdam, FedYogi, and FedAdagrad strategies.

3.1.5. FedProx

This strategy is used to lessen statistical heterogeneity. It introduces terms that regularize each local training loss and is built on FedAvg [29]. The local model and its deviation are also controlled by this model. Re-parameterization does some minor modifications but it is a better version of FedAvg. It is important for both theoretical and practical fields. In terms of practical field, robust convergence is demonstrated compared to FedAvg [30]. Meanwhile, in theory, in device-level systems, convergence is provided that allows each device. Between FedProx and FedAvg, the first one is more stable and provides accurate convergence. The FedProx algorithm is represented in Appendix A - Algorithm 3.

3.1.6. Scaffold

Stochastic Controlled Averaging for Federated Learning is also known as Scaffold. The training process is coordinated across different nodes across the network as expected. The Scaffold algorithm is given in Appendix A - Algorithm 4. This approach reduces the need for multiple rounds of communication and resolves the issue of heterogeneity [31]. Furthermore, a server-side learning rate is employed to regulate variates and reduce control variates. Ultimately, it allows for collaboration between multiple devices with different data sources.

3.2. Comparison of the FL strategies

In this section, the applications given in the literature are examined and a comparative evaluation of the FL strategies used is accomplished.

In a study by Gao et al. [32], in addition to the FL models trained with the previously mentioned known strategies such as FedAvg, FedProx, and Scaffold, the model trained with the FedDC strategy they proposed was also compared. Thus, the effects of these algorithms for a more convergent model due to heterogeneous datasets were investigated. As a result of studies conducted with different datasets, the FedAvg method had the most round numbers to converge to the target accuracy. Although a higher or approximate number of rounds were obtained with FedProx, trainings were completed faster than FedAvg in some cases. With Scaffold, fewer and faster training processes were completed compared to both methods, and the same percentage of accuracy was achieved with all of them. The FedDC strategy, a combination of FedProx and Scaffold, showed much better results than others for both IID and non-IID datasets. In the final comparison results, they determined that FedDC is robust and effective in full and partial client participation [32].

Nguyen et al. conducted a study with the CIFAR-10 dataset using strategies such as FedAvg, FedProx, Scaffold, FedNova, FedAdam, FedAdagrad, and FedYogi and their combinations [33]. In the obtained test accuracy graph, it was observed that the best results were achieved with the ProxYogi strategy, which is the combination of FedProx and FedYogi [33]. This study was also conducted with a non-IID dataset.

In another study, training was carried out with medical data [34]. FedAvg, FedAdam, FedYogi, FedAdagrad, FedProx, and FedAvgM strategies were used to predict in-hospital mortality and acute kidney injury. In the results, it was given that the training with FedProx had the lowest accuracy for the prediction of acute kidney injury. Nearly all of them obtained highly accurate training results for predicting hospital mortality, but FedAdam had a slightly lower result. They also highlighted that the basic FL algorithms FedAvg and FedAvgM are the best for machine learning tasks, and FedAvg performs well with the IID datasets [34].

As can be understood from the literature, there are cases where the use of each FL strategy is advantageous. An overview of these advantages for commonly used strategies is given in Table 3.

4. Security concerns for federated learning

FL is a distributed machine learning approach in the AI/ML concept that involves multiple parties, which introduces various security concerns regarding the privacy, confidentiality and integrity of the data used in the collaborative learning process. In this section, the main security concerns associated with federated learning are discussed along with potential solutions to mitigate these risks.

4.1. Data privacy

To mitigate the risk of data privacy, several privacy-preserving techniques have been proposed. One of these methods is differential privacy [35], which adds random noise to the updates sent by each device to obfuscate individual contributions. For example, let X be the input data from a device and $f(X)$ be the corresponding model update. Then, the noise added to $f(X)$ is drawn from a probability distribution $N(0,\sigma^2)$, where σ is the privacy parameter that controls the level of noise added. Mathematically, the noisy update can be expressed as $f'(X) = f(X) + \eta$, where $\eta \sim N(0,\sigma^2)$. The privacy guarantee of differential privacy can be quantified using the concept of ϵ-differential privacy, which limits the amount of information that can be inferred about any individual device's data from the overall output of the federated learning algorithm. Specifically, a federated learning algorithm is said to be ϵ-differentially private if, for any two input datasets X and X' that differ by a single record, and for any output S of the algorithm, the following condition holds:

$$\frac{Pr[S(X)]}{Pr[S(X')]} \leq e^{\epsilon} \tag{1}$$

where $Pr[S(X)]$ and $Pr[S(X')]$ denote the probability of obtaining the output S on datasets X and X', respectively.
Another method is secure multi-party computation (SMC) [36], which enables collaborative computation on encrypted data without revealing the raw inputs. For example, it is assumed that $X_1, X_2, ..., X_n$ present the private inputs of n devices participating in the federated learning process. In SMC, first, each device encrypts its input utilizing a public key, i.e., generates the encrypted data, and then sends the data to a central server. The central server is responsible for computation on the encrypted data as well as returning the encrypted result to the devices to decrypt the data using the private key.
The security of SMC is based on cryptographic techniques, such as homomorphic encryption (HE), which allows computations on encrypted data. It is assumed that $Enc(X)$ is the encrypted version of input X, $Dec(X)$ is the decryption function, and let f be a function operating on encrypted data. HE ensures the function can be performed on encrypted data without revealing the underlying inputs, i.e., $Dec(f(Enc(X_1), Enc(X_2), ..., Enc(X_n))) \approx Enc(f(X_1, X_2, ..., X_n))$.

4.2. Model poisoning attacks

FL includes model updates from multiple devices to form a global model. However, FL can be exploited by malicious participants, i.e., injecting poisoned updates to corrupt the global model or model poisoning attacks. This can lead to incorrect predictions or leakage of sensitive information. Several defense mechanisms have been proposed for model poisoning attacks.
One mechanism is anomaly detection methods based on statistical methods (clustering or outlier detection) or AI/ML (such as neural networks or decision trees). The anomaly detection method is used to identify and remove malicious updates. This can be explained mathematically as follows:

- Given $U_t = u_1, u_2, ..., u_n$: the set of updates received at time t, and f_t: the global model at time t.
- Identify updates $u_i \in U_t$, which do not align with the current global model f_t, i.e., $||f_t - u_i|| > \theta$, where θ is a predefined threshold.
- Define the distance metric, i.e., $||f_t - u_i||$, by using the Euclidean distance or the cosine similarity.
- Identify and remove malicious updates if needed.

An alternative mechanism is robust aggregation algorithms, which can be defined as an optimization problem solved using various techniques, such as gradient descent or stochastic optimization. The robust aggregation algorithms aim to reduce the effect of malicious updates on the global model. This can be simply expressed mathematically as follows:

- Given f_t: the global model at time t, and $U_t^m = u_1^m, u_2^m, ..., u_{n_m}^m$: the set of malicious device updates at time t, $L(f_t, U_t)$: the loss function of the global model in the set of updates U_t, and k: a constraint that limits the maximum number of compromised updates.
- Define the optimization problem as follows:

$$\min_{f_t} \mathbb{E}_{U_t \setminus U_t^m}[L(f_t, U_t)] \quad \text{subject to} \quad |U_t^m| \leq k \tag{2}$$

- Apply the defined optimization algorithm to the global model.

4.3. Model inversion and membership inference

FL can be vulnerable to model inversion and membership inference attacks. Model inversion attempts to uncover sensitive information from the trained model, while membership inference attempts to determine if a particular data point is used in the training. These attacks can be more critical particularly high-risk applications, such as healthcare or finance.
Fortunately, a variety of countermeasures have been developed to enhance the security of federated learning against these attacks, which include model regularization, adversarial training, and gradient obfuscation. Each countermeasure method is briefly explained below.

- **Model regularization** is an additional regularization term to the loss function during training, which penalizes the model for learning sensitive information. This can be given mathematically as:

$$\mathcal{L}_{\text{regularized}} = \mathcal{L} + \lambda \cdot \mathrm{R}(f) \tag{3}$$

  where $\mathcal{L}$ is the original loss function, λ is a regularization parameter, and $\mathrm{R}(f)$ is a regularization term to encourage the model to forget sensitive information.
- **Adversarial training** is the training process against adversarial examples generated specifically to extract sensitive information. Mathematically, this can be given as a min-max optimization problem:

$$\min_{f} \max_{x' \in \mathcal{X}'} \mathcal{L}(f, x, x') \tag{4}$$

  where x is the original input, x' is the adversarial example, and $\mathcal{L}(f, x, x')$ is the loss function that captures the vulnerability of the model to model inversion attacks.
- **Gradient obfuscation** protects sensitive information by perturbing gradients during the training process, making it more difficult for an adversary to perform model inversion attacks.

Additionally, a variety of countermeasures such as privacy-preserving data synthesis and differentially private aggregation can also be utilized for membership inference attacks. Each countermeasure method is briefly explained below.

- **Privacy-preserving data synthesis** refers to the process of generating synthetic data that maintains the statistical characteristics of the original data while ensuring the confidentiality of individual data points. It uses techniques such as generative adversarial networks (GANs) or differential privacy mechanisms. Mathematically, this can be given as:

$$\hat{X} = \text{Synthesize}(X) \tag{5}$$

  where X is the original dataset and $\hat{X}$ is the synthesized dataset. The synthesized dataset $\hat{X}$ can be used for training the federated model while protecting the privacy of individual data points.
- **Differentially private aggregation** is to protect membership privacy by adding random noise to the aggregation process. Mathematically, this can be given as:

$$\text{Agg}(U_1, U_2, ..., U_n) = \frac{1}{n} \sum_{i=1}^{n} U_i + \text{noise} \tag{6}$$

  where $U_1, U_2, ..., U_n$ are the updates from the participating devices, and noise is a random noise term that ensures differential privacy guarantees.

These techniques provide additional security measures to mitigate the risks associated with model inversion and membership inference attacks in federated learning.

5. Federated learning-based applications

FL can be used in different fields, and any field that deals with a person's private information can benefit from FL. Below are some of the applications that can use and benefit from FL models.

5.1. Healthcare

FL shares its data without revealing the identity of the users. Each medical institute might have some data about their patient. Still, the amount of data lacks sufficiency to train a model [37]. This makes the use of FL in this field very attractive. Some of the applications in healthcare are discussed as follows:

- **Electronic Health Records (EMR):** It has a lot of clinical components, the authors [38] suggested an attempt to use tensor factorization models. It would do phenotype analysis and obtain concealed medical data of patients. It is the first attempt at FL learning. Their proposed graph-based attention model was used in disease diagnosis prediction with the records of adult patients

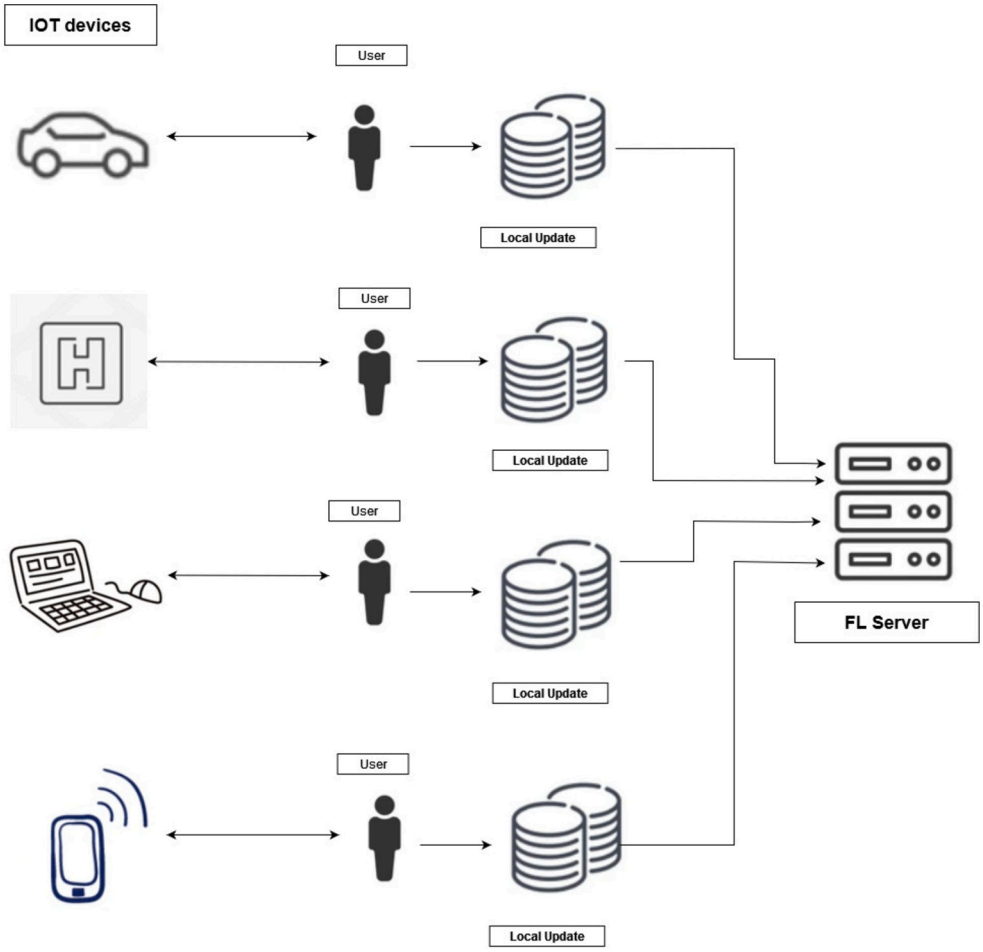

Fig. 3. Applications of federated learning with different IoT devices.

and intensive care unit patients, and heart failure prediction studies. The model they proposed as a result of studies conducted with the data of patients between the ages of 20-40 in intensive care disease prediction ensured them to achieve 10% higher accuracy than training with RNN [38]. The authors [39] explored EMR differently while using Federated Settings. It was used to predict the mortality rate of cardiac patients by using EMRs from different hospitals.

- **Biomedical Image Analysis:** FL has been used in this sector. For example, Silva et al. [40] put forward FPCA (Federated principal components analysis). It extracted features from MRI that come from different medical centers.
- **EEG Classification:** A Hierarchical Heterogeneous Horizontal FL (HHHFL) was proposed by Gao et al. [25]. They overcame the challenge of limited labeled samples along with the provision of privacy [2].
- **Clinical Prediction:** To predict disease, beforehand FL can be used. As described by the authors Pfohl et al. [39], FL establishes efficacy over centralized and local data. It also performs FL differently.

Overall, FL can be used to provide effective ways in healthcare that will help to make life easier.

5.2. Internet of Things (IoT)

Federated learning has several potential applications in the IoT industry. Fig. 3 shows various applications of FL and how different IoT devices take part in training an FL model. Here are some of the potential applications of federated learning in IoT:

- **Smart Transportation:** Intelligent Transportation System (ITS) has been implemented by using AI/ML for the past few years. It is done using centralized vehicular data learned from the data center [41]. This requires sharing data in an untrusted environment, which is a safety issue for FL and is used to provide privacy and safety [42].
- **Smart City:** The use of AI has been increasing over the years. The creation of a smart city that can enhance the quality of life for citizens living in urban areas is needed. By supplying seamless delivery of food, water, and other necessary items to their users [43]. To create a smart city, many sensors are needed. Also, AI and ML have the ability to take care of huge numbers of

big data that are usually generated from sensors [44]. FL offers a more attractive and effective way by enabling decentralized methods. For example, an environmental monitoring framework based on fog computing was proposed in study [45]. The studied multi-source heterogeneous dataset was collected from the IoT sensors in Beijing.

- **Smart Grid:** Smart grid is an important factor in terms of increasing industrial systems as it provides the necessary power to the energy sources of the household and industry [46]. Smart grid operations can be helped by FL with intelligent solutions. For example, FL can be used to create a federated predictive power scheme that can run a recent neural network locally. That can estimate the amount of power demands by inspecting the history of a customer's usage. This can help construct a global prediction model [47]. In the research [47], a power consumption prediction model for GPU-based edge data centers was constructed. Their model is based on ANN and achieves a normalized root mean square deviation error of less than 7.4% compared to actual measurements.

The FL model can help in the IoT sector in different ways. The use of different technological products is increasing day by day, and this requires privacy, too. FL can provide this by leveraging IoT products.

5.3. Internet of Underwater Things (IoUT)

IoUT stands for Internet of Underwater Things and has garnered rapid momentum recently [48]. The use of FL in IoT networks has been expanding rapidly in recent years. It spans applications on defense, environmental monitoring, exploration of the sea, etc. Like any other FL system, IoUT also uses a traditional ML model. Below, the various applications of IoUT are discussed.

- **Environmental Monitoring:** The environment consists of different elements like water, air, soil, etc. In terms of them, water is an essential one, and it is used in everyday life. The quality of the water always needs to be monitored because contaminated water can cause some serious problems. FL can be used to monitor the quality of water at all times. So that, no man-made pollution happens, like contaminating the water due to oil spills.
- **Underwater Exploration:** This helps to discover lost treasures such as underwater ruins of civilization and other natural resources. The underwater environment is used to experiment with some new technology. How these technologies are affected is also investigated using several sensors. H. Zhao et al. [49] proposed to use a "federated meta-learning enhanced acoustic radio cooperative framework", called as ARC/FML, for data collecting from distributed sources. This stated technique helps in sharing sensitive data regarding underwater exploration. DeepSink and RF Channel were used to conduct the experiment. The proposed model provided an accuracy of 97%.
- **Disaster Prevention and Mitigation:** The underwater is always susceptible to various disasters. They can be both natural and man-made. Disasters that are man-made are mostly oil spills, and poisonous chemical spills which mostly occur due to underwater experiments. On the other hand, natural disasters are tsunamis, earthquakes, and underwater volcano eruptions. The Deepwater Horizon incident in 2010 [50] is one of the worst man-made disasters to be recorded. It occurred due to the leakage of a significant amount of natural gas, which resulted in the gas rising and catching fire [51]. On the other hand, the most horrific natural disaster occurred in 2004, which was the earthquake and tsunami in the Indian Ocean [51]. FL-enabled IoUT solutions can significantly provide help in preventing these disasters by sending the data in real-time.
- **Defenced Military:** Defensive naval operations traditionally involve human-operated underwater vessels to carry out tasks such as detecting submarines, mine warfare, recovery operations, and surveillance. However, advancements in technology have led to the development of Underwater Wireless Sensors, or UWSN, which are a type of wireless sensor network that allows for underwater activities to be carried out without human intervention. UWSN are capable of detecting and classifying objects in the underwater environment. The US Navy places great emphasis on real-time data sharing between ships, submarines, drones, and intelligence analysts onshore [52]. FL is used for this purpose, too.

These are some of the applications of FL in terms of IoUT. There are many more ways that FL can be used.

5.4. Industrial engineering

FL has several application sectors in different industries, and some of them are mentioned below.

- **6G Industry:** In the recent growth in data traffic, ML has garnered a huge amount of attention. In the development of the sixth generation (6G), it has become vital [53].
- **Image Detection:** FL can also be applied to perform visual inspection task [54]. It can be used to help detect defective products in the manufacturing industry.
- **Image Representation:** In image field, Liu et al. [55] describes about how vision-and-language as a flashpoint. They also discuss how FL can diversify these tasks for better grounding.
- **Unmanned Aerial Vehicles:** The work of unmanned aerial vehicles by using FL is given by Mowla et al. [56]. It was discussed how these vehicles can detect malicious attacks by using FL.
- **Electrical Vehicles:** Electric vehicles are getting popularized now. An FL is designed by Saputra et al. [57] that can predict the vehicles' energy demand and methods of efficient charging stations. This can prevent energy congestion in transmissions.

- **Financial Field:** The use of credit cards and transactions in various fields is increasing day by day. Yang et al. [58] leveraged FL to detect credit card fraud efficiently. This is a significant contribution to the field of finance.
- **Text Mining:** An industrial grade federated framework was used by Wang et al. [59] on Latent Dirichlet Allocation. It is used to assess real data for spam filtering.

5.5. Integrating FL with technologies

FL is used in studies in different fields and with different data, as well as in integration with different technologies.

- **Blockchain:** Blockchain-supported FL applications are being implemented to solve problems, such as centralized processing, data manipulation, and lack of motivation [60]. With blockchain technology, improvement of the security and scalability of FL is also being realized [42]. Singh et al. [61] proposed a framework to use in the IoT healthcare field, taking advantage of the unanimity efficiency improvement of FL and the stability, originality, transparency, distributed, and decentralized features of blockchain. It has been stated that although the study was developed only as a theoretical model, it is planned to focus on delay and storage needs in the future [61]. BLADE-FL, a blockchain-based decentralized FL architecture, was proposed in [62]. The standard FL frameworks work with a single central server. Hence, a malicious client may cause the training to fail. In BLADE-FL, FL and blockchain are integrated into each client for training and mining tasks, respectively. Studies based on blockchain-based FL were overviewed in [63]. It was concluded from the literature that integrating blockchain with FL increases security in training, but is not sufficient to protect privacy.
- **Edge Computing:** Since FL is a collaborative learning and model optimization method, it can be an encouraging factor for edge computing technology that extends computing services closer to the dataset [64]. Poisoning attacks, an attack model using edge computing in FL, were designed by Zhang et al. [65]. Updating the global model parameters by repeating the samples of the clients, increasing the applicability of the attacks by reducing the attack assumptions, and poisoning attack strategies such as label flipping and backdoor were implemented. As a result, it was observed that these attacks effectively risked the global model [65]. Ye et al. [66] proposed the EdgeFed algorithm to optimize the high computational cost encountered on mobile devices during the implementation of FL based on edge computing with the FedAvg strategy. In this algorithm, the process of updating the local model is separated, and the outputs of mobile devices are collected at the edge server to reduce the frequency of global communication [66]. Another study for FL model aggregation of image classification was conducted in the field of vehicular edge computing [67]. The model was selected using the two-dimension contract theory as a distributed framework. As a result of studies involving the MNIST dataset, model aggregation showed better results in accuracy and efficiency than the original FedAvg strategy [67].

6. FL-based frameworks and tools

In FL research, we find numerous libraries, many of which are open-source, that act as both frameworks and tools. They assist in training models, ensuring security, facilitating communication, and quickly compiling results. Using these libraries helps reduce repetitive tasks in the code, leading to more streamlined models. With the growing variety of these tools, their capabilities also expand. Therefore, choosing the right tool or framework becomes essential for better model outcomes.

6.1. Existing frameworks and tools

Various software tools tailored for Federated Learning (FL) are emerging, each with distinct methods and capabilities. The literature showcases several notable frameworks and tools, including but not limited to: TensorFlow Federated (TFF) [68], PySyft [69], NVIDIA Federated Learning Application Runtime Environment (NVFlare) [70], Federated AI Technology Enabler (FATE) [71], Flower [72], IBM Federated Learning (IBM FL) [73], FedLab [74], FedML [75], Federated Learning Utilities and Tools for Experimentation (FLUTE) [76], Open Federated Learning (OpenFL) [77], SecureBoost [78], Interplanetary File System (IPFS) [79], Fed-BioMed [80], and FederatedScope [81]. Subsequent subsections focus on the unique features of these frequently mentioned frameworks.

- **TFF:** TensorFlow Federated (TFF) is an open-source framework introduced by Google, supplying decentralized data to ML [68]. It promotes global model training, leveraging local data without requiring server uploads. The results from individual devices are then centralized for consolidation. Being rooted in TensorFlow, TFF is able for mathematical tasks as well as learning [82]. It offers the FL API and Federated Core (FC) APIs, enhancing the flexibility in developing and applying FL algorithms. While it allows the use of Regression and Neural Network (NN) models in horizontal FL, vertical FL is not supported [83]. Unlike many, TFF is designed for single-host deployments, emphasizing its utility where data privacy and security are paramount. The TFF architecture in Fig. 4 includes a comprehensive framework designed to support federative learning scenarios. This architecture includes optimization strategies such as FedAvg and FedSGD, differential privacy with security features, various model types, and several protocols between an executor and client that enable communication at run-time.
- **PySyft:** Developed by the OpenMined community, PySyft is an open-source framework that prioritizes data privacy and security. It employs methods, such as anonymization, encryption, and differential privacy to partition data and models [69]. Since it performs data anonymization, this framework is used in encrypted privacy-preserving deep learning studies. The differential privacy

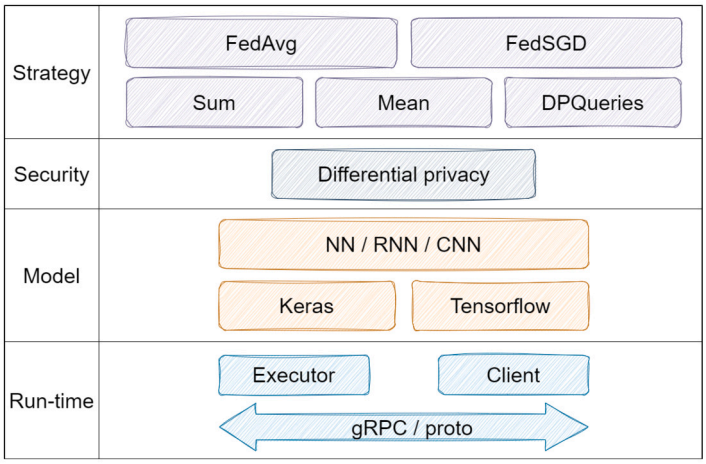

Fig. 4. Architecture of TFF [82].

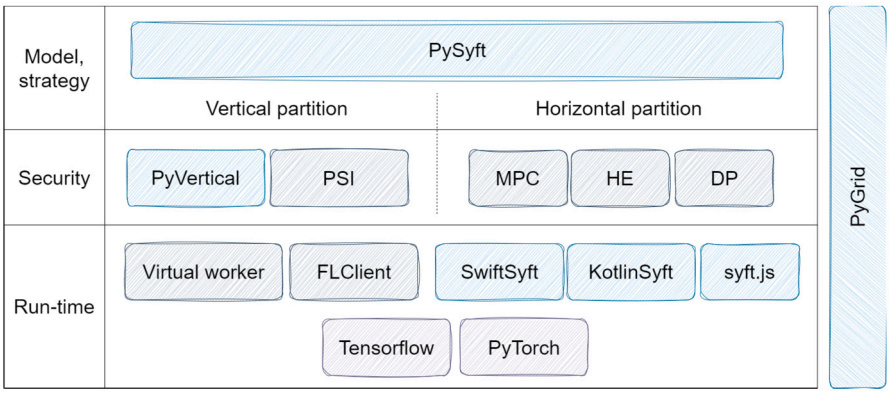

Fig. 5. Architecture of PySyft [82].

techniques included in PySyft help prevent attacks against FL models that may be endangered despite being encrypted, by confusing steps that increase the complexity of the training process [84]. During remote data operations, it facilitates learning without direct data copying, maintaining privacy boundaries. Moreover, PySyft is compatible with established ML frameworks like TensorFlow and PyTorch [69]. The image describing the PySyft architecture (Fig. 5) contains the main structure of this framework. In this structure, there are special modules to process horizontally and vertically split data and different techniques used to ensure computational, security and confidentiality. PySyft offers a powerful framework, supporting a wide range of features including secure multi-stakeholder computations, different data partitioning strategies, encryption techniques, and privacy-preserving federative learning.

- **NVFlare:** Designed primarily for collaborative computing, this framework ensures a secure and confidential environment for researchers [85] [70]. It operates using Python. During model training, clients independently train using their datasets and subsequently transmit results to a central server. The server model is updated with this information, and this iterative process persists until the main model convergence [70]. Fig. 6, which shows the NVFlare architecture in detail, includes a wide range of features under the headings of learning algorithms, Federation workflows, FLARE programming API, privacy protection with security management, tools, and FL simulator.
- **FATE:** FATE is an algorithm-centric framework focusing on machine privacy protection, with capabilities in both simulation and federated modes [82]. It operates under the assumption of a semi-trusted server, prioritizing combined parameters over private data [71]. Designed for AI and big data industrial applications, FATE finds usage in sectors like finance and healthcare [87] [88]. The framework is versatile, supporting NN, decision trees, regression, and transfer learning [71]. Fig. 7, which shows the FATE framework, reveals in detail a broad structure covering major components such as cloud-based services, workflow management, model distribution, federated ML components, security protocols, computational resources, and storage.
- **Flower:** The Flower FL framework is friendly, easy to use, adaptable to FL models, can be used in similar studies, and can be used to run these studies on lots of heterogeneous devices [72]. Flower has built-in algorithms such as FedAvg, FedProx, QFedAvg, and FedOptim, addressing challenges such as client connectivity issues and adapting to varied network conditions. These core strategies also serve as foundations for deriving algorithms for additional functionalities [72]. The flow in Fig. 8 demonstrates in

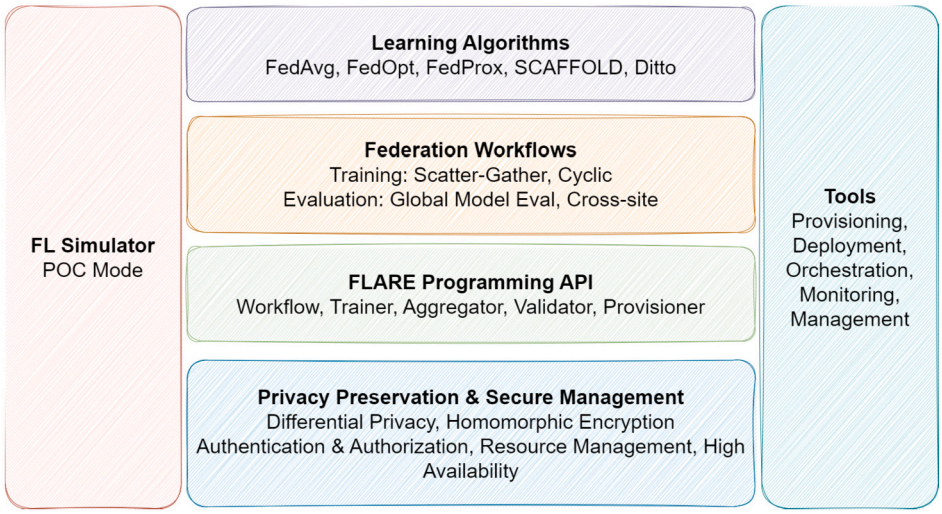

Fig. 6. Architecture of NVFlare [86].

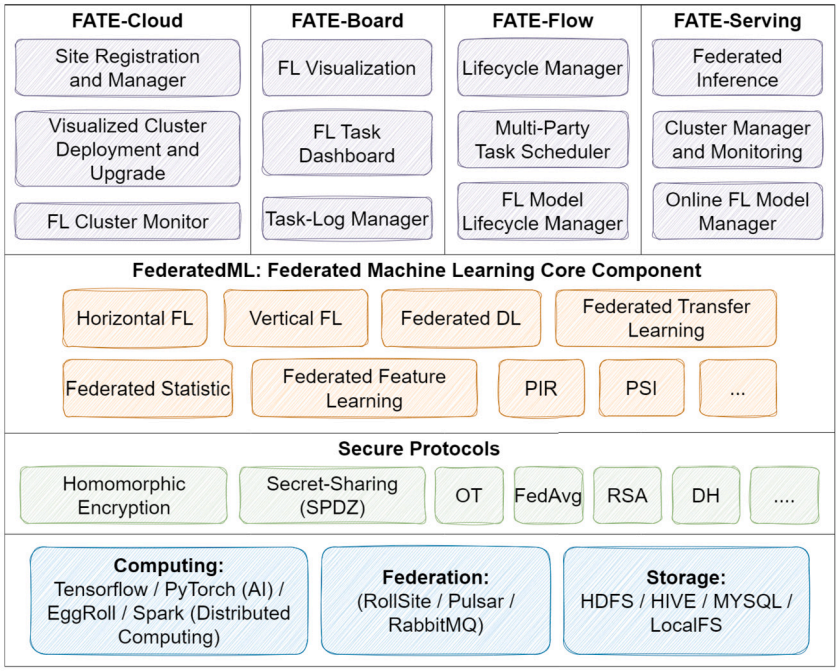

Fig. 7. Architecture of FATE [89].

detail the interaction between the core components of the Flower framework: Strategy, Client Manager and Remote Procedure Call (RPC) Server and RPC Clients, revealing a comprehensive design that shows the workings of Flower's internal structure.

- **IBM FL:** IBM FL, an independent ML library, supports Python and is adaptable for use with decision trees, support vector machines (SVMs), NNs, and RL [73] [82]. Since this FL was developed by IBM company, it is a proprietary framework, distinct from open-source options [82]. FL algorithms are executed by easily configuring all the basic substructures in their infrastructure, and implementation of grouping and aggregation of these algorithms are provided. Its capacity to collate data from various sources enhances its viability for use in institutional areas [73]. Notably, it incorporates FL strategies such as FedAvg and iterative average, predominantly for NN and RL applications [82]. Fig. 9 demonstrates the IBM FL architecture in detail. Data sources of different countries are connected to each country's own Remote Training System (RTS), and it is shown how the model parameters obtained from here are integrated and used within this framework. This flow clearly describes how data sources are integrated and how model updates are coordinated by ensuring secure cross-country collaboration.
- **FedLab:** It is an open-source framework that allows model optimization, data partition, and communication between algorithms and their efficiency can be examined with FL simulations [74]. Because of the flexible structure of the FedLab framework, users

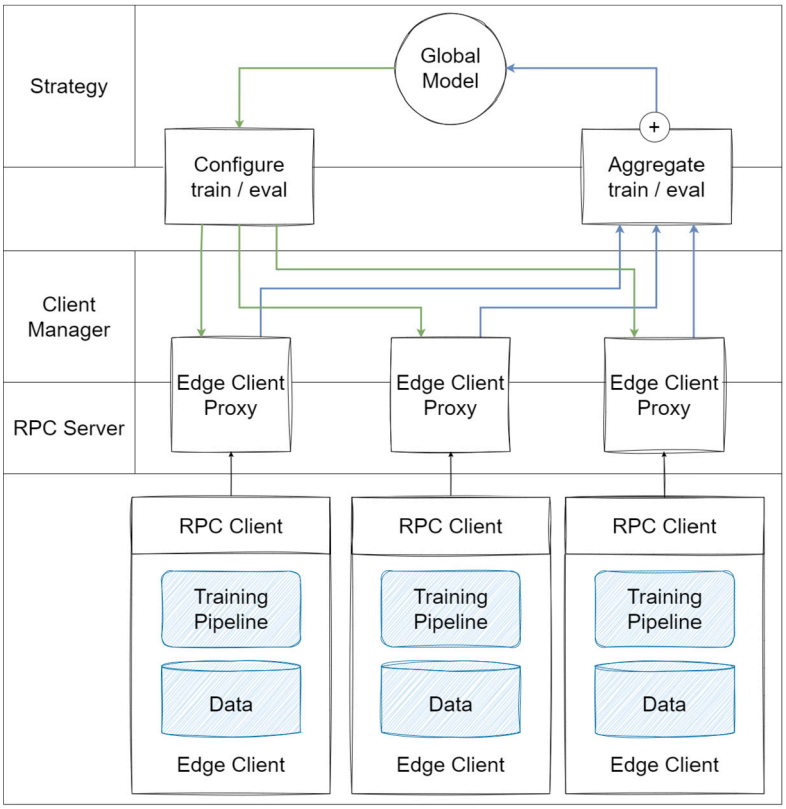

Fig. 8. Architecture of Flower [90].

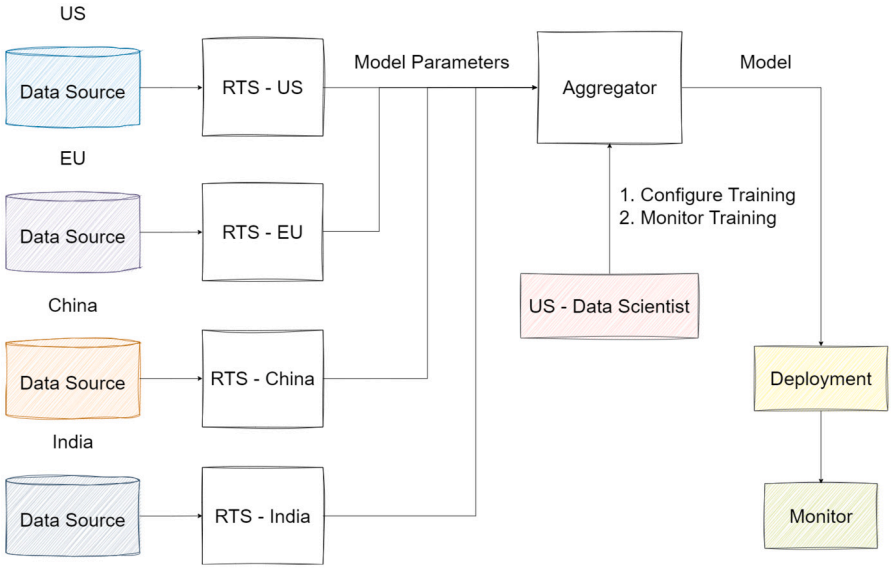

Fig. 9. Architecture of IBM FL [91].

can construct their own FL simulation algorithms by modifying the substructures. Three different simulation plans have been added, namely Standalone, Cross-process, and Hierarchical [74]. These modes are used to simulate a single operation by multiple clients with limited possibilities, to process multiple large FL models in the same environment, and to provide communication between local and global servers, respectively [74]. The communication contract definition and FL optimization stages are performed during the model development process. Fig. 10 shows the FedLab architecture with the interaction between the server

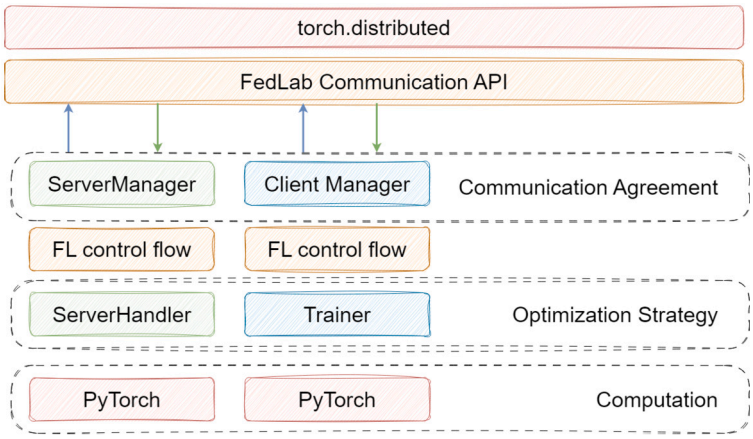

Fig. 10. Architecture of FedLab [74].

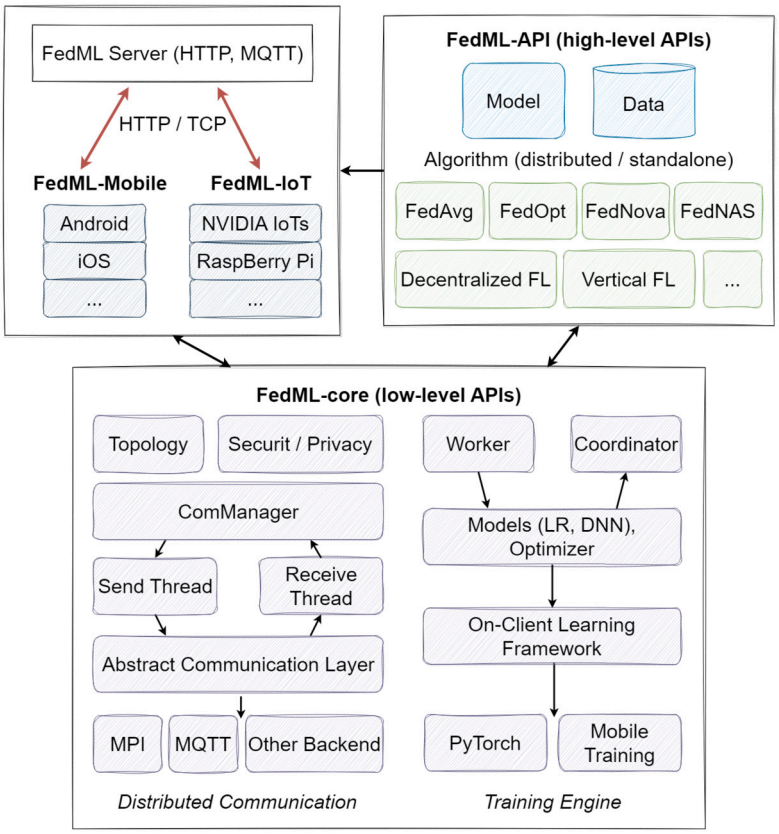

Fig. 11. Architecture of FedML [75].

and the client via the FedLab Communication API. Focusing on FedLab's communication agreement, optimization strategies, and computation steps, the flowchart of the framework is expressed.

- **FedML:** FedML is an open-source framework that offers flexibility in training. It supports different computing methods, including distributed computing, on-device training for edge devices, and single-machine simulation [75]. It provides FL training by supporting various algorithmic operations. It consists of two components, FedML-API, which represents the high-level API and enables the implementation of new algorithms in accordance with the client-oriented programming interface, and FedML-core, which represents the low-level API and divides communication and training modules [75]. The design objective behind the low-level APIs is to amplify security, privacy, and resilience. The architecture of the FedML framework consists of FedML Server, FedML-API, and FedML-core structures (as shown in Fig. 11). FedML Server coordinates FL processes, FedML-API performs optimization operations, and FedML-core manages the framework's communication and training operations. Integrated

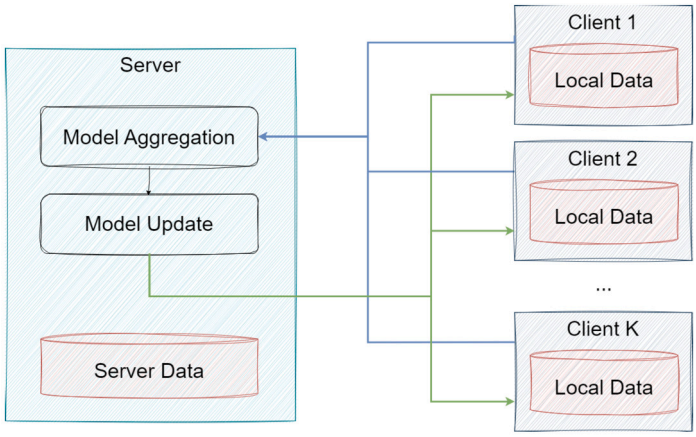

Fig. 12. Architecture of FLUTE [92].

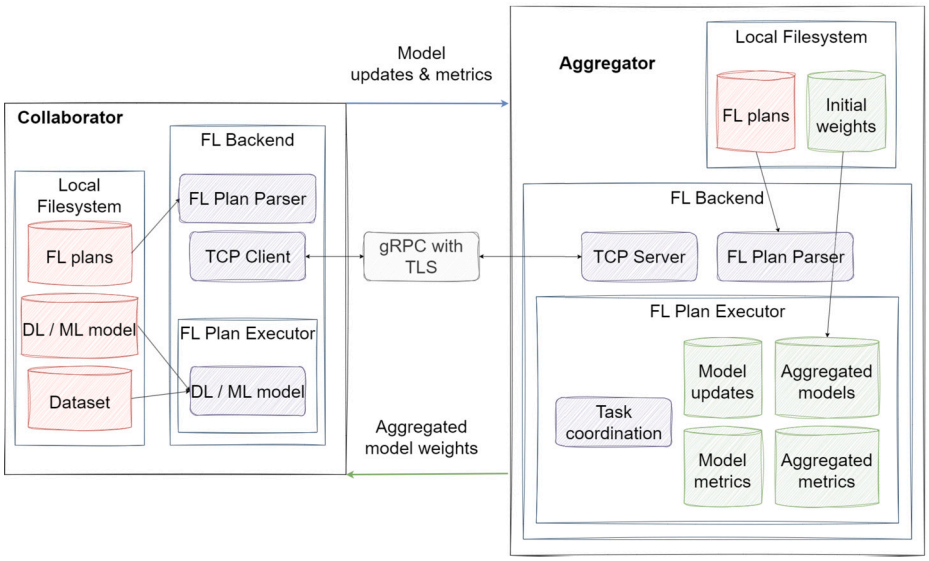

Fig. 13. Architecture of OpenFL [77].

work between and within these structures ensures that the framework provides training in a powerful and efficient manner. These dynamic structures in the architecture of the FedML framework are designed to optimize the learning process and ensure effective communication between different components.

- **FLUTE:** This framework was developed by the Microsoft Research team to ensure enhanced performance in FL research [76]. Tailored for FL simulations, it aims to design a framework that works better in security, optimization, and communication scales [76]. Training procedures can be changed according to the customer's request, and they can accommodate complex algorithms. During the training phase, only gradients are sent to the centralized server, and the client or server can change the gradient values for privacy [76]. FedAvg strategy is used as the basis for federated optimization. Furthermore, it seamlessly integrates with the NVIDIA Collective Communications Library, presenting potential advantages in temporal efficiency and memory consumption [76]. Fig. 12 is a diagram describing the architecture of the FLUTE framework. Here, it is shown how the model on the server is updated, how the data from the clients are integrated, and how the federative approach is applied in the model training process.

- **OpenFL:** OpenFL framework was initially developed for applications in healthcare but has subsequently been updated for broader industrial usage, primarily due to its robust security features [77]. Throughout the model training phase, the model owner can only access model weight updates and measurements. The data used here is retained at the collaborating center [77]. The FL plan and model code can be accessed prior to training with an OpenFL command [77]. It also offers developers the ability to easily work with remote data. However, it doesn't provide differential privacy. Fig. 13 gives the architecture of the OpenFL framework. Here, the flow of model parameters between the collaborator and the aggregator ensures the training process. In addition, the local file system, FL backend, and FL plan executor, which are included in the internal structure of both collaborator

Table 4
Comparison of the properties of the frameworks [93].

Framework	Properties							
	Developer	Open source	Data sharing[1]	Usability[2]	Integration[3]	Industry applications	Safety-focused design	Scalability
TFF	Google	Yes	H	3	T	Yes	Yes	Yes
PySyft	OpenMined	Yes	H & V	3	K, P, T	Yes	Yes	Yes
NVFlare	NVIDIA	Yes	H & V	2	P, T	Yes	Yes	Yes
FATE	WeBank	Yes	H & V	3	K, P, T	Yes	Yes	Yes
Flower	Adap gmbh	Yes	H	1	K, P, T	Yes	Yes	Yes
IBM FL	IBM	No	H	4	K, P, T	Yes	Yes	Yes
FedLab	FedLab	Yes	H	3	K, P, T	Yes	Yes	Yes
FedML	FedML Inc.	Yes	H	5	J, M, P, T	Yes	Yes	Yes
FLUTE	Microsoft	Yes	H	2	P, T	Yes	Yes	Yes
OpenFL	Intel	Yes	H	5	P, T	Yes	Yes	Yes

[1] H = Horizontal Federated Learning, V = Vertical Federated Learning.
[2] The ease of usage rating is high if the framework has paper, GitHub, website, and video tutorials for its use.
[3] J = Jax, K = Keras, M = MXNet, P = PyTorch, T = TensorFlow.

and aggregator, constitute the basic infrastructure components of this framework. The workflows within these infrastructures are critical elements that support the general training mechanism of OpenFL.

6.2. Criteria to choose framework

As can be seen from the frames given in the previous section, each frame has its own pros and cons. These features allow users to make different choices according to the FL model they will train. In Table 4, comparisons of these frameworks in terms of various features are given. For example, one of the most essential features that users can benefit from is whether the framework is open source or not. When this matter is examined, it is seen that the sample frameworks given here, except the IBM framework, are open source. As the next feature, when the data sharing permission is investigated, it is stated that many frameworks do not allow vertical FL applications. The ease of use of these frameworks is rated according to the level of availability of manuals such as paper, GitHub, website, and video tutorials. FedML and OpenFL are highly rated for having a large number of manuals. In terms of integration, it has been observed that while TFF can be integrated into the least environment, FedML can be integrated into the most. It is understood from the table that the last three features are taken into account in all frameworks. These can be interpreted as follows. Applications can be implemented with these frameworks in the industrial field. The designs of the models are formed to secure privacy. All of them have a scalable structure.

7. Opportunities, challenges, and future directions

This section provides an overview of the opportunities, challenges, and future directions of federated learning. Federated learning is a decentralized learning model approach offering improved privacy protection while allowing the collective intelligence of various devices or data resources. However, there are still issues to be addressed, such as communication, system and data heterogeneity, cyber-attacks, etc. Future research and direction in Federated Learning will primarily focus on these challenges.

7.1. Opportunities

Federated learning is one of the promising fields in the AI/ML concept. It also offers many opportunities. The primary ones can be summarized below.

- Preserving privacy: Training provided with federated learning is carried out on the local devices of the clients. Only the updated parameters of the model are shared with the server, and the data used in training does not need to be shared [94]. With this method, the confidentiality of the data can be ensured. To provide privacy secure aggregation protocols can be introduced that devise a training framework that can protect the local model updates [95] [96]. These protocols can enable the server to create a combination between the global model's update and each individual local model without learning about any information. This prevents the local model updates from staying concealed from the server which prevents the server from exploiting the updates of any user to infer their private training data [97].
- Improved data security: The data does not leave the device and is processed locally. The information shared with the server is limited to only updated model parameters. Therefore, the server does not have access to any client's own training data.
- Scalability: During the FL process, model training can be carried out in parallel on different devices of the clients. Here, the processed dataset can be divided into users, and its size can be reduced. Additionally, the location of the devices does not matter as long as they have a connection to the server. As a result of the convenience brought by such examples, it can be said that model training with FL is scalable.

- Reduced communication costs: Especially for AI training, the processed datasets must be quite large. Hence, the accuracy of the training increases. In addition, in studies developed collaboratively, a large dataset can be obtained by combining the datasets of different clients. In studies conducted with FL, working with a large amount of data requires less communication cost. Because each client processes its own dataset and shares only the training parameters with the server without sharing the dataset. For this, low bandwidth is sufficient [64].
- Edge computing integration: With Edge Computing (EC), IoT data obtained from devices such as sensors is used in studies [64]. However, this data is directly shared with third parties. Since the data is shared, communication costs are high. Data confidentiality also cannot be ensured. However, it is possible to integrate this architecture with FL. Thus, EC studies can be performed using the opportunities provided by FL.
- Real-time adaptability: As aforementioned, model training is done on local devices. The model parameters obtained as a result of the training are instantly shared with the server. Therefore, clients can directly access the current model. In this way, efficient and fast training processes are carried out in real-time.
- Decreased data bias: In the training performed by different clients on their own devices, data that do not have the same content and type are used. These data can be IID (independent and identically distributed) or non-IID. Thus, data diversity increases. This increases the generalization performance of the updated model and reduces data bias.

7.2. Challenges

Research on FL is still in its primary stages. Federated Learning has several challenges. Some of them are discussed below.

- Security and Privacy Challenges: The fundamental premise of Federated Learning is to provide privacy to the local datasets. A secure aggregation algorithm is proposed that can aggregate encrypted local models without decrypting the data in the aggregator [98]. However, a specific local learner's identity can be disclosed. It can be done by analyzing the global model [99]. To prevent this, differentially private federated learning has been proposed [100]. Necessary privacy is provided at the local learner level instead of providing protection to a single data sample.
- Wireless Communication and Settings: The other challenge is wireless communication (widely used in real-world applications) due to limited channel capacity, noise, and interference. The information is quantized before being sent over to the channels. This is done by exchanging model parameters between the local learners and the aggregator. The federated learning paradigm takes off with parameter quantization [101].
- Communication Overloads: In federated learning, it is one of the major challenges. Existing studies tried to solve this issue by applying data compression [4] or by allowing clients to add only relevant output [102,103].
- System and Data Heterogeneity: In the FL, the system and data heterogeneity in the network along with the non-identically distributed data from the nodes significantly affect the overall model and system performance. An FL model trained on heterogeneous, i.e., non-IID datasets may bias training in the direction that causes heterogeneity of datasets stored locally by clients. As observed in the literature, the problem of data heterogeneity is tried to be overcome by developed strategies used in FL. For example, while FedAvg is more suitable for training with IID datasets, the developed FedProx, FedYogi, Scaffold, etc. strategies were improved to adapt the strategies to non-IID datasets [32–34].
- Membership Inference Attacks: In this case, the raw data stays in the local device. Even with this step, there are several ways, which can infer the training data used in FL. It is possible to extract the information regarding the training data [104]. It looks for mechanisms that offer a differential privacy guarantee. Some of the defense mechanisms are discussed below.
  - Secure Computation: There are two main techniques that fall under this category, i.e., Secure Multiparty Computation (SMC) and Homomorphic Encryption (HE). In SMC, with the inputs provided by the participants only two or more parties comes to an agreement to perform. Also, the output is only revealed exclusively to a subset of participants. In HE, computation is performed on encrypted inputs without decrypting it first [104].
  - Differential Privacy: In this scheme, before model aggregation, noise is added to mask a user's contribution. [100].
  - Trusted Execution Environment (TEE): It is a secure platform, which runs a process when provided with low computational overhead. This is executed when it is compared with a secure computation technique.
- Data Poisoning Attacks: It is widely recognized as the most prevalent form of attack against ML models. In the context of the FL model, this attack is carried out during the training process by introducing malicious behavior to a subset of participating devices. This results in the compromisation of model accuracy. The adversary has the ability to directly inject poisoned data into the devices or inject poisoned data through other devices [105]. Identifying malicious participants is the defense mechanism against this kind of attack. It is executed in each round of learning before averaging.
- Model Poisoning Attacks: These attacks are similar to data poisoning attacks. In this case, the main objective is to corrupt the local models rather than manipulating the local data. The global model is affected by introducing errors. The attacker carries out a model poisoning attack by compromising certain devices and altering the parameters of the local model. The defense mechanism is almost similar to data poisoning attacks. The most common defense mechanisms are based on rejections. It is mostly based on two factors one is error rate and another one is loss function [106]. To overcome this challenge, Bagdasaryan et al. have developed a model-replacement technique that is accomplished by injecting backdoors [107]. However, the persistence and stealth of this methodology are not good enough [108]. Thus, stealth was tried to be ensured with another methodology developed in [109]. The method was improved with the use of an alternating minimization strategy. Additionally, they noted that their methodology is very vulnerable to Byzantine-resilient aggregation strategies [109]. In the study [108], an attack based

on optimizing stealthy and persistent was proposed. This was developed by bypassing defense methods and avoiding forgetting, respectively.

- Backdoor Attacks: Secured Federated Learning provides the devices anonymity during the process of the model update. Backdoor Attacks use the same functionality, a device or a group of devices introduces an adversary in the global model [107]. An attacker can manipulate certain tasks without impacting the overall accuracy of the global model. This can be done by assigning a particular label to a data instance with specific characteristics, which is commonly referred to as a targeted attack.

7.3. Future directions

In federated learning, there are many open research directions required to explore, which can address the challenges discussed earlier. They can be briefly explained in the following.

- Wireless Settings: One of the important considerations in paradigm quantization is the robustness of the models present in the quantization error. It also includes communication bandwidth, noise, and interference. Robustness to these channel effects might be a consideration [101].
- System and Data Heterogeneity: FedAvg was introduced as a method to tackle this challenge. However, due to the large differences in the structure of the datasets used for different applications, this method is not efficient in overcoming this. Modifications in the model aggregation methods are addressed to solve this issue [110,111].
- Incentive Mechanism: FL methods operate under the assumption that devices will collaborate during the learning process as needed, without taking rewards into account. On the contrary, nodes need to be economically compensated for their participation. This reputation-based incentive method, where devices get rewarded according to their model accuracy, data reliability, and contribution, might help get better models [112,113].
- Federated Learning as a Service: Federated Learning as a cloud service is recommended for collaborating among third parties. This framework allows the third party applications to contribute and collaborate on an ML model. This was developed in a recent work [114]. For any operating environment, the framework claims to be suitable.
- Asynchronous Federated Learning: Current FL aggregation techniques are for devices that has the ability to work in a synchronized manner. However, due to factors such as systems and data heterogeneity along with training and model transfer a synchronized manner is followed. In this manner, the feasibility to scale federated optimization has a chance of decreasing. This can happen in a synchronized manner [115]. Asynchronous federated averaging techniques have the capability to accommodate a larger number of devices, enabling updates to be received at different times in comparison to FedAvg.
- Blockchain in FL: In order to enhance the global model's capability to handle the asynchronous arrival of device parameters, the inclusion of an aggregator is essential. The presence of this component serves as a requirement for the widespread adoption of FL models. Blockchain is an example of a decentralized network. This means devices have the ability to learn collaboratively without using the central aggregator. Federated Learning was proposed by some works in blockchain framework [116].
- Backdoor Attacks: The model remains vulnerable to backdoor attacks by attackers placing backdoor triggers on local models during the training stage. Then, at the prediction stage, attackers triggered by crafted inputs cause misclassification [117]. To overcome this challenge, various backdoor defense methods are being developed [107].
- Poisoning Attacks: It has been determined that FL frameworks have vulnerabilities in active attacks. One of these, the poisoning attack, occurs when attackers damage the global model with local updates prepared by attackers [118]. In addition, data poisoning attacks are sometimes performed by injecting malicious data into the training dataset before the learning stage begins [119].
- Adversarial Example Attacks: In this type of attack, malicious output is produced by adding little disturbances to the input samples [120].

8. Conclusion

This survey paper provides a comprehensive overview of Federated Learning (FL), i.e., a distributed machine learning approach, which enables collaborative training of a shared model without sharing raw data. Unlike traditional approaches, FL facilitates collaborative model training without the insecure exchange of raw data, thereby safeguarding critical data privacy and security. FL offers a solution by allowing data to remain decentralized while facilitating model training through collaborative efforts and various strategies of FL. Security concerns related to FL are addressed in the survey, recognizing the importance of safeguarding data and models in distributed learning environments. Furthermore, the survey explores the FL-based applications, revealing their transformative potential across diverse domain(s). Additionally, the survey recognizes both the opportunities and challenges associated with working with FL, underscoring the potential for enhanced machine-learning applications while acknowledging the complexities involved. It is expected that the survey paper will contribute to the advancement and adoption of this innovative approach in the field of federated learning in terms of data privacy, security, and scalability.

CRediT authorship contribution statement

Betul Yurdem: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Murat Kuzlu:** Writing – review & editing, Writing – original draft, Supervision, Data curation,

Conceptualization. **Mehmet Kemal Gullu:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Formal analysis, Conceptualization. **Ferhat Ozgur Catak:** Writing – review & editing, Writing – original draft, Supervision, Investigation, Formal analysis, Conceptualization. **Maliha Tabassum:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization.

Declaration of competing interest

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

Data availability statement

Not applicable.

Appendix A. Federated learning strategies - algorithms

Algorithm 1 FedAvg [18].

Server performs:
k is indexed as the K clients,
the minibatch size is denoted by B,
the number of local epochs is E,
the learning rate is denoted as η
ω_0 is initialized by the server
for $t = 1, 2, \ldots, T$ **do**
  $m \leftarrow \max(C.K, 1)$
  $S_t \leftarrow m$ clients (random set)
  for $k \in S_t$ **do**
    $\omega_{t+1}^k \leftarrow \text{ClientUpdate}(k, \omega_t)$
    $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k$
    ClientUpdate(k, ω):
    $\beta \leftarrow$ split P_k into batches of size B
    for each local epoch i from 1 to E **do**
      for batch $b \in \beta$ **do**
        $\omega \leftarrow \omega - \eta \nabla l(\omega; b)$ ▷ l represents the loss function
      end for
    end for
  end for
end for
return ω to server ▷ The updated model ω is returned to the server for the next iteration

Algorithm 2 FedYogi, FedAdam and FedAdagrad [121].

Initialization: $x_0, v_{-1} \geq \tau^2$,
Decay parameters $\beta_1, \beta_2 \in [0, 1]$
for $t = 0, \ldots., T-1$ **do**
  S_t: Sample subset of clients
  $x_{i,0}^t = x_t$
  for each client $i \in S$ **in parallel do**
    for $k = 0, \ldots, K-1$ **do**
      Compute an unbiased estimate $g_{i,k}^t$ of $\nabla F_i(x_{i,k}^t)$
      $x_{i,k+1}^t = x_{i,k}^t - \eta_l g_{i,k}^t$
      $\Delta_i^t = x_{i,K}^t - x_t$
      $\Delta_t = \beta_1 \Delta_{t-1} + (1-\beta_1)(\frac{1}{|S|} \sum_{i \in S} \Delta_i^t)$
      $v_t = v_{t-1} - (1-\beta_2)\Delta_t^2 sign(v_{t-1} - \Delta_t^2)$ **(FedYogi)**
      $v_t = \beta_2 v_{t-1} + (1-\beta_2)\Delta_t^2$ **(FedAdam)**
      $v_t = v_{t-1} + \Delta_t^2$ **(FedAdagrad)**
      $x_{t+1} = x_t + \eta \frac{\Delta_t}{\sqrt{v_t} + \tau}$
    end for
  end for
end for

Algorithm 3 FedProx [122].

Input Parameters:
The number of devices are denoted by K,
Total round is denoted as T,
μ is used as proximal term,
γ is used for step size,
ω^0 is donated for initial model,
N is donates for the total number of device,
ρ_k is used as device-specific weights for K = 1,....,N
for $t = 0, ..., T-1$ **do**
  A random subset is selected by subset S_t of K
  (each device k is chosen with probability p_k)
  ω^t is send by server to all chosen devices
  Each chosen device $k \in S_t$ finds a ω_k^{t+1} which is a γ_k^t - inexact minimizer of:
  $\omega_k^{t+1} \approx argmin_\omega h_k(\omega;\omega^t) = F_k(\omega) + \frac{\mu}{2} || \omega - \omega^t ||^2$
  $k \in S_t$ is sent by each device ω_k^{t+1} back to the server
  Each server is aggregated by the ω as ω^{t+1}
  $\omega^{t+1} = \frac{1}{K} \sum_{k \in S_t} \omega_k^{t+1}$
end for

Algorithm 4 Scaffold [123].

Initial Input Parameters:
Initial model is denoted by x_o
Scalar value is denoted by c
A local learning rate or step size is denoted by $\eta\imath$
A global learning rate or step size is denoted by η
for $t = 0, ..., T-1$ **do**
  A subset S of clients is sampled
  $x_i^t = x_t = x_t$ 5 ▷ Parallel processing over clients is done
  for each client $i \in S$ **do** in parallel
    for $e = 1, ..., E$ **do**
      for $b \in B_i$ **do**
        $g_i^t = \nabla f_i(x_i^t; b)$
        $x_i^t = x_i^t - \eta\imath(g_i^t - c_i + C)$
        $c_i^+ = c_i - c + (E \mid B_i \mid \eta\imath)^{-1}(x_i^t - x_i)$
        $\Delta x_i = x_i^t - x_t, c_i = c_i^+ - c_i$
        $c_i = c_i^+$
        $n = \sum_{i \in S} n_i, \Delta x = \sum_{i \in S} \frac{n_i}{n} \Delta x_i, \Delta c = \sum_{i \in S} \frac{n_i}{n} \Delta c_i$
        $x_{t+1} = x_t + \eta \Delta x, c = c + \frac{|S|}{N} \Delta c$
      end for
    end for
  end for
end for

References

[1] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature 529 (28) (2016) 484–492.
[2] L. Li, Y. Fan, M. Tse, K.-Y. Lin, A review of applications in federated learning, Comput. Ind. Eng. 149 (2020) 106854.
[3] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (1–2) (2021) 1–210.
[4] J. Konečnỳ, H.B. McMahan, D. Ramage, P. Richtárik, Federated optimization: distributed machine learning for on-device intelligence, arXiv preprint, arXiv: 1610.02527, 2016.
[5] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 1310–1321.
[6] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.
[7] P. Regulation, Regulation (eu) 2016/679 of the European Parliament and of the council, Regulation (eu) 679 (2016) 2016.
[8] M. Boban, Digital single market and eu data protection reform with regard to the processing of personal data as the challenge of the modern world, in: Economic and Social Development: Book of Proceedings, 2016, p. 191.
[9] C. Zhang, X. Hu, Y. Xie, M. Gong, B. Yu, A privacy-preserving multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, Front. Neurorobot. 13 (2020) 112.
[10] Y. Xie, H. Wang, B. Yu, C. Zhang, Secure collaborative few-shot learning, Knowl.-Based Syst. 203 (2020) 106157.
[11] W.G. Van Panhuis, P. Paul, C. Emerson, J. Grefenstette, R. Wilder, A.J. Herbst, D. Heymann, D.S. Burke, A systematic review of barriers to data sharing in public health, BMC Public Health 14 (1) (2014) 1–9.
[12] S. Sarp, M. Kuzlu, E. Wilson, O. Guler, Wg2an: synthetic wound image generation using generative adversarial network, J. Eng. 2021 (5) (2021) 286–294.
[13] S. Sarp, Y. Zhao, M. Kuzlu, Artificial Intelligence-Powered Chronic Wound Management System: Towards Human Digital Twins, 2022.
[14] S.D. Holcomb, W.K. Porter, S.V. Ault, G. Mao, J. Wang, Overview on deepmind and its alphago zero ai, in: Proceedings of the 2018 International Conference on Big Data and Education, 2018, pp. 67–71.

[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.
[16] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[17] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, arXiv preprint, arXiv:1811.03604, 2018.
[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.
[19] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318.
[20] J. Wang, Z. Charles, Z. Xu, G. Joshi, H.B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, et al., A field guide to federated optimization, arXiv preprint, arXiv:2107.06917, 2021.
[21] J. Qi, Q. Zhou, L. Lei, K. Zheng, Federated reinforcement learning: techniques, applications, and open challenges, arXiv preprint, arXiv:2108.11887, 2021.
[22] Q. Ho, J. Cipar, H. Cui, S. Lee, J.K. Kim, P.B. Gibbons, G.A. Gibson, G. Ganger, E.P. Xing, More effective distributed ml via a stale synchronous parallel parameter server, Adv. Neural Inf. Process. Syst. 26 (2013).
[23] S. Lee, M.E. Lacy, M. Jankowich, A. Correa, W.-C. Wu, Association between obesity phenotypes of insulin resistance and risk of type 2 diabetes in African Americans: the Jackson heart study, J. Clin. Transl. Endocrinol. 19 (2020) 100210.
[24] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, D. Evans, Privacy-preserving distributed linear regression on high-dimensional data, Cryptol. ePrint Arch. (2016).
[25] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, Q. Yang, Hhhfl: hierarchical heterogeneous horizontal federated learning for electroencephalography, arXiv preprint, arXiv:1909.05784, 2019.
[26] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, T. Huang, Fairness and accuracy in horizontal federated learning, Inf. Sci. 589 (2022) 170–185.
[27] H. Zhu, H. Zhang, Y. Jin, From federated learning to federated neural architecture search: a survey, Complex Intell. Syst. 7 (2) (2021) 639–657.
[28] M. Kuzlu, Z. Xiao, M. Tabassum, F.O. Catak, A robust diabetes mellitus prediction system based on federated learning strategies, in: 2023 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS), 2023, pp. 246–253.
[29] J.O.d. Terrail, S.-S. Ayed, E. Cyffers, F. Grimberg, C. He, R. Loeb, P. Mangold, T. Marchand, O. Marfoq, E. Mushtaq, et al., Flamby: datasets and benchmarks for cross-silo federated learning in realistic healthcare settings, arXiv preprint, arXiv:2210.04620, 2022.
[30] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc. Mach. Learn. Syst. 2 (2020) 429–450.
[31] S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A.T. Suresh, Scaffold: stochastic controlled averaging for federated learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.
[32] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, C.-Z. Xu, Feddc: federated learning with non-iid data via local drift decoupling and correction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10112–10121.
[33] H. Nguyen, H. Warrier, Y. Gupta, A novel approach for federated learning with non-iid data, in: 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), IEEE, 2022, pp. 62–67.
[34] T.K. Dang, X. Lan, J. Weng, M. Feng, Federated learning for electronic health records, ACM Trans. Intell. Syst. Technol. 13 (5) (2022) 1–17.
[35] C. Dwork, Differential privacy, in: International Colloquium on Automata, Languages, and Programming, Springer, 2006, pp. 1–12.
[36] F. Bayatbabolghani, M. Blanton, Secure multi-party computation, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 2157–2159.
[37] G. Szegedi, P. Kiss, T. Horváth, Evolutionary federated learning on eeg-data, in: ITAT, 2019, pp. 71–78.
[38] E. Choi, M.T. Bahadori, L. Song, W.F. Stewart, J. Sun, Gram: graph-based attention model for healthcare representation learning, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 787–795.
[39] S.R. Pfohl, A.M. Dai, K. Heller, Federated and differentially private learning for electronic health records, arXiv preprint, arXiv:1911.05861, 2019.
[40] S. Silva, B.A. Gutman, E. Romero, P.M. Thompson, A. Altmann, M. Lorenzi, Federated learning in distributed medical databases: meta-analysis of large-scale subcortical brain data, in: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), IEEE, 2019, pp. 270–274.
[41] H. Ye, L. Liang, G.Y. Li, J. Kim, L. Lu, M. Wu, Machine learning for vehicular networks: recent advances and application examples, IEEE Veh. Technol. Mag. 13 (2) (2018) 94–101.
[42] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H.V. Poor, Federated learning for Internet of things: a comprehensive survey, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1622–1658.
[43] A. Gharaibeh, M.A. Salahuddin, S.J. Hussini, A. Khreishah, I. Khalil, M. Guizani, A. Al-Fuqaha, Smart cities: a survey on data management, security, and enabling technologies, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2456–2501.
[44] M. Mohammadi, A. Al-Fuqaha, Enabling cognitive smart cities using big data and machine learning: approaches and challenges, IEEE Commun. Mag. 56 (2) (2018) 94–101.
[45] W. Wang, C. Feng, B. Zhang, H. Gao, Environmental monitoring based on fog computing paradigm and Internet of things, IEEE Access 7 (2019) 127154–127165.
[46] M. Masera, E.F. Bompard, F. Profumo, N. Hadjsaid, Smart (electricity) grids for smart cities: assessing roles and societal impacts, Proc. IEEE 106 (4) (2018) 613–625.
[47] S. Pérez, J. Pérez, P. Arroba, R. Blanco, J.L. Ayala, J.M. Moya, Predictive gpu-based adas management in energy-conscious smart cities, in: 2019 Ieee International Smart Cities Conference (isc2), IEEE, 2019, pp. 349–354.
[48] N. Victor, M. Alazab, S. Bhattacharya, S. Magnusson, P.K.R. Maddikunta, K. Ramana, T.R. Gadekallu, et al., Federated learning for iout: concepts, applications, challenges and opportunities, arXiv preprint, arXiv:2207.13976, 2022.
[49] H. Zhao, F. Ji, Q. Guan, Q. Li, S. Wang, H. Dong, M. Wen, Federated meta learning enhanced acoustic radio cooperative framework for ocean of things underwater acoustic communications, arXiv preprint, arXiv:2105.13296, 2021.
[50] C.L. Hagerty, Deepwater Horizon Oil Spill: Selected Issues for Congress, Diane Publishing, 2010.
[51] J. Telford, J. Cosgrave, The international humanitarian system and the 2004 Indian Ocean earthquake and tsunamis, Disasters 31 (1) (2007) 1–28.
[52] M. Karagoz, Maritime security operations and the combined joint operations from the sea center of excellence, in: Maritime Security and Defence Against Terrorism, IOS Press, 2012, pp. 87–91.
[53] W. Saad, M. Bennis, M. Chen, A vision of 6g wireless systems: applications, trends, technologies, and open research problems, IEEE Netw. 34 (3) (2019) 134–142.
[54] X. Han, H. Yu, H. Gu, Visual inspection with federated learning, in: Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part II 16, Springer, 2019, pp. 52–64.
[55] F. Liu, X. Wu, S. Ge, W. Fan, Y. Zou, Federated learning for vision-and-language grounding problems, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 11572–11579.
[56] N.I. Mowla, N.H. Tran, I. Doh, K. Chae, Federated learning-based cognitive detection of jamming attack in flying ad-hoc network, IEEE Access 8 (2019) 4338–4350.
[57] Y.M. Saputra, D.T. Hoang, D.N. Nguyen, E. Dutkiewicz, M.D. Mueck, S. Srikanteswara, Energy demand prediction with federated learning for electric vehicle networks, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[58] W. Yang, Y. Zhang, K. Ye, L. Li, C.-Z. Xu, Ffd: a federated learning based method for credit card fraud detection, in: Big Data–BigData 2019: 8th International Congress, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 8, Springer, 2019, pp. 18–32.

[59] M.R. Sprague, A. Jalalirad, M. Scavuzzo, C. Capota, M. Neun, L. Do, M. Kopp, Asynchronous federated learning for geospatial applications, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 21–28.
[60] Y. Qu, M.P. Uddin, C. Gan, Y. Xiang, L. Gao, J. Yearwood, Blockchain-enabled federated learning: a survey, ACM Comput. Surv. 55 (4) (2022) 1–35.
[61] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, B. Yoon, A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology, Future Gener. Comput. Syst. 129 (2022) 380–388.
[62] C. Ma, J. Li, L. Shi, M. Ding, T. Wang, Z. Han, H.V. Poor, When federated learning meets blockchain: a new distributed learning paradigm, IEEE Comput. Intell. Mag. 17 (3) (2022) 26–33.
[63] D. Hou, J. Zhang, K.L. Man, J. Ma, Z. Peng, A systematic literature review of blockchain-based federated learning: architectures, applications and issues, in: 2021 2nd Information Communication Technologies Conference (ICTC), IEEE, 2021, pp. 302–307.
[64] H.G. Abreha, M. Hayajneh, M.A. Serhani, Federated learning in edge computing: a systematic survey, Sensors 22 (2) (2022) 450.
[65] J. Zhang, B. Chen, X. Cheng, H.T.T. Binh, S. Yu, Poisongan: generative poisoning attacks against federated learning in edge computing systems, IEEE Int. Things J. 8 (5) (2020) 3310–3322.
[66] Y. Ye, S. Li, F. Liu, Y. Tang, W. Hu, Edgefed: optimized federated learning based on edge computing, IEEE Access 8 (2020) 209191–209198, https://doi.org/10.1109/ACCESS.2020.3038287.
[67] D. Ye, R. Yu, M. Pan, Z. Han, Federated learning in vehicular edge computing: a selective model aggregation approach, IEEE Access 8 (2020) 23920–23935, https://doi.org/10.1109/ACCESS.2020.2968399.
[68] Tensorflow federated: machine learning on decentralized data, https://www.tensorflow.org/federated, 2020. (Accessed 21 May 2023).
[69] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose, et al., Pysyft: a library for easy federated learning, in: Federated Learning Systems: Towards Next-Generation AI, 2021, pp. 111–139.
[70] H.R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu, et al., Nvidia flare: federated learning from simulation to real-world, arXiv preprint, arXiv:2210.13291, 2022.
[71] Y. Liu, T. Fan, T. Chen, Q. Xu, Q. Yang, Fate: an industrial grade platform for collaborative learning with data protection, J. Mach. Learn. Res. 22 (1) (2021) 10320–10325.
[72] D.J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K.H. Li, T. Parcollet, P.P.B. de Gusmão, et al., Flower: a friendly federated learning research framework, arXiv preprint, arXiv:2007.14390, 2020.
[73] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn, et al., Ibm federated learning: an enterprise framework white paper v0. 1, arXiv preprint, arXiv:2007.10987, 2020.
[74] D. Zeng, S. Liang, X. Hu, H. Wang, Z. Xu, Fedlab: a flexible federated learning framework, J. Mach. Learn. Res. 24 (100) (2023) 1–7.
[75] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, et al., Fedml: a research library and benchmark for federated machine learning, arXiv preprint, arXiv:2007.13518, 2020.
[76] M. Hipolito Garcia, A. Manoel, D. Madrigal Diaz, F. Mireshghallah, R. Sim, D. Dimitriadis, Flute: a scalable, extensible framework for high-performance federated learning simulations, arXiv e-prints (2022) arXiv–2203.
[77] G.A. Reina, A. Gruzdev, P. Foley, O. Perepelkina, M. Sharma, I. Davidyuk, I. Trushkin, M. Radionov, A. Mokrov, D. Agapov, et al., Openfl: an open-source framework for federated learning, arXiv preprint, arXiv:2105.06413, 2021.
[78] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, Q. Yang, Secureboost: a lossless federated learning framework, IEEE Intell. Syst. 36 (6) (2021) 87–98.
[79] J. Benet, Ipfs-content addressed, versioned, p2p file system, arXiv preprint, arXiv:1407.3561, 2014.
[80] S. Silva, A. Altmann, B. Gutman, M. Lorenzi, Fed-biomed: a general open-source frontend framework for federated learning in healthcare, in: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2, Springer, 2020, pp. 201–210.
[81] Y. Xie, Z. Wang, D. Gao, D. Chen, L. Yao, W. Kuang, Y. Li, B. Ding, J. Zhou, Federatedscope: a flexible federated learning platform for heterogeneity, arXiv preprint, arXiv:2204.05011, 2022.
[82] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, M. Nordlund, Open-source federated learning frameworks for iot: a comparative review and analysis, Sensors 21 (1) (2020) 167.
[83] X. Liu, T. Shi, C. Xie, Q. Li, K. Hu, H. Kim, X. Xu, B. Li, D. Song, Unifed: a benchmark for federated learning frameworks, arXiv preprint, arXiv:2207.10308, 2022.
[84] Z. Zuo, M. Watson, D. Budgen, R. Hall, C. Kennelly, N. Al Moubayed, Data anonymization for pervasive health care: systematic literature mapping study, JMIR Med. Inform. 9 (10) (2021) e29871.
[85] NVIDIA, NVFlare: a framework for federated learning, https://developer.nvidia.com/nvflare, 2021. (Accessed 6 July 2023).
[86] Experimenting with novel distributed applications using NVIDIA Flare 2.1, https://developer.nvidia.com/blog/experimenting-with-novel-distributed-applications-using-nvidia-flare-2-1/, 2022. (Accessed 22 August 2023).
[87] F. Zheng, K. Li, J. Tian, X. Xiang, et al., A vertical federated learning method for interpretable scorecard and its application in credit scoring, arXiv preprint, arXiv:2009.06218, 2020.
[88] Z. Xiong, Z. Cheng, C. Xu, X. Lin, X. Liu, D. Wang, X. Luo, Y. Zhang, N. Qiao, M. Zheng, et al., Facing small and biased data dilemma in drug discovery with federated learning, BioRxiv (2020) 2020–03.
[89] Architecture, https://fate.readthedocs.io/en/latest/architecture/, 2021. (Accessed 22 August 2023).
[90] Flower architecture, https://flower.dev/docs/framework/contributor-explanation-architecture.html, 2022. (Accessed 22 August 2023).
[91] Federated learning architecture, https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fl-arch.html, 2023. (Accessed 22 August 2023).
[92] FLUTE overview, https://microsoft.github.io/msrflute/overview, 2021. (Accessed 22 August 2023).
[93] W. Riviera, I.B. Galazzo, G. Menegaz, Felebrities: a user-centric assessment of federated learning frameworks, IEEE Access (2023).
[94] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.
[95] Y. Aono, T. Hayashi, L. Wang, S. Moriai, et al., Privacy-preserving deep learning via additively homomorphic encryption, IEEE Trans. Inf. Forensics Secur. 13 (5) (2017) 1333–1345.
[96] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191.
[97] T. Nguyen, M.T. Thai, Preserving privacy and security in federated learning, IEEE/ACM Trans. Netw. (2023).
[98] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for federated learning on user-held data, arXiv preprint, arXiv:1611.04482, 2016.
[99] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: user-level privacy leakage from federated learning, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 2512–2520.
[100] R.C. Geyer, T. Klein, M. Nabi, Differentially private federated learning: a client level perspective, arXiv preprint, arXiv:1712.07557, 2017.
[101] S. Niknam, H.S. Dhillon, J.H. Reed, Federated learning for wireless communications: motivation, opportunities, and challenges, IEEE Commun. Mag. 58 (6) (2020) 46–51.
[102] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G.R. Ganger, P.B. Gibbons, O. Mutlu, Gaia: geo-distributed machine learning approaching LAN speeds, in: 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 2017, pp. 629–647.

[103] W. Luping, W. Wei, L. Bo, Cmfl: mitigating communication overhead for federated learning, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, pp. 954–964.
[104] P.M. Mammen, Federated learning: opportunities and challenges, arXiv preprint, arXiv:2101.05428, 2021.
[105] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, J. Liu, Data poisoning attacks on federated machine learning, IEEE Int. Things J. 9 (13) (2021) 11365–11375.
[106] M. Fang, X. Cao, J. Jia, N. Gong, Local model poisoning attacks to Byzantine-robust federated learning, in: 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1605–1622.
[107] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2938–2948.
[108] X. Zhou, M. Xu, Y. Wu, N. Zheng, Deep model poisoning attack on federated learning, Future Internet 13 (3) (2021) 73.
[109] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: International Conference on Machine Learning, PMLR, 2019, pp. 634–643.
[110] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan, et al., Towards federated learning at scale: system design, Proc. Mach. Learn. Syst. 1 (2019) 374–388.
[111] Y. Liu, J. James, J. Kang, D. Niyato, S. Zhang, Privacy-preserving traffic flow prediction: a federated learning approach, IEEE Int. Things J. 7 (8) (2020) 7751–7763.
[112] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, Q. Yang, A sustainable incentive scheme for federated learning, IEEE Intell. Syst. 35 (4) (2020) 58–69.
[113] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, D.I. Kim, Incentive design for efficient federated learning in mobile networks: a contract theory approach, in: 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), IEEE, 2019, pp. 1–5.
[114] N. Kourtellis, K. Katevas, D. Perino, Flaas: federated learning as a service, in: Proceedings of the 1st Workshop on Distributed Machine Learning, 2020, pp. 7–13.
[115] C. Xie, S. Koyejo, I. Gupta, Asynchronous federated optimization, arXiv preprint, arXiv:1903.03934, 2019.
[116] H.B. Desai, M.S. Ozdayi, M. Kantarcioglu, Blockfla: accountable federated learning via hybrid blockchain architecture, in: Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, 2021, pp. 101–112.
[117] C. Zhu, J. Zhang, X. Sun, B. Chen, W. Meng, Adfl: defending backdoor attacks in federated learning via adversarial distillation, Comput. Secur. 132 (2023) 103366.
[118] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 2019, pp. 374–380.
[119] M. Fang, X. Cao, J. Jia, N.Z. Gong, Local model poisoning attacks to byzantine-robust federated learning, CoRR, arXiv:1911.11815, 2019.
[120] B. Luo, Y. Liu, L. Wei, Q. Xu, Towards imperceptible and robust adversarial example attacks against neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.
[121] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, H.B. McMahan, Adaptive federated optimization, arXiv preprint, arXiv:2003.00295, 2020.
[122] Should the global model replace the client model? · Issue #25 · litian96/FedProx — github.com, https://github.com/litian96/FedProx/issues/25. (Accessed 2 April 2023).
[123] S. Praneeth Karimireddy, S. Kale, M. Mohri, S.J. Reddi, S.U. Stich, A. Theertha Suresh, Scaffold: stochastic controlled averaging for federated learning, arXiv e-prints (2019) arXiv–1910.