Neurocomputing 597 (2024) 128019

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Survey Paper

Recent advances on federated learning: A systematic survey

Bingyan Liu, Nuoyan Lv, Yuanchun Guo, Yawen Li *

Beijing University of Posts and Telecommunications, HaiDian, Beijing, 100871, China

ARTICLE INFO

Communicated by J. Andreu-Perez

Keywords:
Artificial intelligence
Federated learning
Survey

ABSTRACT

Federated learning has emerged as an effective paradigm to achieve privacy-preserving collaborative learning among different parties. Compared to traditional centralized learning that requires collecting data from each party, in federated learning, only the locally trained models or computed gradients are exchanged, without exposing any data information. As a result, it is able to protect privacy to some extent. In recent years, federated learning has become more and more prevalent and there have been many surveys for summarizing related methods in this hot research topic. However, most of them focus on a specific perspective or lack the latest research progress. In this paper, we provide a systematic survey on federated learning, aiming to review the recent advanced federated methods and applications from different aspects. Specifically, this paper includes four major contributions. First, we present a new taxonomy of federated learning in terms of the pipeline and challenges in federated scenarios. Second, we summarize federated learning methods into several categories and briefly introduce the state-of-the-art methods under these categories. Third, we overview some prevalent federated learning frameworks and introduce their features. Finally, some potential deficiencies of current methods and several future directions are discussed.

1. Introduction

Over the past few years, deep neural networks (DNNs) have received a lot of attention due to their remarkable performance on various tasks such as Computer Vision (CV) [1–4], Natural Language Processing (NLP) [5–7], Recommendation Systems (RS) [8–10] and Data Mining (DM) [11–13]. However, the superiority of DNNs depends on the support of big data, which is hard to access in a certain party considering the limitation of the storage space and the difficulty of data collection. Gathering data from different parties to a central server for training is a direct solution to the issue. Nevertheless, data in each party may be sensitive or include some user privacy information. For example, medical images in a hospital are prohibited from outsourcing due to their privacy property. Besides, policies such as General Data Protection Regulation (GDPR) [14] also highlight the importance of protecting privacy when sharing information among different organizations. Thus, how to aggregate the data knowledge from different parties while ensuring privacy is an important and practical problem in real-world scenarios.

Federated learning (FL) [15], which enables multiple parties to collaboratively train a DNN with the help of a central server, can be regarded as an effective solution to the aforementioned problem. Different from the traditional centralized learning that needs to collect data from each party, in FL, data do not need to upload for a joint training. Instead, the local trained models are exchanged with a central server, which are used to aggregate the knowledge from all of the uploaded models and then distribute the global model to each party. As a result, each party is able to benefit from other parties, improving the model accuracy. In recent years, there have been many applications based on FL in practice, such as loan status prediction, health situation assessment, and next-word prediction [16–18].

We take Fig. 1 as an example to illustrate a typical FL pipeline. First, each hospital (party) trains the local model distributed from a central cloud. The training process is usually implemented based on SGD with local data and then generates corresponding local updates. Second, the local updates rather than local data are transferred to the cloud, where the updates are sampled in terms of some heuristic rules to ensure the overhead and some aggregation algorithms (e.g., FedAvg [15]) are conducted to achieve effective knowledge integration. In this way, the cloud can get an improved new global model and distributes it to each hospital for further tuning. These steps may repeat several times until the healthcare service can be satisfied (e.g., the accuracy of the learned model is acceptable for practical deployment).

There have been other surveys on FL over the past few years. For instance, Li et al. [19] summarized related FL methods from the system perspective, where the authors provided the definition of federated learning systems and analyzed the system components. Lim et al. [20] focused on the FL application in mobile edge networks. Lyu et al. [21]

* Corresponding author.
E-mail addresses: bingyanliu@bupt.edu.cn (B. Liu), lvnuoyan@bupt.edu.cn (N. Lv), gyc2001@bupt.edu.cn (Y. Guo), warmly0716@126.com (Y. Li).

https://doi.org/10.1016/j.neucom.2024.128019
Received 9 February 2023; Received in revised form 3 June 2024; Accepted 7 June 2024
Available online 11 June 2024

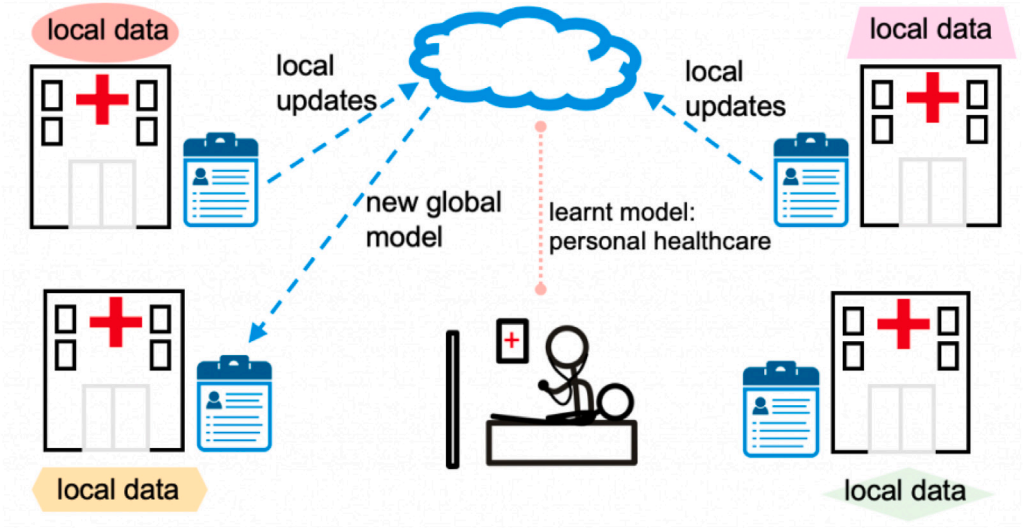

Fig. 1. An example of the FL pipeline [23].

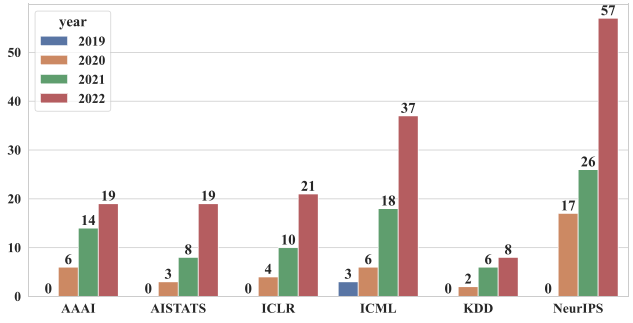

Fig. 2. The number of published FL papers in top-tie conference from 2019–2022.

paid more attention to the security and privacy issues existed in current FL schemes. However, these surveys only review a specific aspect of federated learning, failing to give readers a comprehensive understanding on FL. Towards the general FL overviews, most of them are out of date and cannot catch the latest trend in FL research. For example, Yang et al. [18] divided FL methods into three categories (i.e., horizontal federated learning, vertical federated learning and federated transfer learning) and described their features respectively. Kairouz et al. [22] gave a comprehensive introduction of federated learning theory and application. Notice that both of the surveys mainly cited papers published before 2020, which is impossible to track the latest research progress on FL considering the rapid development in this field. As shown in Fig. 2, we can clearly see that the number of accepted FL papers in top-tier conferences increases dramatically after 2020, which calls for a timely survey to summarize the advances in the FL community. Besides, the rapid update of FL frameworks also requires us to highlight their latest features.

In this paper, we attempt to provide a systematic survey on federated learning, targeting at reviewing the recent advanced federated methods and applications from different aspects. Specifically, the key contributions of this survey are as follows: (1) we present a new taxonomy based on the federated learning pipeline and challenges, which includes four typical aspects: *aggregation optimization, heterogeneity, privacy protection, fairness*. We will give detailed explanation in the following sections. (2) we summarize different federated learning methods into the proposed categories and briefly describe the state-of-the-art methods under these categories. (3) we overview the latest federated learning frameworks and introduce their features. (4) we discuss some potential deficiencies of current methods and several future directions.

The remainder of this survey is structured as follows. In Section 2, we first introduce preliminaries of federated learning. In Section 3, we propose the taxonomy of federated learning according to different aspects, in which various federated learning approaches are discussed and categorized. Then, in Section 4, we introduce some prevalent frameworks to show the practical deployment of federated learning. Finally, Section 5 and Section 6 discuss the future work and concludes this paper.

2. Preliminaries

2.1. Problem formulation

In this section, we first introduce some notations and symbols used in this survey to formally define federated learning. In general, there are two ends participated in the round of federated learning: *client end* and *server end*. The client end holds a series of local private data $\mathcal{D} = \{D_1, D_2, \dots, D_N\}$, which are then used to train the model in each client and generate local models $\mathcal{M} = \{M_1, M_2, \dots, M_N\}$. Here N denotes the number of clients. After the local training process, the local models $\mathcal{M}$, rather than the data $\mathcal{D}$, are uploaded to the server end, where aggregation algorithms are implemented to obtain a global model M_{global}. The process can be defined as

$$M_{global} = AGG(M_1, M_2, \dots, M_N), \tag{1}$$

where AGG represents the aggregation algorithms. In this way, we finish one round of federated learning and distribute the global model to each client side for further local training. The concrete number of round is usually determined by the model performance (i.e., we stop the process until the model can achieve desirable accuracy). In addition, to provide a more rigorous privacy protection, each client may enforce some encryption techniques to the models before uploading them. Differential privacy (DP) [24] and homomorphic encryption (HE) [25] are widely used to conduct such protection.

Based on the aforementioned statement, we can see that the performance of federated learning largely depends on the aggregation algorithm in the server end. Formally, the goal of federated learning is to optimize the following objective function

$$\min_{w} \; L(w), \; where \; L(w) = \sum_{i=1}^{N} f_i L_i(w), \tag{2}$$

where w is the weights of DNNs, $L(w)$ is the global loss function and $L_i(w)$ is the local loss function in the i_{th} client. f_i represents the importance of the i_{th} client and $\sum_{k=1}^{N} = 1$. In federated learning, the aggregation algorithm determines the value allocation for f_i. Many research papers that try to improve the accuracy performance of federated learning are focused on this aspect.

2.2. Key challenges

Different from traditional centralized learning or distributed learning, federated learning faces the following key challenges:

- **Heterogeneity problem.** In federated learning, the heterogeneity comes from three aspects: (1) Data heterogeneity. Considering that each participator collects data from its local end, the overall data distribution inevitably conforms to the non-independent identically distribution (non-iid) situation. For example, the same object image collected from different environments, or the same activity coming from different people, can lead to different data distributions, which will further affect the performance of federated aggregation [26]. (2) Model heterogeneity. In real-world scenarios, it is hard to limit the federated clients to use an identical model architecture. Instead, each client may prefer a distinctive model architecture for improved task performance. Therefore, how to aggregate these heterogeneous models is challenging in practical federated learning conditions. (3) System heterogeneity. Because of the variability in hardware, different parties may have different storage space, computation power, and communication capabilities. As a result, the server end needs to decide whether to wait for all parties to upload their models for better accuracy or remove stragglers (i.e., the parties with weak hardware performance) for accelerating the federation process.
- **Privacy leakage.** The key idea of federated learning is to achieve collaborative learning in a privacy-preserving manner, which differs from the traditional paradigm that exchanges data or other sensitive information. Keeping data in the local end and transferring corresponding models is the original privacy protection design in federated learning. However, the parameters of the uploaded models may also be exploited by attackers to infer the user privacy information [27]. So we require more rigorous encryption or obfuscation methods to ensure privacy.
- **Unfairness.** In traditional centralized learning or distributed learning, the unfairness problem does not exist since the participants belong to a same organization. However, the participants in federated learning come from various parties with different data resources. According to a previous work [28], if individuals with similar preferences and characteristics receive substantially different outcomes, then we say that the model violates individual fairness. Thus, it is necessary to generate federated models that go beyond average accuracy to further consider the fairness performance.

3. Approaches of federated learning

In this section, we first present a taxonomy of federated learning and allocate different federated approaches into different categories according to the taxonomy. Then for each category, we describe in detail how various methods achieve their goal.

3.1. Taxonomy

In this survey, we propose a new taxonomy to classify the existing federated learning methods (Fig. 3). Our taxonomy is motivated by the pipeline and challenges in federated learning. As stated in the previous section, the key step in the federated learning pipeline is the aggregation algorithm and the key challenges come from three different aspects. Therefore, in our taxonomy, federated learning approaches can be summarized into four cases: aggregation optimization, heterogeneous federated learning, secure federated learning and fair federated learning.

- **Aggregation optimization.** Considering that the number of participants in a federated learning system is usually large, it is essential to implement an effective aggregation optimization for outputting a better global model compared to the ones with local training. This survey investigates various aggregation methods such as FedAvg [15,26,29], FedMA [30] and FedProx [31], with a focus on how to combine local models into an improved global model.
- **Heterogeneous federated learning.** In real-world scenarios, federated clients may come from different environments or equip with various hardware, leading to the heterogeneity problem. In the following sections, we respectively explore how related research efforts address the issue of data heterogeneity, model heterogeneity and system heterogeneity. In particular, techniques such as meta-learning [32–37], multi-task learning [38–47], transfer learning [48–51] and clustering [52–58] are incorporated to achieve our goal.
- **Secure federated learning.** Although traditional federated learning has attempted to protect data privacy by only exchanging parameters of the local trained models, malicious attackers can still design some scheme to infer the properties of raw data. In our survey, we first summarize a series of attacks targeting federated learning, where we describe how backdoor attacks [59–65], gradient attacks [27,66–72] and poison attacks [73–76] are applied to compromise federated learning. Then we introduce how to combine federated learning, differential privacy (DP) [30,36,77–83], homomorphic encryption (HE) [84,85], trusted execution environment (TEE) [86,87] and other algorithms [64,88–90] to defend aforementioned attacks.
- **Fair federated learning.** During federated learning, it is possible that the performance of the global model varies significantly across the devices, resulting in the fairness problem. This survey reviews literature about how to ensure fair federated learning, such as designing minimax optimization strategies [91,92] and sample reweighting approaches [93,94].

3.2. Aggregation optimization

The goal of aggregation optimization is to improve the performance of the final global model, which is the core output in federated learning. There have been a large number of aggregation algorithms proposed to combine these local models to a better global one. In the following parts, we will describe in detail how different types of aggregation methods work.

3.2.1. Weight-level aggregation

A typical and prevalent weight-level aggregation method called FedAvg [15] is mostly adopted by developers. The key idea of FedAvg is to aggregate these local models in a coordinate-based weight averaging manner, which can be denoted as

$$W_g^r = \frac{1}{N}\sum_{k=1}^{N} w_k^r, \tag{3}$$

where N is the number of federated clients. w_k denotes the weight parameters of the k_{th} client and W_g^r is the final aggregated model at the r_{th} round. Researchers have shown the remarkable performance of FedAvg on a variety of public datasets (e.g., MNIST [95] and CIFAR-10 [96]) and provided some theoretical analyses to prove why FedAvg works well [97].

Despite being widely applied, FedAvg still suffers from the weight divergence problem [26]: the weight in the same coordinates (e.g.,

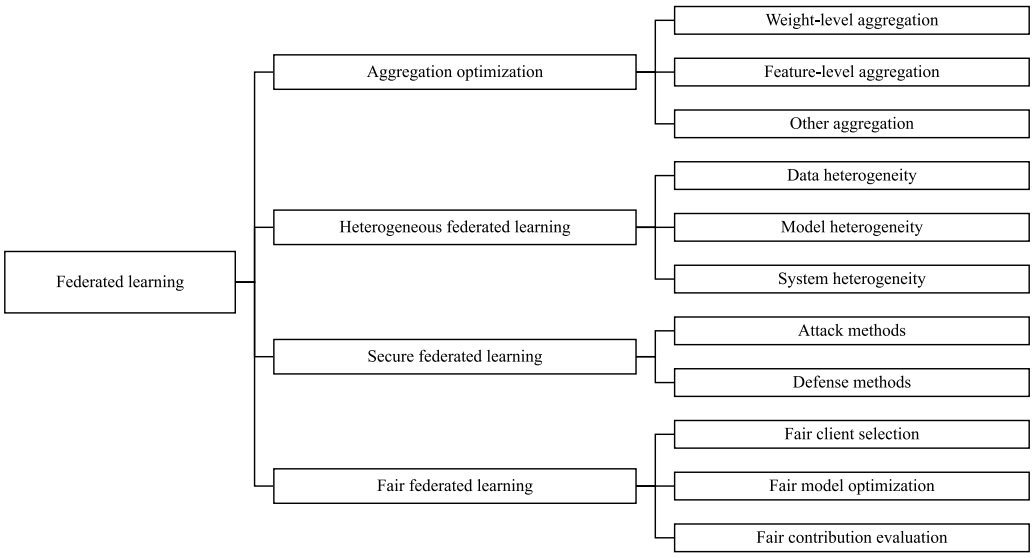

Fig. 3. Our taxonomy of different federated learning methods.

same layer or same filter) may have a large mismatching due to the highly skewed data distribution in each distinctive client/party. Therefore, directly averaging them will degrade the accuracy of the generated global model. To solve the issue, researchers leverage a particular DNN principle, *weight permutation invariance*, which has been mentioned and discussed by recent works [30,98,99]. The key idea of this principle is that the weights in a DNN can be specially shuffled without incurring much accuracy drop. Concretely, suppose l_j and l_{j+1} are the weight of two continuous layers in a DNN model, where the output function can be denoted as

$$O_{j+1} = l_{j+1} l_j I, \tag{4}$$

where I is the input and O_{j+1} is the output of the $j+1_{th}$ layer. Note that for each weight matrix l, it can be further decomposed as follows

$$l = l\mathbf{1} = l\Pi\Pi^T, \tag{5}$$

where Π represents the permutation matrix. In terms of this equation, we can transform Eq. (4) to the following form

$$O_{j+1} = (l_{j+1}\Pi_{j+1}\Pi_{j+1}^T) l_j I = (l_{j+1}\Pi_{j+1})(\Pi_{j+1}^T l_j) I, \tag{6}$$

Based on Eq. (6), we can clearly see that the original layer weight can be losslessly transformed with a pair of well-designed permutation matrices, which we call it *weight permutation invariance*.

In federated learning, traditional aggregation methods fuse local models according to their weight location, which may be sub-optimal since the *weight permutation invariance* principle indicates that we can change the weight value in a specific location while ensuring the same performance. Thus, the location-based aggregation cannot achieve accurate knowledge fusion, leading to the weight mismatching problem.

To address this problem, a large number of federated optimization works attempt to achieve weight-level alignment. For example, Yurochkin et al. [99] developed Probabilistic Federated Neural Matching (PFNM). As shown in Fig. 4, the key idea is to identify subsets of neurons in each local model that matches neurons in other local models and then combine the matched neurons to an improved global model by leveraging Bayesian nonparametric machinery. For single-layer neural matching, they presented a Beta Bernoulli Process [100] based model of MLP weight parameters, where the corresponding neurons in the output layer are used to convert the neurons in each batch and form a cost matrix. Then the matched neurons can be aggregated to generate the final global model. For multilayer neural matching, they extended the single strategy by defining a generative model of deep neural network weights from outputs back to inputs. In this way, they could adopt a

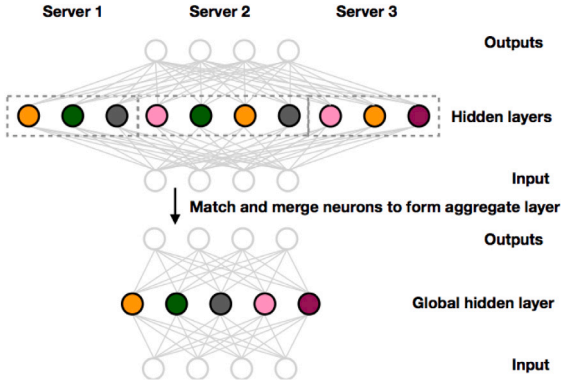

Fig. 4. The illustration of PFNM [99].

greedy inference procedure that first infers the matching of the top layer and then proceeds down the layers of the model.

Unfortunately, PFNM only performs well on simple architectures (e.g. fully connected feedforward networks). For more complex CNNs and LSTMs, it just receives minor improvements over location-based methods (e.g., FedAvg). To further achieve the weight alignment goal, Wang et al. [30] proposed Federated Matched Averaging (FedMA) to effectively align advanced CNNs and LSTMs in a layer-wise manner. The key idea is to search for the best permutation matrices by addressing the following optimization problem

$$\min_{\left\{\pi_{li}^j\right\}} \sum_{i=1}^{L} \sum_{j,l} \min_{\theta_i} \pi_{li}^j c\left(w_{jl}, \theta_i\right) \tag{7}$$

$$\text{s.t.} \sum_i \pi_{li}^j = 1 \forall j, l; \sum_l \pi_{li}^j = 1 \forall i, j, \tag{8}$$

where θ_i is the i_{th} neuron in the current global model, w_{jl} is the output weights processed by permutation matrix π_{li}^j. $c()$ is the distance metric served as determining the similarity between neurons. To solve this optimization problem, unlike PFNM that used heuristic choices, FedMA addressed it by the Hungarian matching algorithm [101].

3.2.2. Feature-level aggregation

Despite effectiveness, the performance of weight-level aggregation/alignment largely depends on the selection of distance metrics, which may not fully reflect the inherent feature information embedded

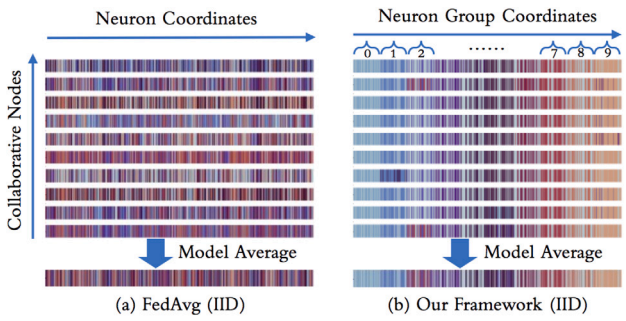

Fig. 5. Comparison between FedAvg and Fed^2 [102].

in the neurons. In addition, the computation cost of the matching process is significantly heavy. To address these limitations, Yu et al. [102] designed a feature-level alignment method, named Fed^2 which is composed of a feature-oriented structure adaptation and a model fusion algorithm. As shown in Fig. 5, compared with traditional weight alignment, Fed^2 paid more attention to the neuron features and then aggregated the corresponding neurons. As a result, similar knowledge can be fused to achieve better performance.

Concretely, the authors developed two schemes to accomplish feature-based federated learning. Fig. 6 shows the pipeline of the proposed Fed^2. The first scheme is *model structure adaptation*, where Fed^2 takes advantage of the group-convolution technique to allocate and learn the distinctive neuron features. Next, a feature paired averaging policy is presented to aggregate different neurons according to the partitioned group features. In this way, Fed^2 enables more accurate feature alignment as well as avoiding the expensive distance-based optimization.

3.2.3. Other aggregation

The aforementioned works mainly focus on alignment, in fact, there are also many other literatures targeting federated aggregation. For example, Yin et al. [103] proposed a robust aggregation method for distributed learning. In the beginning, this work mainly analyzed two robust distributed gradient descent (GD) algorithms, including the coordinate-wise median and the coordinate-wise trimmed mean. They proved statistical error rates for three kinds of population loss functions: strongly convex, non-strongly convex, and smooth non-convex. Furthermore, to reduce the communication cost, the authors designed a median-based distributed algorithm and demonstrate its effectiveness by extensive experiments. Chen et al. [104] further considered the federated learning scenario, and found that heterogeneous data in different nodes will harm the training convergence to some degree. Based on this observation, they developed a novel gradient correction mechanism that can perturb the local gradients with noise. The main advantage of the proposed scheme is that it offers a provable convergence guarantee even when data are non-iid.

Besides, Yurochkin et al. [105] leveraged Bayesian nonparametrics to design a meta-model that can potentially capture the global structure through statistical parameter matching. The authors pointed out that their approach is model-independent and is applicable to a wide range of model types. Chen et al. [106] proposed FEDBE, a novel method to apply bayesian model ensemble into conventional federated learning, aiming at making the aggregation more robust. Motivated by prior work [107], the authors utilized bayesian inference to construct an improved global model. In addition, stochastic weight average (SWA) [108] is also used to further boost the performance.

3.3. Heterogeneous federated learning

Heterogeneous federated learning aims to effectively aggregate models generated from heterogeneous environments. Here the heterogeneous property could be reflected from data, models or device systems. We will dive into each aspect in the next parts.

3.3.1. Data heterogeneity

Data heterogeneity indicates that collaborative clients might be in different situations, resulting in various data distributions. For example, the dog images collected from indoors and outdoors display highly heterogeneous data distribution. To address the issue, the research community borrows the idea from other AI techniques to alleviate the heterogeneity influence, which we list as follows.

Multi-task learning based methods. Multi-task learning enables learning models for multiple related tasks at the same time [109–111]. The core design principle is to capture the relationship among tasks and leverage the relationship to facilitate the learning process. In federated learning, clients with different data distributions could also be considered as a type of multi-task learning, where each task has a distinctive statistical representation [39,41,42,112–114]. For instance, Smith et al. [38] first proposed to combine federated learning and multi-task learning. By a series of concept formulations and theoretical analyses, they suggested multi-task learning is a natural choice to handle the statistical problem in the federated setting. Based on the combination, they further developed a novel approach MOCHA, in order to accomplish their goal. Specifically, the authors formulated the problem as a dual optimization problem as follows

$$\min_{\boldsymbol{\alpha}} \left\{ D(\boldsymbol{\alpha}) := \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^* \left(-\boldsymbol{\alpha}_t^i\right) + \mathcal{R}^*(\mathbf{X}\boldsymbol{\alpha}) \right\}, \tag{9}$$

where l_t^* and $\mathcal{R}^*$ are the conjugate dual functions of l_t and $\mathcal{R}$, respectively. To solve (9), they carefully designed the quadratic approximation of the dual problem to separate computation across the nodes.

Despite federated multi-task learning being demonstrated effective, it has been applied only on convex models. To address the limitation, Corinzia et al. [40] proposed a more general approach, named VIRTUAL, to achieve federation on non-convex models. The key idea is to construct a hierarchical Bayesian network in terms of the central server and the clients, such that the inference could be performed with variational methods. In this way, each client can obtain a task-specific model that benefits from the server model in a transfer learning manner.

Marfoq et al. [44] further proposed to study federated multi-task learning under the flexible assumption that each local data distribution is a mixture of unknown underlying distributions, which is a more challenging and practical scenario. In the beginning, the authors showed the fact that t federated learning is impossible without assumptions on local data distributions. Then they made the flexible assumption and developed Federated Expectation-Maximization to accomplish their objective. Besides, the proposed approach is proven generalizable to unseen clients.

Meta-learning based methods. Meta-learning is commonly considered as learning to learn [115]. Compared with conventional deep learning algorithms that learn specific feature knowledge, meta-learning focus more on learning the learning ability. In the field of federated learning, meta-learning techniques can also be applied to generate a more personalized federation model. Jiang et al. [34] first proposed to combine them, where they believed meta-learning had a number of similarities with the objective of addressing the statistical challenge in FL. Concretely, they developed a novel algorithm to further combine FedAvg [15] and Reptile [116], with two modifications: the first one is to decrease the local learning rate to make training more stable; another is to design a fine-tuning stage based on Reptile with smaller K and Adam as the server optimizer, which could improve the initial model as well as preserving and stabilizing the personalized model.

Khodak et al. [35] built a theoretical framework to further characterize meta-learning methods and apply them into federated learning. They introduced Average Regret-Upper-Bound Analysis (ARUBA), which enables meta-learning to leverage more sophisticated structures. With ARUBA, researchers could improve the results of many ML tasks,

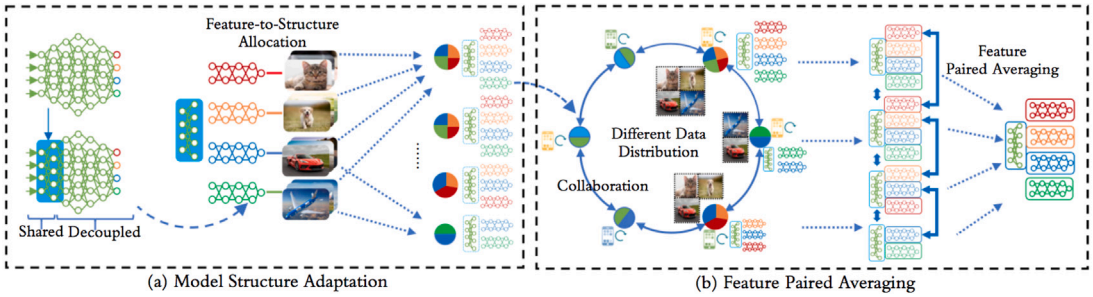

Fig. 6. The illustration of Fed^2 [102].

including adapting to the task-similarity, adapting to dynamic environments, adapting to the inter-task geometry and statistical learning-to-Learn. Towards FL, they improved meta-test-time performance on few-shot learning and effectively added user-personalization to FedAvg.

Fallah et al. [32] aims to find an initial shared model that can be easily fitted to their local data with one or a few steps of gradient descent. They achieved their objective by incorporating Model-Agnostic Meta-Learning (MAML) [117,118] into current FL pipelines. Specifically, the authors proposed a personalized variant of the FedAvg algorithm, named Per-FedAvg, which can be formulated as optimizing the following equation

$$\min_{w\in\mathbb{R}^d} F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i\left(w - \alpha\nabla f_i(w)\right), \tag{10}$$

where n is the number of clients and α is the learning rate. The detailed solution for the optimization problem can be seen in the paper if readers have an interest.

Acar et al. [33] further modified meta-learning to benefit federated learning. As shown in Fig. 7, they proposed PFL, a gradient correction method based on prior works, which explicitly de-biased the meta-model in the distributed heterogeneous data setting to learn a more personalized device model. During the process, convergence guarantees of PFL for strongly convex, convex and nonconvex meta objectives are provided.

Transfer learning based methods. Transfer learning aims to transfer the information learned from a source task to a target task [119]. A large number of research works have been proposed to advance this promising field [120–123]. In federated learning, transferring the knowledge of the federated model to each client model will significantly facilitate the personalization performance under the data heterogeneity environment. Wang et al. [48] proposed to use fine-tuning, a typical transfer learning algorithm to achieve personalization. They first conducted traditional FL to obtain a global model. Then the federated model is regarded as the source model and further retrained using individual client's training cache data. In this way, each client model can acquire and benefit the transferred knowledge, outputting an improved customized model.

Based on the aforementioned work, Yu et al. [49] extended the simple fine-tuning strategy. They investigated how three adaptation mechanisms: fine-tuning, multi-task learning, and knowledge distillation affect the personalization performance. The authors characterized these mechanisms as *local adaptation*. In addition, different model protection techniques such as differential privacy and robust aggregation were applied to further validate the effectiveness of local adaptation. Finally, they used both CV and NLP datasets to demonstrate the superiority and necessity to conduct local adaptation.

Peng et al. [124] considered a new FL+TL scenario beyond fine-tuning. Instead, they paid more attention to domain shift, which means that the labeled data collected by source nodes statistically differ from the target node's unlabeled data. Based on this setting, they proposed the problem of federated domain adaptation and address it by Federated Adversarial Domain Adaptation (FADA). The key idea is

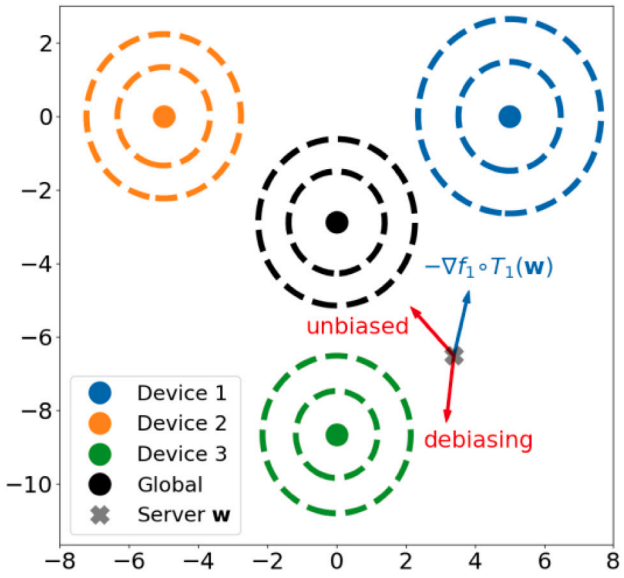

Fig. 7. The illustration of PFL [33].

Fig. 8. The illustration of IFCA [53].

to apply adversarial adaptation and representation disentanglement to FL settings.

Ozkara et al. [51] introduced a quantized and personalized FL algorithm to deal with the data issue. The quantized training process is conducted via knowledge distillation (KD) among clients who have access to heterogeneous data and resources. Besides, they developed an alternating proximal gradient update to address this compressed personalization challenge and analyzed its convergence properties.

Clustering-based methods. Clustering-based FL attempts to tackle the data heterogeneity issue via partitioning clients into different clusters, each of which conforms to a similar distribution. In terms of this key idea, much research effort is made to explore cluster-based FL. Sattler et al. [52] proposed Clustered Federated Learning (CFL), to utilize geometric properties of the FL loss surface, in order to group the client population into clusters with jointly trainable data distributions. It is worth noting that CFL is orthogonal to the current FL communication

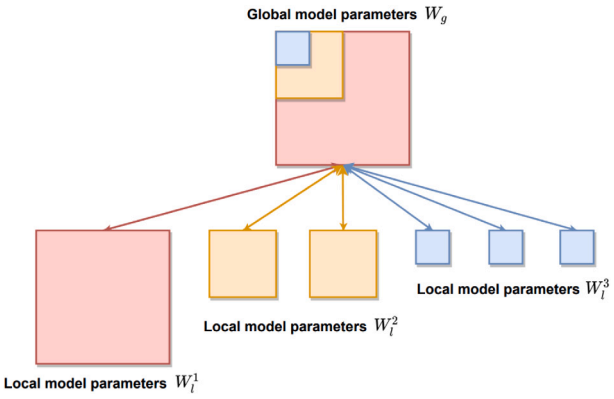

Fig. 9. The illustration of HeteroFL [129].

protocol and can be applied to general non-convex objectives beyond DNNs.

Ghosh et al. [53] proposed the Iterative Federated Clustering Algorithm (IFCA), which alternately estimated the cluster identities of the users and optimized model parameters for the user clusters via gradient descent. As shown in Fig. 8, the server broadcasted models and the workers dynamically identified their cluster memberships and run local updates. This process will continue to operate until the clusters become stable.

To train high-quality cluster models, Ruan et al. [56] suggested FedSoft, which uses proximal updates to restrict client burden by asking a subset of clients to complete just one optimization task per communication round.

Liu et al. [125] proposed a framework to accomplish privacy-preserving federated adaptation. The key idea is to group the clients with similar distribution to collaboratively adapt the federated model, rather than just adapting it with the data in a single device. PFA leveraged the sparsity property of neural networks to generate privacy-preserving representations and used them to efficiently identify clients with similar data distributions. In this way, PFA can conduct an FL process in a group-wise way on the federated model to achieve adaptation.

Besides, in order to achieve clustering without uploading any extra information, Liu et al. [126] further proposed DistFL, targeting at finishing accurate, automated and efficient cluster-based FL in terms of distribution feature. Specifically, they extracted the distribution knowledge from the uploaded model via existing synthesis techniques [127] and then compared them to obtain the clustering results. Finally, they aggregated models in each cluster, getting rid of the influence of heterogeneous data.

3.3.2. Model heterogeneity

Model heterogeneity means that the federated model might not be identical due to the different hardware and data distributions of clients. For example, in order to fit various computation capabilities of clients, we require deploying different model architectures to match each client. On the other hand, NAS techniques [128] have been widely used to search a crafted architecture based on the data in each device, thus leading to the model heterogeneity situation.

To tackle the problem, Li et al. [130] used transfer learning and knowledge distillation to develop a universal framework, which enabled federated learning with uniquely designed models. Lin et al. [131] further proposed a distillation framework for robust federated model fusion and leveraged entropy-reduction to accelerate convergence. Diao et al. [129] designed HeteroFL to address heterogeneous clients equipped with highly different computation and communication capabilities. As shown in Fig. 9, the federation is achieved by aggregating parameters on the same location while unlearning the other non-overlapping area.

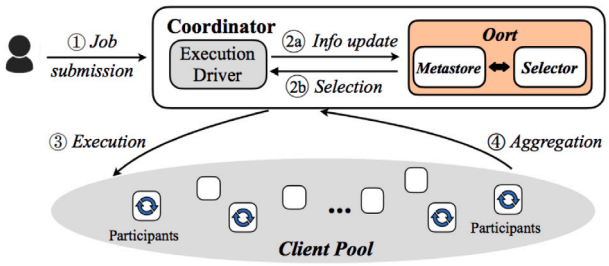

Fig. 10. The illustration of Oort [132].

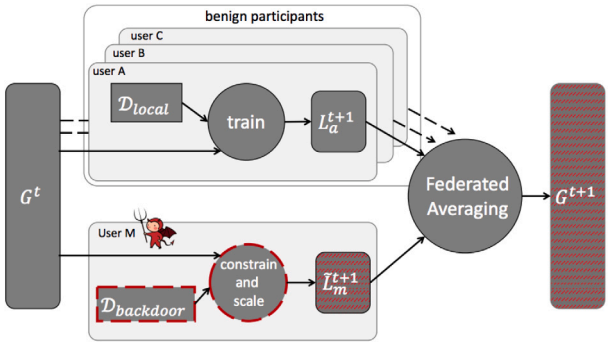

Fig. 11. The illustration of model replacement [59].

3.3.3. System heterogeneity

System heterogeneity is a practical property in FL scenarios because different clients/parties naturally own heterogeneous hardware and memory limitation. Therefore, how to accomplish FL under the condition of system heterogeneity is worth exploring.

A key design for system acceleration is to develop different client selection strategies for avoiding the influence of latency stragglers. Here stragglers refer to the clients with weak computing power and thus could slow down the overall FL process. Lai et al. [132] proposed Oort, a system to improve the performance of federated training and testing with guided participant selection. As shown in Fig. 10, Oort cherry-picked participants according to the tradeoff between statistical and system efficiency. Specifically, they defined "Client Statistical Utility" to measure the importance of each client. Shin et al. [133] developed FedBalancer, a framework to actively select clients' training samples in terms of the more "informative" data. Besides, they introduced an adaptive deadline control scheme to predict the optimal deadline for each round, in order to further speed up global training. Li et al. [134] observed that current client selection was coarse-grained due to their under-exploitation on the clients' data and system heterogeneity. Based on this finding, they proposed PyramidFL, a fine-grained client selection framework to speed up the FL training. The key idea is to not only focus on the divergence of those selected participants but also fully exploited the data and system heterogeneity within selected clients to profile their utility more efficiently. As a result, PyramidFL is able to achieve better performance compared to other baselines.

3.4. Secure federated learning

The original design of federated learning considers the security problem via exchanging parameters while keeping raw data in their own devices. However, recent studies have proved that attackers might steal the privacy information from the uploaded models. Therefore, more rigorous secure FL should be investigated. In the following parts, we will introduce the attack methods and defense methods in FL scenarios.

3.4.1. Attack methods

Backdoor attack. The goal of backdoor attacks is to manipulate a subset of training data by injecting adversarial triggers such that DNN models will output incorrect prediction on the test set when the same trigger occurs. In federated learning, directly applying current backdoor attacks is unsuitable since the aggregation process might destroy the triggers. Bagdasaryan et al. [59] is the first to backdoor federated learning. They achieved their objective by proposing model replacement, which means the backdoor is injected to the joint model rather than raw data. As shown in Fig. 11, the attacker trained a model on the backdoor data using the constrain-and-scale technique. In this way, the averaging function is largely affected by this attack model. Wang et al. [61] proposed edge-case backdoors, which forced a model to misclassify on seemingly easy inputs that are unlikely to be part of the training or testing data. For example, they may exist on the tail of the input distribution. As a result, it is extremely hard to detect them. Xie et al. [62] further developed distributed backdoor attack (DBA) to compromise FL. They mainly took advantage of the distributed nature of FL, decomposing a global trigger pattern into separate local patterns and introducing them into the training set of different adversarial parties respectively. Therefore, DBA is more persistent and stealthy compared to centralized ones. In FL models, backdoors can be inserted, but these backdoors are often not durable, i.e., they do not remain in the model after poisoned updates stop being uploaded. Since training occurs gradually in FL systems, an inserted backdoor may not survive until deployment. Zhang et al. [63] proposed Neurotoxin, which is a simple modification to existing backdoor attacks that target parameters that are not changed in magnitude as much during training.

Gradients attack. Gradients attack targets at reverse some privacy information from gradients. In federated learning, exchanging gradients is a typical step for knowledge update and aggregation. Therefore, gradient attack poses a high risk to the federal participants. Zhu et al. [27] found since training occurs gradually in FL systems, an inserted backdoor may not survive until deployment. that it is possible to obtain the private training data from the publicly shared gradients. They first randomly generated a pair of "dummy" inputs and labels and used them to compute corresponding gradients. Then the gradients were compared to the shared ones and continually optimize the dummy inputs and labels to minimize the distance between them. As a result, the dummy data are close to the original ones and can peek into user privacy. Lam et al. [66] further realized gradients attack from the aggregated model updates/gradients. The authors leveraged the summary information from device analytics and reconstructed the user participant matrix, which invalided the current secure aggregation protocols [135]. Zhu et al. [71] proposed Recursive Gradient Attack on Privacy (R-GAP), an approach to analyze how and when the target gradients can lead to the unique recovery of original data. Concretely, the authors designed a recursive, depth-wise algorithm for recovering training data from the gradient information, which is the first closed-form algorithm that works on both CNN layers and FC layers. Li et al. [70] found that under certain defense settings, generative gradient leakage can still leak private training information. Besides, Truex et al. [136] presented a comprehensive study towards demystifying membership inference attacks, which can be considered as an expanded way to gradient attacks. In their study, they showed that membership inference vulnerability was data-driven and corresponding attack models were largely transferable.

Model poison attack. The goal of poison attacks is to induce the FL model to output the target label specified by the adversary. For example, Tolpegin et al. [137] implemented data poison attack by flipping the labels of training data from one class to another class in the local training epoch to mislead the global model output. Although the aggregation process in FL can mitigate the attack to some extent, when the number of malicious clients becomes large, FL is inevitably poisoned. Fang et al. [138] conducted the first systematic study on local model poisoning attacks to federated learning. Based on this study, they proposed local model poisoning attacks to Byzantine robust federated learning via manipulating the local model parameters on compromised worker devices during the learning process. Besides, the authors further stated two defense strategies and test their performance on the proposed attack.

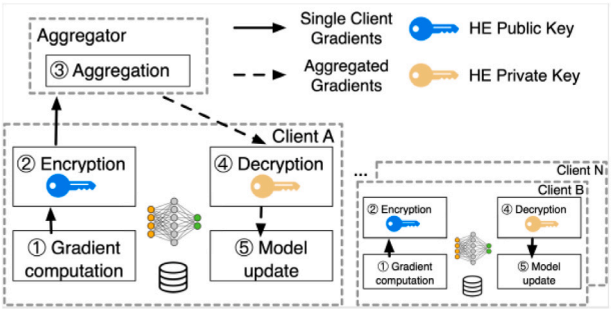

Fig. 12. The illustration of BatchCrypt [85].

3.4.2. Defense methods

DP-based defense. Differential privacy (DP) [24] has been widely used to prevent information leakage. The key idea is to add some noises to obfuscate the original information. As a result, attackers are hard to infer the privacy properties. Federated learning also requires this type of protection since the uploaded model parameters can be easily exploited to extract sensitive information. Wei et al. [77] proposed NbAFL, a framework that applied DP into FL. Specifically, they added noises to parameters of the local model at the client side before aggregation. Besides, the authors theoretically analyzed the convergence property of differentially private FL algorithms and proved the effectiveness of the proposed framework.

Kairouz et al. [80] presented a comprehensive end-to-end system, where they discretized the data and added discrete Gaussian noise before conducting secure aggregation. In addition, the authors provided a novel privacy analysis for sums of discrete Gaussians and carefully analyzed the effects of data quantization and modular summation arithmetic. Experiments demonstrated that their method can achieve comparable performance with 16 bits of precision per value. Agarwal et al. [81] proposed a multi-dimensional Skellam mechanism, where two independent Poisson random variables are used to measure the difference. The authors applied their mechanism to FL and provided a novel algorithm that appropriately discretized the data and used the Skellam mechanism along with modular arithmetic to bound the range of the data and communication costs before secure aggregation. As a result, they could achieve better privacy-accuracy trade-offs in a more efficient manner.

HE-based defense. HE-based FL aims to combine traditional Homomorphic Encryption (HE) and FL in a more suitable way. By applying HE, FL is able to aggregate client models without revealing the information of the concrete model parameters. Therefore, it is impossible to infer user privacy from the model. Hardy et al. [84] proposed to encrypt FL with the homomorphic scheme in the field of privacy-preserving entity resolution and federated logistic regression. They bounded the difference between the empirical loss of their classifier on the true data and showed an improved convergence speed. Besides, their experiments found that even rates for generalization cannot be significantly affected by entity resolution. Liu et al. [139] designed a secure FL framework through leveraging the additive property of partial homomorphic encryption, which effectively avoids the exposure of client models at the server side. Besides, the authors introduced two optimization mechanisms to further enhance efficiency. Zhang et al. [85] proposed BatchCrypt, an efficient homomorphic encryption system for cross-Silo federated learning. As shown in Fig. 12, there

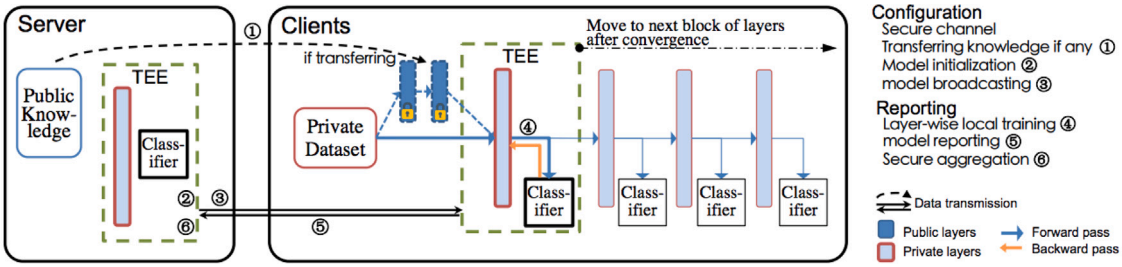

Fig. 13. The illustration of PPFL [86].

exist five typical steps to achieve a cross-silo FL system. In the beginning, the aggregator needed to select a client to generate an HE key-pair and distribute it to others. Then for each iteration, clients conducted local gradient updates and further encrypted them by the public key. These encrypted parameters were uploaded to the server where aggregation happened and the aggregated model is transferred to each client. Finally, the client side decrypted the received information and implemented the local training as the next round. BatchCrypt proposed two novel schemes to further improve efficiency. First, a feasible batch encryption scheme was presented to directly sum up the ciphertexts of two batches. Second, an efficient analytical model dACIQ was presented to choose optimal clipping thresholds with the minimum cumulative error. As a result, BatchCrypt achieved 23×-93× training speedup while reducing the communication overhead by 66×-101×. In addition, Xu and Baracaldo et al. [140] proposed HybridAlpha, an approach for privacy-preserving federated learning employing an SMC protocol based on functional encryption, which is simple, efficient and resilient to participants dropping out.

TEE-based defense. The aforementioned secure FL approaches provide security guarantee mainly from the perspective of software. In real-world scenarios, hardware protection is also widely applied by designing crafted architecture. Trusted Execution Environment (TEE) is a trusted component that establishes an isolated region on the main processor to ensure the confidentiality and integrity of data and programs [141,142]. Compared to traditional encryption schemes such as homomorphic encryption, TEE is more efficient with respect to the computation cost since it only requires some simple operations to connect the trusted and untrusted part in OS. Recently there have been a large number of works targeting at applying TEE to deep/federated learning, in order to achieve protection from hardware level. For example, Mo et al. [87] proposed DarkneTZ that enabled executing DNNs more secure with TEE in an edge device. They partitioned DNNs into a set of non-sensitive layers and sensitive layers, which are respectively processed by TEE or normal OS. Here the partition choice is based on the underlying system's CPU execution time, memory usage, and accurate power consumption of different DNN layers. Besides, the authors developed a threat model to validate DarkneTZ's robustness under the membership inference attack and the results showed that DarkneTZ could defend against this type of attack with negligible performance overhead.

Based on the combination of DNNs and TEE, Mo et al. [86] further attempted to apply TEEs to federated learning. Specifically, they proposed PPFL, a framework that limited privacy leakages in federated learning via implementing local training in TEEs. As shown in Fig. 13, to address the challenge of limited memory size of TEEs, the authors designed a greedy layer-wise training to conduct local updates until convergence. In this way, this approach could support sophisticated settings such as training one or more layers (block) each time, which potentially speed up the training process. Zhang et al. [143] proposed TEESlice, a system to provide a strong security guarantee while maintaining low inference latency with the help of TEEs. Concretely, TEESlice executed the more private model slices on TEEs and others on normal AI accelerators. As a result, TEESlice can achieve more than 10× throughput promotion with the same level of strong security guarantee.

3.5. Fair federated learning

Existing works of federated learning pay more attention to improving learning performance based on the accuracy of the model and the time of learning task completion. However, the interests of the FL clients are often ignored and this may lead to unfairness. The problem of fairness can occur in the whole FL training process, including client selection, model optimization, incentive distribution, and contribution evaluation. The unfairness can have a negative impact on both the FL clients and the FL server, as clients are discouraged to join FL training, and servers are less likely to attract potentially high-quality clients. Recently, to achieve fairness from different angles, various Fairness-Aware Federated Learning (FAFL) approaches have been proposed. In this section, we will discuss recent FAFL methods in detail.

3.5.1. Fair client selection

Unfairness in FL Client Selection mainly consists of three types, over-representation, under-representation, and never-representation. Suppose an FL system prefers to select clients with high performance (such as a faster GPU), and clients with the highest performance may be selected much more than any other clients (i.e., over-representation), while clients with poor performance may be selected just a few times (i.e., under-representation). At the same time, the client with the lowest performance may never be selected (i.e., never-representation). Additionally, due to the heterogeneity among clients, fairness does not indicate giving everyone the same possibility to be selected. It is important to balance the interests of the server and the interests of the clients. If clients from specific groups are oversampled, the global FL model will be partial to their data, so the model's performance will deteriorate [144]. Existing FAFL client selection methods can be partitioned into two categories, considering fairness factors and customization for each client.

(1) Fairness factors. Fairness factors are designed to allow rarely selected clients, such as clients with lower computational abilities or smaller datasets, to join the FL training more frequently. Yang et al. [145] proposed a client selection algorithm based on the Combinatorial Multi-Armed Bandit (CMAB) framework to reduce the class imbalance effect. Inspired by [146], Huang et al. [147] converts the original offline problem to an online Lyapunov optimization problem and uses dynamic queues to quantify the long-term guarantee of the client participation rate. Moreover, Huang introduces a long-term fairness constraint to make sure the average client's long-term chosen rate is above a constant. After [147], Huang et al. [148] improves the performance by replacing dynamic queues to the Exp3 algorithms [149], and the fairness parameter determining the selection possibility in each round can be different. However, these works all design the fairness factor without considering the real-time contribution of individual clients. Song et al. [150] addresses this problem and proposed a client selection policy with fairness constraints based on reputation, using a fairness parameter to balance reputation and the number of successful transmissions.

(2) Client customization. This approach pays attention to customized model settings or customized model procedures. Clients often receive

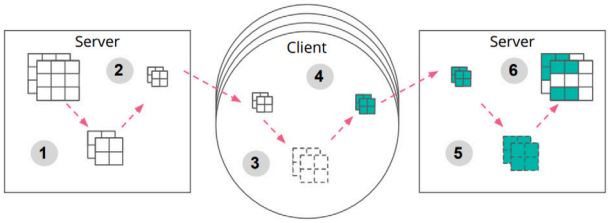

Fig. 14. The summary of the Federated Dropout (FD) training procedure [151].

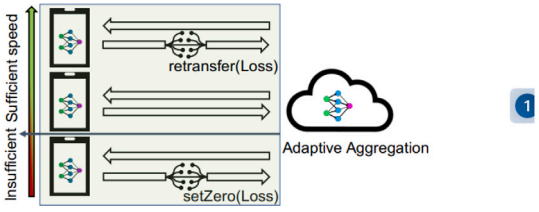

Fig. 15. The illustration of ThrowRightAway (TRA) scheme [155].

the same initial models at the first training round in most current FL paradigms. Therefore, clients with lower capabilities, such as bad network connections, require more time to complete each training round and are likely to be kept out of subsequent rounds, leading to under-presented and never-presented problems. To alleviate this problem, dynamically adapting the FL model framework or the training procedure based on client capabilities is often used.

Caldas et al. [151] proposed Federated Dropout (FD), which distributes sub-models with sizes suitable for each client based on their computational resources. The process of FD is shown in Fig. 14. Although FD diminishes communication and local computation costs largely, it uses dropout operations and treats the neural networks as black-box functions. Bouaciada et al. noticed this problem and proposed Adaptive Federated Dropout (AFD) [152]. AFD keeps an activation score map to generate the best-fit sub-model for each client. FD and AFD both make sure clients with low capabilities could participate in FL training, but they do not provide custom pruned submodels to different clients. To address this limitation, Horvath et al. [153] augmented FD to Ordered Dropout (OD). Different from FD, OD drops neighboring components of the model despite random neurons. OD divides clients with comparable computational capabilities into clusters, and clients in the same cluster apply the same dropout rate. Moreover, OD applies the knowledge distillation method [154] to enhance feature extraction for smaller submodels.

Clients' communication capabilities can also affect client selection. A poor network may cause too much retransmission and lead to extra delays in FL model training, which makes clients with a poorer network less likely to aggregate their model updates into the final model and leads to model bias. To deal with this issue, Zhou et al. [155] proposed ThrowRightAway (TRA), a loss-tolerant FL framework that makes the FL training faster by ignoring few lost packets. As is shown in Fig. 15, at first every participating FL client reports their network conditions to the FL server, and the server divides the clients into two categories: sufficient type and insufficient type. Only the clients in the sufficient type can get a re-transmission request and then re-transmit their loss packets. Apparently, the method can only be effective when the category is accurate.

This method means assigning less work to clients with lower capabilities to make them available to pass threshold-based FL client selection. Li et al. [23] proposed FedProx which allowed each client performed partial training based on its accessible resources. FedProx allows various local epochs, and thus more clients are encouraged to join the training process.

3.5.2. Fair model optimization

In the optimization during FL model training, the model may discriminate against definite preserved groups, or overfit some clients at the expense of others. Recent works dealing with this issue can be approximately divided into two types: (1) objective function-based and (2) gradient-based.

(1) Objective function-based methods: Objective function-based methods focus on the global/local objectives of the FL model, such as minimizing the loss function. Mohri et al. [156] proposed AFL, which aims to prevent the model overfitting any specific client at the expense of others. AFL just optimizes the global model for the target distribution made up of a mixture of clients. However, this method only works for a small number of clients. Zhou et al. [155] proposed q-FFL to diminish the scalability limitation of AFL. q-FFL adds parameter q to reweigh the aggregate loss. To improve the model robustness and maintain good-intent fairness at the same time, Hu et al. [157] proposed fedMGDA+ which optimizes each FL client's loss function respectively and simultaneously. Addressing the same issue, Li et al. [43] proposed Ditto, which improves fairness and robustness at the same time.

While the methods mentioned all pay attention to the accuracy parity notion of fairness, there are also many kinds of research focusing on group fairness. Du et al. [158] proposed AgnosticFair, which incorporates an agnostic fairness constraint. Although it has good accuracy and fairness on unknown testing data distribution, it needs prior knowledge to design the re-weighting function, which limits its application in dynamic systems. Cui et al. [159] proposed FCFL, a multi-objective optimization framework that achieves good-intent fairness and group fairness at the same time. Different from AFL, it minimizes the loss of the client with the worst performance and uses a smooth surrogate maximum function considering all clients. A fairness constraint is also added to calculate the disparities among all clients.

(2) Gradient-based approaches: Here, gradient means the local updated gradient of each client in every local iteration. Wang et al. [160] proposed the federated fair averaging (FedFV) algorithm, which aims to average clients' gradients after mitigating potential conflicts among clients. FedFV detects gradient conflicts through the cosine similarity and modifies both the direction and magnitude of the gradients by iteratively eliminating such conflicts. However, the estimated gradients may be incompatible with the latest updates.

3.5.3. Fair contribution evaluation

Contribution evaluation in FL learning indicates that an FL system can evaluate the contribution of different clients without accessing data from the clients. Many methods designed for non-privacy machine learning environments cannot be applied to FL scenarios directly. A general method is to evaluate each client's model contribution to the aggregated FL model, and a fair evaluation is critical. Unfairness in contribution evaluation may lead to the free-rider issue [161], which implies that clients contribute little but can get similar benefits as the clients who contribute more. In this part we will introduce five types of existing FL contribution evaluation methods with their typical works.

(1) Self-reported information: This method of evaluation contribution is based on clients reporting their information actively. Most works based on this method believe their clients are reliable, which is not always correct in practice. Proposed by Zhang et al. [162], Hierarchically fair federated learning (HFFL) follows the idea of 'contribute more, get more reward', which is proved effective in social psychology [163], game theory [164] and bandwidth allocation [165]. Hence, it is critical to figure out how to evaluate a client's contribution and how much proportion of reward a client should get to ensure fairness. Data Shapley can be used to evaluate contribution in machine learning, but Shapley value is model-dependent [166] and incompatible with FL tasks. As a result, Zhang proposes evaluating contributions based on publicly verifiable factors of clients, such as cost of data collection, data volume, and data quality, to avoid the inconsistency of model-dependent methods. To distribute proportional rewards to

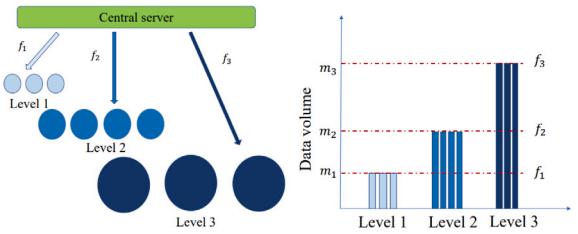

Fig. 16. The illustration of hierarchically fair federated learning (HFFL) [162].

clients, Zhang introduces hierarchically fair federated learning (HFFL), as is shown in Fig. 16. The publicly verifiable factors determined by the clients' consensus about each client are reported to the FL server, and the FL server then uses the information to rate each client, which at the same level are supposed to contribute to the model equally and will get the equal reward.

(2) Individual evaluation: Individual evaluation implies evaluating contribution through performance on specific tasks and pays more attention to individual performance instead of global performance. The method often adopts two assumptions that both the server and the client are reliable and clients with a similar model to others are regarded to supply more contribution, which is not always feasible. To achieve fairness without sacrificing the model performance, Lyu et al. [167] proposed a Collaborative Fair Federated Learning (CFFL) framework based on reputation, which uses a reputation mechanism to achieve collaborative fairness. Lyu definite collaborative fairness as the reward is proportional to the client's contribution. Different standard FL process, CFFL allows clients to receive only the allocated aggregated updates according to their reputations, and the server is in charge of a reputation list which is updated in each communication round relying on the quality of the uploaded gradients of each participant.

(3) Utility game: The utility game [168] refers to a game where each player chooses an available team to maximize their payoffs, while the universal social welfare is the total utility produced by all the teams cumulatively. FL contribution evaluation methods based on utility games have a deep connection with profit-sharing schemes, and there are three diffusely used profit-sharing schemes:

(1) Egalitarian: any part of the utility produced by a team is separated equally between the members.
(2) Marginal gain: the payoff of a player in a team is equal to the team gained when the player joined.
(3) Marginal loss: the payoff of a player in a team is equal to the team will lose if the player leaves.

Among the three types above, the marginal loss scheme is the most commonly adopted. Wang et al. [169] proposed a deletion method to evaluate contributions in horizontal federated learning. This evaluation method consists of removing the instances supplied from one definite party, retraining the model, calculating the difference between the original model and the new model, and using this difference to define the contribution of this party. Wang formulates the influence measure as follows,

$$Influence^{-i} = \frac{1}{n}\sum_{j=1}^{n}\left|\hat{y}_j - \hat{y}_j^{-i}\right|, \tag{11}$$

where n is the size of the dataset, $\hat{y}_j$ is the model trained on all data prediction on jth instance, and $\hat{y}_j^{-i}$ s the model trained without the ith instance prediction on jth instance.

Then Huang defines a party's contribution as the total influence of all instances it possesses.

$$Influence^{-D} = \sum_{i\in D} Influence^{-i}, \tag{12}$$

For vertical horizontal learning, Huang uses shapley value which will be introduced in the next part.

(4) Shapley value: Shapley value (SV) was first introduced in cooperative game theory [170]. Different from marginal loss, SV-based FL contribution evaluation approaches can reflect the contribution of a client's own data, in spite of its joining order, and can produce a fairer evaluation. However, SV's computational complexity is $O(2^n)$, so many approaches have been proposed to improve efficiency.

3.6. Summary

We want to highlight that the main focus of our paper is on research focusing on device FL, which means that our federated clients are closely tied to specific user device attributes. For instance, Abbe et al. [171] have extensively explored how to reduce the running time of distributed learning tasks. Achieving this requires a comprehensive consideration of the device's computation load, straggler tolerance, and communication cost. In our survey, we primarily target four perspectives: aggregation optimization, heterogeneous federated learning, secure federated learning, and fair federated learning. We categorize and describe various works in these areas, considering both the hardware capabilities of devices and user preferences. We believe that summarizing and explaining device-based FL will be highly beneficial for practical FL deployment and large-scale industrial applications.

4. Prevalent frameworks of federated learning

In this section, we will introduce several prevalent frameworks of federated learning, including FedLab, Flower, FedML, FATE, and FedScale.

FedLab. Since most FL schemes follow the same basic steps and just a few changes in some steps are needed in different scenarios, Zeng et al. [172] proposed FedLab [173], which is designed flexible and customizable, offers essential functional modules, and has highly customizable interfaces. Two main roles in FL settings are provided: Server and Client, and both of them are made up of two components, NetworkManager and ParameterServerHandler/Trainer. The design focuses more on communication efficiency and FL algorithm effectiveness. To support methods improving Communication Efficiency, FedLab uses tensor-based communication, supports customizable communication agreement, and implements both Synchronous and Asynchronous communication patterns according to Federated Optimization algorithms. For Optimization Effectiveness, FedLab applies a "high-cohesion and low-coupling" optimization module which provides aggregation and data partition methods. Additionally, FedLab can be used in various scenarios, such as Standalone, Cross-process and Hierarchical FL simulation.

Flower. Due to the lack of frameworks that are able to support scalably executing FL methods on mobile and edge devices, Beutel et al. [174] proposed Flower [175], which can run large-scale FL experiments on different FL device scenarios. Flower makes it possible to smoothly transition from experimental research to system research on a large group of real edge devices. Designed to be scalable, client-agnostic, communication-agnostic, privacy-agnostic, and flexible, Flower has extensive implementations, such as communication stack, serialization, ClientProxy, and Virtual Client Engine(VCE).

FedML. Proposed by He et al. [176], FedML [177] aims to solve the lack of support for diverse FL computing paradigms, support of diverse FL configurations, and standardized FL algorithm implementations and benchmarks. FedML library is mainly made up of high-level API FedML-API and low-level API FedML-core. To support FL on real-world hardware platforms, FedML offers on-device FL testbeds called FedML-Mobile and FedML-IoT which are built upon real-world hardware platforms. FedML programming interface allows worker/client-oriented programming, message definition beyond gradient and model, topology

management, trainer and coordinator, privacy, security, and robustness, so users can just pay attention to algorithms implementations and ignore the backend details.

FATE. Since most open-sourced frameworks are research-oriented and lack the implementation on industry, Liu et al. [178] proposed FATE(Federated AI Technology Enabler) [179], which is the first production-oriented platform. Built on FederatedML, FATE provides Private Set Intersection(PSI), and uses distributed computation framework Eggroll to improve computation efficiency. FATE provides three main components, scheduling system FATE-Flow, visualization tool FATE-Board, and high-performance inference platform FATE-Serving. In addition, kinds of deployments are supported, including building FATE on top of Kubernetes in data centers through KubeFATE, manual or docker deployments on Mac and Linux, and cross-cloud deployment and management through FATE-cloud.

FedScale. Lai et al. [180] proposed FedScale [181], which contains many realistic FL datasets for different tasks, and FedScale Runtime which is an automated evaluation platform aiming to simplify and standardize FL evaluation in more realistic environments. The raw data of FedScale datasets are collected from various sources, processed into consistent formats, sorted into different FL use cases and packed into standardized APIs for users to easily use in other frameworks. The evaluation platform, FedScale Runtime, is equipped with both mobile and cluster backends to enable both on-device FL evaluation on smartphones, and FL evaluations in real deployments and in-cluster simulations.

Frameworks in production use. Here we list several frameworks in production use that are not open source but worthy a mention for completeness: (1) Microsoft used Azure Machine Learning–based federated learning approach to built Text Analytics for Health to empower healthcare organizations to leverage their clinical data, to provide better care to patients [182]. (2) Amazon designed Amazon SageMaker SDK to support deploying federated learning models on AWS without needing to write custom code [183]. (3) Nvidia developed a federated learning Library for GPUs, providing developers with all of the tools required for training and deploying FL models on Nvidia GPUs [184].

5. Discussion

This section summarizes some limitations of current FL approaches and discusses possible future directions.

Dynamic federated learning. Current federated learning approaches assume that data in each client are stable and unchanged. However, in real-world scenarios, clients may be in an ever-changing environment, where the local data are continuously observed and processed by sensors. Under this condition, directly conducting conventional training and aggregation will suffer from the catastrophic forgetting problem, which indicates that the prior knowledge learned by the model might be forgotten as new data arrive. Incremental learning [185–188] is a hot research topic to address the issue, targeting at learning new knowledge while maintaining the ability to recognize previous ones. In the future, how to effectively combine federated learning and incremental learning is worth exploring.

Decentralized federated learning. A central server is of vital importance to traditional federated learning since aggregation needs to be conducted in this side. Considering that the third-party server may not be honest, uploading parameters or gradients to it potentially exists security risks. Therefore, it is necessary to achieve federated learning without a server involved. Although He et al. [189] has made a preliminary attempt to decentralized FL, they only target logistic regression and the experiments are insufficient. How to accomplish general decentralized FL still remains an open problem.

Scalability of federated learning. Recent FL papers paid more attention to designing new algorithms to improve FL performance under different conditions. However, they ignore the scalability property, which determines whether we could operate large-scale FL. In many cooperation scenarios, there might be a huge number of parties and we should provide guidance to the cooperation improvement as the number of participants increases. In a word, FL scalability deserves future investigation.

Unified benchmark. Although a large number of datasets have been used for evaluating the performance of FL, there is still a lack of a unified benchmark to align the results for a fair comparison. On one hand, in order to achieve different federated goals (e.g., personalization, robustness), researchers use different datasets to test the performance. On the other hand, two typical types of FL, horizontal FL and vertical FL, also apply distinctive datasets to demonstrate the performance of different FL types. Thus a unified benchmark will definitely benefit the FL community.

6. Conclusion

Federated learning has gained more and more attention due to its ability of collaboratively generating a global model without leaking sensitive information. Recent surveys have summarized many related works devoted to offering a comprehensive understanding to developers and readers in this community. However, most of them focus on a specific aspect of FL or fail to catch the latest progress of this hot research topic. This paper provides a systematic survey, which investigates recent development on federated learning. By analyzing the pipeline and challenges of FL, we propose a taxonomy with different FL aspects involved. In addition, we also explore some practical FL frameworks and characterize their features. Finally, some limitations and future direction are concluded in order to promote the evolution of the FL community.

CRediT authorship contribution statement

Bingyan Liu: Investigation, Writing – original draft, Writing – review & editing. **Nuoyan Lv:** Investigation, Writing – review & editing. **Yuanchun Guo:** Investigation, Software, Writing – review & editing. **Yawen Li:** Supervision.

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

No data was used for the research described in the article.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (62302054).

References

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[2] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[3] Bingyan Liu, Yao Guo, Xiangqun Chen, WealthAdapt: A general network adaptation framework for small data tasks, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 2179–2187.
[4] Bingyan Liu, Yifeng Cai, Yao Guo, Xiangqun Chen, TransTailor: Pruning the Pre-Trained Model for Improved Transfer Learning, AAAI, 2021.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[7] Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, Lei Li, Vocabulary learning via optimal transport for neural machine translation, in: Chengqing Zong, Fei Xia, Wenjie Li, Roberto Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Association for Computational Linguistics, 2021, pp. 7361–7373.

[8] Chong Chen, Fei Sun, Min Zhang, Bolin Ding, Recommendation unlearning, in: Proceedings of the ACM Web Conference 2022, WWW '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 2768–2777.

[9] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, Shaoping Ma, Efficient non-sampling factorization machines for optimal context-aware recommendation, in: Proceedings of the Web Conference 2020, 2020, pp. 2400–2410.

[10] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, Shaoping Ma, Jointly non-sampling learning for knowledge graph enhanced recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 189–198.

[11] Bingyan Liu, Yuanchun Li, Yunxin Liu, Yao Guo, Xiangqun Chen, Pmc: A privacy-preserving deep learning model customization framework for edge computing, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (4) (2020) 1–25.

[12] Yuanchun Li, Ziqi Zhang, Bingyan Liu, Ziyue Yang, Yunxin Liu, ModelDiff: testing-based DNN similarity comparison for model reuse detection, in: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, 2021, pp. 139–151.

[13] Yingxia Shao, Bin Cui, Lei Chen, Lin Ma, Junjie Yao, Ning Xu, Parallel subgraph listing in a large-scale graph, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 2014, pp. 625–636.

[14] Jan Philipp Albrecht, How the GDPR will change the world, Eur. Data Prot. L. Rev. 2 (2016) 287.

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Aguera y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[16] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, Daniel Ramage, Federated learning for mobile keyboard prediction, 2018, arXiv preprint arXiv:1811.03604.

[17] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, Françoise Beaufays, Applied federated learning: Improving google keyboard query suggestions, 2018, arXiv preprint arXiv:1812.02903.

[18] Qiang Yang, Yang Liu, Tianjian Chen, Yongxin Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.

[19] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, Bingsheng He, A survey on federated learning systems: vision, hype and reality for data privacy and protection, IEEE Trans. Knowl. Data Eng. (2021).

[20] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, Chunyan Miao, Federated learning in mobile edge networks: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2031–2063.

[21] Lingjuan Lyu, Han Yu, Qiang Yang, Threats to federated learning: A survey, 2020, arXiv preprint arXiv:2003.02133.

[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., Advances and open problems in federated learning, Found. Trends® Mach. Learn. 14 (1–2) (2021) 1–210.

[23] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, Virginia Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.

[24] Cynthia Dwork, Differential privacy: A survey of results, in: International Conference on Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.

[25] Craig Gentry, Fully homomorphic encryption using ideal lattices, in: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, 2009, pp. 169–178.

[26] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, Vikas Chandra, Federated learning with non-iid data, 2018, arXiv preprint arXiv:1806.00582.

[27] Ligeng Zhu, Zhijian Liu, Song Han, Deep leakage from gradients, Adv. Neural Inf. Process. Syst. 32 (2019).

[28] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, Richard Zemel, Fairness through awareness, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 214–226.

[29] Lokesh Nagalapatti, Ramasuri Narayanam, Game of gradients: Mitigating irrelevant clients in federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 9046–9054.

[30] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, Yasaman Khazaeni, Federated learning with matched averaging, in: International Conference on Learning Representations (ICLR), 2020.

[31] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, Virginia Smith, Federated optimization in heterogeneous networks, in: I. Dhillon, D. Papailiopoulos, V. Sze (Eds.), Proceedings of Machine Learning and Systems, Vol. 2, 2020, pp. 429–450.

[32] Alireza Fallah, Aryan Mokhtari, Asuman Ozdaglar, Personalized federated learning: A meta-learning approach, 2020, arXiv preprint arXiv:2002.07948.

[33] Durmus Alp Emre Acar, Yue Zhao, Ruizhao Zhu, Ramon Matas, Matthew Mattina, Paul Whatmough, Venkatesh Saligrama, Debiasing model updates for improving personalized federated training, in: International Conference on Machine Learning, PMLR, 2021, pp. 21–31.

[34] Yihan Jiang, Jakub Konečnỳ, Keith Rush, Sreeram Kannan, Improving federated learning personalization via model agnostic meta learning, 2019, arXiv preprint arXiv:1909.12488.

[35] Mikhail Khodak, Maria-Florina F. Balcan, Ameet S. Talwalkar, Adaptive gradient-based meta-learning methods, Adv. Neural Inf. Process. Syst. 32 (2019).

[36] Wenbo Zheng, Lan Yan, Chao Gou, Fei-Yue Wang, Federated meta-learning for fraudulent credit card detection, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4654–4660.

[37] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, Xiuqiang He, Federated meta-learning with fast convergence and efficient communication, 2018, arXiv preprint arXiv:1802.07876.

[38] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, Ameet S Talwalkar, Federated multi-task learning, Adv. Neural Inf. Process. Syst. 30 (2017).

[39] Paul Vanhaesebrouck, Aurélien Bellet, Marc Tommasi, Decentralized collaborative learning of personalized models over networks, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 509–517.

[40] Luca Corinzia, Ami Beuret, Joachim M. Buhmann, Variational federated multi-task learning, 2019, arXiv preprint arXiv:1906.06268.

[41] Valentina Zantedeschi, Aurélien Bellet, Marc Tommasi, Fully decentralized joint learning of personalized models and collaboration graphs, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 864–874.

[42] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, Yong Zhang, Personalized cross-silo federated learning on non-IID data, in: AAAI, 2021, pp. 7865–7873.

[43] Tian Li, Shengyuan Hu, Ahmad Beirami, Virginia Smith, Ditto: Fair and robust federated learning through personalization, in: International Conference on Machine Learning, PMLR, 2021, pp. 6357–6368.

[44] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, Richard Vidal, Federated multi-task learning under a mixture of distributions, Adv. Neural Inf. Process. Syst. 34 (2021) 15434–15447.

[45] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, Salman Avestimehr, Spreadgnn: Serverless multi-task federated learning for graph neural networks, 2021, arXiv preprint arXiv:2106.02743.

[46] Chendi Zhou, Ji Liu, Juncheng Jia, Jingbo Zhou, Yang Zhou, Huaiyu Dai, Dejing Dou, Efficient device scheduling with multi-job federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 9971–9979.

[47] Jiayi Chen, Aidong Zhang, FedMSplit: Correlation-adaptive federated multi-task learning across multimodal split networks, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 87–96.

[48] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, Daniel Ramage, Federated evaluation of on-device personalization, 2019, arXiv preprint arXiv:1910.10252.

[49] Tao Yu, Eugene Bagdasaryan, Vitaly Shmatikov, Salvaging federated learning by local adaptation, 2020, arXiv preprint arXiv:2002.04758.

[50] Daniel Peterson, Pallika Kanani, Virendra J. Marathe, Private federated learning with domain adaptation, 2019, arXiv preprint arXiv:1912.06733.

[51] Kaan Ozkara, Navjot Singh, Deepesh Data, Suhas Diggavi, QuPeD: Quantized personalization via distillation with applications to federated learning, Adv. Neural Inf. Process. Syst. 34 (2021) 3622–3634.

[52] Felix Sattler, Klaus-Robert Müller, Wojciech Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE Trans. Neural Netw. Learn. Syst. 32 (8) (2020) 3710–3722.

[53] Avishek Ghosh, Jichan Chung, Dong Yin, Kannan Ramchandran, An efficient framework for clustered federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 19586–19597.

[54] Avishek Ghosh, Justin Hong, Dong Yin, Kannan Ramchandran, Robust federated learning in a heterogeneous environment, 2019, arXiv preprint arXiv:1906.06629.

[55] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, Jose M Alvarez, Personalized federated learning with first order model optimization, 2020, arXiv preprint arXiv:2012.08565.

[56] Yichen Ruan, Carlee Joe-Wong, Fedsoft: Soft clustered federated learning with proximal local updating, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 8124–8131.

[57] Ekdeep Singh Lubana, Chi Ian Tang, Fahim Kawsar, Robert P Dick, Akhil Mathur, Orchestra: Unsupervised federated learning via globally consistent clustering, 2022, arXiv preprint arXiv:2205.11506.

[58] Bingyan Liu, Yifeng Cai, Hongzhe Bi, Ziqi Zhang, Ding Li, Yao Guo, Xiangqun Chen, Beyond fine-tuning: efficient and effective fed-tuning for mobile/web users, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 2863–2873.

[59] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, Vitaly Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2938–2948.

[60] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, H Brendan McMahan, Can you really backdoor federated learning? 2019, arXiv preprint arXiv:1911.07963.

[61] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, Dimitris Papailiopoulos, Attack of the tails: Yes, you really can backdoor federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 16070–16084.

[62] Chulin Xie, Keli Huang, Pin-Yu Chen, Bo Li, Dba: Distributed backdoor attacks against federated learning, in: International Conference on Learning Representations, 2020.

[63] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, Joseph Gonzalez, Neurotoxin: Durable backdoors in federated learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 26429–26446.

[64] Chulin Xie, Minghao Chen, Pin-Yu Chen, Bo Li, Crfl: Certifiably robust federated learning against backdoor attacks, in: International Conference on Machine Learning, PMLR, 2021, pp. 11372–11382.

[65] Mustafa Safa Ozdayi, Murat Kantarcioglu, Yulia R. Gel, Defending against backdoors in federated learning with robust learning rate, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 9268–9276.

[66] Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, Michael Mitzenmacher, Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix, in: International Conference on Machine Learning, PMLR, 2021, pp. 5959–5968.

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

[68] Bo Zhao, Konda Reddy Mopuri, Hakan Bilen, Idlg: Improved deep leakage from gradients, 2020, arXiv preprint arXiv:2001.02610.

[69] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, Pavlo Molchanov, See through gradients: Image batch recovery via gradinversion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16337–16346.

[70] Zhuohang Li, Jiaxin Zhang, Luyang Liu, Jian Liu, Auditing privacy defenses in federated learning via generative gradient leakage, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10132–10142.

[71] Junyi Zhu, Matthew Blaschko, R-gap: Recursive gradient attack on privacy, ICLR (2021).

[72] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, Michael Moeller, Inverting gradients-how easy is it to break privacy in federated learning? Adv. Neural Inf. Process. Syst. 33 (2020) 16937–16947.

[73] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, Seraphin Calo, Analyzing federated learning through an adversarial lens, in: International Conference on Machine Learning, PMLR, 2019, pp. 634–643.

[74] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, Hai Li, Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective, Adv. Neural Inf. Process. Syst. 34 (2021) 12613–12624.

[75] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, SparseFed: Mitigating model poisoning attacks in federated learning with sparsification, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 7587–7624.

[76] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, Xing Xie, FedAttack: Effective and covert poisoning attack on federated recommendation via hard sampling, 2022, arXiv preprint arXiv:2202.04975.

[77] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, H Vincent Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3454–3469.

[78] Robin C. Geyer, Tassilo Klein, Moin Nabi, Differentially private federated learning: A client level perspective, 2017, arXiv preprint arXiv:1712.07557.

[79] H Brendan McMahan, Daniel Ramage, Kunal Talwar, Li Zhang, Learning differentially private recurrent language models, in: International Conference on Learning Representations, 2018.

[80] Peter Kairouz, Ziyu Liu, Thomas Steinke, The distributed discrete gaussian mechanism for federated learning with secure aggregation, in: International Conference on Machine Learning, PMLR, 2021, pp. 5201–5212.

[81] Naman Agarwal, Peter Kairouz, Ziyu Liu, The skellam mechanism for differentially private federated learning, Adv. Neural Inf. Process. Syst. 34 (2021) 5052–5064.

[82] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Steven Wu, Jinfeng Yi, Understanding clipping for federated learning: Convergence and client-level differential privacy, in: International Conference on Machine Learning, PMLR, 2022, pp. 26048–26067.

[83] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, Ananda Theertha Suresh, Shuffled model of differential privacy in federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 2521–2529.

[84] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, Brian Thorne, Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption, 2017, arXiv preprint arXiv:1711.10677.

[85] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, Yang Liu, {*BatchCrypt*}: Efficient homomorphic encryption for {*Cross* − *Silo*} federated learning, in: 2020 USENIX Annual Technical Conference (USENIX ATC 20), 2020, pp. 493–506.

[86] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, Nicolas Kourtellis, PPFL: privacy-preserving federated learning with trusted execution environments, in: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, 2021, pp. 94–108.

[87] Fan Mo, Ali Shahin Shamsabadi, Kleomenis Katevas, Soteris Demetriou, Ilias Leontiadis, Andrea Cavallaro, Hamed Haddadi, DarkneTZ: towards model privacy at the edge using trusted execution environments, in: Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services, 2020, pp. 161–174.

[88] Gilad Baruch, Moran Baruch, Yoav Goldberg, A little is enough: Circumventing defenses for distributed learning, Adv. Neural Inf. Process. Syst. 32 (2019).

[89] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, Sanjeev Arora, Evaluating gradient inversion attacks and defenses in federated learning, Adv. Neural Inf. Process. Syst. 34 (2021) 7232–7241.

[90] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, Xiaowen Chu, Virtual homogeneity learning: Defending against data heterogeneity in federated learning, 2022, arXiv preprint arXiv:2206.02465.

[91] Pranay Sharma, Rohan Panda, Gauri Joshi, Pramod Varshney, Federated minimax optimization: Improved convergence analyses and algorithms, in: International Conference on Machine Learning, PMLR, 2022, pp. 19683–19730.

[92] Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, Samet Oymak, FEDNEST: Federated bilevel, minimax, and compositional optimization, 2022, arXiv preprint arXiv:2205.02215.

[93] Zhiyuan Zhao, Gauri Joshi, A dynamic reweighting strategy for fair federated learning, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2022, pp. 8772–8776.

[94] David Enthoven, Zaid Al-Ars, Fidel: Reconstructing private training samples from weight updates in federated learning, 2021, arXiv preprint arXiv:2101.00159.

[95] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[96] Alex Krizhevsky, Geoffrey Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Toronto, ON, Canada, 2009.

[97] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, Zhihua Zhang, On the convergence of fedavg on non-iid data, in: International Conference on Learning Representations (ICLR), 2020.

[98] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, Yasaman Khazaeni, Probabilistic federated neural matching, 2018.

[99] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, Yasaman Khazaeni, Bayesian nonparametric federated learning of neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7252–7261.

[100] Yuchen Zhang, John Duchi, Michael I Jordan, Martin J Wainwright, Information-theoretic lower bounds for distributed statistical estimation with communication constraints, Adv. Neural Inf. Process. Syst. 26 (2013).

[101] Harold W. Kuhn, The Hungarian method for the assignment problem, Nav. Res. Logist. Q. 2 (1–2) (1955) 83–97.

[102] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, Xiang Chen, Fed2: Feature-aligned federated learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2066–2074.

[103] Dong Yin, Yudong Chen, Ramchandran Kannan, Peter Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: International Conference on Machine Learning, PMLR, 2018, pp. 5650–5659.

[104] Xiangyi Chen, Tiancong Chen, Haoran Sun, Steven Z Wu, Mingyi Hong, Distributed training with heterogeneous data: Bridging median-and mean-based algorithms, Adv. Neural Inf. Process. Syst. 33 (2020) 21616–21626.

[105] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, Statistical model aggregation via parameter matching, Adv. Neural Inf. Process. Syst. 32 (2019).

[106] Hong-You Chen, Wei-Lun Chao, Fedbe: Making bayesian model ensemble applicable to federated learning, ICLR (2020).

[107] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, Andrew Gordon Wilson, A simple baseline for bayesian uncertainty in deep learning, Adv. Neural Inf. Process. Syst. 32 (2019).

[108] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, Andrew Gordon Wilson, Averaging weights leads to wider optima and better generalization, 2018, arXiv preprint arXiv:1803.05407.

[109] Sylvestre-Alvise Rebuffi, Hakan Bilen, Andrea Vedaldi, Learning multiple visual domains with residual adapters, in: Advances in Neural Information Processing Systems, 2017, pp. 506–516.

[110] Hakan Bilen, Andrea Vedaldi, Universal representations: The missing link between faces, text, planktons, and cat breeds, 2017, arXiv preprint arXiv:1701.07275.

[111] Arun Mallya, Svetlana Lazebnik, Packnet: Adding multiple tasks to a single network by iterative pruning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7765–7773.

[112] Canh T. Dinh, Nguyen Tran, Josh Nguyen, Personalized federated learning with moreau envelopes, Adv. Neural Inf. Process. Syst. 33 (2020) 21394–21405.

[113] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, Peter Richtárik, Lower bounds and optimal algorithms for personalized federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 2304–2315.

[114] Filip Hanzely, Peter Richtárik, Federated learning of a mixture of global and local models, 2020, arXiv preprint arXiv:2002.05516.

[115] Sebastian Thrun, Lorien Pratt, Springer Science & Business Media, 2012.

[116] Alex Nichol, John Schulman, Reptile: a scalable metalearning algorithm, 2018, p. 4, arXiv preprint arXiv:1803.02999, 2 (3).

[117] Chelsea Finn, Pieter Abbeel, Sergey Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[118] Chelsea Finn, Kelvin Xu, Sergey Levine, Probabilistic model-agnostic meta-learning, Adv. Neural Inf. Process. Syst. 31 (2018).

[119] Sinno Jialin Pan, Qiang Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.

[120] Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, How transferable are features in deep neural networks? in: Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.

[121] Xuhong Li, Yves Grandvalet, Franck Davoine, Explicit inductive bias for transfer learning with convolutional networks, in: International Conference on Machine Learning, 2018, pp. 2825–2834.

[122] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, Jun Huan, Delta: Deep learning transfer using feature map with attention for convolutional networks, in: International Conference on Learning Representations (ICLR), 2019.

[123] Ziqi Zhang, Yuanchun Li, Jindong Wang, Bingyan Liu, Ding Li, Yao Guo, Xiangqun Chen, Yunxin Liu, ReMoS: Reducing defect inheritance in transfer learning via relevant model slicing, in: 2022 IEEE/ACM 44th International Conference on Software Engineering, ICSE, IEEE, 2022, pp. 1856–1868.

[124] Xingchao Peng, Zijun Huang, Yizhe Zhu, Kate Saenko, Federated adversarial domain adaptation, ICLR (2020).

[125] Bingyan Liu, Yao Guo, Xiangqun Chen, PFA: Privacy-preserving federated adaptation for effective model personalization, in: Proceedings of the Web Conference 2021, 2021, pp. 923–934.

[126] Bingyan Liu, Yifeng Cai, Ziqi Zhang, Yuanchun Li, Leye Wang, Ding Li, Yao Guo, Xiangqun Chen, DistFL: Distribution-aware federated learning for mobile scenarios, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 5 (4) (2021) 1–26.

[127] Aravindh Mahendran, Andrea Vedaldi, Understanding deep image representations by inverting them, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196.

[128] Barret Zoph, Quoc V. Le, Neural architecture search with reinforcement learning, 2016, arXiv preprint arXiv:1611.01578.

[129] Enmao Diao, Jie Ding, Vahid Tarokh, HeteroFL: Computation and communication efficient federated learning for heterogeneous clients, ICLR (2021).

[130] Daliang Li, Junpu Wang, Fedmd: Heterogenous federated learning via model distillation, 2019, arXiv preprint arXiv:1910.03581.

[131] Tao Lin, Lingjing Kong, Sebastian U. Stich, Martin Jaggi, Ensemble distillation for robust model fusion in federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 2351–2363.

[132] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, Mosharaf Chowdhury, Oort: Efficient federated learning via guided participant selection, in: 15th $\{USENIX\}$ Symposium on Operating Systems Design and Implementation ($\{OSDI\}$ 21), 2021, pp. 19–35.

[133] Jaemin Shin, Yuanchun Li, Yunxin Liu, Sung-Ju Lee, FedBalancer: Data and pace control for efficient federated learning on heterogeneous clients, MobiSys (2022).

[134] Chenning Li, Xiao Zeng, Mi Zhang, Zhichao Cao, PyramidFL: A fine-grained client selection framework for efficient federated learning, Mobicom (2022).

[135] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, Karn 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.

[136] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, Wenqi Wei, Demystifying membership inference attacks in machine learning as a service, IEEE Trans. Serv. Comput. 14 (6) (2019) 2073–2089.

[137] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, Ling Liu, Data poisoning attacks against federated learning systems, in: European Symposium on Research in Computer Security, Springer, 2020, pp. 480–501.

[138] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, Neil Gong, Local model poisoning attacks to $\{Byzantine-Robust\}$ federated learning, in: 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1605–1622.

[139] Changchang Liu, Supriyo Chakraborty, Dinesh Verma, Secure model fusion for distributed learning using partial homomorphic encryption, in: Policy-Based Autonomic Data Governance, Springer, 2019, pp. 154–179.

[140] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, Heiko Ludwig, Hybridalpha: An efficient approach for privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 13–23.

[141] Architecure ARM, Security Technology Building a Secure System Using Trustzone Technology, (white Paper), ARM Limited, 2009.

[142] Victor Costan, Srinivas Devadas, Intel SGX explained, Cryptol. ePrint Arch. (2016).

[143] Ziqi Zhang, Lucien KL Ng, Bingyan Liu, Yifeng Cai, Ding Li, Yao Guo, Xiangqun Chen, TEESlice: slicing DNN models for secure and efficient deployment, in: Proceedings of the 2nd ACM International Workshop on AI and Software Testing/Analysis, 2022, pp. 1–8.

[144] Yae Jee Cho, Jianyu Wang, Gauri Joshi, Client selection in federated learning: Convergence analysis and power-of-choice selection strategies, 2020, arXiv preprint arXiv:2010.01243.

[145] Miao Yang, Ximin Wang, Hongbin Zhu, Haifeng Wang, Hua Qian, Federated learning with class imbalance reduction, in: 2021 29th European Signal Processing Conference, EUSIPCO, IEEE, 2021, pp. 2174–2178.

[146] Fengjiao Li, Jia Liu, Bo Ji, Combinatorial sleeping bandits with fairness constraints, IEEE Trans. Netw. Sci. Eng. 7 (3) (2019) 1799–1813.

[147] Tiansheng Huang, Weiwei Lin, Wentai Wu, Ligang He, Keqin Li, Albert Y Zomaya, An efficiency-boosting client selection scheme for federated learning with fairness guarantee, IEEE Trans. Parallel Distrib. Syst. 32 (7) (2020) 1552–1564.

[148] Tiansheng Huang, Weiwei Lin, Li Shen, Keqin Li, Albert Y Zomaya, Stochastic client selection for federated learning with volatile clients, IEEE Internet Things J. (2022).

[149] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, Robert E Schapire, The nonstochastic multiarmed bandit problem, SIAM J. Comput. 32 (1) (2002) 48–77.

[150] Zhendong Song, Hongguang Sun, Howard H Yang, Xijun Wang, Yan Zhang, Tony QS Quek, Reputation-based federated learning for secure wireless networks, IEEE Internet Things J. 9 (2) (2021) 1212–1226.

[151] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, Ameet Talwalkar, Expanding the reach of federated learning by reducing client resource requirements, 2018, arXiv preprint arXiv:1812.07210.

[152] Nader Bouacida, Jiahui Hou, Hui Zang, Xin Liu, Adaptive federated dropout: Improving communication efficiency and generalization for federated learning, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021, pp. 1–6.

[153] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, Nicholas Lane, Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout, Adv. Neural Inf. Process. Syst. 34 (2021) 12876–12889.

[154] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al., Distilling the knowledge in a neural network, 2015, arXv preprint arXiv:1503.02531, 2 (7).

[155] Pengyuan Zhou, Pei Fang, Pan Hui, Loss tolerant federated learning, 2021, arXiv preprint arXiv:2105.03591.

[156] Mehryar Mohri, Gary Sivek, Ananda Theertha Suresh, Agnostic federated learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 4615–4625.

[157] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, Yaoliang Yu, Federated learning meets multi-objective optimization, IEEE Trans. Netw. Sci. Eng. (2022).

[158] Wei Du, Depeng Xu, Xintao Wu, Hanghang Tong, Fairness-aware agnostic federated learning, in: Proceedings of the 2021 SIAM International Conference on Data Mining, SDM, SIAM, 2021, pp. 181–189.

[159] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, Fei Wang, Addressing algorithmic disparity and performance inconsistency in federated learning, Adv. Neural Inf. Process. Syst. 34 (2021) 26091–26102.

[160] Zheng Wang, Xiaoliang Fan, Jianzhong Qi, Chenglu Wen, Cheng Wang, Rongshan Yu, Federated learning with fair averaging, 2021, arXiv preprint arXiv:2104.14937.

[161] Russell Hardin, Garrett Cullity, The free rider problem, 2003.

[162] Jingfeng Zhang, Cheng Li, Antonio Robles-Kelly, Mohan Kankanhalli, Hierarchically fair federated learning, 2020, arXiv preprint arXiv:2004.10386.
[163] Kjell Y. Tornblom, Dan R. Jonsson, Subrules of the equality and contribution principles: Their perceived fairness in distribution and retribution, Soc. Psychol. Q. (1985) 249–261.
[164] Matthew Rabin, Incorporating fairness into game theory and economics, Am. Econ. Rev. (1993) 1281–1302.
[165] Li Li, Martin Pal, Yang Richard Yang, Proportional fairness in multi-rate wireless LANs, in: IEEE INFOCOM 2008-the 27th Conference on Computer Communications, IEEE, 2008, pp. 1004–1012.
[166] Amirata Ghorbani, James Zou, Data shapley: Equitable valuation of data for machine learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 2242–2251.
[167] Lingjuan Lyu, Xinyi Xu, Qian Wang, Han Yu, Collaborative fairness in federated learning, in: Federated Learning, Springer, 2020, pp. 189–204.
[168] Sreenivas Gollapudi, Kostas Kollias, Debmalya Panigrahi, Venetia Pliatsika, Profit sharing and efficiency in utility games, in: 25th Annual European Symposium on Algorithms (ESA 2017), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
[169] Guan Wang, Charlie Xiaoqian Dang, Ziye Zhou, Measure contribution of participants in federated learning, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 2597–2604.
[170] Lloyd S. Shapley, A value for n-person games, Class. Game Theory 69 (1997).
[171] Min Ye, Emmanuel Abbe, Communication-computation efficient gradient coding, in: International Conference on Machine Learning, PMLR, 2018, pp. 5610–5619.
[172] Dun Zeng, Siqi Liang, Xiangjing Hu, Zenglin Xu, FedLab: A flexible federated learning framework, 2021, arXiv preprint arXiv:2107.11621.
[173] SMILELab-FL, FedLab: A flexible federated learning framework, 2021, https://github.com/SMILELab-FL/FedLab.
[174] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, Nicholas D Lane, Flower: A friendly federated learning research framework, 2020, arXiv preprint arXiv:2007.14390.
[175] adap, Flower - A friendly federated learning framework, 2020, https://github.com/adap/flower.
[176] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al., Fedml: A research library and benchmark for federated machine learning, 2020, arXiv preprint arXiv:2007.13518.
[177] FedML-AI, FedML: The community building open and collaborative AI anywhere at any scale, 2020, https://github.com/FedML-AI/FedML.
[178] Yang Liu, Tao Fan, Tianjian Chen, Qian Xu, Qiang Yang, FATE: An industrial grade platform for collaborative learning with data protection, J. Mach. Learn. Res. 22 (226) (2021) 1–6.
[179] WeBank, An industrial level federated learning framework, 2019, https://github.com/FederatedAI/FATE.
[180] Fan Lai, Yinwei Dai, Xiangfeng Zhu, Harsha V Madhyastha, Mosharaf Chowdhury, FedScale: Benchmarking model and system performance of federated learning, in: Proceedings of the First Workshop on Systems Challenges in Reliable and Secure Federated Learning, 2021, pp. 1–3.
[181] SymbioticLab, FedScale: Benchmarking model and system performance of federated learning at scale, 2021, https://github.com/symbioticlab/fedscale.
[182] Microsoft, Extracting value from siloed healthcare data using federated learning with Azure Machine Learning, 2022, https://customers.microsoft.com/en-us/story/1587521717158304168-microsoft-partner-professional-services-azure.
[183] Amazon, Machine learning with decentralized training data using federated learning on Amazon SageMaker, 2023, https://aws.amazon.com/cn/blogs/machine-learning/machine-learning-with-decentralized-training-data-using-federated-learning-on-amazon-sagemaker/.
[184] NVIDIA, Federated learning powered by NVIDIA clara, 2019, https://developer.nvidia.com/blog/federated-learning-clara/.
[185] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, Karteek Alahari, End-to-end incremental learning, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 233–248.
[186] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Yun Fu, Large scale incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 374–382.
[187] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al., Avalanche: an end-to-end library for continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3600–3610.
[188] Qiang Wang, Bingyan Liu, Yawen Li, Traceable federated continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 12872–12881.
[189] Chaoyang He, Conghui Tan, Hanlin Tang, Shuang Qiu, Ji Liu, Central server free federated learning over single-sided trust social networks, 2019, arXiv preprint arXiv:1910.04956.

Bingyan Liuis a research associate professor at the School of Computer Science (National Pilot School of Software Engineering), Beijing University of Posts and Telecommunications, China. He received his Ph.D. degree in computer science from Peking University in 2022. His research interests include federated learning, data mining, edge intelligence, transfer learning, model compression and privacy computing.

Nuoyan Lyu is an undergraduate at the School of Computer Science (National Pilot School of Software Engineering), Beijing University of Posts and Telecommunications, China. She will graduate in 2023 and continue her Ph.D. degree study at the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include graph neural networks, graph machine learning, and adversarial attack and defense.

Yuanchun Guo is currently pursuing the B.S.degree in computer science from Beijing University of Posts and Telecommunications, China. His current research interests include federated learning and computer vision.

Yawen Li is an associate professor at the School of Economics and Management, Beijing University of Posts and Telecommunications. She received her Ph.D. in Innovation, Entrepreneurship, and Strategy from Tsinghua University in 2018. Her research interest focuses on artificial intelligence, collaborative innovation, the development of science parks, and the scientific productivity of firms.