Heterogeneous Federated Learning: State-of-the-art and Research Challenges

MANG YE, XIUWEN FANG, and BO DU, School of Computer Science, Wuhan University, China
PONG C. YUEN, Department of Computer Science, Hong Kong Baptist University, China
DACHENG TAO, The University of Sydney, Australia

Federated learning (FL) has drawn increasing attention owing to its potential use in large-scale industrial applications. Existing FL works mainly focus on model homogeneous settings. However, practical FL typically faces the heterogeneity of data distributions, model architectures, network environments, and hardware devices among participant clients. Heterogeneous Federated Learning (HFL) is much more challenging, and corresponding solutions are diverse and complex. Therefore, a systematic survey on this topic about the research challenges and state-of-the-art is essential. In this survey, we firstly summarize the various research challenges in HFL from five aspects: statistical heterogeneity, model heterogeneity, communication heterogeneity, device heterogeneity, and additional challenges. In addition, recent advances in HFL are reviewed and a new taxonomy of existing HFL methods is proposed with an in-depth analysis of their pros and cons. We classify existing methods from three different levels according to the HFL procedure: data-level, model-level, and server-level. Finally, several critical and promising future research directions in HFL are discussed, which may facilitate further developments in this field. A periodically updated collection on HFL is available at https://github.com/marswhu/HFL_Survey.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Security and privacy** → **Privacy-preserving protocols**; • **Computing methodologies** → *Computer vision;*

Additional Key Words and Phrases: Survey, federated learning, trustworthy AI

ACM Reference format:
Mang Ye, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao. 2023. Heterogeneous Federated Learning: State-of-the-art and Research Challenges. *ACM Comput. Surv.* 56, 3, Article 79 (October 2023), 44 pages.
https://doi.org/10.1145/3625558

1 INTRODUCTION

With the popularization of smartphones, wearable devices, mobile networks, and so on, edge devices have become ubiquitous in modern society. An effective method to better utilize the abundant private data in edge devices without compromising privacy is **Federated Learning (FL)**, which

This work is partially supported by Zhejiang lab (NO.2022NF0AB01), CCF-Huawei Populus Grove Fund (CCF-HuaweiTC2022003).

Authors' addresses: M. Ye, X. Fang, and B. Du, School of Computer Science, Wuhan University, Bayi Road, Wuhan, Hubei Province, 430072, China; e-mails: {yemang, fangxiuwen, dubo}@whu.edu.cn; P. C. Yuen, Department of Computer Science, Hong Kong Baptist University, Renfrew Road, Kowloon Tong, Hong Kong, China; e-mail: pcyuen@comp.hkbu.edu.hk; D. Tao, The University of Sydney, City Road, Sydney, New South Wales, 2006, Australia; e-mail: dacheng.tao@gmail.com.

0360-0300/2023/10-ART79 $15.00
https://doi.org/10.1145/3625558

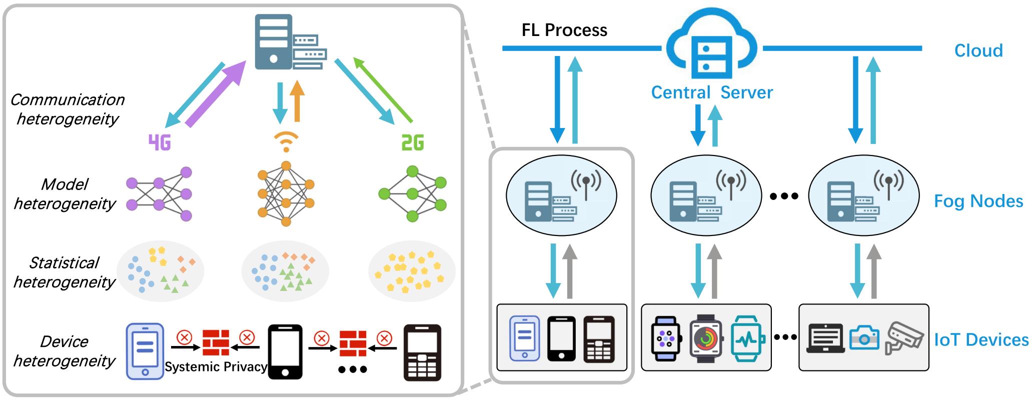

Fig. 1. Schematic of HFL. The figure includes cloud, fog nodes, and IoT edge devices, which constitute the multi-layer HFL framework. Different devices represent participating clients with heterogeneous local models and Non-IID private data. Each client cannot access the private data from others. The clients upload model-related information to the server, and the server aggregates and broadcasts the knowledge.

aims to collaboratively train **Machine Learning** (**ML**) models while keeping the data decentralized [229], i.e., following the data-stay-local policy. The participating devices in the FL system are regarded as different clients. FL is a distributed ML framework with secure encryption technology, which enables multiple institutions to conduct joint ML modeling under the requirement of protecting data privacy [68]. FL has drawn increasing attention in both academia and industry owing to its potential utility in large-scale applications [48, 87, 187], which has been widely explored in the fields of healthcare [39, 180, 216], medical analysis [55, 99], and data security [207], and so on.

Despite the great success in homogeneous FL, where it heavily relies on the assumption that all the participants share the same network structure and possess similar data distributions [125]. However, in practical large-scale situations, there may be considerable differences between data distributions, model structures, communication networks, and system edge devices, which make it challenging to realize federated collaboration. The FL associated with these situations is denoted as **Heterogeneous Federated Learning** (**HFL**), where this heterogeneity can be categorized into four classes according to the FL process: statistical heterogeneity, model heterogeneity, communication heterogeneity, and device heterogeneity. These are shown in Figure 1 and detailed as follows. (1) *Statistical heterogeneity*: the collected data in different participants may be **Non-Independent Identically Distributed** (**Non-IID**) [257] or unbalanced, resulting in inconsistent update optimization directions of participants. Original methods would fail in biased collaboration. Therefore, many recent approaches [175, 177, 204, 252] attempt to tackle this challenge from various aspects. (2) *Model heterogeneity*: Clients may have different tasks and specific requirements. Consequently, each client may expect to design its local model independently [152, 219], resulting in knowledge transfer barriers among heterogeneous participants, where the widely-used model aggregation or gradients operation cannot be applied. (3) *Communication heterogeneity*: Considering that clients may be deployed under varying network environments [184], this brings in communication inconsistency and unsynchronization, which is also ignored in existing works. This challenge might affect the learning efficiency, especially when the client number is large, which greatly limits the application in large-scale industry scenarios. (4) *Device heterogeneity*: The storage and computation capabilities of the devices for different participants may be diverse [125], which may cause faults and inactivation of some participating nodes, i.e., stragglers. There are several methods [75, 155, 236] developed to deal with this challenge at different stages. Compared with traditional homogeneous FL environments, HFL has several benefits. First and foremost, it is more flexible and adaptable to the diverse and dynamic scenarios of different clients, which may have inconsistent data distributions, model structures, communication networks, and hardware capabilities. Second, it takes full advantage of the complementary information and knowledge among heterogeneous clients, thus improving learning performance and robustness under complex and uncertain environments. Finally, it can also handle heterogeneous tasks, that is, different clients can pursue different learning tasks or objectives according to their own needs, thus, improving the adaptability and flexibility of the model.

Several surveys have been released on FL in general or specific aspects of FL. However, none of them provide a reasonable and comprehensive taxonomy of the research challenges and state-of-the-art of HFL, which is an important and emerging research direction in FL. HFL aims to address the heterogeneity issues that arise from different aspects, such as statistical distribution, model architecture, communication setting, and device condition. Therefore, we discuss the main contributions and limitations of existing related surveys and highlight the unique contributions of our work in Table 1. Kairouz et al. [98] discuss recent advances in FL and provide a survey of open problems and challenges, including communication efficiency, privacy preservation, attack defense, and federated fairness. Li et al. [125] discuss the challenges faced by FL from four perspectives: communication efficiency, system heterogeneity, statistical heterogeneity, and privacy concerns, and briefly listed several future research directions. Wahab et al. [210] provide a fine-grained classification scheme of existing challenges and approaches. Li et al. [121] classify FL systems from six aspects, including data distribution, ML model, privacy mechanism, communication architecture, federated scale, and federated motivation. Lim et al. [131] studied FL in mobile edge networks and divided the existing methods into methods that solve the fundamental problems of FL and methods that use FL to solve edge computing problems. Niknam et al. [167] mainly enumerate and discuss several possible applications of FL in 5G networks, and described the key issues faced by FL in wireless communication settings. There are some surveys [102, 164] exploring FL for IoT Networks. Khan et al. [102] present advances in FL for IoT applications and provide a taxonomy using various operation modes as parameters (global aggregation, resource, local learning model, etc.). Besides, they identify important issues (robustness, privacy, and communication cost) and open challenges in FL, and propose corresponding guidelines. Nguyen et al. [164] provide a survey and analysis of FL in IoT services (IoT data sharing, data offloading and caching, attack detection, etc.) and IoT applications (smart healthcare, smart transportation, unmanned aerial vehicles, etc.). Yang et al. [229] divide FL into three categories: horizontal FL, vertical FL, and federated transfer learning according to the distribution characteristics of data. However, they mainly introduce the concept and application of FL, lacking a detailed classification and summary of existing methods. Gao et al. [59] discuss data space, statistics, system, and model heterogeneity in FL, respectively, and provide a classification and introduction of scenarios, goals, and methods under each heterogeneity problem. Zhu et al. [257] analyze the impact of Non-IID data in FL and provide a survey of the researche on handling Non-IID data, but overlook several other heterogeneity issues and related research. Kulkarni et al. [108] point out that statistical heterogeneity can deprive high-performance clients of incentives to participate in FL, highlighting the need for personalization, and survey work on this topic. They focus on the challenges posed by statistical heterogeneity while ignoring other issues. Tan et al. [200] explore the field of personalized FL, which studies the problem of learning personalized models to handle statistical heterogeneity and conduct a taxonomic survey of existing methods. However, it lacks a comprehensive taxonomy and systematic analysis of the challenges in FL. Wu et al. [219] provide a personalized FL framework in a cloud-edge architecture for intelligent IoT applications. But their classification of existing methods is not reasonable enough, which is only a small part of HFL. In addition, the existing methods are diverse and vary

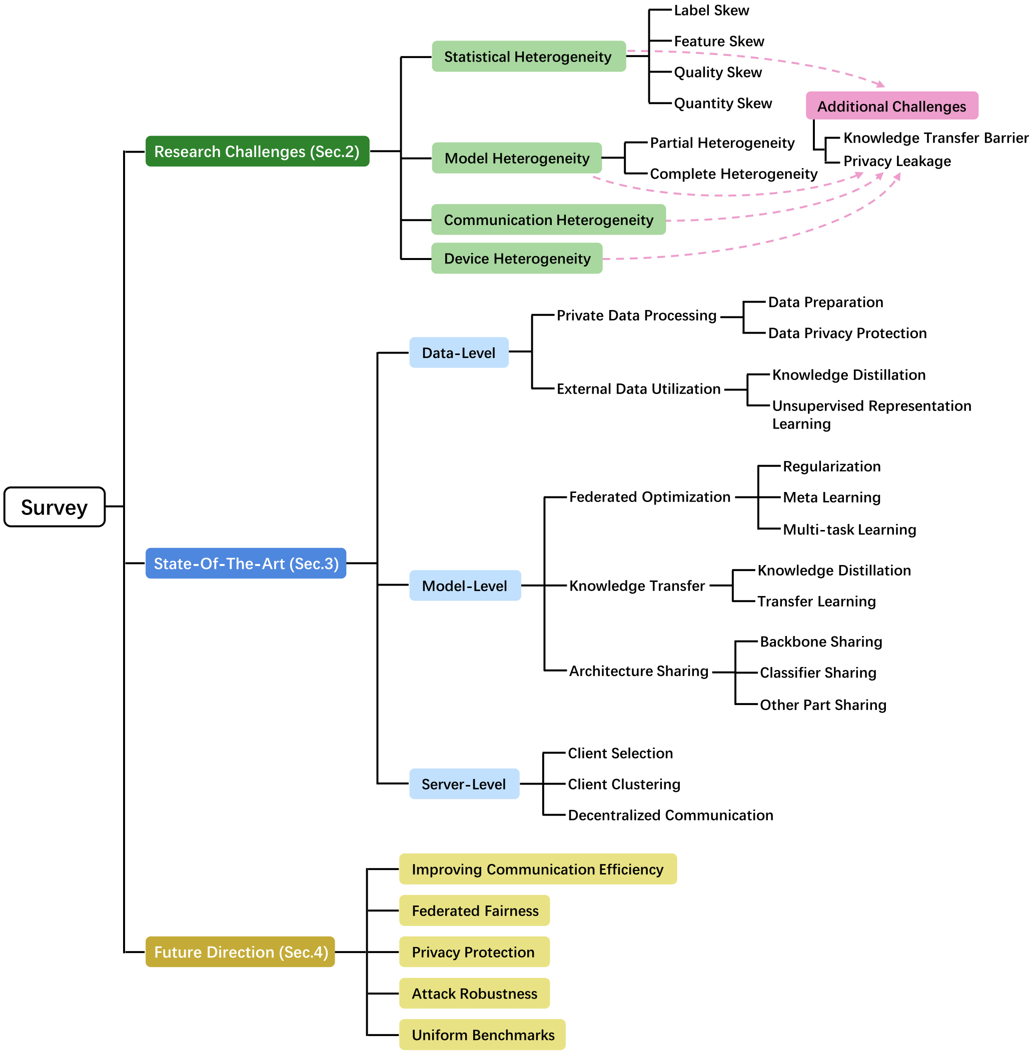

Fig. 2. The outline structure of our survey. It contains three different parts: Research challenges, State-of-the-Art, and Discussion of future directions.

widely in their own settings without a standard setting, making it challenging for readers to keep abreast of advancements in this field. Consequently, a comprehensive and systematic survey on the research challenges, methods, limitations, and future directions associated with HFL is urgently needed.

In this article, we survey recently published or pre-printed articles on HFL from top-tier conferences and journals. In particular, we not only investigate the problem of statistical heterogeneity and model heterogeneity but also analyze the aspects of privacy preservation and storage computational capacity during federated communication, which are particularly important for HFL. Unlike other related surveys [125, 200], this survey consists of three major parts (Figure 2). (1) We firstly systematically summarize the research challenges from five different aspects (Section 2). (2) We then review the current state-of-the-art methods with in-depth discussions about their advantages and limitations in the context of a new taxonomy (Section 3). (3) Finally, we will present a thorough outlook analysis of the unsolved issues and open problems for future development (Section 4). An outline structure of our survey is illustrated in Figure 2.

For the research challenges of HFL, we focus on the above-mentioned five aspects, i.e., statistical heterogeneity, model heterogeneity, communication heterogeneity, device heterogeneity, and additional challenges. Considering that the data distribution of each client may be different, we discuss the Non-IID data from four perspectives: label skew, feature skew, quality skew, and quantity skew. Model heterogeneity is divided into partial heterogeneity and complete heterogeneity, according to the architectural models trained in the FL process. Communication heterogeneity refers to the differences in communication resources and environments of clients, which are affected by the bandwidth, reliability, and topology of communication channels. Device heterogeneity is mainly caused by differences in the storage and computational capability of devices. Furthermore, the above four heterogeneous challenges may exacerbate two additional challenges, namely knowledge transfer barriers and privacy leakage. Knowledge transfer barriers indicate difficulties in effectively learning from each other. Privacy leakage refers to the sensitive information of local data sources being exposed to other parties. By extensively analyzing the research challenges in HFL, the research priorities in this field can be identified.

To review the state-of-the-art methods, we introduce a new taxonomy to categorize existing HFL approaches (Figure 5) into three levels, i.e., data-level, model-level, and server-level. Data-level approaches focus on smoothing the statistical heterogeneity of local data across clients at the data level to support HFL, such as data augmentation [43, 94, 237]. Model-level methods tend to operate at the model level for HFL, e.g., sharing partial structures [35, 141], model optimization [46, 120, 126], and knowledge transfer [48, 49, 87]. The server-level methods require server participation, such as participating client selection [211, 228], or client clustering [61, 182, 208]. This new taxonomy of existing methods will facilitate the understanding of the state-of-the-art in HFL, providing further guidelines for our following discussion.

Last but not least, this survey also provides several perspectives to highlight the scope for the future development of HFL. For example, how to reduce resource consumption and training time while improving model performance in heterogeneous scenarios is a key challenge, that is, we need to improve the efficiency and effectiveness of FL by conquering the heterogeneities. Besides, the emphasis on fairness will continue to grow as practical deployments of FL expand to more users and enterprises. This aspect is especially important for heterogeneous participants with unequal initial states, as they may have personalized requirements and characteristics. To ensure the privacy protection of HFL, it is crucial to establish stricter and more flexible privacy constraint policies, enforcing secure federated communication in all zones. In addition, the robustness of federated systems against attacks and failures requires increasing attention, especially in cases involving heterogeneous models with varying patterns against the attacks. At present, there is a lack of widely recognized benchmark datasets and benchmark testing frameworks for HFL. This highlights the need to establish systematic evaluation metrics to promote the research on and the development of HFL.

2 PROBLEMS: RESEARCH CHALLENGES IN HETEROGENEOUS FEDERATED LEARNING

First, this section provides a formulaic definition of FL and illustrates a typical FL process. Additionally, we present a detailed taxonomy of the problems encountered in HFL. (1) Data is the primary element in HFL. Considering that the data distribution may differ across clients, we discuss *statistical heterogeneity* from the four perspectives of label skew, feature skew, quality skew, and quantity skew in Section 2.1. Besides, this Non-IID phenomenon may hinder subsequent model training. (2) According to the different architectural models trained in the FL process,

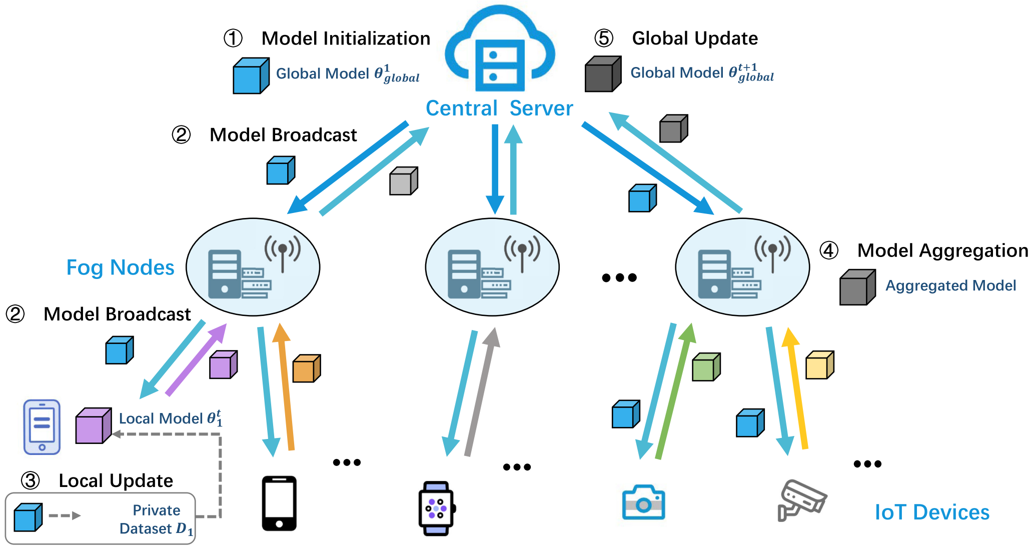

Fig. 3. General multi-layer FL architecture diagram.

model heterogeneity can be divided into partial heterogeneity and complete heterogeneity, as described in Section 2.2. (3) For the problem of *communication heterogeneity* caused by different network environments such as high communication costs and low communication efficiency, we will discuss this issue in Section 2.3. (4) The challenges of *device heterogeneity*, which result from differences in device storage and computation capabilities and may lead to stragglers, fault issues, and low communication efficiency, are discussed in Section 2.4. The various forms of heterogeneities introduce additional challenges associated with knowledge transfer barriers and privacy leakage, as presented in Section 2.5.

Preliminaries. In the typical FL framework, we assume that it involves K participating clients $\{C_1, C_2, \ldots, C_K\}$. For each client, the kth client C_k has a private dataset $D_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ with $|x^k| = N_k$ and $N = \sum_{k=1}^{K} N_k$. Moreover, the client C_K usually has a learned local network model or initialized model, denoted by $f(\theta_k)$. Therefore, $f(x^k, \theta_k)$ represents the predicted output of the private sample x^k on the local model θ_k. Traditional centralized ML frameworks are typically built on a larger centralized dataset $D_{central} = D_1 \cup D_2 \cup \ldots \cup D_K$ by directly integrating the private datasets of each client, which is then used to train a better performing centralized model $\theta_{central}$. However, owing to the constraints of data silos and data privacy, traditional centralized learning cannot be applied in real-world privacy-sensitive scenarios. As an alternative, FL enables each client C_k to collaboratively train ML models without exposing the private data D_k to other clients $C_{k_0 \neq k}$. Here we take FedAvg [157] as a typical FL process (Figure 3). In the following, we show the steps involved in FL at the tth epoch, in which the federated system iterates the steps for several epochs until the end of the federated training process.

- **Model Initialization.** The server selects eligible clients $\{C_1, C_2, \ldots, C_K\}$ as participants, and initializes the global model θ^1_{global} in the first round.
- **Model Broadcast.** The server sends the current global model θ^{t-1}_{global} to all participating clients as the initialization of local models $\{\theta_1^{t-1}, \theta_2^{t-1}, \ldots, \theta_K^{t-1}\}$.
- **Local Update.** Each participating client C_k utilizes the private dataset D_k for local model updating as follows:

$$\theta_k^t \leftarrow \theta_k^{t-1} - \alpha \nabla_\theta \mathcal{L}_k(f(x^k, \theta_k^{t-1}), y^k), \tag{1}$$

where α represents the learning rate and $\mathcal{L}_k(\cdot)$ denotes the calculated loss for each client k.

- **Model Aggregation.** The server calculates the aggregation $\sum_{k=1}^{K} \frac{N_k}{N} \theta_k^t$ of the updated client model parameters.
- **Global Update.** The server updates the global model for the next epoch based on the aggregated result, as follows:

$$\theta_{global}^{t+1} \leftarrow \sum_{k=1}^{K} \frac{N_k}{N} \theta_k^t. \tag{2}$$

- **Model Deployment.** The server distributes the global model to the participating clients.

In the cloud-fog-IoT computing environment, FL is usually considered as decentralized and multi-layered [78], as illustrated in Figure 3. The participants will be divided into different layers according to their roles and capabilities, including the cloud layer, fog layers, and IoT layers. The cloud layer is the central server that performs global model aggregation and updates [166]. It has high computing and storage capabilities, but the communication cost with edge nodes is also high. The fog layers are intermediate layers composed of multiple edge servers (e.g., base stations) that can communicate with the cloud layer and the IoT layers [171]. The IoT layers are composed of edge devices (e.g., smartphones, sensors), performing local model training and communicating with the fog layers. And the IoT devices are allowed to communicate with their neighbors in a peer-to-peer manner [131]. Compared with the general FL process, the cloud-fog-IoT FL introduces the middle layer of the fog server between the cloud server and the IoT device [136]. Therefore, there will be an additional step of model distribution and model aggregation in the fog layers, which can relieve the communication pressure between the cloud layer and the IoT layers [176]. In addition, the cloud-fog-IoT systems have greater flexibility and adaptability in handling various heterogeneities in FL.

2.1 Statistical Heterogeneity

Statistical heterogeneity refers to the case where the data distribution across clients in FL is inconsistent and does not obey the same sampling, i.e., Non-IID. To explore the difficulties of the statistical heterogeneity with the Non-IID phenomenon, we classify statistical heterogeneity from a distribution perspective [98, 119]. Specifically, we distinguish different categories of Non-IID data in terms of four different skew patterns as shown in Figure 4, including label skew, feature skew, quality skew, and quantity skew. We define two different clients i and j. Therefore, the local data distribution of client i is denoted as $P_i(x, y)$, where x and y represent the features and labels of the data samples, respectively. Numerous studies [101, 119, 126] indicate that the local optimization objectives of the clients are inconsistent with the global optimization objective due to the differences in the local data distribution of the clients. Therefore, statistical heterogeneity may cause local models to converge in different directions, reaching local optima rather than global optima, thus degrading the FL performance, which might be even worse than the local learning stage without federated communication.

Label Skew. It means that the label distributions across participating clients are different. This phenomenon is commonly encountered in practical applications in which the data collection or annotation is inconsistent. To characterize the various label skew scenarios, we introduce two different settings: *label distribution skew* [246] and *label preference skew* [98]. A visual example is illustrated in Figure 4(a).

Label distribution skew indicates that the label distributions may be different for different clients, i.e., $P_i(y) \neq P_j(y)$, even if the feature distribution is shared (the features of the data samples are similar for each label, regardless of which client they belong to), i.e., $P_i(x|y) = P_j(x|y)$. For example, in handwriting number recognition, different users may contain different numbers.

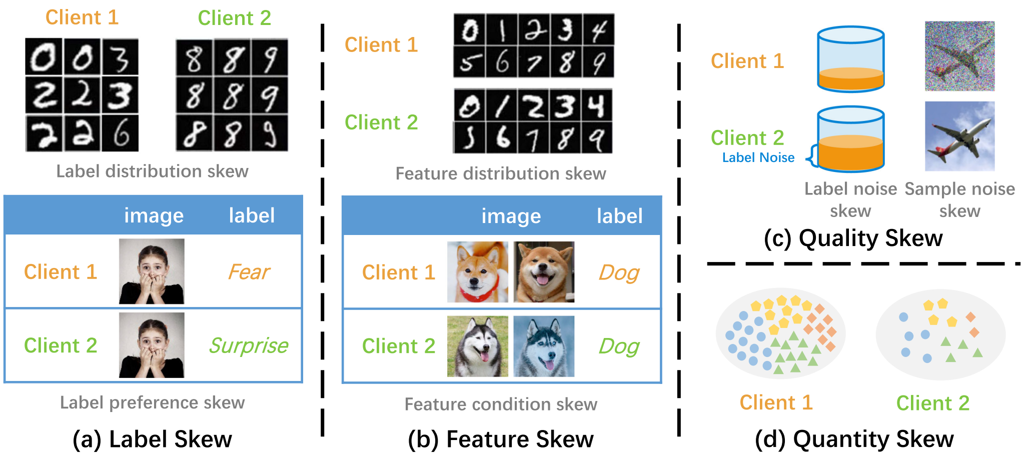

Fig. 4. Illustration of four different skew patterns in statistical heterogeneity.

Label preference skew implies that even when the feature distribution is consistent across clients, i.e., $P_i(x) = P_j(x)$, the label distribution may be different for different clients, i.e., $P_i(y|x) \neq P_j(y|x)$. The local training datasets of different clients may overlap horizontally, leading to label preference skew, i.e., the same features have different labels. That is, different clients may annotate the same data samples with different labels owing to individual annotation preferences. For example, for a visual intent understanding task [95], the same image may be annotated with different labels according to the personal preferences of users.

Feature Skew. It refers to the scenario that the feature distributions across participant clients are different. This phenomenon often occurs in complex real-world environments, implying that the feature distribution of local data on individual clients may be significantly different [119, 142]. Feature skew can be divided into the following two settings, which are shown in Figure 4(b):

Feature distribution skew implies that the label distribution is consistent, i.e., $P_i(y|x) = P_j(y|x)$, but the feature distribution may be different, i.e., $P_i(x) \neq P_j(x)$. For instance, in handwriting number recognition, different users may write the same number with different styles, stroke thicknesses, and so on.

Feature condition skew means that the feature distributions may vary across clients, i.e., $P_i(x|y) \neq P_j(x|y)$, even if $P_i(y) = P_j(y)$. The data features across clients may not fully overlap, and this situation is mainly related to vertical FL [27], which is commonly performed in medical applications. For example, when individual clients link regions, the Japan region contains a large number of Shiba Inu samples, while the Siberia region contains a large number of Husky samples, but their labels are all dogs.

Quality Skew. It demonstrates that the annotation or data collection qualities are inconsistent for different clients. FL typically involves a large number of clients, each of whom may have different data synthesis capabilities. There is no guarantee that all clients have data samples of the same quality, and they may hold unequal proportions of noisy data [218, 230]. Therefore, we divide the quality skew into label noise skew and sample noise skew, as shown in Figure 4(c).

Label noise skew represents that the proportion of noisy labels contained differs across clients. Owing to the differences in expertise and input costs, the quality of data annotations tends to vary widely across clients, which means that clients contain data with varying degrees of label noise. This challenge intensifies when the architectures of participating clients are different since the decision boundaries are inconsistent.

Sample noise skew refers that each client holds private data with different qualities, where the data collection process inevitably introduces varying levels of sample noise. Owing to differences in the abilities of clients to synthesize and collect data, several clients may collect data containing redundant or noisy information, which makes communication across different clients uncertain and complex.

Quantity Skew. It means that the amount of local data may be extremely unbalanced across clients [257], i.e., long-tail distributed data [186]. An example is illustrated in Figure 4(d). In such a situation, several clients may have problems with data scarcity [88], which reinforces the need for FL. However, existing methods cannot adaptively balance the contribution of each client in the server aggregation process.

2.2 Model Heterogeneity

In the widely-studied FL paradigm, each participating client is required to use a local model with the same architecture. Thus, the network parameters of the local model can be aggregated into a global model on the server side. In practical IoT applications, each client may expect to design their own local model architecture in a unique manner owing to differences in individual requirements and hardware constraints [219]. Additionally, clients are often reluctant to reveal or share the details of model design, as they wish to safeguard their commercial proprietary information and privacy. For example, when healthcare institutions conduct collaborative learning without sharing patient information, they may design models with different structures to meet the properties of different tasks. Therefore, model HFL requires learning knowledge from others without sharing private data or disclosing local model structure information. The main challenge of model heterogeneity would be the difficulty of transferring knowledge between model heterogeneous clients in a model-agnostic manner. To this end, we categorize model heterogeneity into partial heterogeneity and complete heterogeneity.

Partial Heterogeneity. It is commonly encountered in real-life scenarios, where some of the clients are using the same model structure while others do not. When client subsets are divided based on the local model structure, at least one client subset is expected to have no less than two elements [182]. In this analysis, we consider the federated system to be a partial model heterogeneity. A FL model needs to be trained for each client subset whose models are isomorphic. Through the clustering algorithms, participating clients can be divided into many clusters, i.e., the structures are the same within each cluster. Therefore, some common techniques, such as weighted averaging of model parameters, can be directly used to realize the aggregation of intra-cluster models. However, the communication of inter-cluster models requires the design of some special techniques, such as knowledge distillation.

Complete Heterogeneity. It is a special case of partial heterogeneity, in which all the network structures of participant models are completely different in the FL framework. Complete model heterogeneity occurs when the individual clients in a federated system have local model structures that differ from each other [193]. In this context, it is necessary to learn a unique model for each client, which can better handle different data distributions but may eventually lead to high learning costs and low communication efficiency. Ensuring communication between complete heterogeneous models is challenging because the widely-used network parameter aggregation or gradient operations cannot be performed.

2.3 Communication Heterogeneity

In practical IoT applications, the devices are typically deployed in different network environments and have different network connectivity settings (3G, 4G, 5G, Wi-Fi) [26, 125], which leads to inconsistent communication bandwidth, latency, and reliability, i.e., communication heterogeneity [24].

For example, a central hospital may have a high-speed fiber optic network, while a rural clinic may only have a low-speed wireless network. This leads to the problem of communication heterogeneity. When these medical institutions perform operations such as uploading and downloading with the server, delays, and failures may occur, thereby hindering the effect of FL [157].

Communication heterogeneity increases communication cost and complexity to some extent. Considering the differences in the network connectivity settings of IoT devices, different devices may require different amounts of data or time to connect and communicate with the server [79, 91]. Some devices may connect slowly, rendering them expensive and unreliable to communicate with. Besides, in the training process, there may be offline devices due to network bandwidth and energy constraints. Communication heterogeneity may also reduce communication efficiency and effectiveness [76]. Some IoT devices have problems such as low-quality network environments, slow device connection, and limited network bandwidth. Therefore, the clients may encounter varying degrees of noise, delay, or loss during the communication process, which severely reduces communication efficiency [215, 240]. To enhance communication efficiency, stragglers and offline devices with a significant time difference may be discarded after a sufficient number of clients have transmitted their feedback results to the server side. Notably, communication heterogeneity can be viewed as a strategy to address the differences in device computing power [38] by using bounded-delay assumptions to control device latencies. Communication heterogeneity is very prevalent in complex IoT environments, which may lead to high-cost and low-efficiency communication, thereby diminishing the effectiveness of FL. Therefore, how to adaptively adjust federated communication in heterogeneous network environments is worth studying.

2.4 Device Heterogeneity

In practical applications, FL networks may involve a large number of participating IoT devices. The differences in device hardware capabilities (CPU, memory, battery life) may lead to different storage and computation capabilities [125], which inevitably lead to device heterogeneity. For example, one client is a smartphone, while another client is a smartwatch. Smartphones have larger storage capacity and stronger computing capacity than smartwatches, which brings device heterogeneity. Therefore, during federated communication, smartwatches may have longer local runtimes and may be dropped or lost, reducing system efficiency and stability.

In FL, clients need to perform local updates and seed the updates back to the server side. However, the participating clients may encounter faults during this process. When the model is updated synchronously, devices with limited computation capabilities may consume a significant amount of time to update the model and may become stragglers. Overall, device heterogeneity poses several challenges to FL. First, it leads to system imbalance and inefficiency, as different clients may have different computing speeds or resources, causing system lags or bottlenecks. Second, it introduces uncertainty and instability to the system, since different clients may have different device states, including states that lead to system failure or loss [125]. Therefore, device heterogeneity introduces constraints such as straggler mitigation and fault tolerance into the FL process, which necessitates the adaptive adjustment of the feedback for different devices in large-scale FL scenarios.

2.5 Additional Challenges

Apart from the above-mentioned heterogeneities, this part also discusses some additional challenges in HFL, including knowledge transfer barriers and privacy leakage.

Knowledge Transfer Barriers. FL is aimed at transferring knowledge between different clients to collaboratively learn models with superior performance. However, the above-mentioned heterogeneous characteristics cause knowledge transfer barriers to different degrees. The client collects data in a Non-IID way, which leads to statistical heterogeneity. Under the combined influence of multiple skew patterns, the domain knowledge of all clients cannot be sufficiently learned. When the clients have models with different structures, the general average parameter strategy cannot be used for model aggregation, resulting in barriers to knowledge exchange between heterogeneous models. Besides, communication heterogeneity and device heterogeneity in the complex **Internet of Things** (**IoT**) environments weaken the efficiency of knowledge transfer. Therefore, how to achieve efficient knowledge transfer in heterogeneous scenarios is a problem that current researches need to focus on.

Privacy Leakage. It is now well-understood that privacy is one of the first-order concerns in FL [98] because protecting the local data of clients from being leaked is a fundamental principle of FL. In the communication process, each client never shares private data with the server or other clients to ensure basic privacy [162]. However, FL by itself cannot guarantee perfect data security, as there are still potential privacy risks or attacks on data privacy. Moreover, the above-mentioned four types of heterogeneity inevitably exacerbate privacy leakage in different learning stages. For example, when clients implement FL by sharing model gradient updates, logits output, and so on, attackers can infer the private information of clients by injecting malicious data or models into the system, or by analyzing their model gradients. This may result in the unavoidable leakage of sensitive information to the server or other clients.

Several methods have been investigated to enhance the privacy protection of FL, mainly containing anonymization, secure aggregation, **Differential Privacy** (**DP**), homomorphic encryption, **Secure Multi-party Computation** (**SMC**), and so on. Anonymization conceals the identity of the client by using cryptography so that model updates or gradients cannot reveal anything unique to the client. Data aggregation enhances the privacy of FL by combining data or gradients from multiple clients, reducing the information from a specific client in the shared information [97]. DP hides real original information by adding noise to the data or gradients before sending them to the server, including local DP [206], hybrid DP [137], shuffle model [62], and so on. Homomorphic encryption allows a server to perform computations on encrypted data or gradients without decrypting them. Secure multi-party computing is based on the SMC encryption protocol, enabling multiple clients to jointly calculate functions applicable to their private data without sharing the original data. Typically, they achieve privacy protection at the expense of model performance and communication efficiency. Therefore, how to ensure privacy protection without compromising the model performance is a key challenge in federated communication.

3 METHODS: STATE-OF-THE-ART

This section reviews existing HFL approaches by dividing them into three parts (Figure 5), i.e., data-level, model-level, and server-level methods. The data-level methods refer to operations at the data level that smooth the statistical heterogeneity of local data across clients or improve data privacy, such as data augmentation and anonymization techniques. The model-level methods refer to operations designed at the model level, e.g., sharing partial structures, and model optimization. The server-level methods require server engagement, such as participating client selection, or clients clustering.

3.1 Data-Level Methods

In this subsection, we introduce the classification of data-level methods and some representative methods in each category, as shown in Table 2. Data-level methods refer to operations performed at the data level, including private data processing and external data utilization. Private data processing means that clients internally process private data to improve data quality, diversity, and security, thereby optimizing the performance of FL. These methods include data preparation and data privacy protection. Data preparation includes operations such as data collection, filtering,

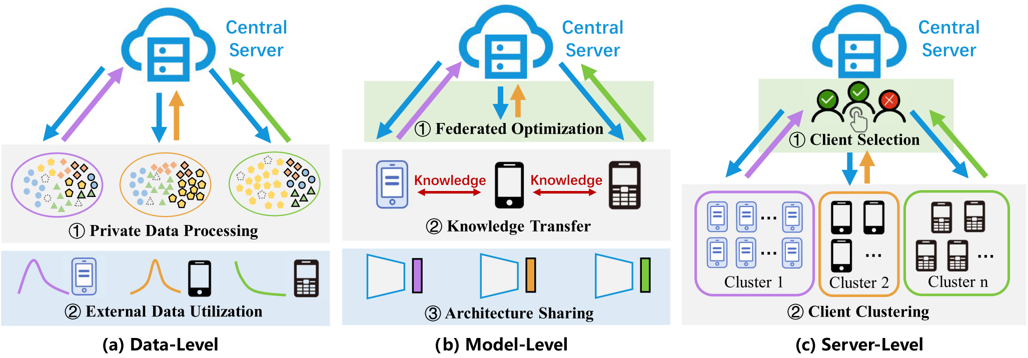

Fig. 5. Illustration of the state-of-the-art methods in our taxonomy at three different levels.

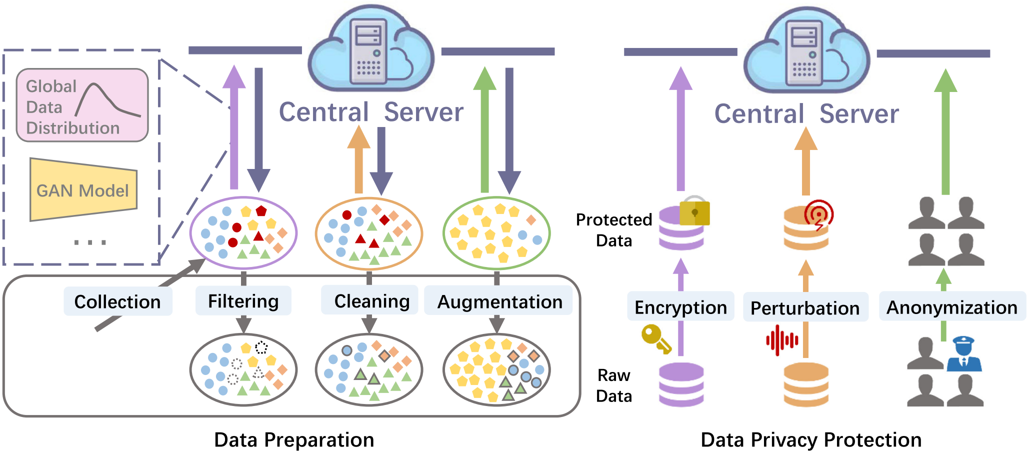

Fig. 6. Illustration of private data processing methods in HFL.

cleaning, and augmentation, which can directly alleviate statistical heterogeneity. Data privacy protection aims at ensuring that the original data information is not disclosed. External data utilization refers to performing model knowledge distillation or imposing constraints on model updates by introducing additional data. Knowledge distillation is usually employed to deal with communication difficulties caused by model heterogeneity and can alleviate statistical heterogeneity and communication heterogeneity to some extent. Unsupervised representation learning can alleviate the statistical heterogeneity between the local data.

3.1.1 Private Data Processing.

Data Preparation. Private data preparation in FL includes data collection, filtering, cleaning, augmentation, and more (Figure 6). These operations are aimed at ensuring the data quality and security of each client, thereby improving the efficiency and effectiveness of FL. Data collection refers to the process of obtaining data from participating clients. Data filtering refers to the removal of irrelevant data. Data cleaning refers to the correction of inaccurate data. Data augmentation is the process of enhancing or augmenting data with additional information or features, and it has been widely explored in FL to address statistical heterogeneity.

Data collection refers to the process of obtaining data from participating clients. In FL, the quantity, quality, and diversity of data collection determine how much useful information the client provides to the FL system. Therefore, local data collection is an important part to be optimized in FL. Data filtering is the process of removing or excluding irrelevant, noisy, or malicious data from FL. Therefore, Li et al. [113] consider various data-related factors (error rate, classification distribution, content diversity, and data size) that affect model performance to measure the data quality of samples. Then they select the optimal sample combination with the highest total quality under the monetary budget constraint. Besides, data filtering can effectively prevent the FL system from being negatively affected by malicious data. For example, Xu et al. [227] propose a collaborative data filtering method, Safe, for data selection, which can detect and filter out poisoned data from attacked devices in an FL system. Specifically, Safe first clusters the local data, then measures the distance between each sample and its cluster center, and finally discards samples far from the cluster center as poisoned data. Data cleaning is the process of correcting or improving incomplete, inaccurate, or inconsistent data in FL, usually by applying relevant techniques locally on the client side to impute missing values, resolve conflicts, and standardize data.

Data augmentation is a technique of artificially expanding training datasets by generating more data from limited raw data, which can effectively alleviate the problem of data deficiency in deep learning. In addition, popular data augmentation operations include flipping, rotating, scaling, cropping, shifting, Gaussian noise introduction, and MixUp [191, 243]. Additional data can also be artificially synthesized using **Generative Adversarial Networks** (**GANs**) [64]. However, statistical heterogeneity of client datasets is commonly encountered in federated settings, and private data augmentation techniques can be directly used to smooth the data distribution across multiple clients and mitigate statistical heterogeneity. Federated data augmentation typically requires users to upload a few local data samples, which increases the risk of data privacy breaches. To circumvent this risk, several approaches require a proxy dataset that can represent the overall data distribution of a federated system. Owing to these aspects, data augmentation in federated settings is highly challenging.

In a Non-IID environment with uneven data distributions on the clients, Yoon et al. [237] construct a **Mean Augmented Federated Learning** (**MAFL**) framework, in which clients exchange mean local data to obtain global information while maintaining privacy requirements. Furthermore, they designed a data augmentation algorithm FedMix under the MAFL framework, which approximates the loss function of the global mixup through Taylor expansion without accessing the raw data of other clients. Consider the client i has a private local dataset (x_i, y_i), and $f(,)$ is the model output. Therefore, the approximated FedMix loss can be expressed as

$$\mathcal{L}_{FedMix} = (1-\lambda)\mathcal{L}(f((1-\lambda)x_i), y_i) + \lambda\mathcal{L}(f((1-\lambda)x_i), \bar{y}_j) + \lambda\frac{\partial\mathcal{L}}{\partial x}\bar{x}_j, \tag{3}$$

where λ represents the mixup rate, and $\bar{x}_j$ and $\bar{y}_j$ refer to the means of all inputs and labels received from client j. However, MAFL may pose a threat to privacy security. Especially when there are insufficient local data, the averaged data contain a large amount of raw relevant information, and adequate privacy restrictions for the raw data cannot be ensured. In Astraea [43], the server collects the local data distributions of the clients in the initialization phase and then performs data augmentation based on the global data distribution. To alleviate data distribution imbalance, Astraea creates mediators that rearrange the training of clients based on the **Kullback–Leibler** (**KL**) divergence of the local data distributions. **Federated augmentation** (**FAug**) [94] is a data augmentation scheme using GAN. Each client can identify target labels that are lacking in data samples. Subsequently, the clients upload partial data samples of the target labels to the server, and the server oversamples the uploaded data samples to train a conditional GAN. In this manner, the clients effectively enhance the statistical homogeneity of the local data by generating missing data samples using the received GAN.

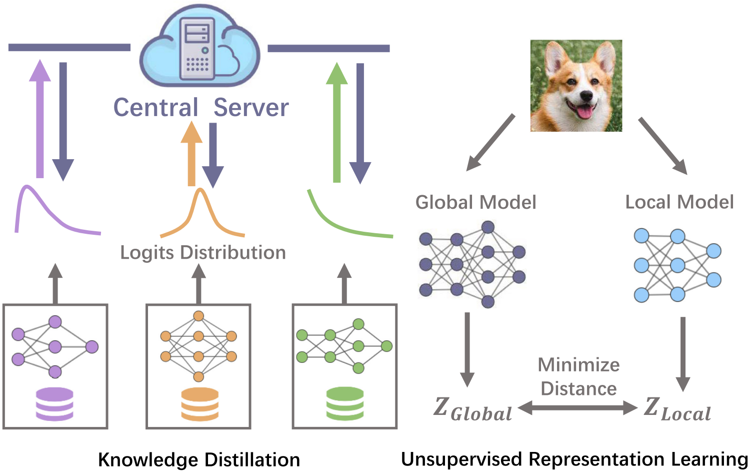

Fig. 7. Illustration of external data utilization methods in HFL.

Data Privacy Protection. To ensure that original information about commercial encryption, user privacy, and so on is not leaked to other clients, methods for data privacy protection at the local level have been extensively studied. These methods generally fall into three categories [235], namely, data encryption, perturbation, and anonymization (Figure 6).

Homomorphic encryption is a commonly used data encryption method that allows computations to be performed using encrypted data without decryption. Asad et al. [7, 47] apply homomorphic encryption to FL, enabling the client to encrypt its local model with a private key and then send it to the server. So the server can only get encrypted model parameters and cannot deduce any private information. DP is the most commonly used method when using data perturbation to achieve privacy protection. It protects the client's private information by clipping and adding noise to local updates. Hu et al. [81] propose an FL method for personalized local DP, PLDP-PFL. It allows each client to choose an appropriate privacy budget for personalized differential privacy according to the sensitivity of its private data. To alleviate the model performance degradation caused by DP, Shi et al. [189] propose a FL framework with DP, DP-FedSAM. It leverages the sharpness aware minimization optimizer while updating locally to generate a local flatness model with better stability and robustness to weight perturbations, thus, improving its robustness to DP noise. Data anonymization makes it difficult for data subjects to be identified by removing or replacing identifiable sensitive information in the data. Choudhury et al. [33] make the clients generate their own anonymous data mapping according to the characteristics of the local dataset, that is, convert the original data into some random numbers or symbols, so as to desensitize the original data.

3.1.2 External Data Utilization.

Knowledge Distillation from External Data. This method leverages external data sources for knowledge distillation to improve FL performance (Figure 7), where external data usually refers to public data. The idea is to use a global teacher model trained on an external dataset to help clients generate soft labels for local data. Then, clients utilize these soft labels as additional supervision for local updates, thereby improving the generalization ability of the models and mitigating the impact of statistical heterogeneity. Besides, these methods are often used to address model heterogeneity by utilizing external easily accessed data [48, 87]. Specifically, each client computes the output prediction distribution of the local model on external data and sends it as local knowledge to the server or other clients. And distillation means that other clients update their own local model parameters based on the received knowledge. In this way, clients with heterogeneous models can share local information in a blackbox manner, thereby alleviating the impact of model heterogeneity.

FAug [94] and FedMD [115] utilize **Federated Distillation** (**FD**), also known as **Co-Distillation** (**CD**), to learn knowledge from other clients. Each client stores a local model output and considers the average local model output of all clients as the global output. Huang et al. [88] adopt a federated communication strategy similar to FedMD and innovatively add a latent embedding adaptation module, which alleviates the impact of the large domain gap between public datasets and private datasets. Considering the global model as a teacher and the local model as a student, Yu et al. [239] attempt to mitigate overfitting in personalized updates by enhancing the logit similarity between the global model and the local models. FedGKT [72] periodically transfers the knowledge of small CNNs on edges to the large server-side CNN through knowledge distillation, thereby decreasing the burden of edge training. Besides, several recent approaches utilize knowledge distillation to mitigate statistical heterogeneity among clients. FedFTG [248] trains a conditional generator to fit the input space of a local model, which is then used to generate pseudo data. These pseudo data are input to the global model and the local model for knowledge distillation, and the knowledge of the local model is transferred to the global model by narrowing the KL divergence between their output predictions.

Unsupervised Representation Learning. Since the private data from clients are usually difficult to annotation [233] and it usually involves huge manual cost, **Federated Unsupervised Representation Learning** (**FURL**) [139, 208] is discussed to learn a common representation model while keeping private data decentralized and unlabeled (Figure 7).

FedCA [242] is a FURL algorithm based on contrastive loss, which can address data distribution inconsistencies and representation misalignment across clients. FedCA includes a dictionary module that aggregates sample representations from all clients and sharing them with clients, and an alignment module that aligns the representations of each client on public data. Briefly, the clients generate local dictionaries through the above two contrastive learning modules, and then the server aggregates the trained local models and integrates the local dictionaries into the global dictionary. Similarly, MOON [120] and FedProc [163] also use contrastive learning to address statistical heterogeneity in FL. MOON corrects the update direction by introducing a model-contrastive loss. Its objective is to drive the representation learned by the current local model to be consistent with that learned by the global model, while increasing the distance between the representation learned by the current local model and that learned by the previous local model. z, z_p, and z_g denote the representations from the current local model, previous local model, and global model, respectively. Therefore, the model-contrastive loss can be defined as

$$\mathcal{L}_{con} = -\log \frac{\exp(sim(z, z_g)/\tau)}{\exp(sim(z, z_g)/\tau) + \exp(sim(z, z_p)/\tau)}, \tag{4}$$

where τ represents a temperature parameter. The main idea of FedProc is to treat the global prototype as global knowledge, and use a local network architecture and a global prototype contrastive loss to constrain the training of the local model. Different from MOON, Tan et al. [201] propose FedProto, which only transfers the prototype to the client without transferring the model parameters and gradients. This method can handle various heterogeneous problems more efficiently.

3.2 Model-Level Methods

In this subsection, we categorize model-level methods, introduce representative methods in each category, and discuss their contributions and limitations, as shown in Table 3. Model-level methods represent methods for innovative design at the model level, mainly including federated optimization, knowledge transfer across models, and architecture sharing. Federated optimization aims at

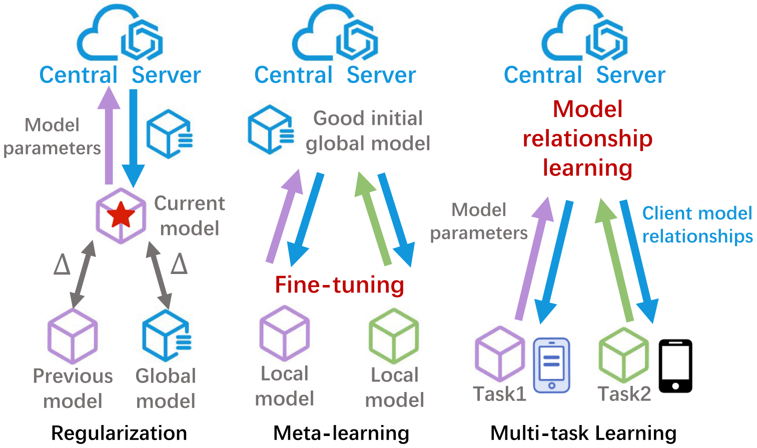

Fig. 8. Illustration of federated optimization in HFL.

adapting the model to the local distribution while learning the global information. They can effectively realize local model personalization under statistical heterogeneity. Knowledge transfer across models enables multi-party collaboration in a model-agnostic manner and thus is usually used to solve model and communication heterogeneity. Architecture sharing realizes personalized FL by sharing part of the model structure and can solve statistical, model and device heterogeneity simultaneously to some extent. The data-level methods can solve the problem associated with large differences in data distribution to a certain extent and accelerate convergence by smoothing the statistical heterogeneity among data. In contrast, the model-level methods aim at learning a local model for each client that adapts to its private data distribution, and thus, such methods have been extensively researched.

3.2.1 Federated Optimization.

Regularization. Regularization is a technique that helps prevent overfitting by adding a penalty term to the loss function. This strategy decreases the model complexity by dynamically estimating the parameter values and decreases variance by adding bias terms (Figure 8). Therefore, many FL frameworks implement regularization to provide convergence guarantees when learning under statistical heterogeneity [104].

FedProx [126] adds a proximal term on the basis of FedAvg [157]. The server distributes the global model θ^t of the previous epoch t to clients, and the client k computes the local empirical risk $f_k(\theta)$ on the private dataset. Furthermore, the client k approximately minimizes the objective h_k as follows:

$$\min_{\theta} h_k(\theta, \theta^t) = f_k(\theta) + \frac{\lambda}{2}||\theta - \theta^t||^2, \tag{5}$$

where λ is the regularization parameter to control the strength of θ^t to the personalized model, and FedAvg is a special case with $\lambda = 0$. The essence of the proximal term is to constrain the difference between the local model and the global model, so as to effectively increase the stability of model training and accelerate model convergence. FedCurv [192] and FedCL [232] are adaptations of the **Elastic Weight Consolidation** (**EWC**) [105] algorithm to FL scenarios. FedCurv uses the EWC algorithm to prevent catastrophic forgetting when transferring to different learning tasks. Its main idea is to selectively penalize certain parameter vectors far from the network parameters learned in the previous task. FedCL adopts the EWC algorithm to estimate the importance weight matrix of the global model and integrates the knowledge of each client into the global model. pFedME [199] utilizes the Moreau envelope function as a regularized loss function, which decouples the personalized model optimization process from global model learning. Zhang et al. [249] highlight that the clients should consider the suitability of other models to their goals when downloading personalized weighted combinations, and thus devised a personalized FL framework named FedFomo. FedAMP [90] builds a positive feedback loop, iteratively promotes similar client models to collaborate more strongly than dissimilar client models, and adaptively groups similar clients to promote effective collaboration. FedBN [128] solves the problem of feature skew in statistical heterogeneity by adding a batch normalization layer to the local model. SCAFFOLD [101] uses variance reduction to correct the client-drift caused by statistical heterogeneity. Specifically, SCAFFOLD estimates the update direction of the server model and the local model and then uses the difference between the server model and the local model to correct the local model update. Instead of learning a single global model, Hanzely et al. [70] find a tradeoff between the global and local models by adding a regularization term, and learn an implicit mixture model of the global and local models. Unlike these methods, MOON [120] considers not only the regularization between the global model and the local model but also the regularization between the current local model and the previous local model. To deal with statistical heterogeneity and to improve training stability, Xu et al. [225] introduce an adaptive weighted proximal regularization term based on the estimated noise level. To address the problem of statistical heterogeneity and device heterogeneity, Pillutla et al. [109, 110, 172] improve the performance of the worse-off clients while maintaining the average performance using a risk measure known as the superquantile (or CVaR), which is able to capture the tail statistical characteristics of the client error distribution. Huang et al. [89] propose FPL to make sample embeddings closer to cluster prototypes of the same domain and category. Meanwhile, a consistency regularization is introduced to align sample embeddings with homogeneous unbiased prototypes that do not contain domain information. Chen et al. [23] designed an elastic aggregation strategy that adaptively interpolates client-side models based on parameter sensitivity, measured by computing the impact of each parameter variation on the output of the overall prediction function. It is an implicit regularization method. In terms of practical technical applications, FedHumor [67] applies FL to personalized humor recognition in texts and considers the distribution of humor preferences of different clients to perform domain adaptive fine-tuning training, achieving personalized FL.

Meta-learning. This technique is also known as "learning to learn". Previous experience is used to guide the learning of new tasks, thereby allowing a machine to learn a model by itself for different tasks (Figure 8). Recently, a meta-learning algorithm named **Model-Agnostic Meta-Learning** (**MAML**) [52] has attracted widespread attention, as it can be directly applied to any method based on gradient descent. In brief, MAML first trains the initialized model. When training on new tasks, a satisfactory learning performance can be achieved by performing fine-tuning based on the initial model with only a small amount of data. In this manner, meta-learning has sufficient personalization ability to handle statistical heterogeneity in FL.

Jiang et al. [96] point out that the MAML setting is consistent with the personalized objectives of HFL. The MAML algorithm is divided into two phases: meta-training and meta-testing, corresponding to the global model training and local model personalization in FL, respectively. They also observe that the FedAvg algorithm is very similar to the Reptile [3] algorithm. Careful fine-tuning yields a global model with a high accuracy, and the local models are easy to personalize. Per-FedAvg [46] is a personalized variant of the FedAvg algorithm based on the MAML formula. Per-FedAvg aims at learning a high-performance initial global model to ensure that each heterogeneous client can obtain a high-performance local model after personalized updating on the global model. Compared with Per-FedAvg, which performs only one-step gradient updates to obtain personalized models, pFedMe [199] implements multi-step updates. Chen et al. [25] propose a federated meta-learning framework, FedMeta. An algorithm is maintained on the server and distributed to the client for training, after which the test results on the query set are uploaded to the server for algorithm update. ARUBA [103] utilizes online convex optimization and sequence prediction algorithms to adaptively learn the task-similarity and test the FL performance. To combat the possible vulnerabilities of meta-learning algorithms, a federated meta-learning method named FedML [134] is established based on **distribution robust optimization** (**DRO**). Zheng et al. [256] apply federated meta-learning to fraudulent credit card detection. This method enables the collaboration between different banks through FL, improves the triplet loss function, and designs a meta-learning-based classifier for local model updates. Chu et al. [34] propose a multi-layer personalized FL method, MLPFL, to optimize the inference accuracy of different levels of device grouping criteria. MLPFL trains a personalized model with meta-gradient updates for all groups of edge devices.

Multi-task Learning. Multi-task learning enables models learned on a single task to help learn other tasks by using shared representations or models for relevant tasks. If the local model learning for each client is considered as a separate task, the idea of multi-task learning can be implemented to solve the FL problem. To this end, multi-task learning aims at solving different tasks on multiple clients simultaneously and trains a model that jointly learns the relevant tasks (Figure 8). All participating clients collaboratively train their local models, thereby effectively mitigating statistical heterogeneity and yielding high-performance personalized local models.

MOCHA [193] is a system-aware optimization framework for **federated multi-task learning** (**FMTL**), which attempts to address the problems of high communication cost, stragglers, and fault tolerance for distributed multi-task learning. To address statistical heterogeneity and system challenges, MOCHA employs the distributed optimization method COCOA [92, 148], and trains a unique model for each client. However, MOCHA has some limitations in that it requires all clients to participate in each iteration, which is clearly impractical, and this method cannot be applied to non-convex deep learning models. To ensure that the FMTL algorithm can be applied to general non-convex models, VIRTUAL [36] considers the federation of a central server and clients as a Bayesian network and employs approximated variational inference for training. OFMTL [122] simulates the relationship between different devices by introducing an accuracy matrix, through which personalized models for new devices can be inferred without revisiting the original device data. Besides, Dinh et al. [41] advise a communication centralized FMTL algorithm FedU, which uses Laplacian regularization to capture the relationship between client models. Ditto [124] is a scalable FMTL framework with two tasks, a global objective and a local objective. This method ensures that the personalized model is close to the global optimization model by introducing a regularization term. Marfoq et al. [154] study FMTL under the assumption that each local data distribution is a mixture of unknown underlying distributions. Therefore, each client can benefit from the knowledge distilled from the local data of other clients by modeling the distributions.

3.2.2 *Knowledge Transfer across Models.*

Knowledge Distillation across Models. In practical FL applications, clients may expect to design unique model structures and be reluctant to share the model details. To address the challenges associated with model heterogeneity, knowledge distillation is widely applied in model HFL. The objective is to refine the knowledge distribution on clients, and then transfer the learned knowledge in a model-agnostic way (Figure 9).

FedMD [115] implements client-side communication by leveraging knowledge distillation and transfer learning. The logits output of the local model f_k on the public dataset $D_0 = \{x_i^0\}$ can be considered as the knowledge distribution of the client k. The server collects this knowledge to

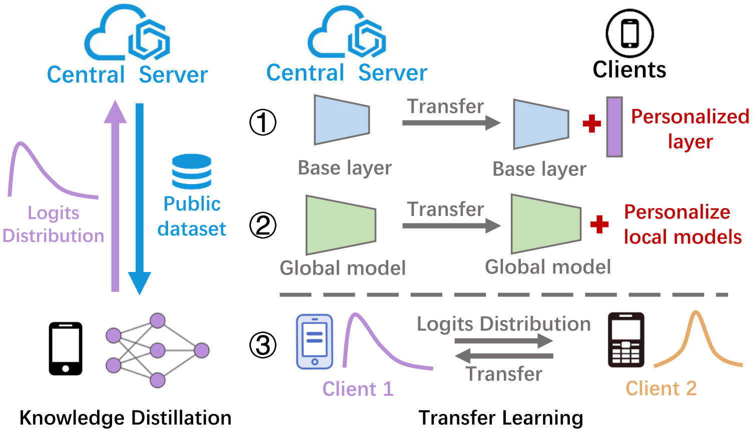

Fig. 9. Illustration of the knowledge transfer approaches across models in HFL.

compute the average logits output as a global consensus $\tilde{f}$, which can be expressed as

$$\tilde{f}(x_i^0) = \frac{1}{K} \sum_{k=1}^{K} f_k(x_i^0), \tag{6}$$

Subsequently, each client learns the information from other clients by training its local model to approach the global consensus. The basic ideas of Cronus [21] and FedMD are similar as they both operate on public data via knowledge distillation. However, FedMD calculates a global consensus through a common averaging strategy, whereas Cronus designs an aggregation algorithm that is robust against poisoning attacks. The performance of the aforementioned methods depends heavily on the public data quality. To alleviate the reliance on public data, FedDF [135] utilizes unlabeled or generated data for ensemble distillation. To perform FL with heterogeneous clients without relying on a global consensus or shared public models, RHFL [48, 87] learns the knowledge distribution of other clients by aligning models feedback on irrelevant public data. To improve the communication efficiency of FL, Sattler et al. [181] developed a compressed federated distillation method CFD, which employs distilled data curation, soft-label quantization, and delta-encoding to reduce the communication from client to server. Unlike these approaches, FedGEN [258] does not rely on the server side to possess a proxy dataset. FedGEN performs statistical HFL through a data-free knowledge distillation approach, in which the server learns a lightweight generator derived only from the prediction rules of client models, thereby integrating client information in a data-free manner. The generator is then broadcast to the clients, using the learned knowledge as an inductive bias to guide local model training.

Transfer Learning. The goal of transfer learning is to apply the knowledge learned on the source domain to different but related target domains (Figure 9). In FL scenarios, the clients typically belong to different but related domains and wish to learn knowledge from other domains. Therefore, transfer learning is widely applied in the FL field. Knowledge distillation is an effective strategy for transfer learning. Accordingly, federated transfer learning aims at transferring the knowledge learned on clients to the server for aggregation or to transfer the global consensus to clients for personalization.

In the healthcare domain, FedHealth [30] is established as a federated transfer learning framework. FedHealth performs data aggregation through FL and then builds personalized models through transfer learning. Considering the large distribution differences between the server model and the client models, FedHealth allows clients to train personalized models through transfer

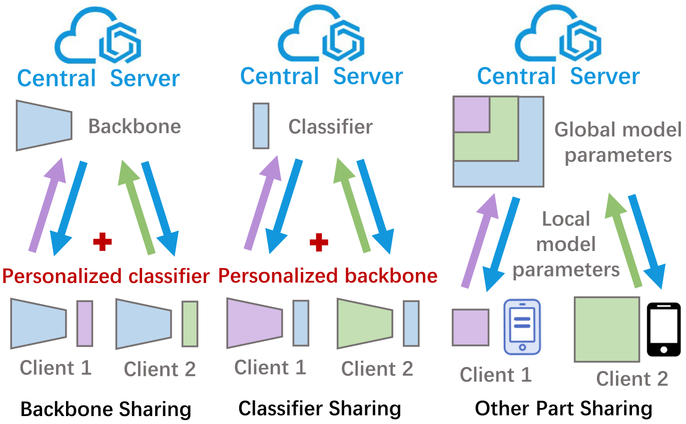

Fig. 10. Illustration of the architecture sharing approaches in HFL.

learning. In FedMD, transfer learning is a key approach to address the problem of private data scarcity and personalize the local models. In the transfer learning phase, the clients first fully train the local models on the public dataset and then train them on the private datasets until convergence. To solve the problem of slow convergence and degraded learning performance in heterogeneous scenarios, the decentralized FL via mutual knowledge transfer (Def-KT) algorithm is designed by Li et al. [114]. Def-KT does not require the participation of the server, and each client directly transfers the knowledge in a point-to-point manner. In the personalized knowledge transfer phase, KT-pFL [245] linearly combines the soft predictions of all clients through the knowledge coefficient matrix to identify the mutual contribution of clients, thereby enhancing the collaboration between clients with similar data distributions. The soft prediction of the local model θ_n on the public dataset D_0 represents the collaborative knowledge $f(\theta_n, D_0)$ from client n. Furthermore, KL divergence is utilized to transfer knowledge across clients, and then the local update can be phrased as

$$\theta_n \leftarrow \theta_n - \alpha \nabla_\theta \mathcal{L}_{kl} \left(\sum_{m=1}^{N} \mathcal{M}_m \cdot f(\theta_m, D_0), f(\theta_n, D_0) \right), \tag{7}$$

where D_0 represents a mini-batch of the public dataset, α is the learning rate, and $\mathcal{M}_m$ denotes the knowledge coefficient vector of client m. Gao et al. [58] design a privacy-preserving transfer learning method to remove covariate shifts in homogeneous feature spaces and bridge heterogeneous feature spaces of different clients. Unlike the above methods that transfer the entire global model to the client, FedPer [6] trains the model base layers on the server side based on FedAvg and then transfers the globally shared base layers to the client. The clients train the model personalization layers on local data with stochastic gradient descent, thereby mitigating the impact of statistical heterogeneity.

3.2.3 *Architecture Sharing.*

Backbone Sharing. In heterogeneous scenarios, the private datasets of clients may be Non-IID. To mitigate the negative effects caused by statistical heterogeneity, clients may share the backbone, but they design personalized layers in the neural network models for personalized demands (Figure 10). Furthermore, the clients only need to upload the backbone to the server-side in the aggregation phase, thereby decreasing the communication cost to some extent.

For example, the aforementioned FedPer [6], which combines the base layers and the personalized layers for federated training of deep feedforward neural networks, can effectively capture the personalization aspects of clients. Intuitively, FedPer first uses a FedAvg-based approach to globally train the base layers on the public dataset. Subsequently, each client updates the personalized layers with the private dataset using an SGD-style algorithm. The base layers consist of shallow neural networks for high-level feature extraction, and the personalized layers are deep neural networks for classification. This framework can avoid the problem of retraining in federated transfer learning. In the training process of federated representation learning (FedRep) [35], all clients jointly train the global representation learning structure and then use their private datasets to train their own client-special heads. Here the heads of clients represent personalized, low-dimensional classifiers. **Classifier Calibration with Virtual Representations** (**CCVR**) [141] generates approximate **Gaussian mixture model** (**GMM**)-based virtual representations in feature space via learned feature extractors. To mitigate the problem that the classifier can be easily biased to the heterogeneous local data, CCVR eliminates the bias of the classifier by regularizing or calibrating the classifier weights. To further consider the problem of long-tail distribution, **Classifier Re-training with Federated Features** (**CReFF**) [186] learns federated features to re-train the classifier, which approximates training the classifier on real data.

Classifier Sharing. To handle heterogeneous data and tasks, several methods share a classifier instead of a backbone (Figure 10). Intuitively, the clients perform feature extraction through their own backbones and share a public classifier for classification. In this way, clients can learn from each other while satisfying personalization and without compromising their data privacy or task specificity.

The recently devised LG-FedAvg [130] jointly learns compact local representations of all clients and a global model across all clients. In contrast to FedPer, LG-FedAvg uses personalized layers to extract high-level, compact features that are important for prediction and uses the base layers shared by the server for classification. In LG-FedAvg, the client k possesses a private dataset $\{(x, y)|x \in X_k, y \in Y_k\}$. Furthermore, the client k extracts the feature $\mathcal{F}_k = l_k(X_k, \theta_k^l)$ of the private data samples through the local model θ_k^l. These features $f \in \mathcal{F}_k$ are predicted by the global model θ_k^g. The overall loss on the client k can be expressed as

$$\mathcal{L}_k(\theta_k^l, \theta_k^g) = \mathbb{E}\left[-\log \sum_f (p_{\theta_k^g}(y|f), p_{\theta_k^l}(f|x))\right]. \tag{8}$$

LG-FedAvg thus effectively decreases the communication cost by extracting useful lower-dimensional representations. Xu et al. [226] propose FedPAC, which reduces inter-client feature variance by constraining each sample feature vector to be close to the global feature centroid of its category. Then, the server performs an optimal weighted aggregation of the personalized classifier headers of the clients.

Other Part Sharing. Apart from the case of backbone or classifier sharing, several methods employ other part sharing strategies, that is, adaptively share a subset of the local model parameters according to local conditions (data distribution, computing capability, network bandwidth, privacy preference, etc.) as shown in Figure 10. This approach effectively enhances the applicability of FL in scenarios involving different resource and network environment constraints across clients, thereby alleviating device heterogeneity and communication heterogeneity. Furthermore, sharing part of the model for personalization can avoid catastrophic forgetting to some extent [156, 173], which is extremely important for clients with large differences in data distributions.

To alleviate the effect of communication heterogeneity and device heterogeneity, HeteroFL [40] allocates local models of different sizes according to the computing and communication capabilities of each client. The local model parameters are a subset of the global model parameters, which effectively decreases the computation of local clients. Different from the prevailing methods that

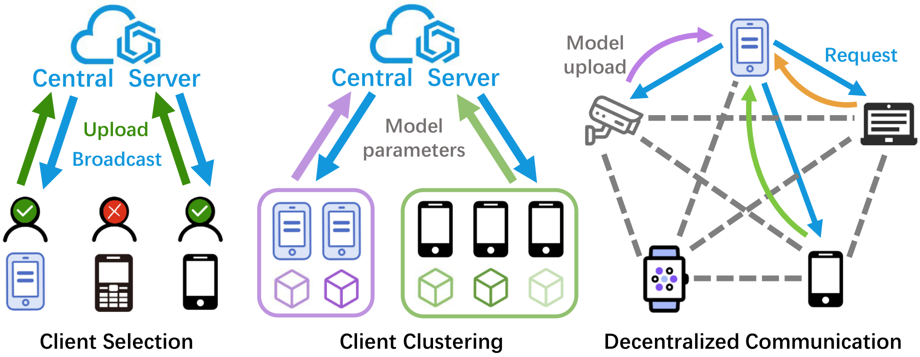

Fig. 11. Illustration of the server-level methods in HFL.

divide the shared layers and the personalization layers in a layer-wise mechanism, CD^2-pFed [188] dynamically decouples the global model parameters for personalization, which is known as channel decoupling. Concretely, a personalization rate is also defined to assign global shared weights and local private weights to each layer of the objective model. FedLA [149] leverages a hypernetwork on the server to evaluate the importance of each client model layer and generate aggregation weights for each model layer, thereby realizing personalized layer-wised model aggregation.

3.3 Server-Level Methods

In this subsection, we categorize server-level methods and discuss the advantages and disadvantages of existing methods in Table 4. Server-level methods refer to methods that rely on server-side operations, including client selection, client clustering, and decentralized communication. Client selection aims at selecting the appropriate client to participate in the FL process for each iteration, which can solve various heterogeneous challenges. Client clustering improves the FL efficiency by aggregating similar clients, thereby alleviating communication and device heterogeneity. Decentralized communication enables peer-to-peer collaboration between devices without relying on a central server.

3.3.1 Client Selection. Client selection is typically performed by the server so that data can be sampled from clients with uniform data distributions (Figure 11). Moreover, constraints such as the network bandwidth, computation capability, and local resources of different clients are considered when formulating selection strategies. Consequently, client selection can accelerate convergence and substantially improve the model accuracy.

The performance of traditional FL on Non-IID datasets is inferior to its performance on IID datasets, and the convergence speed on Non-IID datasets is also much slower than that on IID datasets. Several methods [10, 83, 214, 250] are devised to alleviate the bias introduced by Non-IID data. Favor [211] is an experience-driven control framework that actively selects the best subset of clients to participate in FL iterations. Innovatively, they define device selection for FL as a deep reinforcement learning problem that aims at training an agent to learn an appropriate selection policy. In addition, class imbalance may occur when the client data distribution is inconsistent. To address this problem, Yang et al. [228] devise an estimation scheme that clarifies the client class distribution based on the gradient of client service updates without considering the original data. Moreover, they designed a client selection algorithm for the minimal class imbalance to improve the global model convergence. Tang et al. [202] propose a correlation-based client selection strategy, using a Gaussian process to model loss changes of clients, and then selecting one client in each iteration to reduce the overall loss expectation. Qin et al. [174] believe that when selecting clients for collaborative training, in addition to considering the independent characteristics of clients, it is necessary to pay attention to the synergy between clients.

The differences in the hardware and the network connectivity capabilities across clients may lead to high communication costs, low training efficiency, wasted computing resources, and so on. Thus, a large number of client selection strategies [1, 14, 32, 84, 238] aim at addressing these issues. In FedSAE [117], the server estimates the reliability of each device based on the complete information of the client training history tasks, thereby adjusting the training load of each client. Furthermore, the server selects the clients with higher values based on the training losses of the clients to improve communication efficiency. FedCS [168] performs the client selection operation based on the differences in data resources, computing capabilities, and wireless channel conditions across client models and uses the selected clients $\mathbb{C}$ for aggregation. The core idea of client selection is to solve the following maximization problem:

$$\max_{\mathbb{C}} |\mathbb{C}| s.t. \quad T_{epoch} \geq T_{cs} + T_{\mathbb{C}}^{d} + T_{\mathbb{C}}^{u} + T_{agg}, \tag{9}$$

where T_{round} is the deadline for each round. In addition, T_{cs}, $T_{\mathbb{C}}^{d}$, $T_{\mathbb{C}}^{u}$, and T_{agg} represent the time of client selection, distribution, scheduled update and upload, and aggregation, respectively. In this manner, FedCS effectively alleviates the effects of communication heterogeneity and device heterogeneity, while maximizing the number of participants in a round and improving the training efficiency. To better deal with the deviation problem caused by statistical, communication and device heterogeneity, TiFL [20] adopts an adaptive layer selection method that divides clients into different layers according to the training time and then selects clients from the same layer in each training round. Li et al. [118] propose a multi-layer online coordination framework for high-performance energy-efficient FL, MCFL, which selects suitable devices by simultaneously considering the training data volume, computing power, and runtime training behavior of each device. Wu et al. [220] propose a multi-layer FL protocol, HybridFL, where each edge node randomly selects a subset of clients according to a specific probability distribution depending on the region slack factor. Regulating the proportion of selected clients mitigates the straggler and dropout problems caused by communication and device heterogeneity.

3.3.2 *Client Clustering.*

The model performance for all clients may not be satisfactory if the entire federated system shares one global model. Therefore, many existing approaches [179, 182] perform the personalized clustering of all clients by considering the similarities of the data distributions, the local models and, the parameter updates of different clients (Figure 11).

Briggs et al. [15] aim to decrease the number of communication rounds required to reach convergence while improving the test accuracy. To this end, a hierarchical clustering step is introduced to separate client clusters by the similarity of client updates to the global joint model. In addition, Xie et al. [223] leverage a novel multi-center aggregation mechanism to address the problem of statistical heterogeneity. It utilizes a distance-based multi-center loss function to minimize the distance between a local model and its nearest global model while constraining the variations in the local model updates. The local models $\{\theta_m\}_{m=1}^{M}$ are divided into K clusters, i.e., cluster 1,..., cluster k. The cluster k has an aggregated model $\tilde{\theta}^k$ of multiple local models, and the multi-center loss function can be formulated as

$$\min_{\tilde{\theta}^k} \frac{1}{M} \sum_{k=1}^{K} \sum_{m=1}^{M} r_m^k Dist(\theta_m, \tilde{\theta}^k), \tag{10}$$

where r_m^k represents the client m is assigned to the cluster k. Then they propose FeSEM, which employs **Stochastic Expectation Maximization** (**SEM**) [18] optimization to calculate the distance between the local models and the cluster centers to ensure that an optimal match can be derived. To alleviate the high communication cost incurred by clustering, FedFMC [107] dynamically groups devices of similar prototypes in certain epochs and then merges them into a single model. To improve the clustering efficiency, FedGroup [44] designs a Euclidean distance of **Decomposed Cosine similarity** (**EDC**) to perform clustering based on the similarities between the optimization directions of the clients. A novel method called **Clustered Federated Learning** (**CFL**) [182] measures the similarity between the data distributions of different clients in terms of the cosine similarity between their gradient updates. The above operations can effectively identify the cluster structure, and thus clients with similar data can benefit from one another, while weakening harmful interference between clients with different data. Other methods measure the similarity of the models by comparing their loss values. For example, **Iterative Federated Clustering Algorithm** (**IFCA**) [61] minimizes the loss functions by alternately optimizing the cluster model parameters through gradient descent while estimating the client's cluster identity. Lim et al. [132, 133] propose a hierarchical FL framework to solve a two-level resource allocation and incentive mechanism design problem. At the edge layer, edge devices choose any cluster to join, and intermediate nodes provide rewards for the participation of edge devices. In the cloud layer, each intermediate node chooses any server, and the servers have to compete with each other for the service of the intermediate node. Feraudo et al. [51] propose CoLearn, which leverages manufacturer usage description profiles of IoT devices to cluster devices with similar learning tasks and network requirements. CoLearn also uses a publish/subscribe messaging system to coordinate the learning process between edge devices and cloud servers.

Several recent methods aim at improving the attack robustness of FL systems through client clustering. Sattler et al. [183] apply CFL to the Byzantine scenario, in which a subset of clients interferes with federated training in a detrimental way. A large number of clients are declared benign, and other clients are declared adversarial and excluded from training. This method decreases the computation cost while enhancing the robustness and flexibility of the federated framework. However, it is vulnerable to backdoor attacks (including data poisoning and model poisoning) in FL scenarios. To address this problem, Nguyen et al. [165] propose FLAME, which does not rely on the underlying data distributions for benign and adversarial datasets. FLAME detects adversarial model updates through a clustering strategy that limits the noise scale of backdoor noise removal. To improve the performance of the aggregated model, FLAME implements a weighting method to limit the adversary models.

3.3.3 *Decentralized Communication.*

The general FL algorithm relies on a central server, which requires all clients to trust a central institution, and the failure of this institution will destroy the entire FL process. Therefore, some algorithms adopt decentralized communication [190], which conducts peer-to-peer communication between various devices without relying on a central server (Figure 11).

Roy et al. [178] propose a peer-to-peer FL framework without a central server, BrainTorrent, that is, direct communication between clients. Specifically, in each round, a client is randomly selected as a temporary server and then cooperates with other clients that have completed the model update for collaborative update. In this way, any client can dynamically start an update process at any time. Lalitha et al. [112] propose a fully distributed FL algorithm, where clients can only communicate with their one-hop neighbors without relying on a centralized server. Clients update their local models by aggregating information obtained from one-hop neighbors to obtain an optimal resulting model. To sufficiently utilize the bandwidth capacity between clients while maintaining convergence performance, some decentralized federated training algorithms [80, 203] are designed based on the Gossip protocol. For example, in GossipFL [203], each client dynamically selects a peer client based on its network bandwidth to fully utilize the global bandwidth resource. Then, the client exchanges a highly compressed model with its peer client, reducing communication traffic. Hu et al. [80] propose a segmented gossip approach, Combo, where the client k divides the local model W_K into S segments, and then randomly selects some clients to transfer the model segments. K_s represents the set of clients providing segment s with $s \in S$ and D_k represents the private dataset of client k. The segment s can be aggregated with the private dataset size as the weights to get $\tilde{W}[s]$. Then, merge all aggregated segments to get the complete aggregated model W as follows:

$$W = (\tilde{W}[1], \tilde{W}[2], \ldots \tilde{W}[s], \ldots \tilde{W}[S]), \tilde{W}[s] = \frac{\sum_{k \in K_s} W_k[s] \cdot |D_k|}{\sum_{k \in K_s} |D_k|}. \tag{11}$$

Similar to the idea of this model segmentation, Pappas et al. [170] introduce a fully decentralized FL framework, IPLS. Each client retains some model segments. Before model training, it obtains model segments that are not available locally from other clients and combines them into a complete model. Then, the client uses the full model to update locally and then shares the update gradient with other clients. Kalra et al. [100] design a proxy-based decentralized FL scheme, ProxyFL, in which each client maintains two models, a private model, and a publicly shared proxy model. During collaborative learning, clients communicate with others by exchanging their proxy models. The way of sharing the proxy model can effectively deal with the challenge of model heterogeneity. To improve the security of decentralized FL, Li et al. [129] propose a blockchain-based decentralized FL framework, BFLC. The framework leverages blockchain for global model storage and local model update exchange. A committee consisting of honest clients verifies other clients so that the most reliable clients learn from each other, and a small number of malicious client updates are ignored.

4 FUTURE DIRECTIONS

4.1 Improving Communication Efficiency

In FL, the heterogeneous environment can decrease the training efficiency to some extent. Therefore, improving communication efficiency and effectiveness is the focus of many existing approaches [11, 60, 69, 213]. For example, Konečný et al. [106] study two methods, one to learn an update from a finite space, and the other to update the model and send the compressed model to the server. **Communication-Mitigated Federated Learning** (**CMFL**) [143] avoids the transmission of irrelevant updates to the server by measuring whether local updates are consistent with global updates, which effectively decreases the overhead of communication transmission. **Federated Maximum and Mean Discrepancy** (**FedMMD**) [231] decreases the number of communication rounds by introducing the **Maximum Mean Discrepancy** (**MMD**) constraint into the loss function. In addition, Caldas et al. [17] developed a Fed-Dropout scheme by extending the concept presented in [194]. Fed-Dropout derives a small sub-model of the global model for local updates and exchanges the sub-models between the server and the clients to decrease the communication cost. Xiong et al. [224] propose FedDM to construct some synthetic data locally on the client so that it has a similar distribution to the original data on the loss function, and then send these synthetic data to the server for global model update. Compared with transmitting model parameters, transmitting a small amount of synthetic data can effectively reduce communication overhead and increase the amount of information. Dai et al. [37] adopt a decentralized sparse training technique, so that each local model uses a personalized sparse mask to select its own active parameters, and maintains a fixed number of active parameters during local training and peer-to-peer communication. This way, each local model only needs to transmit the index of its active parameters once. In the subsequent communication process, only the values of these active parameters need to be transmitted instead of the entire model, thus greatly reducing the communication overhead. Besides, the multi-layer decentralized FL framework can alleviate the communication load and overhead on the server and client to some extent. However, it still faces challenges such as multi-hop communication delay, unbalanced communication load, and asynchronous communication.

Nevertheless, HFL encounters the following challenges to communication efficiency [8, 185]. The existence of numerous edge nodes can increase computing costs and the required computing power and storage capacity. The differences in the network bandwidth can lead to delays or even losses in sending the local models from clients to the server [209, 234]. In addition, the differences in the sizes of private datasets can also lead to delays in model updates. Thus, a good tradeoff between communication efficiency and model accuracy needs to be guaranteed.

4.2 Federated Fairness

In real-world scenarios, HFL encounters the security issues associated with model fairness [42, 241]. The contribution of participating clients to collaborative learning varies and can be exacerbated by heterogeneity. However, the differences in the contribution of the participating clients in the collaborative process are ignored in most of the existing FL frameworks. In a fair collaborative federated system, clients with more contributions should be able to learn more from other clients and obtain superior models through collaborative training. In the case of general ML, the training data and the training patterns of models may be biased [71], and some data sample groups may be discriminated against. In the FL scenario, there may exist several free-riding participants [54] in the system who want to learn from others in federated communication without providing useful information. Additionally, a global model learned under the constraints common to all clients may be biased towards clients with larger amounts of data or frequent occurrences, and the overall loss function may implicitly advantage or disadvantage certain clients [212]. Therefore, the emphasis on fairness is expected to continue growing with extended practical deployments of FL to more users and enterprises. Furthermore, multi-layer decentralized FL scenarios will face more difficult fairness problems, for example, the tradeoff between fairness and efficiency in multi-layer communication protocols, and the inconsistency of fairness criteria between different layers.

Several recently developed FL paradigms [77, 85, 160] aim at ensuring fairness while maintaining high accuracies. FPFL [57] regards the fairness problem in FL as an optimization problem subject to the constraints of fairness and enhances the system fairness by improving the differential multiplier MMDM. q-FedAvg [127] improves the fairness by narrowing the differences between the accuracies of client models. Besides, CFFL [144] achieves collaborative fairness by assigning models with different performances based on the contributions of each client in FL. However, these methods cannot be well applied to HFL scenarios. Besides, the potential relationship between collaborative fairness and privacy protection [93, 147] needs to be further investigated.

4.3 Privacy Protection

Privacy protection is the first and foremost principle of FL, and the clients typically safeguard basic privacy constraints by never sharing local data [28, 50]. However, the clients may leak private information to the server, for instance, to infer sensitive private data samples as the client has memorized the previous model and gradient updates [151] or information feedback [87]. In addition, many methods, such as sharing few samples during data augmentation, and sharing local data distributions during knowledge transfer, inevitably result in privacy leakage to some extent [29, 45].

Currently, many researches [12, 145, 251] are dedicated to addressing privacy issues in FL. In FL, one of the most popular privacy-preserving techniques is DP, which adds noise to local updates and clips the norm of local updates to preserve the original private information. In FedAvg, larger updates introduce more noise, and a smaller number of iterations decreases the privacy consumption. Therefore, DP-FedAvg [158] applies a Gaussian mechanism to add user-level privacy protection in FedAvg, making large updates by user-level data. To ensure user-level privacy protection without compromising model performance, Cheng et al. [31] improve DP-FedAvg by adding regularization and sparsification processes to the local updates. Similarly, FedMD-NFDP [197] claims that random noise perturbations can alleviate privacy concerns, but this process may sacrifice the model performance. Therefore, FedMD-NFDP adds a noise-free DP mechanism to FedMD [115], which protects data privacy without introducing noise. Moreover, in practical heterogeneous scenarios, different clients or data samples may have varying privacy concerns. Consequently, it is critical to build stricter and more flexible privacy constraint policies, which can measure and set more fine-grained privacy constraints for each client and sample while providing sufficient privacy guarantees. Compared with the general FL scenario, it is more urgent to solve the privacy protection problem in the multi-layer decentralized FL scenario. Since edge devices may communicate directly with their neighbors, this may expose their raw data or model updates to other devices.

Besides, additional privacy concerns regarding personally sensitive information need to be considered when processing and using biometric data in FL. The corresponding solutions include raw data anonymization and feature template protection. Raw data anonymization means that when raw biometric data is preprocessed, a method is adopted so that personally identifiable information or other sensitive information such as gender, age, and health status cannot be extracted from the data [247]. This information may be used by malicious attackers, or correlated with information from other sources, thereby revealing personal privacy. Although existing FL frameworks adopt a data-stay-local policy, this cannot completely prevent local clients or other third parties from accessing and analyzing raw data. Therefore, several methods [65, 235] remove identifiable information through anonymization techniques, thereby protecting local sensitive information from being leaked while maintaining the practicability of published data. However, novel raw data anonymization schemes need to be designed, which can effectively protect the privacy of raw data and retain enough personally identifiable information for model training in FL. Meanwhile, the tradeoff between privacy and practicality of data anonymization will also be a major challenge. Feature template protection means that after feature templates are extracted from the original biometric data, a method is adopted so that the original biometric data or other sensitive information cannot be reconstructed from these feature templates. A feature template is a representation that encodes and compresses raw biometric data. However, recent studies have indicated that raw biometric data can be reconstructed from feature templates [150]. Therefore, an irreversible, updateable, and verifiable feature template protection scheme is needed, while maintaining the distinguishability of the protected feature templates under the FL framework. In summary, in order to improve the security and efficiency of processing and using biometric data in FL systems, the raw data anonymization and the feature template protection in FL need further investigation.

4.4 Attack Robustness

Federated systems may be vulnerable to two major types of attacks [146]: poisoning attacks and inference attacks. (1) Poisoning attacks attempt to prevent the models from being learned and make the learning directions of the models deviate from the original goal. Such attacks involve data poisoning and model poisoning. Data poisoning [13] means that the adversaries compromise the integrity of the training data through methods such as label flipping [56] and backdoor insertion [66, 198], thereby deteriorating the model performance. In model poisoning [9], adversaries cannot directly operate on private data, but they can change the learning direction of the model by destroying client updates. (2) Inference attacks infer information about private user data, thereby compromising user privacy. For example, in the process of parameter transmission, the malicious clients can infer the sensitive data of other clients according to the differences of gradient parameters in each round [5]. Moreover, compared with the server-client FL scenario, there are more intermediate nodes that can be attacked in the multi-layer decentralized FL scenario, so it may face more serious malicious attacks and require stricter defense mechanisms.

Attack methods [63, 138, 205, 244] in FL settings should be studied to improve the attack robustness of federated systems. Xie et al. [222] propose the **Distributed Backdoor Attack** (**DBA**) strategy, in which a global trigger is decomposed into local triggers, and they are injected into multiple malicious clients. Such DBAs are more stealthy and effective than centralized backdoor attacks. Moreover, Nguyen et al. [212] propose edge-case backdoors, which are considered poisoning edge-case samples. Edge-case samples typically represent the tail data of the data distributions and are unlikely to be used as training or test data. Fowl et al. [53] propose an attack method based on the imprint module so that the server directly obtains a copy of the original data from the gradient uploaded by the clients. The imprint module is a special convolutional layer that can directly copy the input feature map to the output feature map so that the original input information is contained in the gradient.

These attacks pose a significant threat to FL, and thus, several defense strategies [153, 169, 195, 254] have been developed to enhance system robustness. Xie et al. [221] improve the robustness against backdoor attacks by clipping the model and adding smooth noise. To counteract model poisoning attacks, Li et al. [123] learn a detection model on the server side to identify and remove malicious model updates, thereby removing irrelevant features while retaining valid basic features. Wu et al. [217] propose a robust blockchain multi-layer decentralized FL framework, RBML-DFL, which can prevent central server failures or malfunctions through blockchain encrypted transactions. To deal with inference attacks, ResSFL [116] is trained by experts through attacker perception to obtain a resistant feature extractor that can initialize the client models. Besides, Soteria [196] performs attack defense by generating perturbed data representations, thereby decreasing the quality of reconstructed data. Several methods [161, 255] implement attack detection to proactively identify the malicious intrusions in federated systems. In BaFFle [4], the server trains backdoor filters and sends them randomly to clients to detect backdoor instances. The client then removes the backdoor instances from the training data, thereby training a backdoor-free local model. Different from other methods that are studied in the federated setting, Abeshu et al. [2] apply FL to network attack detection, and propose a network attack detection model based on FL-authorized edge network. This enables edge devices to learn from each other without sharing data to improve the accuracy of network detection attacks. Recently, hardware-based **Trusted Execution Environment** (**TEE**) that allocates an isolated block of memory for private computing of sensitive data has attracted significant attention from the industry. Unlike general privacy-preserving technologies, TEE is committed to providing a secure platform for FL with low computational overhead and high computational efficiency, and protecting models from inference attacks. Therefore, it is suitable for FL scenarios with limited computing resources [65]. Mo et al. [159] propose a privacy-preserving FL framework for mobile systems, PPFL, which uses TEE for secure training of local models and secure aggregation on the server, thereby hiding gradient updates and preventing adversary attacks. Zhang et al. [253] utilize TEE to protect the integrity and privacy of gradients and prevent adversary models from inferring or modifying gradients through side-channel attacks. Specifically, they use TEE for local training, randomly group gradients into gradient segments for encryption, and then send them to the server. The server decrypts and securely aggregates gradient segments from the same group with TEE.

The inference attacks aim at obtaining the information of the clients, and thus, inevitably threaten user privacy and security. Therefore, the attack defense strategies can decrease the risk of privacy leakage. Several privacy-preserving mechanisms can not only reduce inference attacks but also improve robustness against poisoning attacks. Future work should be aimed at exploring the close relationships between attack robustness and privacy protection to enable a better tradeoff between these two aspects.

4.5 Uniform Benchmarks

The FL concept was first proposed by McMahan et al. [157] in 2016. As this field has been developed for a relatively short period of time, there is a lack of widely recognized benchmark datasets and benchmark testing frameworks for heterogeneous scenarios. The development of unified benchmark datasets and benchmark testing frameworks can facilitate the reproduction of experimental results and widespread application of novel algorithms. Heterogeneous benchmark frameworks provide various possibilities for client-side data distributions and model structures. Moreover, the statistical and model discrepancies of different clients can validate the generalization ability of HFL algorithms to some extent. Systematic evaluation indicators should be developed to promote the research and development of HFL. The indicators can fairly and comprehensively evaluate the security, convergence, accuracy, and generalization ability of different algorithms. Therefore, benchmark is very important to promote the development of the HFL field. Several recent works have explored benchmark frameworks and datasets, which we group into the following three categories:

General Federated Learning Systems. FedML [74] is a research library that supports distributed training, mobile on-device training, and stand-alone simulation training. It provides standardized implementations of many existing FL algorithms and provides standardized benchmark settings for a variety of datasets, including Non-IID partition methods, number of devices, and baseline models. FedScale [111] is an FL benchmark suite that provides real-world datasets covering a wide range of FL tasks, including image classification, object detection, language modeling, and speech recognition. Additionally, FedScale includes a scalable and extensible FedScale Runtime to enable and standardize real-world end-point deployments of FL. OARF [82] leverages public datasets collected from different sources to simulate real-world data distributions. In addition, OARF quantitatively studies the preliminary relationship among various design metrics such as data partitioning and privacy mechanisms in FL systems. FedEval [19] is an FL evaluation model with five metrics including accuracy, communication, time consumption, privacy, and robustness. FedEval is implemented and evaluated on two of the most widely used algorithms, FedSGD and FedAvg.

Specific Federated Learning Systems. FedReIDBench [259] is a new benchmark for implementing FL to person ReID, which includes nine different datasets and two federated scenarios. Specifically, the two federated scenarios are federated-by-camera scenario and federated-by-dataset scenario, which respectively represent the standard server-client architecture and client-edge-cloud architecture. pFL-Bench [22] is a benchmark for personalized FL, which covers twelve different dataset variants, including image, text, graph, and recommendation data, with unified data partitioning and realistic heterogeneous settings. And pFL-Bench provides more than 20 competitive personalized FL baseline implementations to help them with standardized evaluation. FedGraphNN [73] is a benchmark system built on a unified formulation of graph FL, including extensive datasets from seven different fields, popular **Graph Neural Network** (**GNN**) models, and FL algorithms [86].

Datasets. LEAF [16] contains six types of federated datasets covering different fields, including image classification (FEMNIST, Synthetic Dataset), image recognition (Celeba), sentiment analysis (Sentiment140) and next character prediction (Shakespeare, Reddit). In addition, LEAF provides two sampling methods of “IID” and “Non-IID” to divide the dataset to different clients. Luo et al. [140] introduce a federated dataset for object detection. The dataset contains over 900 images generated from 26 street cameras and 7 object categories annotated with detailed bounding boxes. Besides, the article provides the data division of 5 or 20 clients, in which their data distribution is Non-IID and unbalanced, reflecting the characteristics of real-world FL scenarios.

Although several FL benchmarks are devised, more realistic datasets containing extensive ML tasks should be established to facilitate the development of FL. In heterogeneous scenarios, the scales and the distributions of local data on different clients may differ significantly, and it is generally difficult to benchmark the Non-IID performance under different algorithms. The Non-IID setting requires a sufficient understanding of the statistical heterogeneity in real-world scenarios, including the four statistical heterogeneity cases we discussed in Section 2.1. Therefore, developing a realistic benchmark framework that incorporates the four heterogeneities is a challenging task.

5 CONCLUSION

This survey aims to provide a comprehensive and systematic understanding of HFL. First, a detailed overview of the research challenges in HFL is demonstrated in Section 2. Different from existing surveys on FL, we focus on the heterogeneity problems in FL and categorize them into four categories: statistical heterogeneity, model heterogeneity, communication heterogeneity, and device heterogeneity. Subsequently, we survey the recently published and pre-printed articles on HFL and provide a reasonable and comprehensive taxonomy of existing techniques in Section 3. This taxonomy divides the state-of-the-art methods into three different levels: data level, model level, and server level. Finally, we provide an outlook analysis on the directions worthy of further exploration and open problems for future development in HFL in Section 4. We believe that these valuable discussions can promote the high-quality development of the HFL community.

APPENDIX

Table 1. Related FL Surveys

Surveys	Key Contributions	Differences from Our Survey
Kairouz et al. [98]	They discuss recent advances in FL and provide a survey of open problems and challenges.	This work comprehensively demonstrates the advance in FL, but lacks the fine-grained classification of existing methods.
Li et al. [125]	They discuss the challenges of FL from the perspectives of efficiency, heterogeneity and privacy, and list several future directions.	Our survey focuses on the challenges of heterogeneity and provides a more comprehensive and detailed classification of heterogeneity.
Wahab et al. [210]	They provide a fine-grained classification scheme of existing challenges and approaches.	Our survey focuses on the challenges of heterogeneity and provides a more comprehensive classification of heterogeneity.
Li et al. [121]	They provide a comprehensive analysis on FL from systems perspective, including system components, taxonomy, summary, design, and vision.	The taxonomy proposed in this work is not based on a uniform standard.
Lim et al. [131]	They survey FL in mobile edge networks.	This work is based on FL in mobile edge network optimization, while our work investigates from a more general perspective.
Niknam et al. [167]	They discuss the applications and challenges of FL in wireless communication environments.	This work discusses applications of FL in wireless communication, while we surveys FL from a more general perspective.
Khan et al. [102]	They present advances in FL for IoT applications and provide a taxonomy using various parameters.	This work explores FL for IoT networks and identifies issues of robustness, privacy and communication cost, while our work targets heterogeneity issues
Nguyen et al. [164]	They survey and analyze the services applications of FL in IoT networks.	This work focuses on the characteristics and requirements of IoT networks, rather than covering all possible FL scenarios.
Yang et al. [229]	They divide FL into three categories according to the data distribution characteristics.	This work provides an overview of FL but lacks a detailed classification and summary of existing methods.
Gao et al. [59]	They investigate the domain of heterogeneous FL in terms of data space, statistical, system, and model heterogeneity.	This work classifies existing methods based on problem settings and learning objectives, while our survey classifies methods based on specific techniques.
Zhu et al. [257]	They analyze the impact of non-IID data in FL and provide a survey of the research on handling non-IID data.	This work analyzes the impact of Non-IID data on FL, but ignores other challenges and related research.
Kulkarni et al. [108]	They point out that statistical heterogeneity can hinder FL and highlight the need for personalization.	This work focuses on the challenges posed by statistical heterogeneity while ignoring other issues.
Tan et al. [200]	They explore the field of personalized FL and conduct a taxonomic survey of existing methods.	This work briefly explains statistical heterogeneity, but lacks a comprehensive taxonomy and analysis of the challenges in FL.
Wu et al. [219]	They provide a personalized FL framework in a cloud-edge architecture for intelligent IoT applications.	This work focuses on personalized FL schemes, whereas our survey encompasses broader FL schemes.

Table 2. Data-level Methods

Methods		Advantages	Ref.	Key Contributions	Limitations
Private Data Processing 3.1.1	Data Preparation	These methods effectively improve the quality and security of private data, and alleviate statistical heterogeneity.	[227]	Safe detects and filters out poisoned data from attacked devices through a clustering algorithm.	Filtering poisoned data by clustering requires proper selection of attacking clients.
			[237]	FedMix performs data augmentation based on the MixUp strategy.	Collecting local data distributions may bring potential information leakage.
			[43]	Astraea performs data augmentation based on the global data distribution, generated by collecting the local data distribution.	Uploading local data distributions may expose potential backdoors to attacks.
			[94]	FAug studies the tradeoff between privacy leakage and communication overhead through a GAN-based data augmentation scheme.	Transferring local data to the server violates the privacy requirements of FL.
	Data Privacy Protection	These methods ensure the local data privacy.	[81]	PLDP-PFL performs personalized differential privacy according to the sensitivity of private data.	DP can degrade model performance due to its clipping and noise-adding operations
			[33]	They use anonymization technology to desensitize local private data.	Anonymization processing may may reduce data quality and availability.
External Data Utilization 3.1.2	Knowledge Distillation	These methods overcome model heterogeneity to achieve communication between heterogeneous models and effectively decrease communication overhead.	[115]	FedMD enables clients to independently design their models and implements communication between heterogeneous models.	Global knowledge cannot describe rich domain knowledge in feature skew scenarios.
			[239]	They mitigate overfitting in personalized updates by enhancing the logits similarity between the global model and the local models.	Logits Exploitation may lead to insufficient learning of local information.
			[72]	FedGKT extracts knowledge on resource-constrained edge devices through knowledge distillation.	Uploading prediction vectors to the server may not satisfy DP guarantees.
			[248]	FedFTG trains a conditional generator to fit the input space of a local model, and uses it to generate pseudo data.	Training the generator and using FL-div to learn the global knowledge may increase the computation cost.
	Unsupervised Representation Learning	These approaches enable local models to learn consistent representations while keeping private data decentralized and unlabeled.	[242]	A novel problem named FURL is proposed. And the corresponding solution algorithm, namely FedCA, is designed.	Directly sharing the features of local data may introduce potential privacy risks.
			[120]	MOON solves the Non-IID data issues through model-based contrastive learning.	Storing multiple models simultaneously may require significant additional resources.
			[163]	FedProc mitigates statistical heterogeneity through prototype-based contrastive learning.	Unprocessed logits transmissions may lead to privacy leakage of local data.

Table 3. Model-level Methods

Methods		Advantages	Ref.	Key Contributions	Limitations
Federated Optimization 3.2.1	Regularization	These methods provide convergence guarantee under statistical heterogeneity.	[126]	FedProx is a federated optimization algorithm that adds a proximal term to FedAvg.	Adding a regularization term may result in slower convergence.
			[192]	FedCurv uses the EWC algorithm to prevent catastrophic forgetting when transferring tasks.	It may ignore the differences in the extent to which clients are affected by catastrophic forgetting.
			[199]	pFedME utilizes the Moreau envelope function as a regularized loss function.	Tuning the regularization parameters may be labour intensive.
	Meta-learning	These methods use meta-learning to achieve the personalized objectives under HFL.	[96]	Similarities between the MAML setting and the personalized objectives of HFL are pointed out.	The two stages of meta-training and meta-testing may introduce additional communication overhead.
			[46]	Per-FedAvg is a personalized variant of FedAvg algorithm based on the MAML formula.	It may not apply to scenarios with significant feature skew.
	Multi-task Learning	These methods leverage multi-task learning to learn personalized local models.	[193]	A system-aware optimization framework for FMTL is built.	It cannot be applied to non-convex deep learning models.
			[124]	Ditto is a scalable federated multi-task learning framework with two tasks: global goal and local goal.	Tuning the regularization parameters may take much effort.
Knowledge Transfer 3.2.2	Knowledge Distillation	These methods refine knowledge distribution of each client.	[48]	RHFL uses irrelevant data for knowledge distillation, thereby solving the problem of model heterogeneity.	Using logits output on irrelevant data as local knowledge may underutilize local information.
	Transfer Learning	These methods transfer the local knowledge in a model-agnostic manner.	[245]	FT-pFL achieves personalized knowledge transfer via a knowledge coefficient matrix.	The logits output on public dataset may not describe rich local information.
			[30]	FedHealth is a federated transfer learning framework applied in the healthcare domain.	Its generality may be decreased owing to the model heterogeneity in real medical scenarios.
Architecture Sharing 3.2.3	Backbone Sharing	These methods decrease computation costs while satisfying personalized demands.	[6]	FedPer combines base layers and personalized layers for federated training.	Excessive resource may be consumed owing to the activation of all clients in each round.
			[35]	FedRep enables all clients jointly train global representation learning structure, and then uses private data to train their own heads.	The base layers learn a global representation that may limit the personalization.
	Classifier Sharing	These methods leverage the personalized layers to extract features.	[130]	LG-FedAvg uses personalized layers to extract high-level features and server-shared base layers for classification.	Assuming that all clients have sufficient training data may decrease the generality.
	Other Part Sharing	These methods share part of the model according to local conditions.	[40]	HeteroFL allocates local models of different sizes according to the computational and communication capabilities of each client.	It may not satisfy practical scenarios with high model heterogeneity.

Table 4. Server-level Methods

Methods	Advantages	Ref.	Key Contributions	Limitations
Client Selection 3.3.1	These methods accelerate convergence by formulating client selection strategies.	[211]	Favor is an experience-driven control framework that actively selects the best subset of clients to participate in FL iterations.	Training reinforcement learning models may be data-hungry.
		[228]	CUCB is a client selection algorithm that minimizes the class imbalance and facilitates the global model convergence.	Revealing the class distribution based on updated gradients may be vulnerable to inference attacks.
		[117]	FedSAE estimates the reliability of each device and performs client selection based on training losses.	Adjusting workloads based on the training history of clients may be delayed.
		[168]	FedCS performs client selection operations based on data resources, computing capabilities, and wireless channel conditions.	Estimating training time for complex models may be difficult.
Client Clustering 3.3.2	These methods enhance HFL efficiency by personalized clustering clients.	[15]	FL+HC introduces a hierarchical clustering step to separate client clusters based on the similarity of client updates to the global joint model.	The effect of communication heterogeneity and device heterogeneity may be ignored.
		[223]	FeSEM employs SEM optimization to calculate the distance between local models and cluster centers.	Compared with single-center clustering, multi-center clustering may require higher storage capability.
		[182]	CFL clusters similar clients by the cosine similarity between their gradient updates.	It is vulnerable to backdoor attacks in FL.
		[165]	FLAME detects adversarial model updates through a clustering strategy that limits the noise scale of backdoor noise removal.	Building trusted server in practical settings may be challenging.
Decentralized Communication 3.3.3	These methods can effectively reduce the reliance on the secure central server and alleviate the communication bottleneck.	[178]	BrainTorrent randomly selects a client as a temporary server in each round, and then coordinates updates with other clients.	It requires high computational and storage resources for temporary servers.
		[80]	Combo divides the local model into model segments, and then randomly selects some clients to transfer the model segments.	Transferring model segments alleviate communication delays, but do not reduce overall communication overhead.
		[100]	ProxyFL makes each client maintain two models, a private model and a publicly shared proxy model for exchanges.	Proxy models may not capture all the information or complexity of private models.
		[129]	BFLC utilizes the blockchain for global model storage and local model update exchange to enhance the security of FL.	Maintaining and validating blockchain ledgers can incur high computational and storage costs.

REFERENCES

[1] Sawsan Abdul Rahman, Hanine Tout, Azzam Mourad, and Chamseddine Talhi. 2021. FedMCCS: Multicriteria client selection model for optimal IoT federated learning. *IEEE IoT-J* 8, 6 (2021), 4723–4735.

[2] Abebe Abeshu and Naveen Chilamkurti. 2018. Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Communications Magazine* 56, 2 (2018), 169–175.

[3] Nichol Alex, Achiam Joshua, and Schulman John. 2018. On first-order meta-learning algorithms. arXiv:1803.02999. Retrieved from https://arxiv.org/abs/1803.02999

[4] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. 2021. Baffle: Backdoor detection via feedback-based federated learning. In *ICDCS*.

[5] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE TIFS* 13, 5 (2018), 1333–1345.

[6] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated learning with personalization layers. arXiv:1912.00818. Retrieved from https://arxiv.org/abs/1912.00818

[7] Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. 2020. FedOpt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences* 10, 8 (2020), 2864.

[8] Muhammad Asad, Ahmed Moustafa, Takayuki Ito, and Muhammad Aslam. 2021. Evaluating the communication efficiency in federated learning algorithms. In *IEEE CSCWD*.

[9] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *AISTATS*.

[10] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. 2022. Diverse client selection for federated learning via submodular maximization. In *ICLR*.

[11] Sameer Bibikar, Haris Vikalo, Zhangyang Wang, and Xiaohan Chen. 2022. Federated dynamic sparse training: Computing less, communicating less, yet learning better. In *AAAI*.

[12] Alberto Bietti, Chen-Yu Wei, Miroslav Dudik, John Langford, and Steven Wu. 2022. Personalization improves privacy-accuracy tradeoffs in federated learning. In *ICML*.

[13] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *ICML*.

[14] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards federated learning at scale: System design. *SysML* 1 (2019), 374–388.

[15] Christopher Briggs, Zhong Fan, and Peter Andras. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *IJCNN*.

[16] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2019. Leaf: A benchmark for federated settings. In *NeurIPS Workshop*.

[17] Sebastian Caldas, Jakub Konečny, H. Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. arXiv:1812.07210. Retrieved from https://arxiv.org/abs/1812.07210

[18] Olivier Cappé and Eric Moulines. 2009. On-line expectation–maximization algorithm for latent data models. *J R STAT SOC B* 71, 3 (2009), 593–613.

[19] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. FedEval: A Holistic Evaluation Framework for Federated Learning. arXiv:2011.09655. Retrieved from https://arxiv.org/abs/2011.09655

[20] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. 2020. Tifl: A tier-based federated learning system. In *ACM HPDC*.

[21] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. 2019. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. arXiv:1912.11279. Retrieved from https://arxiv.org/abs/1912.11279

[22] Daoyuan Chen, Dawei Gao, Weirui Kuang, Yaliang Li, and Bolin Ding. 2022. pFL-Bench: A comprehensive benchmark for personalized federated learning. In *NeurIPS Track on Datasets and Benchmarks*.

[23] Dengsheng Chen, Jie Hu, Vince Junkai Tan, Xiaoming Wei, and Enhua Wu. 2023. Elastic aggregation for federated optimization. In *CVPR*.

[24] Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. 2023. Efficient personalized federated learning via sparse model-adaptation. In *ICML*.

[25] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning with fast convergence and efficient communication. arXiv:1802.07876. Retrieved from https://arxiv.org/abs/1802.07876

[26] Mingzhe Chen, Nir Shlezinger, H. Vincent Poor, Yonina C. Eldar, and Shuguang Cui. 2021. Communication-efficient federated learning. *PNAS* 118, 17 (2021), e2024789118.

[27] Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. 2020. Vafl: A method of vertical asynchronous federated learning. In *ICML Workshop*.

[28] Wei-Ning Chen, Christopher A. Choquette Choo, Peter Kairouz, and Ananda Theertha Suresh. 2022. The fundamental price of secure aggregation in differentially private federated learning. In *ICML*.

[29] Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. 2022. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In *ICML*.

[30] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems* 35, 4 (2020), 83–93.

[31] Anda Cheng, Peisong Wang, Xi Sheryl Zhang, and Jian Cheng. 2022. Differentially private federated learning with local regularization and sparsification. In *CVPR*.

[32] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. 2020. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. arXiv:2010.01243. Retrieved from https://arxiv.org/abs/2010.01243

[33] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2020. Anonymizing data for privacy-preserving federated learning. In *ECAI*.

[34] Yun-Wei Chu, Seyyedali Hosseinalipour, Elizabeth Tenorio, Laura Cruz, Kerrie Douglas, Andrew Lan, and Christopher Brinton. 2022. Multi-layer personalized federated learning for mitigating biases in student predictive analytics. arXiv:2212.02985. Retrieved from https://arxiv.org/abs/2212.02985

[35] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *ICML*.

[36] Luca Corinzia, Ami Beuret, and Joachim M. Buhmann. 2019. Variational federated multi-task learning. arXiv:1906.06268. Retrieved from https://arxiv.org/abs/1906.06268

[37] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. 2022. DisPFL: Towards communication-efficient personalized federated learning via decentralized sparse training. In *ICML*.

[38] Wei Dai, Abhimanu Kumar, Jinliang Wei, Qirong Ho, Garth Gibson, and Eric Xing. 2015. High-performance distributed ML at scale through parameter server consistency models. In *AAAI*.

[39] Ittai Dayan, Holger R. Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z. Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J. Wood, Chien-Sung Tsai, Chih-Hung Wang, Chun-Nan Hsu, C. K. Lee, Peiying Ruan, Daguang Xu, Dufan Wu, Eddie Huang, Felipe Campos Kitamura, Griffin Lacey, Gustavo César de Antônio Corradi, Gustavo Nino, Hao-Hsin Shin, Hirofumi Obinata, Hui Ren, Jason C. Crane, Jesse Tetreault, Jiahui Guan, John W. Garrett, Joshua D. Kaggie, Jung Gil Park, Keith Dreyer, Krishna Juluru, Kristopher Kersten, Marcio Aloisio Bezerra Cavalcanti Rockenbach, Marius George Linguraru, Masoom A. Haider, Meena AbdelMaseeh, Nicola Rieke, Pablo F. Damasceno, Pedro Mario Cruz e Silva, Pochuan Wang, Sheng Xu, Shuichi Kawano, Sira Sriswasdi, Soo Young Park, Thomas M. Grist, Varun Buch, Watsamon Jantarabenjakul, Weichung Wang, Won Young Tak, Xiang Li, Xihong Lin, Young Joon Kwon, Abood Quraini, Andrew Feng, Andrew N. Priest, Baris Turkbey, Benjamin Glicksberg, Bernardo Bizzo, Byung Seok Kim, Carlos Tor-Díez, Chia-Cheng Lee, Chia-Jung Hsu, Chin Lin, Chiu-Ling Lai, Christopher P. Hess, Colin Compas, Deepeksha Bhatia, Eric K. Oermann, Evan Leibovitz, Hisashi Sasaki, Hitoshi Mori, Isaac Yang, Jae Ho Sohn, Krishna Nand Keshava Murthy, Li-Chen Fu, Matheus Ribeiro Furtado de Mendonça, Mike Fralick, Min Kyu Kang, Mohammad Adil, Natalie Gangai, Peerapon Vateekul, Pierre Elnajjar, Sarah Hickman, Sharmila Majumdar, Shelley L. McLeod, Sheridan Reed, Stefan Gräf, Stephanie Harmon, Tatsuya Kodama, Thanyawee Puthanakit, Tony Mazzulli, Vitor Lima de Lavor, Yothin Rakvongthai, Yu Rim Lee, Yuhong Wen, Fiona J. Gilbert, Mona G. Flores, and Quanzheng Li. 2021. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature Medicine* 27, 10 (2021), 1735–1743.

[40] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *ICLR*.

[41] Canh T. Dinh, Tung T. Vu, Nguyen H. Tran, Minh N. Dao, and Hongyu Zhang. 2021. FedU: A unified framework for federated multi-task learning with laplacian regularization. arXiv:2102.07148. Retrieved from https://arxiv.org/abs/2102.07148

[42] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. 2021. Fairness-aware agnostic federated learning. In *SDM*.

[43] Moming Duan, Duo Liu, Xianzhang Chen, Yujuan Tan, Jinting Ren, Lei Qiao, and Liang Liang. 2019. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *ICCD*.

[44] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. 2021. FedGroup: Efficient federated learning via decomposed similarity-based clustering. In *IEEE ISPA*.

[45] Anis Elgabli, Chaouki Ben Issaid, Amrit Singh Bedi, Ketan Rajawat, Mehdi Bennis, and Vaneet Aggarwal. 2022. FedNew: A communication-efficient and privacy-preserving newton-type method for federated learning. In *ICML*.

[46] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*.

[47] Chen Fang, Yuanbo Guo, Na Wang, and Ankang Ju. 2020. Highly efficient federated learning with strong privacy preservation in cloud computing. *Computers and Security* 96 (2020), 101889.

[48] Xiuwen Fang and Mang Ye. 2022. Robust federated learning with noisy and heterogeneous clients. In *CVPR*.

[49] Xiuwen Fang, mang Ye, and Xiyuan Yang. 2023. Robust heterogeneous federated learning under data corruption. In *ICCV*.

[50] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. 2020. PMF: A privacy-preserving human mobility prediction framework via federated learning. *ACM IMWUT* 4, 1 (2020), 1–21.

[51] Angelo Feraudo, Poonam Yadav, Vadim Safronov, Diana Andreea Popescu, Richard Mortier, Shiqiang Wang, Paolo Bellavista, and Jon Crowcroft. 2020. CoLearn: Enabling federated learning in MUD-compliant IoT edge networks. In *ACM EdgeSys*.

[52] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

[53] Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. 2022. Robbing the fed: Directly obtaining private data in federated learning with modified models. In *ICLR*.

[54] Yann Fraboni, Richard Vidal, and Marco Lorenzi. 2021. Free-rider attacks on model aggregation in federated learning. In *AISTATS*.

[55] David Froelicher, Juan R. Troncoso-Pastoriza, Jean Louis Raisaro, Michel A. Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. 2021. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature Communications* 12, 1 (2021), 5910.

[56] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2018. Mitigating sybils in federated learning poisoning. arXiv:1808.04866. Retrieved from https://arxiv.org/abs/1808.04866

[57] Borja Rodríguez Gálvez, Filip Granqvist, Rogier van Dalen, and Matt Seigel. 2021. Enforcing fairness in private federated learning via the modified method of differential multipliers. In *NeurIPS Workshop*.

[58] Dashan Gao, Yang Liu, Anbu Huang, Ce Ju, Han Yu, and Qiang Yang. 2019. Privacy-preserving heterogeneous federated transfer learning. In *IEEE Big Data*.

[59] Dashan Gao, Xin Yao, and Qiang Yang. 2022. A survey on heterogeneous federated learning. arXiv:2210.04505. Retrieved from https://arxiv.org/abs/2210.04505

[60] Hongchang Gao, An Xu, and Heng Huang. 2021. On the convergence of communication-efficient local SGD for federated learning. In *AAAI*.

[61] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. In *NeurIPS*.

[62] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. 2021. Shuffled model of differential privacy in federated learning. In *AISTATS*.

[63] Xueluan Gong, Yanjiao Chen, Huayang Huang, Yuqing Liao, Shuai Wang, and Qian Wang. 2022. Coordinated backdoor attacks against federated learning with model-dependent triggers. *IEEE Network* 36, 1 (2022), 84–90.

[64] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*.

[65] Matei Grama, Maria Musat, Luis Muñoz-González, Jonathan Passerat-Palmbach, Daniel Rueckert, and Amir Alansary. 2020. Robust aggregation for adaptive privacy preserving federated learning in healthcare. arXiv:2009.08294. Retrieved from https://arxiv.org/abs/2009.08294

[66] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv:1708.06733. Retrieved from https://arxiv.org/abs/1708.06733

[67] Xu Guo, Han Yu, Boyang Li, Hao Wang, Pengwei Xing, Siwei Feng, Zaiqing Nie, and Chunyan Miao. 2022. Federated learning for personalized humor recognition. *ACM TIST* 13, 4 (2022), 1–18.

[68] Gamze Gürsoy, Tianxiao Li, Susanna Liu, Eric Ni, Charlotte M. Brannon, and Mark B. Gerstein. 2022. Functional genomics data: Privacy risk assessment and technological mitigation. *Nature Reviews Genetics* 23, 4 (2022), 245–258.

[69] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. 2020. Fedboost: A communication-efficient algorithm for federated learning. In *ICML*.

[70] Filip Hanzely and Peter Richtárik. 2020. Federated learning of a mixture of global and local models. arXiv:2002.05516. Retrieved from https://arxiv.org/abs/2002.05516

[71] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*.

[72] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group knowledge transfer: Federated learning of large cnns at the edge. In *NeurIPS*.

[73] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S. Yu, Yu Rong, Peilin Zhao, Junzhou Huang, Murali Annavaram, and Salman Avestimehr. 2021. Fedgraphnn: A federated learning system and benchmark for graph neural networks. In *ICLR Workshop*.

[74] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. Fedml: A research library and benchmark for federated machine learning. arXiv:2007.13518. Retrieved from https://arxiv.org/abs/2007.13518

[75] Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. 2022. Efficient split-mix federated learning for on-demand and in-situ customization. In *ICLR*.

[76] Robert Hönig, Yiren Zhao, and Robert Mullins. 2022. DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning. In *ICML*.

[77] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. 2021. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. In *NeurIPS*.

[78] Seyyedali Hosseinalipour, Christopher G. Brinton, Vaneet Aggarwal, Huaiyu Dai, and Mung Chiang. 2020. From federated to fog learning: Distributed machine learning over heterogeneous wireless networks. *IEEE Communications Magazine* 58, 12 (2020), 41–47.

[79] Charlie Hou, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. 2022. FedChain: Chained algorithms for near-optimal communication cost in federated learning. In *ICLR*.

[80] Chenghao Hu, Jingyan Jiang, and Zhi Wang. 2019. Decentralized federated learning: A segmented gossip approach. In *FML Workshop*.

[81] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. 2020. Personalized federated learning with differential privacy. *IEEE IoT-J* 7, 10 (2020), 9530–9539.

[82] Sixu Hu, Yuan Li, Xu Liu, Qinbin Li, Zhaomin Wu, and Bingsheng He. 2022. The oarf benchmark suite: Characterization and implications for federated learning systems. *ACM TIST* 13, 4 (2022), 1–32.

[83] Tiansheng Huang, Weiwei Lin, Li Shen, Keqin Li, and Albert Y. Zomaya. 2022. Stochastic client selection for federated learning with volatile clients. *IEEE IoT-J* 9, 20 (2022), 20055–20070.

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

[85] Wei Huang, Tianrui Li, Dexian Wang, Shengdong Du, and Junbo Zhang. 2020. Fairness and accuracy in federated learning. arXiv:2012.10069. Retrieved from https://arxiv.org/abs/2012.10069

[86] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. 2023. Federated graph semantic and structural learning. In *IJCAI*.

[87] Wenke Huang, Mang Ye, and Bo Du. 2022. Learn from Others and Be Yourself in Heterogeneous Federated Learning. In *CVPR*.

[88] Wenke Huang, Mang Ye, Xiang Gao, and Bo Du. 2022. Few-shot model agnostic federated learning. In *ACM Multimedia*.

[89] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. 2023. Rethinking federated learning with domain shift: A prototype view. In *CVPR*.

[90] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-iid data. In *AAAI*.

[91] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. 2022. FedPara: Low-rank hadamard product for communication-efficient federated learning. In *ICLR*.

[92] Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I. Jordan. 2014. Communication-efficient distributed dual coordinate ascent. In *NeurIPS*.

[93] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. 2019. Differentially private fair learning. In *ICML*.

[94] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *NeurIPS*.

[95] Menglin Jia, Zuxuan Wu, Austin Reiter, Claire Cardie, Serge Belongie, and Ser-Nam Lim. 2021. Intentonomy: A dataset and study towards human intent understanding. In *CVPR*.

[96] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. 2019. Improving federated learning personalization via model agnostic meta learning. arXiv:1909.12488. Retrieved from https://arxiv.org/abs/1909.12488

[97] Peter Kairouz, Ziyu Liu, and Thomas Steinke. 2021. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *ICML*.

[98] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[99] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, Andreas Saleh, Marcus Makowski, Daniel Rueckert, and Rickmer Braren. 2021. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence* 3, 6 (2021), 473–484.

[100] Shivam Kalra, Junfeng Wen, Jesse C. Cresswell, Maksims Volkovs, and H. R. Tizhoosh. 2023. Decentralized federated learning through proxy model sharing. *Nature Communications* 14, 1 (2023), 2899.

[101] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*.

[102] Latif U. Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. 2021. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys and Tutorials* 23, 3 (2021), 1759–1799.

[103] Mikhail Khodak, Maria-Florina F. Balcan, and Ameet S. Talwalkar. 2019. Adaptive gradient-based meta-learning methods. In *NeurIPS*.

[104] Jinkyu Kim, Geeho Kim, and Bohyung Han. 2022. Multi-level branched regularization for federated learning. In *ICML*.

[105] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* 114, 13 (2017), 3521–3526.

[106] Jakub Konečnỳ, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. arXiv:1610.05492. Retrieved from https://arxiv.org/abs/1610.05492

[107] Kavya Kopparapu and Eric Lin. 2020. Fedfmc: Sequential efficient federated learning on non-iid data. arXiv:2006.10937. Retrieved from https://arxiv.org/abs/2006.10937

[108] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. 2020. Survey of personalization techniques for federated learning. In *IEEE WorldS4*.

[109] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. 2020. Device heterogeneity in federated learning: A superquantile approach. arXiv:2002.11223. Retrieved from https://arxiv.org/abs/2002.11223

[110] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. 2021. A superquantile approach to federated learning with heterogeneous devices. In *IEEE CISS*.

[111] Fan Lai, Yinwei Dai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2022. FedScale: Benchmarking model and system performance of federated learning at scale. In *ICML*.

[112] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. 2018. Fully decentralized federated learning. In *NeurIPS Workshop*.

[113] Anran Li, Lan Zhang, Juntao Tan, Yaxuan Qin, Junhao Wang, and Xiang-Yang Li. 2021. Sample-level data selection for federated learning. In *IEEE INFOCOM*.

[114] Chengxi Li, Gang Li, and Pramod K. Varshney. 2021. Decentralized federated learning via mutual knowledge transfer. *IEEE IoT-J* 9, 2 (2021), 1136–1147.

[115] Daliang Li and Junpu Wang. 2019. Fedmd: Heterogenous federated learning via model distillation. In *NeurIPS Workshop*.

[116] Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. 2022. ResSFL: A resistance transfer framework for defending model inversion attack in split federated learning. In *CVPR*.

[117] Li Li, Moming Duan, Duo Liu, Yu Zhang, Ao Ren, Xianzhang Chen, Yujuan Tan, and Chengliang Wang. 2021. FedSAE: A novel self-adaptive federated learning framework in heterogeneous systems. In *IJCNN*.

[118] Li Li, Jun Wang, Xu Chen, and Cheng-Zhong Xu. 2020. Multi-layer coordination for high-performance energy-efficient federated learning. In *IWQoS*.

[119] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *IEEE ICDE*.

[120] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *CVPR*.

[121] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2023. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE TKDE* 35, 4 (2023), 3347–3366.

[122] Rui Li, Fenglong Ma, Wenjun Jiang, and Jing Gao. 2019. Online federated multitask learning. In *IEEE Big Data*.

[123] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. 2020. Learning to detect malicious clients for robust federated learning. arXiv:2002.00211. Retrieved from https://arxiv.org/abs/2002.00211

[124] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *ICML*.

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

[126] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *MLSys*.

[127] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2020. Fair resource allocation in federated learning. In *ICLR*.

[128] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. Fedbn: Federated learning on non-iid features via local batch normalization. In *ICLR*.

[129] Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. 2020. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network* 35, 1 (2020), 234–241.

[130] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B. Allen, Randy P. Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2019. Think locally, act globally: Federated learning with local and global representations. In *NeurIPS Workshop*.

[131] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials* 35, 1 (2020), 234–241.

[132] Wei Yang Bryan Lim, Jer Shyuan Ng, Zehui Xiong, Jiangming Jin, Yang Zhang, Dusit Niyato, Cyril Leung, and Chunyan Miao. 2021. Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning. *IEEE TPDS* 33, 3 (2021), 536–550.

[133] Wei Yang Bryan Lim, Jer Shyuan Ng, Zehui Xiong, Dusit Niyato, Chunyan Miao, and Dong In Kim. 2021. Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks. *IEEE JSAC* 39, 12 (2021), 3640–3653.

[134] Sen Lin, Guang Yang, and Junshan Zhang. 2020. A collaborative learning framework via federated meta-learning. In *IEEE ICDCS*.

[135] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*.

[136] Lumin Liu, Jun Zhang, S. H. Song, and Khaled B. Letaief. 2020. Client-edge-cloud hierarchical federated learning. In *IEEE ICC*.

[137] Wenyan Liu, Junhong Cheng, Xiaoling Wang, Xingjian Lu, and Jianwei Yin. 2022. Hybrid differential privacy based federated learning for Internet of Things. *JSA* 124 (2022), 102418.

[138] Yang Liu, Zhihao Yi, and Tianjian Chen. 2020. Backdoor attacks and defenses in feature-partitioned collaborative learning. In *FL-ICML Workshop*.

[139] Ekdeep Singh Lubana, Chi Ian Tang, Fahim Kawsar, Robert P. Dick, and Akhil Mathur. 2022. Orchestra: Unsupervised federated learning via globally consistent clustering. In *ICML*.

[140] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. 2019. Real-world image datasets for federated learning. In *FL-NeurIPS Workshop*.

[141] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *NeurIPS*.

[142] Zhengquan Luo, Yunlong Wang, Zilei Wang, Zhenan Sun, and Tieniu Tan. 2022. Disentangled federated learning for tackling attributes skew via invariant aggregation and diversity transferring. In *ICML*.

[143] Wang Luping, Wang Wei, and L. I. Bo. 2019. CMFL: Mitigating communication overhead for federated learning. In *IEEE ICDCS*.

[144] Lingjuan Lyu, Xinyi Xu, and Qian Wang. 2020. Collaborative fairness in federated learning. In *FL-IJCAI'20 Workshop*.

[145] Lingjuan Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S. Yu. 2022. Privacy and robustness in federated learning: Attacks and defenses. *IEEE TNNLS* (2022). DOI: 10.1109/TNNLS.2022.3216981

[146] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to federated learning: A survey. arXiv:2003.02133. Retrieved from https://arxiv.org/abs/2003.02133

[147] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, Jiong Jin, Han Yu, and Kee Siong Ng. 2020. Towards fair and privacy-preserving federated deep models. *IEEE TPDS* 31, 11 (2020), 2524–2541.

[148] Chenxin Ma, Virginia Smith, Martin Jaggi, Michael Jordan, Peter Richtárik, and Martin Takác. 2015. Adding vs. averaging in distributed primal-dual optimization. In *ICML*.

[149] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. 2022. Layer-wised model aggregation for personalized federated learning. In *CVPR*.

[150] Guangcan Mai, Kai Cao, Xiangyuan Lan, and Pong C. Yuen. 2020. Secureface: Face template protection. *IEEE TIFS* 16 (2020), 262–277.

[151] Guangcan Mai, Kai Cao, Pong C. Yuen, and Anil K. Jain. 2018. On the reconstruction of face images from deep face templates. *IEEE TPAMI* 41, 5 (2018), 1188–1202.

[152] Disha Makhija, Xing Han, Nhat Ho, and Joydeep Ghosh. 2022. Architecture agnostic federated learning for neural networks. In *ICML*.

[153] Yunlong Mao, Xinyu Yuan, Xinyang Zhao, and Sheng Zhong. 2021. Romoa: Robust model aggregation for the resistance of federated learning to model poisoning attacks. In *ESORICS*.

[154] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. 2021. Federated multi-task learning under a mixture of distributions. In *NeurIPS*.

[155] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. 2022. Personalized federated learning through local memorization. In *ICML*.

[156] Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation* 24 (1989), 109–165.

[157] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*.

[158] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *ICLR*.

[159] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. 2021. PPFL: Privacy-preserving federated learning with trusted execution environments. In *MobiSys*.

[160] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic federated learning. In *ICML*.

[161] Viraaji Mothukuri, Prachi Khare, Reza M. Parizi, Seyedamin Pouriyeh, Ali Dehghantanha, and Gautam Srivastava. 2021. Federated-learning-based anomaly detection for IoT security attacks. *IEEE IoT-J* 9, 4 (2021), 2545–2554.

[162] Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. 2021. A survey on security and privacy of federated learning. *FGCS* 115 (2021), 619–640.

[163] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. 2023. FedProc: Prototypical contrastive federated learning on non-IID data. *FGCS* 143 (2023), 93–104.

[164] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor. 2021. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys and Tutorials* 23, 3 (2021), 1622–1658.

[165] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. 2022. FLAME: Taming backdoors in federated learning. In *USENIX Security*.

[166] Van-Dinh Nguyen, Symeon Chatzinotas, Björn Ottersten, and Trung Q. Duong. 2022. FedFog: Network-aware optimization of federated learning over wireless fog-cloud systems. *IEEE TWC* 21, 10 (2022), 8581–8599.

[167] Solmaz Niknam, Harpreet S. Dhillon, and Jeffrey H. Reed. 2020. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine* 58, 6 (2020), 46–51.

[168] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC*.

[169] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R. Gel. 2021. Defending against backdoors in federated learning with robust learning rate. In *AAAI*.

[170] Christodoulos Pappas, Dimitris Chatzopoulos, Spyros Lalis, and Manolis Vavalis. 2021. Ipls: A framework for decentralized federated learning. In *IFIP Networking*.

[171] Charith Perera, Yongrui Qin, Julio C. Estrella, Stephan Reiff-Marganiec, and Athanasios V. Vasilakos. 2017. Fog computing for sustainable smart cities: A survey. *ACM CSUR* 50, 3 (2017), 1–43.

[172] Krishna Pillutla, Yassine Laguel, Jérôme Malick, and Zaid Harchaoui. 2023. Federated learning with superquantile aggregation for heterogeneous data. *Springer Machine Learning* (2023), 1–68.

[173] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. 2022. Federated learning with partial model personalization. In *ICML*.

[174] Zixuan Qin, Liu Yang, Qilong Wang, Yahong Han, and Qinghua Hu. 2023. Reliable and interpretable personalized federated learning. In *CVPR*.

[175] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. 2022. Rethinking architecture design for tackling data heterogeneity in federated learning. In *CVPR*.

[176] Youyang Qu, Longxiang Gao, Tom H. Luan, Yong Xiang, Shui Yu, Bai Li, and Gavin Zheng. 2020. Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE IoT-J* 7, 6 (2020), 5171–5183.

[177] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. 2022. Generalized federated learning via sharpness aware minimization. In *ICML*.

[178] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 2019. Braintorrent: A peer-to-peer environment for decentralized federated learning. arXiv:1905.06731. Retrieved from https://arxiv.org/abs/1905.06731

[179] Yichen Ruan and Carlee Joe-Wong. 2022. FedSoft: Soft clustered federated learning with proximal local updating. In *AAAI*.

[180] Adam Sadilek, Luyang Liu, Dung Nguyen, Methun Kamruzzaman, Stylianos Serghiou, Benjamin Rader, Alex Ingerman, Stefan Mellem, Peter Kairouz, Elaine O. Nsoesie, Jamie MacFarlane, Anil Vullikanti, Madhav Marathe, Paul Eastham, John S. Brownstein, Blaise Aguera y. Arcas, Michael D. Howell, and John Hernandez. 2021. Privacy-first health research with federated learning. *NPJ Digital Medicine* 4, 1 (2021), 132.

[181] Felix Sattler, Arturo Marban, Roman Rischke, and Wojciech Samek. 2020. Communication-efficient federated distillation. arXiv:2012.00632. Retrieved from https://arxiv.org/abs/2012.00632

[182] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. In *IEEE TNNLS* 32, 8 (2020), 3710–3722.

[183] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2020. On the byzantine robustness of clustered federated learning. In *IEEE ICASSP*.

[184] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE TNNLS* 31, 9 (2019), 3400–3413.

[185] Osama Shahid, Seyedamin Pouriyeh, Reza M. Parizi, Quan Z. Sheng, Gautam Srivastava, and Liang Zhao. 2021. Communication efficiency in federated learning: Achievements and challenges. arXiv:2107.10996. Retrieved from https://arxiv.org/abs/2107.10996

[186] Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. 2022. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. In *IJCAI*.

[187] Rui Shao, Pramuditha Perera, Pong C. Yuen, and Vishal M. Patel. 2022. Federated generalized face presentation attack detection. In *IEEE TNNLS*. DOI: 10.1109/TNNLS.2022.3172316

[188] Yiqing Shen, Yuyin Zhou, and Lequan Yu. 2022. CD2-pFed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. In *CVPR*.

[189] Yifan Shi, Yingqi Liu, Kang Wei, Li Shen, Xueqian Wang, and Dacheng Tao. 2023. Make landscape flatter in differentially private federated learning. In *CVPR*.

[190] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. 2023. Improving the model consistency of decentralized federated learning. In *ICML*.

[191] MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2020. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. In *FL-ICML Workshop*.

[192] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. 2019. Overcoming forgetting in federated learning on non-iid data. In *NeurIPS Workshop*.

[193] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. In *NeurIPS*.

[194] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.

[195] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, and Hai Li. 2021. Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. In *NeurIPS*.

[196] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *CVPR*.

[197] Lichao Sun and Lingjuan Lyu. 2021. Federated model distillation with noise-free differential privacy. In *IJCAI*.

[198] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. 2019. Can you really backdoor federated learning?. In *NeurIPS Workshop*.

[199] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. 2020. Personalized federated learning with moreau envelopes. In *NeurIPS*.

[200] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards personalized federated learning. *IEEE TNNLS* (2022). DOI: 10.1109/TNNLS.2022.3160699

[201] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*.

[202] Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. 2022. FedCor: Correlation-based active client selection strategy for heterogeneous federated learning. In *CVPR*.

[203] Zhenheng Tang, Shaohuai Shi, Bo Li, and Xiaowen Chu. 2022. GossipFL: A decentralized federated learning framework with sparsified and adaptive communication. *IEEE TPDS* 34, 3 (2022), 909–922.

[204] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. 2022. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *ICML*.

[205] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *ESORICS*.

[206] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. 2020. LDP-Fed: Federated learning with local differential privacy. In *3rd ACM International Workshop on Edge Systems, Analytics and Networking*.

[207] Dmitrii Usynin, Alexander Ziller, Marcus Makowski, Rickmer Braren, Daniel Rueckert, Ben Glocker, Georgios Kaissis, and Jonathan Passerat-Palmbach. 2021. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence* 3, 9 (2021), 749–758.

[208] Bram van Berlo, Aaqib Saeed, and Tanir Ozcelebi. 2020. Towards federated unsupervised representation learning. In *ACM EdgeSys*.

[209] Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben Itzhak, and Michael Mitzenmacher. 2022. Eden: Communication-efficient and robust distributed mean estimation for federated learning. In *ICML*.

[210] Omar Abdel Wahab, Azzam Mourad, Hadi Otrok, and Tarik Taleb. 2021. Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. *IEEE Communications Surveys and Tutorials* 23, 2 (2021), 1342–1397.

[211] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM*.

[212] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. 2020. Attack of the tails: Yes, you really can backdoor federated learning. In *NeurIPS*.

[213] Hui-Po Wang, Sebastian U. Stich, Yang He, and Mario Fritz. 2022. ProgFed: Effective, communication, and computation efficient federated learning by progressive training. In *ICML*.

[214] Yuwei Wang and Burak Kantarci. 2020. A novel reputation-aware client selection scheme for federated learning within mobile environments. In *IEEE CAMAD*.

[215] Yujia Wang, Lu Lin, and Jinghui Chen. 2022. Communication-efficient adaptive federated learning. In *ICML*.

[216] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication-efficient federated learning via knowledge distillation. *Nature Communications* 13, 1 (2022), 2032.

[217] Di Wu, Nai Wang, Jiale Zhang, Yuan Zhang, Yong Xiang, and Longxiang Gao. 2022. A blockchain-based multi-layer decentralized framework for robust federated learning. In *IJCNN*.

[218] Nannan Wu, Li Yu, Xuefeng Jiang, Kwang-Ting Cheng, and Zengqiang Yan. 2023. FedNoRo: Towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity. In *IJCAI*.

[219] Qiong Wu, Kaiwen He, and Xu Chen. 2020. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. *IEEE OJ-CS* 1 (2020), 35–44.

[220] Wentai Wu, Ligang He, Weiwei Lin, and Rui Mao. 2020. Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. *IEEE TPDS* 32, 7 (2020), 1539–1551.

[221] Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. 2021. Crfl: Certifiably robust federated learning against backdoor attacks. In *ICML*.

[222] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2019. Dba: Distributed backdoor attacks against federated learning. In *ICLR*.

[223] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. 2023. Multi-center federated learning: clients clustering for better personalization. *WWW* 26, 1 (2023), 481–500.

[224] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. 2023. Feddm: Iterative distribution matching for communication-efficient federated learning. In *CVPR*.

[225] Jingyi Xu, Zihan Chen, Tony Q. S. Quek, and Kai Fong Ernest Chong. 2022. FedCorr: Multi-stage federated learning for label noise correction. In *CVPR*.

[226] Jian Xu, Xinyi Tong, and Shao-Lun Huang. 2023. Personalized federated learning with feature alignment and classifier collaboration. In *ICLR*.

[227] Xiaolong Xu, Haoyuan Li, Zheng Li, and Xiaokang Zhou. 2022. Safe: Synergic data filtering for federated learning in cloud-edge computing. *IEEE TII* 19, 2 (2022), 1655–1665.

[228] Miao Yang, Ximin Wang, Hongbin Zhu, Haifeng Wang, and Hua Qian. 2021. Federated learning with class imbalance reduction. In *EUSIPCO*.

[229] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM TIST* 10, 2 (2019), 1–19.

[230] Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. 2022. Robust federated learning with noisy labels. *IEEE Intelligent Systems* 37, 2 (2022), 35–43.

[231] Xin Yao, Tianchi Huang, Chenglei Wu, Rui-Xiao Zhang, and Lifeng Sun. 2019. Federated learning with additional mechanisms on clients to reduce communication costs. arXiv:1908.05891. Retrieved from https://arxiv.org/abs/1908.05891

[232] Xin Yao and Lifeng Sun. 2020. Continual local training for better initialization of federated models. In *IEEE ICIP*.

[233] Mang Ye, Jianbing Shen, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. 2020. Augmentation invariant and instance spreading feature for softmax embedding. *IEEE TPAMI* 44, 2 (2020), 924–939.

[234] Liping Yi, Wang Gang, and Liu Xiaoguang. 2022. QSFL: A two-level uplink communication optimization framework for federated learning. In *ICML*.

[235] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM CSUR* 54, 6 (2021), 1–36.

[236] Jaehong Yoon, Geon Park, Wonyong Jeong, and Sung Ju Hwang. 2022. Bitwidth heterogeneous federated learning with progressive weight dequantization. In *ICML*.

[237] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. 2021. Fedmix: Approximation of mixup under mean augmented federated learning. In *ICLR*.

[238] Naoya Yoshida, Takayuki Nishio, Masahiro Morikura, and Koji Yamamoto. 2020. Mab-based client selection for federated learning with uncertain resources in mobile networks. In *IEEE GC Workshops*.

[239] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. 2020. Salvaging federated learning by local adaptation. arXiv:2002.04758. Retrieved from https://arxiv.org/abs/2002.04758

[240] Kai Yue, Richeng Jin, Ryan Pilgrim, Chau-Wai Wong, Dror Baron, and Huaiyu Dai. 2022. Neural tangent kernel empowered federated learning. In *ICML*.

[241] Daniel Yue Zhang, Ziyi Kou, and Dong Wang. 2020. Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *IEEE Big Data*.

[242] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. 2023. Federated unsupervised representation learning. *FITEE* 24, 8 (2023), 1181–1193.

[243] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *ICLR*.

[244] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. 2020. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE IoT-J* 8, 5 (2020), 3310–3322.

[245] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. 2021. Parameterized Knowledge Transfer for Personalized Federated Learning. In *NeurIPS*.

[246] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. 2022. Federated Learning with Label Distribution Skew via Logits Calibration. In *ICML*.

[247] Junwu Zhang, Mang Ye, and Yao Yang. 2022. Learnable Privacy-Preserving Anonymization for Pedestrian Images. In *ACM MM*.

[248] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. 2022. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *CVPR*.

[249] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. 2021. Personalized federated learning with first order model optimization. In *ICLR*.

[250] Wenyu Zhang, Xiumin Wang, Pan Zhou, Weiwei Wu, and Xinglin Zhang. 2021. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access* 9 (2021), 24462–24474.

[251] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Steven Wu, and Jinfeng Yi. 2022. Understanding Clipping for Federated Learning: Convergence and Client-Level Differential Privacy. In *ICML*.

[252] Xu Zhang, Yinchuan Li, Wenpeng Li, Kaiyang Guo, and Yunfeng Shao. 2022. Personalized Federated Learning via Variational Bayesian Inference. In *ICML*.

[253] Yuhui Zhang, Zhiwei Wang, Jiangfeng Cao, Rui Hou, and Dan Meng. 2021. ShuffleFL: Gradient-preserving federated learning using trusted execution environment. In *ACM CF*.

[254] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. 2022. Neurotoxin: Durable Backdoors in Federated Learning. In *ICML*.

[255] Chen Zhao, Yu Wen, Shuailou Li, Fucheng Liu, and Dan Meng. 2021. Federatedreverse: A detection and defense method against backdoor attacks in federated learning. In *IH&MMSec*.

[256] Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. 2021. Federated meta-learning for fraudulent credit card detection. In *IJCAI*.

[257] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.

[258] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*.

[259] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. 2020. Performance optimization of federated person re-identification via benchmark analysis. In *ACM MM*.

Received 15 September 2022; revised 19 July 2023; accepted 21 August 2023