Advancements in Federated Learning: Models, Methods, and Privacy

HUIMING CHEN, Tsinghua University, Beijing, China
HUANDONG WANG, Tsinghua University, Beijing, China
QINGYUE LONG, Tsinghua University, Beijing, China
DEPENG JIN, Tsinghua University, Beijing, China
YONG LI, Tsinghua University, Beijing, China

Federated learning (FL) is a promising technique for resolving the rising privacy and security concerns. Its main ingredient is to cooperatively learn the model among the distributed clients without uploading any sensitive data. In this article, we conducted a thorough review of the related works, following the development context and deeply mining the key technologies behind FL from the perspectives of theory and application. Specifically, we first classify the existing works in FL architecture based on the network topology of FL systems with detailed analysis and summarization. Next, we abstract the current application problems, summarize the general techniques, and frame the application problems into the general paradigm of FL base models. Moreover, we provide our proposed solutions for model training via FL. We have summarized and analyzed the existing FedOpt algorithms, and deeply revealed the algorithmic development principles of many first-order algorithms in depth, proposing a more generalized algorithm design framework. With the instantiation of these frameworks, FedOpt algorithms can be simply developed. As privacy and security are the fundamental requirements in FL, we provide the existing attack scenarios and the defense methods. To the best of our knowledge, we are among the first tier to review the theoretical methodology and propose our strategies since there are very few works surveying the theoretical approaches. Our survey targets motivating the development of high-performance, privacy-preserving, and secure methods to integrate FL into real-world applications.

CCS Concepts: • **Theory of computation** → **Theory and algorithms for application domains**; • **Networks** → **Network architectures**; • **Computing methodologies** → **Distributed computing methodologies**; • **Security and privacy**;

Additional Key Words and Phrases: Federated learning, architecture, communication efficiency, base models, distributed optimization, privacy and security

ACM Reference Format:
Huiming Chen, Huandong Wang, Qingyue Long, Depeng Jin, and Yong Li. 2024. Advancements in Federated Learning: Models, Methods, and Privacy. *ACM Comput. Surv.* 57, 2, Article 46 (November 2024), 39 pages. https://doi.org/10.1145/3664650

Huiming Chen and Huandong Wang authors have equal contributions.

Authors' Contact Information: Huiming Chen, Tsinghua University, Beijing, China; e-mail: chenhuiming@mail.tsinghua.edu.cn; Huandong Wang, Tsinghua University, Beijing, China; e-mail: wanghuandong@tsinghua.edu.cn; Qingyue Long, Tsinghua University, Beijing, China; e-mail: longqy21@mails.tsinghua.edu.cn; Depeng Jin, Tsinghua University, Beijing, China; e-mail: jindp@tsinghua.edu.cn; Yong Li (Corresponding author), Tsinghua University, Beijing, China; e-mail: liyong07@tsinghua.edu.cn.

ACM 0360-0300/2024/11-ART46
https://doi.org/10.1145/3664650

1 Introduction

With the rapid development of information technology, a growing amount of data is generated by people's daily activities, which has become an invaluable asset in modern society. Big data has created enormous benefits for society by providing businesses and individuals with new insights and foresights, but this reliance on data has also raised concerns about privacy and data security.

In recent years, with the European Union's enactment of the General Data Protection Regulation [39] and legislative efforts in data security by the United States [128] and China [31], data security and privacy protection have entered an important stage of development, i.e., the secure computation techniques including differential privacy, secure multi-party computing and FL have been increasingly emphasized and widely applied in various industries.

Federated learning provides an effective solution for ensuring compliant and secure data flow. It is essentially a distributed machine learning framework that collaboratively trainsa specific model over numerous clients which are coordinated by a central server. Typically, the procedures of FL involve client selection, broadcasting, the local computation with client's own data and global aggregation on the server [117]. In general, it has been categorized into the following architectures [185]:

- **Horizontal FL** (**HFL**): HFL is the most common scenario in FL, which we mainly consider for the guidance in algorithmic development in our survey. Here, the clients' data have the common feature space but different samples. HFL has been widely investigated since this scenario can be naturally fit in many real applications. For example, with increasing health consciousness, a large number of people wear smartwatches to monitor their health, these devices will generate massive data sharing the same features for collaboratively learning the model together with coordination of the central server.
- **Vertical FL** (**VFL**): VFL is mainly dealing with the case that the datasets amongst clients have different feature spaces but share a common sample identity space. It was first considered for two-party collaboration and recently has been extended to multiple-party scenarios. VFL matches the real world applications very well. For example, a patient can have a medical record in different hospitals and each may have distinct features. With many samples, these hospitals will have more complete features and are able to jointly learn the model in a more accurate fashion.
- **Federated transfer learning** (**FTL**): FTL deals with the case that the dataset have both different feature spaces and sample identity spaces. For example, in social computing network [201], it requires a highly expensive and time consuming data labeling process. In a federation manner, FTL approaches can be used to reduce compute burden and deliver solutions for the full sample and feature space.

Additionally, FL can conveniently merge multiple techniques for application scenario expansions, further promoting the industrial development. For example, FL has been applied in edge computing, which efficiently leverages the edge device's computational capability for the model training, preventing the leakage of sensitive data without data centralization [180]. FL has been applied in machine unlearning to seek to allow efficient client data removal from the FL models [130]. Moreover, reinforcement learning has been incorporated for optimizing the resource allocations over both clients and servers to reach incentive mechanism [155], which facilitates the clients' participation.

Currently, the field of FL has witnessed many outstanding research achievements, and these include performance [75, 99, 144] and security enhancements [35, 49, 157], and so on. In terms of performance, FL frequently faces the issues of communication bottlenecks caused by heterogeneous networks, physical distance, and communication overhead. For mitigating these issues, novel FL algorithms have been investigated for improving the efficiency of training complex models. They

Table 1. Selected Surveys of FL

Article	Description
[185]	Concepts and applications of FL in terms of its definition, privacy techniques, and categories, including HFL, VFL, and FTL.
[127]	FL in wireless networks in terms of its applications and challenges.
[95]	Unique characteristics and challenges of FL, including expensive communication, systems heterogeneity, statistical heterogeneity, and privacy concerns, and existing studies and directions to solve these challenges.
[176]	On the problem of implementing FL in mobile edge networks regarding to several topics, including optimizing communication, scheduling resource, and guarding privacy and security.
[38]	Survey on comprehensive FL.

include local and global acceleration strategies, asynchronous coordination, compression and sparsification. Regarding security issues, since FL requires participation from multiple parties, it may introduce extra security risks. In recent years, new solutions and technologies have emerged for enhancing privacy preserving capabilities, resisting data leakage and backdoor attacks that may occur during the FL process. In summary, It is foreseeable that the huge demand for secure computing will drive the continuous and fast development of FL, including:

- **The continuous improvement of the system and increasing unification of industry standards:** the application of FL requires standardized specifications. With the rapidly increasing implementation, standard formulation work is also in progress. Various international organizations have started developing the standards of FL since 2018. Currently, IEEE and ITU-T have released framework and function standards [1, 2]. Moreover, the security requirement standards and technical application standards for FL with multi-party secure computation are also in the process of being drafted.
- **Increasingly diverse application scenarios and growing number of applications:** many traditional industries that urgently require digital transformation have the characteristics of data-intensive, such as the financial industry and internet of things. They generally have large-scale application scenarios for data flow and put forward more stringent requirements for security management. These traditional industries are in an urgent need to establish the secure and private data flow to achieve effective data fusion and utilization. Therefore, FL can be a fit for ensuring the secure and compliant data flow and promotes the digital development of traditional industries.

On the one hand, both the developments of applications and techniques in FL become increasingly mature, while on the other hand, the problems faced in FL applications are also becoming increasingly complicated. Therefore, our survey targets at answering the following questions:

- How to categorize the architectures that can be adopted for federated learning in complicated distributed scenarios? (Answered in Section 2)
- How to abstract problems from a specified actual application and propose solutions for FL? (Answered in Section 3)
- How to build an efficient, scalable, secure and trustworthy FL algorithm? (Answered in Sections 4 and 5)

Differences between existing surveys. Currently, numerous surveys have been conducted on FL techniques. Next, we provide the introduction of the representative surveys. As the pioneer FL survey, Yang et al. [185] introduce the concepts and applications of FL in terms of its definition, privacy techniques, and categories, including HFL, VFL, and FTL. Niknam et al. [127] investigate federated machine learning in wireless networks in terms of its applications and challenges. Li et al. [95] focus on the distinctive features and difficulties of FL, such as the communication overhead, the issues of heterogeneity in both statistics and system, and privacy issues, as well as the present research and solutions to these difficulties. Wei et al. [176] focus on FL empowered mobile edge networks regarding several topics, including optimizing communication, scheduling resources, and guarding privacy and security. Kairouz et al. [38] focus on the privacy and security issues in the FL system with the summarization of existing studies in terms of attack methods and defense methods.

Different from existing surveys, we have a thoughtful summary and discussion of existing literatures, which not only digs deep into the existing perspectives, but also provides analysis from different perspectives. Specifically, in terms of the FL architecture, based on the existing taxonomy, we further focus on the topological structures of the FL systems. Moreover, based on the seldom surveys on FedOpt [37], we have summarized and analyzed the existing FedOpt algorithms, which play a key role in improving performance. In particular for many first-order FedOpt, we abstract their common algorithmic patterns and propose the general frameworks which can be instantiated by adopting different acceleration strategies for facilitating FedOpt development. In addition, based on the existing surveys on FL applications, we have provided a generalized federated base model and offered our solutions to facilitate larger-scale applications.

Our contributions. In this survey, we present a comprehensive review for motivating the development of high-performance and privacy-preserving approaches, as to integrate FL into real-world applications. To be specific, from theoretical perspectives, we review the architectures and optimization methods, both of which play a fundamental role in the performance of a FL system. We review the FL applications, abstract the federated base models underlying them, and propose our solutions. Moreover, privacy and security are investigated with the existing attack scenarios and the defense methods. Finally, we classify the existing works in applications and project our insights into challenges and opportunities. Overall, we summarize the contributions we make in this survey as follows:

- **New summarization on FL trends:** We have conducted the extensive review in the topics of FL research articles and surveys from the year 2018 to 2022, **including 2,713 research articles and 93 surveys**, and the results are exhibited in Figure 1. In general, we can draw the conclusions as follows:
  - **A novel taxonomy:** With elaborately partitioning strategy to show the development of FL in each topic, we can classify the existing FL works into knowledge domain, privacy and security, network, real application, architecture, theoretical approach, and graph domain. In particular, optimization theory dominates the theoretical approach in FL.
  - **Rapid development:** FL has been maintaining a fast development speed. In particular, there are 1,389 FL articles released in 2022, which is nearly the sum of works in previous years. Especially with the development of smart cities worldwide, the FL applications will maintain a fast growth speed, which will also drive the theoretical approach development.
  - **Lack of surveys in theoretical approaches:** We also find the FL field lacks significantly the related surveys in theoretical approaches, while they are the underlying basis supporting the FL development. Although the year 2022 has witnessed FL surveys becoming booming on broad topics, there are only exactly two surveys on the theoretical approach topic.
- **Federated base model abstraction and our proposed solutions:** From Figure 1, we can see FL applications constitute a significant proportion in the studies on FL. We project our insights into these applications and abstract the problems, then we have thoroughly analyzed and classified these problems with federated solutions. In general, the underlying issues of these applications can be reduced to the popular machine learning model training, non-smooth regularization, **multi-task learning** (**MTL**), and **matrix factorization** (**MF**), in the

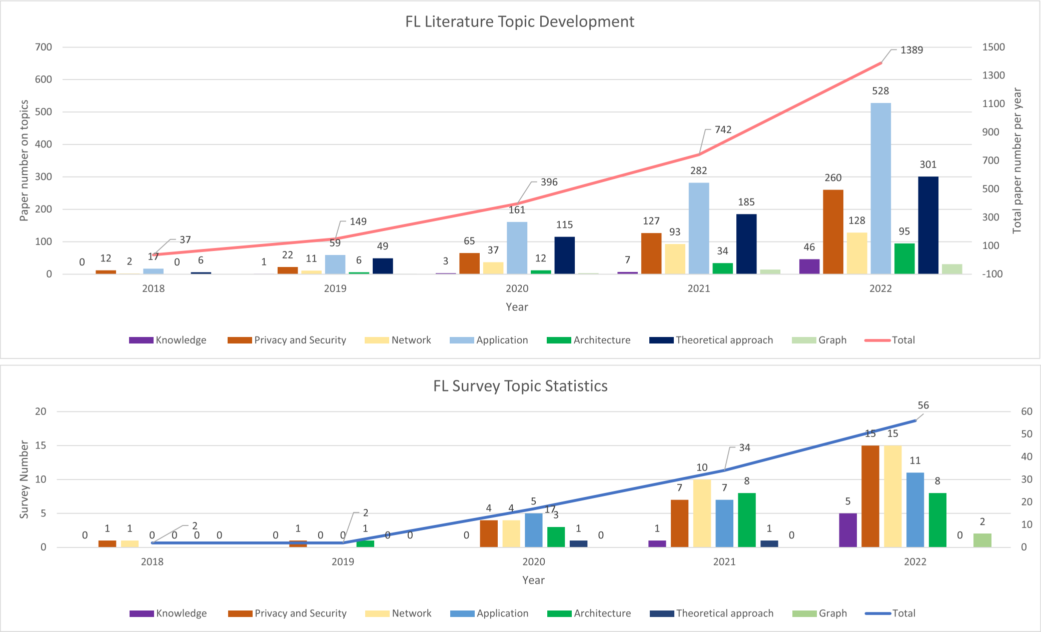


Fig. 1. The statistics on FL related topic articles from years 2018 to 2022. It can be seen FL has been under fast development and can be expected to maintain quick growth in near future. Specifically, the topics of privacy and security, theoretical approaches (optimization in majority), and applications dominate this field, the surveys on privacy and security and applications follow this pattern while this area significantly lacks the optimization related surveys.

context of FL. **It is worth emphasizing that we have provided novel approaches for learning these models under federated settings.**

- **Proposed algorithmic design frameworks:** FedOpt algorithms play the fundamental role in FL and they also occupy a significant proportion in the research efforts of FL. Therefore, we carry out a comprehensive review of the works in FedOpt algorithms, delving deeply into the generalization of the common patterns underlying these algorithms. Specifically, the FedOpt algorithms are identified into first-order (majority in FedOpt), second-order and ADMM based scenarios. In particular, we further generalize the existing first-order works into the local acceleration and global acceleration frameworks respectively, which can be simply instantiated to formulate new FedOpt algorithms. **To the best of our knowledge, we are among the first-tier for comprehensively surveying the theoretical approaches in FedOpt.**

The rest of this survey is organized as follows: Section 2 discusses the FL architectures. Section 3 illustrates the application problem abstraction. In Section 4, FedOpt algorithms are abstracted for their common algorithmic pattern. Based on it, we have proposed the general frameworks, which can be simply instantiated to formulate new FedOpt algorithms. Section 5 studies privacy and security issues. Section 6 presents both the technical and real applications. Section 7 illustrates the challenges and future directions.

2 Architecture

FL architecture plays the fundamental role in FL [164], building FL architectures can be effective and practical for improving the performance. Based on the distribution characteristics of the data,

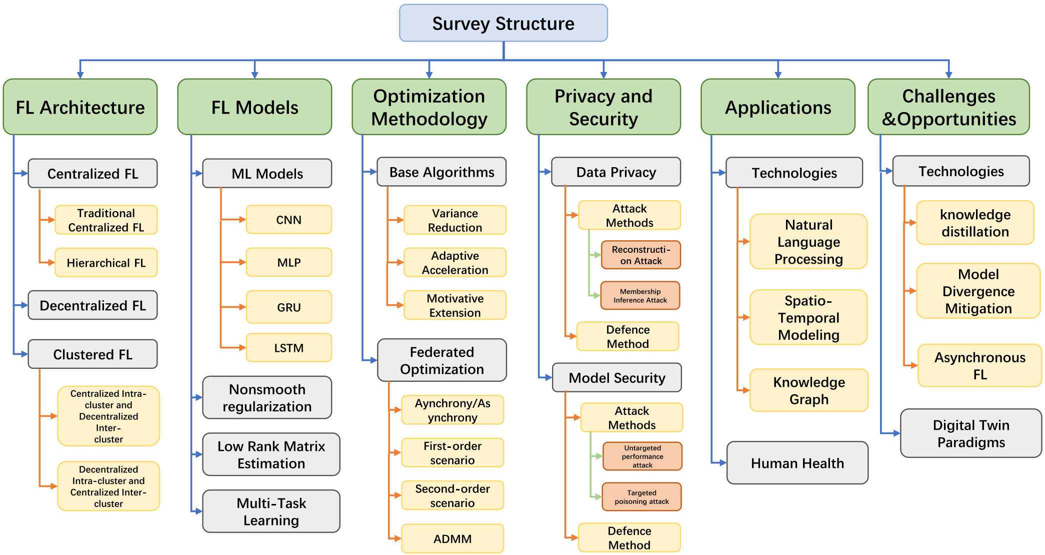


Fig. 2. Organization of this survey.

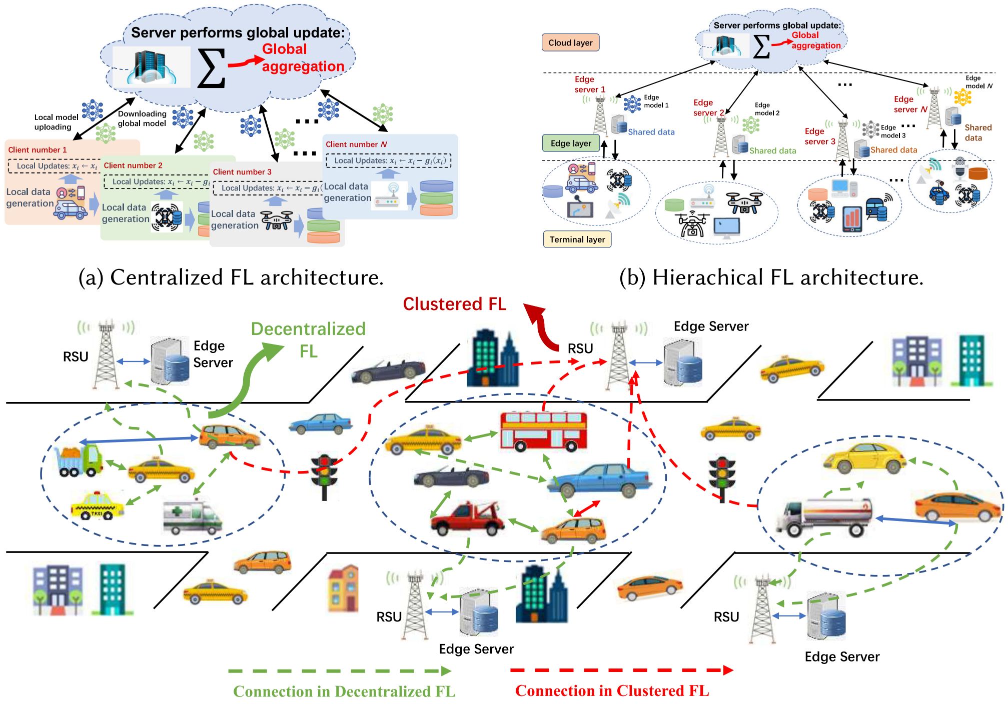
three types of FL techniques have been identified: HFL, VFL, and FTL [185]. Different from this type of categorization, our main focus is on the topological architecture of the network of participating devices in FL. While typical FL architecture consists of a central server coordinating a number of clients for cooperative learning [95, 164], the decentralization [160] or hierarchy [138] have become increasingly popular recently. Therefore, we will categorize the FL architectures into three scenarios: centralized FL, decentralized FL and clustered FL. As the illustration in Figure 3, (a) is the centralized architecture, which is the most popular one in FL; (b) shows the hierarchical version of FL; (c) demonstrates the decentralized and clustered FL in the possible real applications, such as the internet of vehicles.

2.1 Centralized FL

Centralized structures are the most common topology for an FL system, where a centralized server is in charge of coordinating different clients. It was first introduced in [117], where each client trains the local model with its data and communicates with the server for aggregation [188].

Traditional FL. The traditional centralized FL has been among the most popular choice in real applications. Its major bottleneck is the communication overhead. In order to achieve quick global model aggregation, Yang et al. [184] integrate device selection with beamforming design and look into the superposition property of a wireless MAC by taking into account the wireless channel's characteristics. Xu et al. [183] propose a modified FedAvg algorithm with a dynamic learning rate that is able accommodate the fading channel. As for the model aggregation problem, Fan et al. [41] utilize a Markovian probability model to characterize the intrinsic temporal structure of the model aggregation series. Based on it, they design an algorithm with the message passing mechanism, which has a low complexity and a near optimal performance. For further communication efficiency, Jing et al. [72] adopt statistical channel state information. However, traditional centralized FL still has other challenges including statistical heterogeneity, computation overhead, security, and privacy [164].

Hierarchical FL. This scenario generally exists in the well-known cloud-edge-client system and is organized in a tree structure [149], the major goal is to alleviate the above issues in traditional



(a) Centralized FL architecture.

(b) Hierachical FL architecture.

(c) Decentralized and clustered FL architectures.

Fig. 3. Illustration of existing FL architectures, including centralized, hierarchical, decentralized and clustered FL architectures. It can be seen these architectures are generally existing in hybrid mode to handle networks with complicated topology.

FL [106]. Specifically, the edge layers are federated via a cloud layer, while the edge layer coordinates the terminal devices. Since the edge layer has taken on the communication overhead from the cloud layer and the computation from the terminal layer [208], this architecture has shown the capability of high communication efficiency and low latency in the asynchronous and heterogeneous system [173, 205]. For further improving the system robustness, wireless channel has been considered for **hierarchical over-the-air FL (HOTAFL)** [15]. Liu et al. [108] and Xu et al. [182] perform investigations on clustering the users. They also attempt optimizing scheduling the wireless resource in HOTAFL.

2.2 Decentralized FL

In practice, the centralized FL is vulnerable when suffering from server malfunctions, malicious attacks and untrustworthy servers. For addressing these issues, Li et al. [91] propose a decentralized structure that uses the blockchain to entrust the network's nodes with the task of storing the model, which also optimizes the distribution of computing resources. Liu et al. [110] propose a general decentralized FL framework that enables periodic compression and both multiple local updates and multiple inter-node communications. Hashemi et al. [60] propose DeLi-CoCo and demonstrate how the decentralized FL is benefiting from additional gossip stages between gradient iterations for faster convergence. For further improving the performance and alleviating the client-drift problem, Li et al. [88] provide a Def-KT method to fuse models among clients by sharing their acquired knowledge. Xiao et al. [181] create a decentralized FL framework with an inexact stochastic parallel random walk that is somewhat resistant to the time-varying dynamic network.

Table 2. Existing Architectures in FL

Categories	Topological structure	Feature	Ref.
Centralized FL	Traditional Centralized FL	One centralized server aggregates all model parameters updated by clients.	[117], [188], [184], [183], [41]
	Hierarchical FL	The client-edge-cloud structure: The intermediate layer's edge servers are connected to one cloud server. The bottom layer's partial clients are connected to each edge server.	[106], [149], [173], [205], [3], [15], [108], [182], . [33], [207]
Decentralized FL	Decentralized FL	Without a central server. All clients communicate with each other in a distributed way.	[76], [84], [163], [110], [60], [50], [56], [24], . [88], [181]
Clustered FL	Centralized Intra-cluster, Decentralized Inter-cluster	Some clients are connected with one edge server. The edge servers are decentraliedly connected with each other.	[55], [158].
	Decentralized Intra-cluster, Centralized Inter-cluster	Each client is decentraliedly connected with each other. Global aggregations are aggregated by a central server.	[51],[101], [89].

In terms of security, since decentralized clients can only talk to their neighbors, the dissensus attack can poison the clients' collaboration. To overcome this issue, He et al. [63] propose a **Self-Centered Clipping** (**SCClip**) algorithm for Byzantine-robust consensus and optimization. Che et al. [24] adopt the committee mechanism to track the uploaded local gradients, which has shown both performance and security improvement.

2.3 Clustered FL

In this scenario, users are distributed and partitioned into clusters. The client population is organized into clusters with mutually trainable data distributions using geometric characteristics of the FL loss surface [51, 144]. In general, we can regard clustered FL as a hybrid of centralized and decentralized FL or semi-decentralized FL.

Centralized Intra-cluster and Decentralized Inter-cluster. In this architecture, each client node is associated with an edge server according to the specific predefined criteria, and the edge servers are deployed in a decentralized manner. As an exemplar, when this architecture is applied in **Internet of Vehicles** (**IoV**), vehicles in each cluster train their models and communicate with the **service provider** (**SP**) for the aggregation. Then, the SPs are deployed in a decentralized manner and they communicate with their neighboring SPs for information exchange. Here, communication strategy plays a crucial role in the performance. Gou et al. [55] apply gossip protocol for communication between clusters and introduce the leader-follower strategy to make full use of bandwidth and reach a faster convergence.

Decentralized Intra-cluster and Centralized Inter-cluster. In this architecture, clients in each cluster cooperatively learn the local models via **device-to-device** (**D2D**) communications in a decentralized manner, while the central server coordinates the clusters in the centralized fashion. Since client privacy cannot be decrypted by the central server, the privacy is highly improved. Moreover, by clustering the homeomorphic clients together, this scenario is able to alleviate the heterogeneous issue. Based on the architecture, Lin et al. [101] proposed two timescale hybrid FL, which outperforms the current state-of-the-art counterparts in terms of the model accuracy and network energy consumption. To guarantee the clustered structure, we can utilize an adaptive strategy without manual intervention to realize dynamic clustering [54].

2.4 Summarization

Centralized FL is fundamental in this field, but due to the simple structure, it cannot be straightforwardly applied in the complicated client network. Moreover, its scalability is still limited, which will significantly limit the applications in large-scale distributed application scenarios. Decentralized FL can reach the learning tasks by sharing the information with the client's neighbors to reach the consensus in the global models. We can expect it will have great potential for applications in large-scale networks. However, it still demands high performance algorithms and insurance in

Table 3. Training Models via FL Paradigm

Model	Federated training procedure
MLP	The client performs multiple SGD to obtain the local model, and then the server performs aggregation to update the global model.
CNN	
LSTM	
GRU	
TM	The client performs Metropolis Hastings based private computations to obtain the local model, then the server performs integration of the received local topic models.
Nonsmooth regularization	The client performs multiple mirror descent steps to update the local model and then the server performs aggregation to update the global model.
MTL	The client performs ADMM subproblems to update the local model and Lagrangian multiplier and the server performs the aggregation to obtain the global model corresponding to the task.
MF	The client performs ADMM subproblems to update user item vector and Lagrangian multiplier and the server performs the subproblem to update the item profile matrix.

privacy and security. Compared with centralized FL and decentralized FL, the clustered FL is more adaptable to different scenes, and can better balance performance and privacy security, we can expect its significant potentiality in the complicated systems, e.g., the traffic flow prediction over the large-scale dynamic IoV.

3 Federated Base Model

While FL techniques have laid the foundation of massive applications, the fast growing applications will adversely demand the development of techniques. Through comprehensively surveying the applications of FL, we further deeply abstract the application problems and explore the underlying techniques that support these applications. As a result, **they can be generally categorized into a basic paradigm consisting of a federated base model combined with a federated training framework**. Specifically, we first discuss several popular machine learning models, which can be straightforwardly trained through the popular FL methods. Furthermore, we discuss the nonsmooth regularization due to its promising potential in biomedical applications. Considering the more general and practical fashion that different clients may have different tasks, we introduce MTL under FL solution. Next follows MF, which has been widely employed in recommendation systems, our strategy is to formulate the problem with nuclear norm to accurately constrain the matrix being low-ranked. In general, our goal is to provide the FL schemes covering sufficient cases to facilitate larger-scale FL applications.

3.1 Machine Learning Models

Many machine learning models can be collaboratively trained via FL framework, e.g., as the representative models in deep learning, the **multi-layer perceptron** (**MLP**) and **convolutional neural network** (**CNN**) have been widely used. Specifically, with the same structures of these models on each client, they can be locally trained with multiple steps by utilizing their own datasets, then the updated local model is transmitted to the server for further aggregation (such as the weight averaging) [117]. Another popular model is the **long short-term memory** (**LSTM**), which has been utilized in FL for the language modeling [117]. **Gated recurrent unit** (**GRU**) neural network based on FedAvg solution has been applied for the traffic flow prediction [111]. In summary, we can train these models which share the common procedure of multiple local updates and global aggregation. Another popular machine learning model—topic modeling, has been applied in many industrial fields, such as medical informatics [14], and web search log mining [58]. It has been extended to FL settings due to the increasing concerns on the data privacy issue [70, 71]. Specifically, Jiang et al. [70] propose a **federated topic modeling** (**FTM**) framework to discover topics in a collection of documents and classify a document within the collection over distributed massive clients. Moreover, the local clients perform Metropolis Hastings based private computations to obtain the local model, then the server implements integration of the received local topic models. FTM has been demonstrated to simultaneously maintain privacy and improve the training quality.

3.2 Nonsmooth Regularization

Nonsmooth regularization has been widely applied in the biomedical applications with the sparse learning for dealing the challenges of large dimension data, redundant features and sample absence [86, 92]. Moreover, the biomedical application data is highly sensitive. Therefore, FL can be a promising technique for the cooperation over medical organizations, while it helps to reach the high accuracy solution, the privacy and security is preserved without data centralization. To start with, we assume there are N medical organizations and denote $\mathcal{D}_i$ as the local dataset on the ith medical organizations. Subsequently, the federated problem can be derived as follows:

$$\min_{w\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} f_i(w) + \alpha\Psi(w), \tag{1}$$

where $w \in \mathbb{R}^d$ is the model, $f_i(w) : \mathbb{R}^d \rightarrow \mathbb{R}$ is the client i's local loss function, and $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$ is the nonsmooth convex regularizer, and $\alpha > 0$ is the positive scalar, which can be used as the parameter in the proximal gradient descent solver, i.e., $w_i \leftarrow \mathrm{Prox}_{\alpha\eta\Psi}[w_i - \eta\nabla f_i(w_i)]$. When $f_i(w) = \frac{1}{n_i}\sum_{(x_i,y_i)\in\mathcal{D}_i}\left\|w^T x_i - y_i\right\|^2$, with $\Psi = \|\cdot\|_1$, the problem is known as LASSO.

We next discuss the FL solution over the N medical organizations via FedMirror, where the local update utilizes the multiple mirror descent steps [193]. Specifically, we introduce the Bregman distance. Let $h : \mathbb{R}^d \rightarrow \mathbb{R}$ be a continuously-differentiable, strictly convex function, then the Bregman distance can be defined: $\mathcal{D}_h(x,y) = h(x) - h(y) - \langle\nabla h(y), x - y\rangle$. Next, the local model w_i is mapped to the dual space to form the dual model z_i with the carrier ∇h, i.e., $z_i \leftarrow \nabla h(w_i)$. Subsequently, the dual model z_i is updated by utilizing the local gradient queried at w_i, i.e., $z_i \leftarrow z_i - \eta \cdot \nabla f_i(w_i)$. Finally, the updated dual model is mapped back to the primal space, i.e., $w_i \leftarrow \nabla(h + \eta\Psi)^*(w_i)$, where for any convex function g, $g^*(z)$ is the convex conjugate function given by $g^*(z) = \sup_{w\in\mathbb{R}^d}\left\{\langle z, w\rangle - g(w)\right\}$. With multiple local mirror descent steps, the model w_i is transmitted to the server for further aggregation, namely, $w \leftarrow {}^1\!/_N \sum_{i=1}^{N} w_i$.

3.3 Low Rank Matrix Estimation (LRME)

LRME is an efficient technique in dealing with large-scale problems with high-dimensional data and thus raising the significant interests in scientific computing. Exemplar applications include the sparse principal component analysis [131], data compression for natural language processing [29] and distance matrix completion [121]. In this section, we illustrate LRME in federated settings. In particular, we propose to solve the problem via **linearized federated ADMM (L-FedADMM)**, since its base algorithm ADMM can resolve nonsmooth optimization problems fast and effectively [20]. For the adaptation to FedOpt, we first formulated the federated LRME problem. Assume N clients aim to collaboratively recover the matrix X with the data sample $D_j \in \mathbb{R}^{d\times d}$. Then the client i's local loss function is given as $f_i(X) = \sum_{j=1}^{n_i}(\langle X, D_j\rangle - y_j)^2$, where y_j is the noisy observation. Subsequently, L-FedADMM targets at solving the following problem over the N clients:

$$\min_{X} \sum_{i=1}^{N} f_i(X) + \lambda \left\|X\right\|_*, \tag{2}$$

Here, the nuclear norm $\|\cdot\|_*$ mainly constrains the rank of the matrix so as to exploits the dependency. The purpose of the problem (2) is to recover the low rank matrix $X \in \mathbb{R}^{d\times d}$ from the noisy

ALGORITHM 1: L-FedAdmm

```
1:  **server input:** initial $Z^0$, $\eta_g$.
2:  **client $i$'s input:** initial $X_i^0$, $\pi_i^0$ and $\eta_l$.
3:  **for** $r = 1, \ldots, R$ **do**
4:      **Server implements** steps 5-6:
5:      Obtain the update $Z^r$ by performing the iteration (6)
6:      Sample clients $\mathcal{S} \subseteq [N]$ and transmit $Z^r$ to client $i \in \mathcal{S}$.
7:      **Clients implement** steps 8-14 **in parallel for** $i \in \mathcal{S}$:
8:      After receiving $Z^r$, client $i \in \mathcal{S}$ performs
9:      **for** $t = 1, \ldots, T$ **do**
10:         Obtain $X_i^t$ via (5).
11:         Update $\pi_i^t$: $\pi_i^t \leftarrow \pi_i^{t-1} + \rho(X_i^t - Z^r)$.
12:     **end for**
13:     Set $X_i^r \leftarrow X_i^T$ and $\pi_i^r \leftarrow \pi_i^T$ for $i \in \mathcal{S}$, and $X_i^r \leftarrow X_i^{r-1}$ and $\pi_i^r \leftarrow \pi_i^{r-1}$ for $i \notin \mathcal{S}$.
14:     Client $i$ transmits $\pi_i^r + \rho X_i^r$ to the server.
15: **end for**
```

observations y_j. For L-FedADMM, the consensus optimization which is equivalent to (2) is derived:

$$\min_{X} \sum_{i=1}^{N} f_i(X_i) + \lambda \|Z\|_*, \quad \text{s.t. } X_i - Z = 0, i = 1, \ldots, N, \tag{3}$$

where $Z \in \mathbb{R}^{d\times d}$ is the consensus variable, $X_i \in \mathbb{R}^{d\times d}$ can be viewed as the local model. Obviously, (3) and (2) are equivalent. Furthermore, we can derive the augmented Lagrangian function in the following [20]:

$$\mathcal{L}(X_{1:N}, \Pi, Z) = \sum_{i=1}^{N} \left\{ f_i(X_i) + \langle \pi_i, X_i - Z \rangle + \frac{\rho}{2} \|X_i - Z\|_2^2 \right\} + \lambda \|Z\|_*, \tag{4}$$

where $\pi_i \in \mathbb{R}^{d\times d}$ is the Lagrangian multiplier, $\langle , \rangle$ denotes the matrix inner product, $\rho > 0$ is the regularization parameter and $\Pi = \{\pi_i, i = 1, \ldots, N\}$. Subsequently, the decentralized ADMM [20] solves the problem (3) by alternating the update. To be specific, the client i download the consensus matrix Z and perform the update via minimizing (4), i.e., $X_i^+ \leftarrow \operatorname{argmin}_{X_i} \mathcal{L}(X_{1:N}, \Pi, Z)$, then it continues to update the Lagrangian multiplier: $\pi_i^+ \leftarrow \pi_i + \rho(X_i^+ - Z)$. Finally, with X_i and π_i, the server update the consensus matrix Z via $Z^+ \leftarrow \operatorname{argmin}_Z \mathcal{L}(X_{1:N}^+, \Pi^+, Z)$.

We further expand for obtaining the update X_i^+ on client i. To be specific, the efficient linearized approximation strategy for $f_i(X_i^+)$ at X_i can be utilized, namely, $f_i(X_i^+) \approx f_i(X_i) + \langle \nabla f_i(X_i), X_i^+ - X_i \rangle + {}^1/_{2\eta_l} \left\|X_i^+ - X_i\right\|_2^2$, where η_l is the second-order approximate. By expanding, the closed-form solution can be efficiently derived via:

$$X_i^+ \leftarrow \frac{\rho\eta_l Z - \eta_l \pi_i + X_i - \eta_l \nabla f_i(X_i)}{1 + \rho\eta_l}. \tag{5}$$

Furthermore, with the updated X_i^+ and π_i^+, we further derive the update for Z. Specifically, the proximal operator can be utilized for handling the nuclear norm via the following iterations

$$Z^{i+1} \leftarrow \operatorname{Prox}_{\lambda\eta_g \|\cdot\|_*} \left\{ Z^i - \eta_g \left(N\rho Z^i - \sum_{i=1}^{N} \{\pi_i^+ + \rho X_i^+\} \right) \right\} \tag{6}$$

for $i = 0, \dots, I-1$, then we can obtain $Z^+ \leftarrow Z^M$. Next, we mimic the adaptation of SGD to FedAvg for L-FedADMM [117]. Precisely, the participated clients $i \in \mathcal{S}$ performs the local updates for X_i (via (5)) and π_i for multiple times (say T local iterations), and then the quantity $\pi_i^+ + \rho X_i^+$ is committed to the server for further global aggregation according to (6). While clients $i \in \mathcal{S}$ update their local models X_i and Lagrangian multipliers π_i, those clients $i \notin \mathcal{S}$ hold their X_i and π_i, i.e., $X_i^+ \leftarrow X_i$ and $\pi_i^+ \leftarrow \pi_i$. We summarize L-FedADMM in Algorithm 1.

3.4 MTL

In real-world applications, there is an urgent need to solve the problem of multiple learning tasks simultaneously, while exploiting commonalities and differences across tasks [199, 200]. As a running example, consider learning the behaviors of drivers on the road in the transport network based on their individual radar signal, image and trajectory. Each driver may generate data through a distinct distribution, and as a consequence, it is natural to fit the multiple distinct models to the decentralized data. This has resulted in the technique of MTL, where some clients have the shared task, while other clients have different ones. MTL has been applied successfully in many areas, including computer vision [5, 87, 154], bioinformatics and health informatics [62, 97], speech and natural language processing [113, 179, 202], and web applications [8, 9], which have shown to improve the performance. Since MTL learns separate models for each task among the clients, FL is naturally suited to MTL [152, 199, 200]. Next follows the formulated federated MTL problem, existing solution and our proposed approach.

In general, suppose there are N clients with m tasks, and each client i corresponds to the task $\psi(i) \in \{1, \dots, m\}$. For client i, it targets at learning the model $z_{\phi(i)} \in \mathbb{R}^d$ to fit its own data. Therefore, the federated MTL aims at learning the joint model $Z = \{z_1, \dots, z_m\}$ over the N clients and the federated problem is formulated as follows:

$$\min_{Z} \quad \sum_{i=1}^{N} f_i(z_{\phi(i)}) + R(Z), \tag{7}$$

where $f_i(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is the local loss function, and $R : \mathbb{R}^d \to \mathbb{R}$ is the regularization function. MOCHA solves the problem (7) via its dual problem with convex conjugate of the objective function [152]. For the local step in MOCHA, it minimizes a quadratic approximation of the dual subproblem. However, the approach suffers from the strong assumption, e.g., MOCHA assumes one-to-one mapping, namely each client i corresponds to the task i. Moreover, MOCHA requires all clients participating in the update at each iteration, which is unrealistic in many applications.

Federated ADMM for MTL. Since the existing approach MOCHA has limitations for wide applications, we propose a novel strategy based on ADMM to handle MTL. Our proposed approach exactly fits the centralized FL architecture. Moreover, our proposed approach is simple and convenient and thus is promising to be deployed in much wider applications.

To be specific, we view the local model z_i as the consensus variable, and introduce the variable $x_i \in \mathbb{R}^d$ as the new local model for training. Thus the MTL model that N clients aim to collaboratively learn is $\{x_i,\ i = 1, \dots, N\}$. Subsequently, we formulate the federated consensus optimization problem over the N clients as

$$\min_{x_i, i=0,\dots,N;} \quad \sum_{i=1}^{N} f_i(x_i) + R(Z), \qquad \text{s.t.} \quad x_i = z_{\psi(i)}, i = 1, \dots, N. \tag{8}$$

Then, we will use the distributed ADMM to solve the above federated problem. For the local model x_i update, it can be obtained via standard subproblem minimization in ADMM:

$$\min_{x_i} f_i(x_i) + \left\langle \lambda_i, x_i - z_{\phi(i)} \right\rangle + \frac{\rho}{2} \left\| x_i - z_{\phi(i)} \right\|_2^2, \tag{9}$$

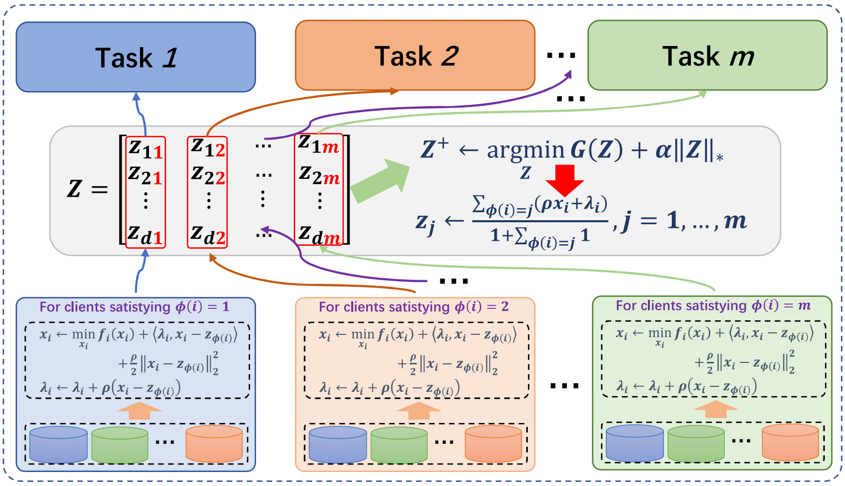

Task 1
Task 2
Task m
$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ z_{d1} & z_{d2} & \cdots & z_{dm} \end{bmatrix}$
$Z^+ \leftarrow \underset{Z}{\operatorname{argmin}}\, G(Z) + \alpha\|Z\|_*$
$z_j \leftarrow \frac{\sum_{\phi(i)=j}(\rho x_i + \lambda_i)}{1+\sum_{\phi(i)=j} 1}, j = 1, \ldots, m$
For clients satisfying $\phi(i) = 1$
For clients satisfying $\phi(i) = 2$
For clients satisfying $\phi(i) = m$
$x_i \leftarrow \min_{x_i} f_i(x_i) + \langle \lambda_i, x_i - z_{\phi(i)} \rangle + \frac{\rho}{2}\|x_i - z_{\phi(i)}\|_2^2$
$\lambda_i \leftarrow \lambda_i + \rho(x_i - z_{\phi(i)})$

Fig. 4. MTL via FL with ADMM strategy, the basic idea is each task corresponding to multiple clients, namely, for task $j, j = 1 \ldots, m$, its corresponding clients satisfy $\psi(i) = j$, where $i = 1, \ldots, N$. For each task j, there can be multiple clients satisfying $\psi(i) = j$ cooperate to obtain the common model z_j.

where λ_i is the Lagrangian multiplier. In addition to local model update, the client also needs to update λ_i: $\lambda_i \longleftarrow \lambda_i + \rho(x_i - z_{\phi(i)})$. It should be noted that while (9) can be solved via standard SGD, linear approach can be adopted for efficient computation. In the sequel, we discuss to solve the consensus matrix Z on the server by considering three scenarios:

- First, consider the case when $\psi(i) = i$ and $R(Z) = \alpha \|Z\|_*$, namely, we choose the nuclear norm to be the regularization function to exploit the commonalities and differences across tasks and obtain a low rank matrix solution Z^*. Moreover, $\psi(i) = i$ means there are N tasks and each task i corresponds to each client i (the same assumption with MOCHA). Then the updated consensus matrix Z^+ with low rank can be derived by solving the following subproblem:

$$Z^+ \longleftarrow \underset{Z}{\operatorname{argmin}}\ \alpha \|Z\|_* + \frac{\rho}{2}\left\|X - Z + \frac{1}{\rho}\Lambda\right\|_F^2, \tag{10}$$

  where $X = [x_1, \ldots, x_n]$, $Z = [z_1, \ldots, z_n]$ and $\Lambda = [\lambda_1, \ldots, \lambda_n]$. Then, we can acquire the closed-form solution $Z^+ \longleftarrow U \cdot \operatorname{diag}\{\operatorname{prox}_{\frac{\alpha}{\rho}\|\cdot\|_1}(\sigma)\} \cdot V^T$, where U, σ and V can be conveniently obtained via the singular value decomposition of the matrix $(X + {}^1/_\rho\Lambda)$.
- Second, consider the scenario of $\psi(i) \neq i$ and $R(Z) = \alpha \|Z\|_*$, then the subproblem is formulated in the following:

$$Z^+ \longleftarrow \underset{Z}{\operatorname{argmin}}\ G(Z) + \alpha \|Z\|_*, \tag{11}$$

  where for simplicity we denote $G(Z)$ as follows:

$$G(Z) := \sum_{i=1}^{N} \left(\left\langle \lambda_i, x_i - z_{\phi(i)} \right\rangle + \frac{\rho}{2} \left\|x_i - z_{\phi(i)}\right\|_2^2 \right). \tag{12}$$

  It can be seen the subproblem (11) has no closed-form solution, therefore, we propose to solve (11) via proximal gradient descent. Specifically, the gradient $\nabla_Z G(Z)$ can be derived column by column, i.e., $\nabla_{z_j} G(Z), j = 1, \ldots, m$ as follows $\nabla_{z_j} G(Z) = \sum_{\phi(i)=j} \rho z_j - (\rho x_i + \lambda_i)$. Subsequently, with the gradient $\nabla_Z G(Z)$, we are able to solve the subproblem (11) via the

ALGORITHM 2: Federated ADMM for MTL

```
1: server input: The communication round $R$ and iteration number $I$, initial $Z$, $\alpha$ and $\rho$.
2: client $i$'s input: The local iteration number $T$, initial $\eta$, $x_i$, and $y_{i,j}$.
3: for $r = 1, \dots, R$ do
4:     Server implements steps 5-6:
5:     Updates $Z$ considering the three scenarios respectively,
         — if $\psi(i) = i$ and $R(Z) = \alpha \|Z\|_*$:
           Obtain $Z^+$ via (10).
         — if $\psi(i) \neq i$ and $R(Z) = \alpha \|Z\|_*$:
           Perform (13) via the iteration $i = 0, I - 1$
         — if $\psi(i) \neq i$ and $R(Z) = Tr(ZZ^T)$:
           Update $Z$ via (14).
6:     Randomly sample clients $\mathcal{S} \subseteq [N]$ and transmit $z_{\phi(i)}$ to each client $i \in \mathcal{S}$.
7:     Clients implement steps 8-10 in parallel for $i \in \mathcal{S}$:
8:     After receiving $z_{\phi(i)}$, client $i \in \mathcal{S}$ updates $x_i$ by solving the subproblem (9), and
9:     updates $\lambda_i$: $\lambda_i \leftarrow \lambda_i + \rho(x_i - z_{\phi(i)})$
10:    Client $i$ transmits $(\rho x_i + \lambda_i)$ to the server.
11: end for
```

iteration $i = 0, I - 1$

$$Z^{i+1} \leftarrow \text{Prox}_{\lambda \eta_g \|\cdot\|_*} \left\{ Z^i - \eta_g \nabla_Z G(Z^i) \right\}, \tag{13}$$

where $\eta_g > 0$ is the step size.

— Third, consider the case when $\psi(i) \neq i$ and the regularization function $R(Z) = Tr(ZZ^T)$, then the consensus variable Z can be obtained via the closed-form solution:

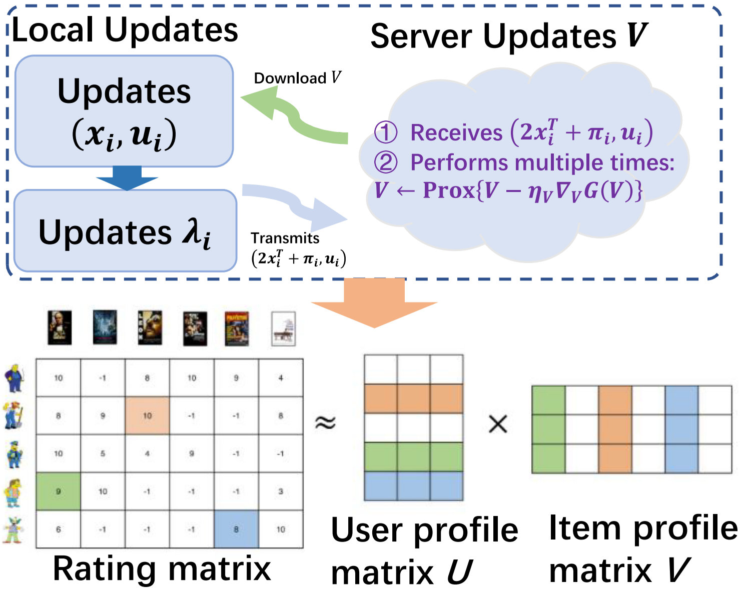
$$z_j \leftarrow \frac{\sum_{\psi(i)=j} (\rho x_i + \lambda_i)}{1 + \rho \sum_{\psi(i)=j} 1}, j = 1, \dots, m. \tag{14}$$

3.5 MF

MF is a prominent machine learning technique which has widely applied in various domains including recommendation system [78, 83] and environment monitoring [169]. While MF has been developed with security guarantee in centralized scenarios [78, 133], its application in FL scenarios is still under investigation. Chai et al. [23] propose a secure federated MF. However, it is susceptible to the error data and has not fully exploited the item correlations. Hence, we propose a robust and efficient strategy for the secure federated recommendation.

In the sequel, we first provide the MF model. Suppose there are N users and each user has rated a subset of M items, which forms the rating matrix $R \in \mathbb{R}^{N \times M}$, with the element $r_{i,j}$ representing the ith user rating the jth item. Moreover, we use $R_{i,:}$ to denote the ith row of the matrix R, which also means the ith user's rating on all the M items. Then, the recommendation system targets at predicting all users' rating on all items. To adopt the MF technique, the problem can be formulated as the decomposition of the rating matrix into the user profile matrix $U \in \mathbb{R}^{N \times d}$ and item profile matrix $V \in \mathbb{R}^{M \times d}$. Subsequently, the prediction of the ith user rating on the jth item can be given as $\langle u_i, v_j \rangle$, where u_i and v_j are the ith and jth row vectors in U and V, respectively. To learn U and V in federated settings, the following nonsmooth regularization problem is formulated:

$$\min_{U,V} \sum_{i=1}^{N} f_i(u_i, V) + \lambda \|U\|_{2,1} + \mu \|V\|_*, \tag{15}$$


Local Updates
Updates (x_i, u_i)
Updates λ_i
Download V
Transmits $(2x_i^T + \pi_i, u_i)$
Server Updates V
① Receives $(2x_i^T + \pi_i, u_i)$
② Performs multiple times:
$V \leftarrow \text{Prox}\{V - \eta_V \nabla_V G(V)\}$
≈
×
Rating matrix
User profile matrix U
Item profile matrix V

10	-1	8	10	9	4
8	9	10	-1	-1	8
10	5	4	9	-1	-1
9	10	-1	-1	-1	3
6	-1	-1	-1	8	10

Fig. 5. Federated MF. Its main ingredient is: first, each user holds there rating vector $R_{i,:}$, moreover, it initializes a user profile vector u_i, $(R_{i,:}, u_i)$ are kept private. The item matrix V can be shared through users. They collaborate with each other to learn (U, V).

where $f_i(u_i, V) = \|R_{i,:} - u_i \cdot V^T\|_2^2$. Here, we use the nuclear norm and $L_{2,1}$ norm instead of the 2-norm in [23] to make our approach exploit well in the item dependency and more robust to the data error [34]. For the FedOpt solution, we propose a three-blocked distributed ADMM based framework. First of all, we incorporate a consensus matrix X, with its ith row $x_i = u_i \cdot V^T$, subsequently the above MF problem can equivalent reformulated in the following three-blocked consensus problem with respect to (X, U, V):

$$\min_{X,U,V} \quad \sum_{i=1}^{N} f_i(x_i) + \lambda \|U\|_{2,1} + \mu \|V\|_*, \text{ s.t.} \quad x_i - u_i \cdot V^T = 0, i = 1, \ldots, N, \tag{16}$$

where $f_i(x_i) = \|x_i - R_{i,:}\|_2^2$. With distributed ADMM, we propose the following FedAvg-like architecture approach, namely, the item profile matrix V is updated on the server side while the ith row of the consensus matrix X and its own user profile vector u_i. Moreover, the Lagrangian multiplier π_i is also updated on the client i.

To be specific, after the server receives $(2x_i^T + \pi_i, u_i)$, it updates V on the server via the minimization:

$$V \leftarrow \operatorname*{argmin}_V \mu \|V\|_* + G_V(V), \tag{17}$$

where for simplicity we denote the summation term as $G_V(V) := \sum_{i=1}^N \left\|x_i - u_i V^T\right\|_2^2 + \sum_{i=1}^N \langle \pi_i, -u_i V^T \rangle$. Subsequently, we aim at solving the problem (17) via the proximal operation. We illustrate this by first deriving of the gradient $\nabla_V G_V(V)$: $\nabla_V G_V(V) = \sum_{i=1}^N \left\{2x_i^T V u_i^T - (2x_i^T + \pi_i) u_i\right\}$. Subsequently, with the gradient $\nabla_V G_V(V)$, we are able to solve the subproblem (17) via the iteration $i = 0, I - 1$

$$V^{i+1} \leftarrow \text{Prox}_{\mu\eta_V \|\cdot\|_*} \left\{V^i - \eta_V \nabla_V G(V^i)\right\}, \tag{18}$$

ALGORITHM 3: 3-Block Federated ADMM for MF

```
1:  server input: The communication round $R$ and iteration number $I$, initial $U$ and $V$, $\alpha$ and $\rho$.
2:  user $i$'s input: The local iteration number $T$, initial $\eta$, $x_i$, and $y_{i,j}$.
3:  for $r = 1, \ldots, R$ do
4:      Server implements steps 5-7:
5:      Receive $(2x_i^T + \pi_i, u_i)$ from users $i \in \mathcal{S}$
6:      Updates $V$ via (18) for $i = 0, \ldots, I-1$.
7:      Randomly sample users $\mathcal{S} \subseteq [N]$ and inform the users to download $V$.
8:      Users implement steps 8-11 in parallel for $i \in \mathcal{S}$:
9:      User $i$ download $V$, and performs the update orderly for $(x_i, u_i)$ via (19).
10:     updates $\pi_i$: $\pi_i \leftarrow \pi_i + \rho(x_i^T - V \cdot u_i^T)$
11:     User $i$ transmits $(2x_i^T + \pi_i, u_i)$ to the server.
12: end for
```

where $\eta_V > 0$ is the step size. After the server finishes the update for item profile matrix V, it can be shared by all users. Then, user i download V for its updating x_i and u_i with order and the problem can be formulated in the following:

$$(x_i, u_i) \leftarrow \underset{x_i, u_i}{\operatorname{argmin}} \left\| x_i - R_{i,:} \right\|_2^2 + \lambda \left\| u_i \right\|_2 + \frac{\rho}{2} \left\| x_i - u_i \cdot V^T \right\|^2 + \langle \pi_i, x_i - u_i \cdot V^T \rangle; \tag{19}$$

Specifically, the update for x_i can be obtained straightforwardly in closed-form. Subsequently, given current x_i, u_i can be updated via standard minimization. Finally, the client i updates π_i via: $\pi_i \leftarrow \pi_i + \rho(x_i^T - V \cdot u_i^T)$.

In general, the procedure can be concluded in the following, first, the user i download V, and performs the update orderly for (x_i, u_i) via (19). It also requires updating π_i via π_i ¸$\pi_i + \rho(x_i^T - V \cdot u_i^T)$. Then, the user transmits $(2x_i^T + \pi_i, u_i)$ to the server, while the server updates V via (18) for $i = 0, \ldots, I-1$. The above ADMM procedure can conveniently solve the MF problem in recommendation system under the federated settings, and the clients can perform their local updates in parallel.

4 Optimization

Federated learning is essentially based on the techniques of distributed optimization for the machine learning, which is abbreviated as FedOpt in this survey [80, 81, 117, 185]. The typical characteristics distinguish FedOpt from classical distributed optimization by the following three aspects:

- To begin with, while FedOpt targets at training a high-quality model from the decentralized data, its primary goal is to ensure the data privacy and security of each participating agent.
- Second, the real world applications of FL generally include massive participants, there is high probability that some participant remains inactive, since they maybe in offline, low-battery, and power off status, while the typical distributed optimization, requires all clients synchronously or a large portion of clients to asynchronously participate in each round update.
- Third, due to each client's own fashion, the local data among them are generally statistically heterogeneous, known as non-IID data. In general, it has been shown partial participation and data heterogeneity bring performance degradation and prevent learning from reaching fast convergence in FL scenarios.

As a result, building efficient and effective FedOpt algorithms has recently drawn substantial interests [81, 94, 96, 117].

4.1 Federated Optimization

In this section, we comprehensively review the existing literatures in FedOpt and characterize FedOpt algorithms from the aspects of search direction accuracy and the **alternating direction of multiplier based methods** (**ADMM**). In particular, the first-order approaches are the most popular choices. **We extensively review these works and find out their common algorithmic patterns, which includes the local acceleration and the global acceleration.** With the basic frameworks, we also provide popular techniques, e.g., quantization and sparsification for performance improvement.

4.1.1 Federated Problem Formulation. We first formulate the federated problem. Suppose there are N workers, and each worker $i \in [N]$ has the local loss function $f_i(x)$ with its own dataset $\mathcal{D}_i$ containing n_i samples. As a result, federated optimization targets at collaboratively solving the following empirical risk minimization problem over N workers and one server:

$$\min_x f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{20}$$

where f is the averaged loss function. The model as the optimization variable satisfies $x \in \mathbb{R}^d$. Moreover, the above functions satisfy $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$.

Implementation Strategy. In terms of the FedOpt implementation, there are two types: synchronous and asynchronous algorithms. In synchronous distributed optimization, all the workers perform the local updates and then transmit their corresponding information (gradients and variables, etc.) to the central server, which subsequently performs aggregation [7, 115, 116, 211]. However, due to the device heterogeneity in workers, the server always needs to wait for the slowest worker to push its information. This has significantly limited the scalability. On the contrary, in asynchronous algorithms, the servers conduct the updates upon receiving the information from the specified bunch of workers with the fastest speed, while it generally does not wait for the slower workers (or seldom wait in some scenarios [196]).

4.1.2 First-Order Scenario. In this scenario, the first-order information is adopted for local search direction. Specifically, momentum and variance reduction techniques can be used for acceleration. According to the acceleration position, we divide them into local and global acceleration schemes respectively, which can be conveniently instantiated to develop new algorithms.

Local acceleration. While the acceleration techniques such as the momentum and variance reduction have been widely utilized to accelerate SGD, e.g., Adam, AdaGrad and SVRG , they are also employed in the FedOpt to speed up the convergence. In particular, Liu et al. [109] has proposed MFL, its key idea is to extend the momentum gradient descent approach to the local update phase, and then the global server performs aggregating all the local models and the momentum terms. Similarly, FedAdam [139] extend Adam [79] to FedOpt. However, the model divergence issue persists in these techniques. [96, 171, 192, 203]. This is explained by the fact that FedOpt uses multiple local updates; hence, the local model may fast approach the local rather than the global optimum of the local loss function [75]. Additionally, the aggregated model will be further biased away from the optimal global one since only a small number of the clients engage in the update during each communication round. Intuitively, it can be attributed to the global aggregation with only the partial information from the participated clients. These algorithms share the common patterns—local acceleration strategy, which can be merged together in the Algorithm 4.

Global acceleration. The control variate strategy is frequently used for the global acceleration and has been shown to effectively mitigate the non-IID issue. Liang et al. [100] propose VRL-SGD

ALGORITHM 4: Local Acceleration Framework

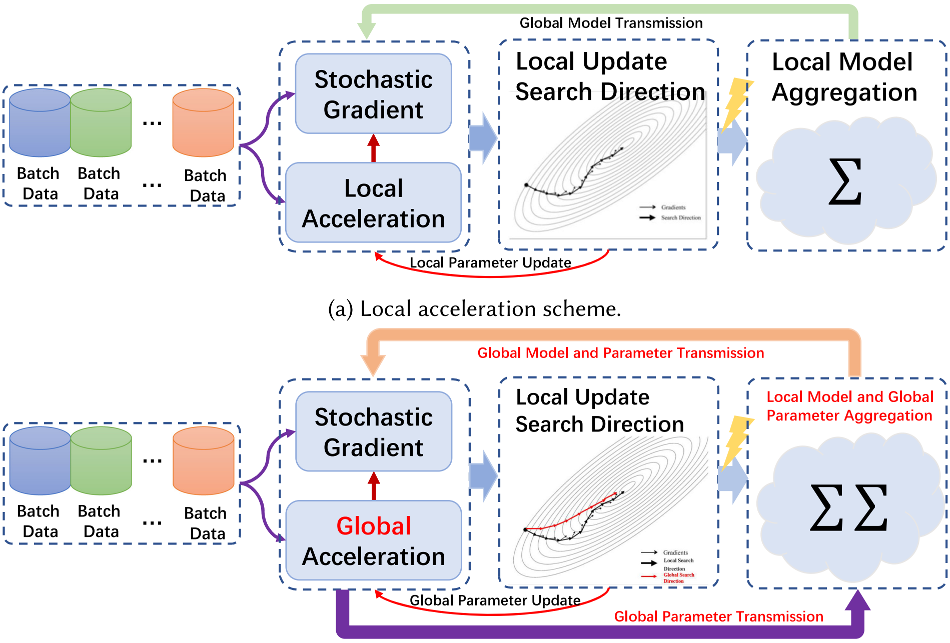
```
1:  server input: The communication round $R$, initial global model $x$.
2:  client $i$'s input: The local iteration number $T$, local model $x_i$, and the acceleration $\mathcal{L}_i$.
3:  for $r = 1, \ldots, R$ do
4:      Server implements steps 5-6:
5:      Updates the global model $x$ by aggregating $x_i$ for $i = 1, \ldots, N$.
6:      Sample clients $\mathcal{S} \subseteq [N]$ and transmit $x$ to each client $i \in \mathcal{S}$.
7:      Clients implement steps 8-15 in parallel for $i \in \mathcal{S}$:
8:      After receiving, initializing the local model $x_i$ with $x$.
9:      for $t = 1, \ldots, T$ do
10:         Sample a data sample from local dataset and calculate the stochastic gradient $g_i$.
11:         Obtain the search direction with $(g_i, \mathcal{L}_i)$: $\mathcal{V}_i(g_i, \mathcal{L}_i)$.
12:         Local model update: $x_i \leftarrow x_i - \eta \mathcal{V}_i(g_i, \mathcal{L}_i)$.
13:         The acceleration update via $\mathcal{U}_i$: $\mathcal{L}_i \leftarrow \mathcal{U}_i(x_i, g_i, \mathcal{L}_i)$.
14:     end for
15:     Client $i$ transmits the local model to the server.
16: end for
```

ALGORITHM 5: Global Acceleration Framework

```
1:  server input: The communication round $R$, initial global model $x$ and the acceleration $\mathcal{M}$.
2:  client $i$'s input: The local iteration number $T$, local model $x_i$, and the acceleration $\mathcal{M}_i$.
3:  for $r = 1, \ldots, R$ do
4:      Server implements steps 5-6:
5:      Updates $x$ and $\mathcal{M}$ by aggregating $x_i$ and $\mathcal{M}_i$ respectively for $i = 1, \ldots, N$.
6:      Sample clients $C \subseteq [N]$ and transmit $x$ and $\mathcal{M}$ to each client $i \in C$.
7:      Clients implement steps 8-15 in parallel for $i \in C$:
8:      After receiving, initializing the local model $x_i$ with $x$ and $\mathcal{M}_i$ with $\mathcal{M}$.
9:      for $t = 1, \ldots, T$ do
10:         Sample a data sample from local dataset and calculate the stochastic gradient $g_i$.
11:         Obtain the search direction with $(g_i, \mathcal{M}_i)$: $\mathcal{V}_i(g_i, \mathcal{M}_i)$.
12:         Local model update: $x_i \leftarrow x_i - \eta \mathcal{V}_i(g_i, \mathcal{M}_i)$.
13:         The acceleration update via $\mathcal{U}_i$: $\mathcal{M}_i \leftarrow \mathcal{U}_i(x_i, g_i, \mathcal{M}_i)$.
14:     end for
15:     Client $i$ transmits $(x_i, \mathcal{M}_i)$ to the server.
16: end for
```

which uses variance reduced convergence path strategy. It has shown the improved performance and has demonstrated the capability in dealing with the non-IID data. However, VRL-SGD does not support random client participation, which is frequent in FL. For addressing this issue, SCAFFOLD [75] has been proposed by utilizing the control variate to estimate the global gradient with the purpose of including as much information from clients as possible. It has demonstrated a notable performance increase for the FedOpt data heterogeneity problem. More recently, popular centralized algorithms (such SGD, Adam, etc.) may now adapt to FedOpt owing to a framework called Mime that has recently been developed [74].

We summarize these algorithms into the general global acceleration strategy in Algorithm 5 to provide the motivation for the development of novel algorithms. It can be seen these algorithms are



(a) Local acceleration scheme.

(b) Global acceleration scheme.

Fig. 6. Illustration of local and global updates in FL. (a) First, the local dataset block is randomly chosen for the gradient calculation; second, obtain the search direction by combining the local acceleration, which is subsequently used for local model update; third, the local acceleration is updated using local information; fifth, the local model is transmitted to the server for global aggregation. (b) Different with the local acceleration scheme, the global acceleration scheme also transmits the global acceleration for global aggregation.

different with the local acceleration scheme in that the acceleration adopts the global information. We further illustrate the global acceleration scheme in Figure 6.

Quantization. Despite the numerous advantages of distributed computing via FedOpt, significant limitations exist in the communication process, since massive workers transmit the gradients or the models for aggregation in the server, resulting in communication overhead. Quantization technique is a straightforward idea for compressing the communication quantities. Reisizadeh et al. [140] has proposed to quantize FedAvg, which has been shown reducing the communication overload. Another strategy, called QSGD [11], quantizes the gradient in each local worker with a specified random quantization operator and it costs only $(2 + \log(2N + 1))Nd$ in each round communication, when training a d-dimensional model with N clients. Jeremy et al. [16] propose SIGNSGD and finds only transmitting the sign of the gradient on each worker can still get the SGD-level convergence rate, while it only costs $2Nd$ bits in each round communication. As for the acceleration strategy for QSGD and SIGNSGD, SVRG is applied in QSGD, and the momentum method is adopted to SIGNSGD, the resultant algorithms are QSVRG and SIGNUM respectively. However, as for the QSGD, its convergence is based on unrealistic assumptions and can diverge in practice. To remedy this issue, Nesterov's momentum has been incorporated to the general distributed SGD with two-way compressions on both the workers and server [206], which has demonstrated numerically reducing the communication by about 32× and can converge as quickly as full-precision distributed momentum SGD to reach the same testing accuracy.

Sparsification. While the quantization is effective in reducing the communication overhead, the sparsification technique is another choice for the communication compression, where only the most important and information-preserving gradient entries are sent when using gradient sparsification technique.

Specifically, Dan et al. [12] and Stich et al. [153] study sparsifying gradients by magnitude with local error correction and selecting top K components. Another algorithm only transmits the large entries to the server [45], which shows the significant reduction in the communication overhead [10]. While the sparsification can bring performance degradation, deep gradient compression [103] combines it with other techniques including momentum correction, local gradient clipping, and momentum factor masking to achieve higher performance with considerable communication cost reduction. In fact, these strategies can deliver good results, but they have a few limitations. First, they currently have no guarantees of convergence; second, although the workers can perform the compression at a high rate, when the server implements aggregation of the disjoint top K, it will again become dense. To alleviate the problem, Sketched-SGD has been proposed by introducing the sketching technique in distributed SGD, which is motivated by the success of sketching methods in sub-linear/streaming algorithms [66]. The main idea of Sketched-SGD is communicating sketches with top K sparsification on both the server and workers. For the communication overhead reduction, Sketched-SGD only communicates $O(logd)$ information, while the general gradient compression method transmits $O(d)$.

Although sparsifying the transmitted gradients can improve the communication efficiency, it also increases the gradient variance, which may slow down the convergence speed. To mitigate the problem, Wangni et al. [175] propose a gradient sparsification technique to find the optimal tradeoff between sparsity and the gradient variance. Specifically, the key idea is to randomly drop out each coordinate g_i^j of the stochastic gradient vector g_i by probability $1 - p_j$ on the ith worker, and appropriately amplify the remaining coordinates to ensure that the sparsified gradient is unbiased. Then the variance of the quantized gradient $Q(g_i)$ can be bounded by $\mathbb{E} \sum_{j=1}^{d} [Q(g_i^j)^2] = \sum_{j=1}^{d} {(g_i^j)^2}/{p_j}$. where $Q(\cdot)$ is the quantization operator, g_i^j is the jth component of g_i. In addition, the expected sparsity $\mathbb{E} \left\| Q(g_i) \right\|_0$ can be computed by $\mathbb{E} \left\| Q(g_i) \right\|_0 = \sum_{j=1}^{d} p_j$. Subsequently, the target is to obtain the gradient as sparsified as possible with the variance as small as possible, which can be formulated as the following problem: $\min\limits_{p} \sum_{j=1}^{d} p_j,\ \text{s.t.}\ \sum_{j=1}^{d} {(g_i^j)^2}/{p_j} \leq \epsilon$, where ϵ is a small value to bound the variance.

It should be emphasized that none of the above compression (i.e., quantization and sparsification) based methods can learn the gradients, preventing them from converging to the true optimum in batch mode, making them incompatible with non-smooth regularizers, and slowing their convergence. To mitigate the problem, DIANA is proposed by compressing the gradient difference and performing the proximal minimization for dealing with the non-smooth regularizers in the local [120]. Another strategy, ADIANA combines the acceleration/momentum and compression strategies [99].

4.1.3 Second-Order Scenarios. The second-order scheme has attracted attention due to the higher accuracy in the search direction. Although they generally need more computation per iteration, they require fewer iterations to achieve similar results [19].

DANE minimizes the linear approximation of the objective function with Bregmen divergence measurement [146]. It has been found that each computational machine in DANE implicitly uses its local Hessian in quadratic objective cases, while no Hessians are explicitly computed. Wang et al. [170] propose a distributed Newton-type optimization method GIANT. Technically, GIANT realizes the local approximated Newton directions by the conjugate gradient descent method (CG),

Table 4. Representative Federated Optimization Methods Based on First-Order Gradient

Algorithm	Local update	Local steps	Local complexity	Transmission quantity	Global update	Convergence speed	Dealing with M.D.	SC/AS	PCP
VRL-SGD	Variance reduced SGD	Multiple	$O(T \cdot d)$	Local and Global models	Weight averaging	$O(\frac{1}{R})$	N/A	SC	×
MFL	MGD	Multiple	$O(Td)$	Local model and momentum Global model and momentum	Weight averaging	$O(\frac{1}{R})$	N/A	SC	×
QSGD	Calculate quantized gradient	Single	$O(d)$	Quantized gradient	Global SGD	$O(\frac{1}{R})$	N/A	SC	×
FedAvg	SGD	Multiple	$O(T \cdot d)$	Local model, Global model	Weight averaging	$O(\frac{1}{R \cdot T})$	✓	AS	×
SCAFFOLD	Variance reduced SGD	Multiple	$O(T \cdot d)$	Local model and gradient, Global model and gradient	Weight averaging	$O(\frac{1}{R \cdot T})$	✓	AS	✓
FedProx	PGD	Single	$O(d)$	Local model, Global model	Weight averaging	$O(\frac{1}{R})$	limited	AS	✓
DIANA	Calculate quantized gradient difference	Single	$O(d)$	Quantized gradient difference, Global model and gradient shift	Global PGD and gradient shift aggregation	$O(\frac{1}{R})$ and linear convergence rate	N/A	SC	×
MimeSVRG	Variance reduced SGD	Multiple	$O(2Td)$	Local model and gradient, Global model and aggregated gradient	Weight averaging	$O(\frac{1}{R \cdot T})$	✓	AS	✓

Specifically, M.D. is an acronym for model divergence, synchrony and asynchrony are abbreviated as SC and AS, respectively, the abbreviation of partial client participation is PCP.

which involves only Hessian-vector products in CG iterations. In [32], a novel method based on second-order information is proposed, known as DINGO and is derived by optimization of the gradient's norm as a surrogate function for the objective function. For further supporting second-order methods towards FL, Safaryan et al. [142] propose FedNL framework, and its main ingredient is to reuse past Hessian information and build the next Hessian estimate by updating the current estimate. Moreover, for reducing communication overload, the compression technique is adopted for Hessian compression. Furthermore, Agafonov et al. [6] propose the similar strategy FLECS-CGD, which also compresses the gradient information. However, they both suffer from high computation complexity, making them impractical in mobile applications with low power consumption. To mitigate this issue, Ma et al. [114] propose FedSSO approach, it updates the local model via SGD, and pushes the Hessian update and global model update to the server.

4.1.4 ADMM.

Another class of FedOpt algorithm is based on the alternating direction method of multipliers (ADMM) [20], which is simple and efficient. ADMM decouples the complicated problems with the coupling constraint(s) into smaller subproblems coordinated by a master problem. Moreover, it utilizes the augmented Lagrangian function [65, 135], which benefits numerical stability since it penalizes the equality constraint to a quadratic term. Therefore, ADMM is adept at solving the complicated problems with high degree of parallelization, and exhibits linear scaling as data is processed in parallel across devices, making it a natural fit in the large-scale distributive applications. In recent years, ADMM has received substantial attention in both academics and industry [20, 161]. With simple and straightforward modification, ADMM can be adapted to FedOpt.

For algorithmic development, Goldstein et al. [53] has incorporated Nesterov's acceleration strategy [125] in ADMM for solving the separate subproblem. It has reached the convergence speed of $O(1/R^2)$ in strongly convex problems. With simple modification, the accelerated ADMM can fit in

Table 5. Representative Federated Optimization Methods Based on Second-Order Gradient

Algorithm	Local update	Local steps	Local complexity	Transmission quantity	Global update	Convergence speed	Dealing with M.D.	SC/AS	PCP
FedNL	Calculate compressed Hessian and gradient	Single	$O(d^2)$	Compressed Hessian and. the gradient	Global model and Hessian update	sub-linear convergence.	N/A	AS	✓
FedSSO	SGD	multiple	$O(Td)$	Local and global models˙	BFGS Hessian and global model update	$O(1/R)$	N/A	AS	✓
FLECS-CGD	Calculate compressed Hessian and gradient	Single	$O(d^2)$	Compressed Hessian and the gradient˙ difference	Global model and Hessian update	sub-linear convergence.	N/A	AS	✓

Table 6. Representative Federated Optimization Methods Based on ADMM

Algorithm	Local update	Local steps	Local complexity	Transmission quantity	Global update	Convergence speed	Dealing with M.D.	SC/AS	PCP
CEADMM	Primal and dual updates	Multiple	$O(Td)$	Local model and Lagrangian multiplier, global model	Primal update	N/A	N/A	AS	✓
Fast ADMM	Primal and dual updates with acceleration	Single	$O(d)$	Local model and gradient, Lagrangian multiplier, global model and gradient	Primal update	$O(\frac{1}{R^2})$	N/A	SC	×
AsyncADMM	Primal and dual updates with acceleration	Single	$O(d)$	Local model and gradient, Global model and gradient	Primal update	$O(\frac{1}{R})$	N/A	AS	×

FedOpt. Another strategy is based on Anderson acceleration [195], it targets at speeding up the convergence of a fixed-point iteration [165] via the quasi-Newton method by adopting previous iterates by approximating its inverse Jacobian [40, 42]. It has shown its superior performance in the nonconvex problems of computer graphics tasks. Considering the topological network, DADMM introduces the undirected graph for modeling the network topology of agents, and the decentralized learning is performed over the agents [150]. For further improving the computational efficiency in DADMM, Ling et al. [104] further propose DLM, which has linearized the local update step in DADMM. Both DADMM and DLM have demonstrated the linear convergence rate under the assumption of strong convexity in the objective function. For supporting ADMM in FL, Zhou et al. [210] propose CEADMM approach, its key idea is to modify the classical ADMM by updating the local model and Lagrangian multiplier multiple times before transmitting to the server.

5 Privacy and Security

The privacy and security issues are important for the FL system, without proper protection, they may result in the consequence of huge losses, e.g., malicious attack in power grid results in catastrophic blackout shown in Figure 7 [156]. In general, privacy and security in FL can be divided into two categories, i.e., data privacy and model security [123, 186]. Specifically, in terms of the privacy issues, existing studies mainly focus on modeling the threat model of leaking information about the data of the participants in the process of FL [49] and developing privacy-preserving mechanisms to prevent privacy leakage [35, 67, 143, 157]. On the other hand, in terms of the security issues, existing studies mainly focus on modeling the threat model of influencing the performance of the model learned based on FL against malicious participants [186]. We will introduce each problem in terms of their corresponding attack models and defense methods, respectively.

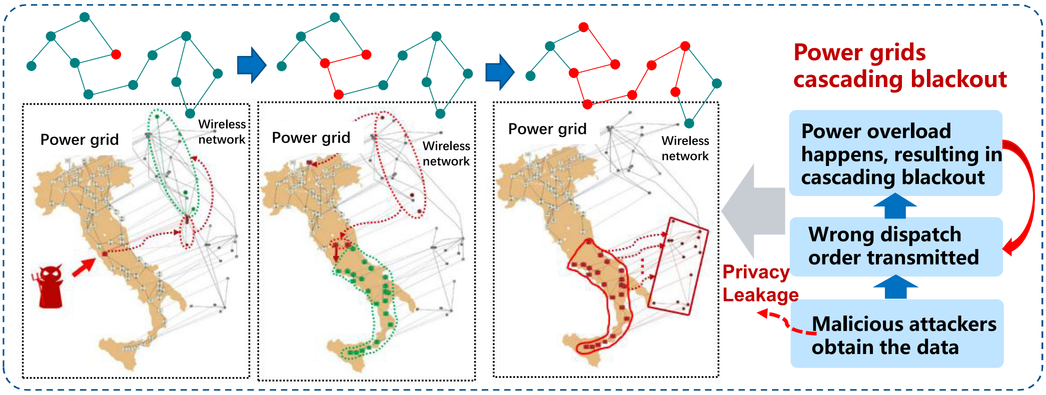

Power grid
Wireless network
Power grid
Wireless network
Power grid
Wireless network
Power grids cascading blackout
Power overload happens, resulting in cascading blackout
Wrong dispatch order transmitted
Malicious attackers obtain the data
Privacy Leakage

Fig. 7. Cascading blackout resulted by information leakage, directly impacting national security.

5.1 Data Privacy

Attack methods. The goal of the attackers in terms of privacy issues is to reveal the information about the data of the participants in the process of FL. Then, the attack methods can be divided into different categories based on what information is leaked. Here, we list major categories of attack methods as follows:

- **Reconstruction Attack:** The goal of this kind of attack is to reconstruct the data samples of the participants based on the transmitted parameters observed by the attacker.
- **Membership Inference Attack:** Different from the reconstruction attack, the goal of this kind of attack is to infer whether a given data sample was included by the dataset of a particular participant.

A number of existing studies focus on designing different attack methods to reveal the privacy leakage risks of FL systems [49, 124]. Nasr et al. [124] conduct a comprehensive analysis of risk of data privacy leakage in terms of the membership inference attack in both centralized learning and FL systems. Geiping et al. [49] focus on the reconstruction attack, and they show that multiple separate input images can be reconstructed from their average gradient in practice, where they propose an privacy attack strategy based on a magnitude-invariant loss along with optimization strategies, indicating that the FL algorithms do not have innate privacy-preserving property and the security guarantee is still necessary, e.g., the provable differential privacy.

Defense method. The mainstream defense methods against privacy attacks are based on perturbation mechanisms, random hash mechanisms, **homomorphic encryption** (**HE**), and **security multi-party computation** (**SMC**), and so on.

Specifically, the perturbation mechanisms protect the privacy of participants by adding random noise to their data or released parameters. Random hash mechanisms map the input data into random features while keeping some properties of the input data unchanged. These two kinds of mechanisms all fall into random mechanisms. In order to provide a criterion to measure the privacy protection level, these random mechanisms are usually required to satisfy the differential privacy, which is defined as follows:

Definition 1 ((ϵ, δ)-differential privacy). A randomized mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{O}$ satisfies (ϵ, δ)-differential privacy if and only if, for arbitrary adjacent datasets D_1 and D_2, and subset $O \in \mathcal{O}$, we have $Pr(\mathcal{M}(D_1) \in O) \leq e^{\epsilon} Pr(\mathcal{M}(D_2) \in O) + \delta$.

Table 7. Privacy and Security in FL

Ref.	Privacy/Security	FL Tasks	Categories	Idea	Architecture
[49]	Privacy	Information disclosure	Attack Method	Magnitude-invariant loss.	Centralized FL
[143]	Privacy	Data Summarization	Defense Method	Random Hash	Centralized FL
[67]	Privacy	Model optimization	Defense Method	SMC, Add perturbation	Centralized FL
[35]	Privacy	Federated contextual bandit	Defense Method	Upper confidence bounds (UCB), differential private perturbations	Centralized and decentralized FL
[157]	Privacy	Model optimization	Defense Method	Adversarial reconstruction, noise regularization, distance correlation minimization	Vertical FL
[17]	Security	Model optimization	Attack Method	Optimization based on indicators of stealth	Centralized FL
[18]	Security	Model optimization	Defense Method	Distance-based aggregation rule	Centralized FL
[190]	Security	Model optimization	Defense Method	Media and trimmed mean operations of the gradient	Centralized FL
[90]	Security	Multi-task learning	Defense Method	Accumulated loss	Decentralized FL
[52]	Security	Second order optimization	Defense Method	Newton's method	Centralized FL
[145]	Privacy	Multi-task learning	Defense Method	Encryption transformation	Clustered FL
[174]	Privacy	Model optimization	Defense Method	Differential privacy	Clustered FL
[122]	Security	Model optimization	Attack Method	label flipping and pixel patch backdoor	Decentralized FL
[209]	Security	Model optimization	Attack Method	Inject deviating gradients into the blockchain	Decentralized FL

As we can observe from the definition, (ϵ, δ)-differential privacy requires a certain level of indistinguishability of the input based on the observed output of the randomized mechanism, i.e., the ratio of the likelihood of generating the same output by the two inputs is bound by e^{ϵ} with at most the probability of δ to not satisfy this constraint. Thus, the parameters (ϵ, δ) quantify the hardness of reconstructing the input based on the output of the randomized mechanism.

On the other hand, HE is a technique that allows us to directly implement computing on the ciphertext and guarantee the correctness of the obtained results after decryption. On the other hand, SMC is another technique that allows us to obtain the correct computation results without leaking individual private data. Based on these methods, we can also preserve user privacy in the computing process of FL.

Sarpatwar et al. [143] protect the privacy of participants by learning their data summarization distributedly with differential privacy guarantee, which is solved based on two hash functions based on the Rahimi–Recht Fourier features [141] and MWEM method [57]. Jayaraman et al. [67] propose two perturbation mechanisms, i.e., the output perturbation method and the gradient perturbation method, of distributed learning to achieve differential privacy, where are all combined with the SMC. Specifically, in the output perturbation method, local models of different participants are aggregated once, and the differential privacy noise is added to the aggregated model before it is revealed. As for the gradient perturbation method, local gradients of different participants are aggregated iteratively to collaboratively train a model, and similarly the differential privacy noise is added to the aggregated gradients before it is revealed. Dubey et al. [35] investigate the FL problem of solving a contextual linear bandit based on cooperation of multiple participants, whose communication privacy should be protected, and they solve this problem based on **upper confidence bounds** (**UCB**) [4, 93] and differential private perturbations [147]. Sun et al. [157] focus on defending the privacy leakage in the VFL by proposing a framework containing three modules of adversarial reconstruction, noise regularization, and distance correlation minimization.

5.2 Model Security

Due to the privacy concern, the dataset as well as the training process of the participants is invisible to the server, which leads to considerable security risk. Specifically, malicious participants (referred to as Byzantine participants) can send elaborately designed parameters to the servers to influence the performance of the finally obtained model, e.g., converging to ineffective sub-optimal models or causing targeted poisoning [17]. In the following part of this section, we will first introduce the attack method in terms of security issues in the FL, and then we will introduce the defense method against these attack methods.

Definition 2 (Model Security). Given the test data $\mathcal{D}_{test}$, the tth communication round and the global model on the server w_G^t, a client model δ_i^t is aggregated: $w_i^t = w_G^t + \alpha_i w_i^t$, where α_i is the weighted parameter, the client model w_i^t is secure if and only if it satisfies: (a) $\mathcal{T}(\mathcal{D}_{test}, w_i^t) - \mathcal{T}(\mathcal{D}_{test}, w_G^t) > \gamma_t$, where the $\mathcal{T}$ is the performance evaluation metric; (b) $\max\{|R_i^u - R^l_{min,[N]-i}|, |R_i^l - R^u_{max,[N]-i}|\} < k_t$, with $R^l_{min,[N]-i}$ and $R^u_{max,[N]-i}$ being the minimum lower bound and maximum upper bound of the distance ranges.

Attack Method. Due to the different goals of the malicious participants, their attack methods can be divided into untargeted performance attack and targeted poisoning attack, which are introduced in details as follows:

- **Untargeted performance attack:** This kind of attack aims at reducing the overall performance of the obtained model through FL.
- **Targeted poisoning attack:** This kind of attacks aims at injecting a backdoor to the obtained model, which leads to its misclassification on a target sample set.

Bhagoji et al. [17] focus on the targeted poisoning attack. Specifically, they propose two indicators of stealth to detect Byzantine participants, and they further incorporate these two indicators in the optimization object of the adversary to derive a stealthy poisoning strategy. Their experiments showed that Byzantine-resilient aggregation strategies [18, 190], which are designed for defending the untargeted performance attack, are not robust to the proposed attacks.

Defense Method. The defense methods against the attacks of model security are mainly based on designing different aggregation rules of the model updated by the participants.

Blanchard et al. [18] focus on the untargeted performance attack, and they propose the concept of (α, f)-Byzantine resilience, where f is the maximum tolerant number of Byzantine participants, and α characterizes the maximum tolerant distance between the optimal model and obtained model under the attack of Byzantine participants. Authors in [18] further propose an aggregation rule to achieve (α, f)-Byzantine resilience. Specifically, this aggregation rule defines a score function based on the distance between the model parameter vector updated by each participant and its $n - f - 2$ closest model parameter vectors, and then the model parameter vector with the smallest score is selected to be the aggregation result. Yin et al. [190] propose two distributed gradient descent algorithms based on media and trimmed mean operations, and they further show that these algorithms have strong ability in defense against Byzantine failures in terms of both experiments and theoretical proofs, where their statistical error rates for strongly convex, non-strongly convex, and non-convex population loss functions are established. Li et al. [90] focus on federated MTL against Byzantine agents, and they propose an online weight adjustment rule based on the accumulated loss to measure the similarities among agents, which is shown to be more robust to Byzantine agents than traditional distant based on similarity measurement. Ghosh et al. [52] propose a federated optimization algorithm based on Newton's method, which is communication-efficient as well as robust against Byzantine failures.

5.3 Summary

For data privacy issues, we mainly discuss two attack methods: reconstruction attack and membership inference attack. Their key difference is that the former tries to reconstruct the data samples of the participants based on the transmitted parameters observed by the attacker; while the later aims at inferring whether a given data sample was included by the dataset of a particular participant. The data privacy preservation in FL can be more challenging due to the sporadic access to power and network connectivity, statistical heterogeneity in the data, and so on. Existing works are mostly developed based on the well-known privacy-preserving techniques, including: (1) HE [129] that allows arithmetic operations to be directly performed on ciphertexts; (2) SMC [189] that enables different participants with private and joint secure computation; and (3) **differential privacy** (**DP**) [148] that implements an overall additive noise mechanism by summing the same mechanism run at each participant (typically with less noise).

In the model security, it aims to send elaborately designed parameters to the servers to influence the performance of the finally obtained mode. Considering the different goals of the malicious participants, their attack methods can be divided into untargeted performance attack and targeted poisoning attack. The defense methods against poisoning attacks are also different. As for defenses against untargeted attacks, it mainly tries to measure the prominent differences between malicious attackers and the normal users and eliminates the malicious attackers. For defenses against targeted attacks, it mainly utilizes detection methods [98], which exploit activation statistics or model properties to determine whether a model is backdoored [166], or whether a training/test example is a backdoor example [162].

6 Application

As we have seen, the FL applications have witnessed rapid development, and we can expect this trend will continue. In this section, we will introduce the applications of FL from the perspectives of both technical and real applications.

6.1 Technical Applications

Reinforcement Learning. **Federated reinforcement learning** (**FRL**) is an emerging paradigm in machine learning that combines the advantages of FL and reinforcement learning. This allows agents to learn optimal strategies through interactions with the environment while protecting user data privacy [136]. Specifically, Wang et al. have adopted FRL to carry out dynamic system-level optimization and application-level enhancement to reach devices and edge node collaboration [172]. Zhang et al. have formulated the vehicular communication problem as a Markov decision process, and proposed FRL strategy to maximize the sum capacity of vehicle-to-infrastructure users while meeting the latency and reliability requirements of V2V pairs [198]. Zhang et al. have combined **Multi-Agent Reinforcement Learning** (**MARL**) and FL to exploit the device performance statistics and training behaviors to produce efficient client selection decisions [197]. Nguyen et al. use MDP to formulate the control optimization problem for each edge node, and develop a DDQN algorithm to quickly achieve the optimal flow rule match-field policy. This is then extended to federated DDQN for traffic monitoring [126].

Knowledge Distillation. **Federated knowledge distillation** (**FKD**) combines the advantages of knowledge distillation and FL, which compresses and improves models by transferring knowledge from a deep network to a smaller network, significantly reducing communication overhead and parameter quantity during model training while maintaining model performance. Wu et al. [178] have presented an FKD approach based on adaptive mutual knowledge distillation and dynamic gradient compression techniques to reach efficient communication. Chen et al. [30] have proposed MetaFed to improve the communication efficiency, which obtains a personalized model for each federation without a central server via the training process of common knowledge accumulation and personalization. Jeong et al. [68] have proposed federated distillation, where each device regards itself as a student and regards the average model output of all other devices as its teacher's output. Furthermore, the method of **federal augmentation** (**FAug**) is proposed to correct non-IID training datasets, which uses a **generative adversarial network** (**GAN**) and traded off between privacy leakage and communication overhead.

Recommendation. Recommendation techniques have been widely used to model user interests and mitigate information overload problems in many real applications[26, 151, 204]. For further improving privacy and security, FL has been introduced in recommendation [159]. HegedHus et al. [64] compare the performance of the recommendation based on FL with centralized structures and the gossip learning with decentralized structures, and show that they have a comparable performance. Wu et al. [177] propose a privacy-preserving recommendation method based on federated GNN, where the privacy is protected based on generating pseudo interacted items and add perturbation to the transmitted embedding vectors based on local differential privacy. In addition, high-order user-item interactions are modeled by matching neighboring users based on the HE technique. Qi et al. [137] propose a privacy-preserving new recommendation method, which protects user privacy based on clipping gradients and adding a Laplace noise to the clipped gradients before aggregation. Liu et al. [107] focus on utilizing the FL techniques to solve collaborative transfer learning problem between different SPs. Specifically, they solve this problem by keeping the CF prediction models private for different domains and using an intermediate transfer agent to aggregate user information across domains. Ammad et al. [13] propose a federated collaborative filtering algorithm for privacy-preserving recommendation. Flanagan et al. [44] propose a federated multi-view MF method to avoid collecting raw user data centralizedly by collaborating between the FL server, item server, and clients. Gao et al. [48] design a privacy-preserving cross-domain location recommendation by proposing a Laplace perturbation mechanism using both semantics and location information. Gao et al. [47] propose a differential private local CF algorithm based on the asymmetric flipping perturbation mechanism.

Natural Language Processing. NLP is extensive in mobile applications. To support various language understanding tasks with privacy-preserving, a foundation NLP model is often fine-tuned in a federated setting [22]. Chen et al. [25] propose FedMatch, a federated **Question Answering** (**QA**) model, which partitions the model into two components of backbone and patch and share only the backbone between different participants to distill the common knowledge while keep the patch component private to retain the domain information for different participants to overcome the data heterogeneity. What's more, they propose four different patch insertion ways and two types of patch architectures.

Spatio-Temporal Modeling. In the real world, especially when it comes to climatic and environmental issues, we frequently have measurements at a number of sites at a single moment and need frequently to forecast the results at unmeasured areas over the course of the following several time steps [59]. We can approach the spatial problem by constructing spatio-temporal modeling via FL. Meng et al. [118] propose a federated GNN-based method to predict traffic flow. Specifically, an encoder-decoder model is trained based on FedAvg to extract node-level temporal dynamics. Then, a GNN model in the server is used to model the uploaded embeddings of the node-level temporal dynamics to further extract the spatial dynamics based on split learning. Finally, the local encoder-decoder model is able to incorporate both temporal and spatial dynamics to implement the traffic flow prediction. Wang et al. [167] propose a federated learning algorithm to train an imitation learning based human trajectory generation algorithm while protecting the privacy of users whose trajectories are utilized as training data. Feng et al. [43] propose a federated mobility prediction neural network model, which adds Laplace noise to the location data to protect user privacy. In addition, they propose to divide the neural network into sub-modules with different groups by their privacy leakage risk level, and train different groups of modules with different privacy protection levels.

Knowledge Graph (KG). KGs, consisting of a large number of triple data with the form of (head, relation, tail), have become essential data supports for an increasing number of applications. To further enhance decentralized privacy-preserved applications of KGs, FL has been incorporated [27]. Chen et al. [28] propose an FL algorithm to complete multiple KG without sharing their private triples, where they propose an optimization-based model fusion procedure at clients. Peng et al. [132] propose a differential private federated algorithm to learn the KG embeddding, which utilizes a module named **privacy-preserving adversarial translation** (**PPAT**) to project embeddings of aligned entities and relations of different KG to capture their relationship. Specifically, this PPAT module is designed based on the technique of PATE-GAN [73] to protect user privacy.

6.2 Real Applications

Human Health. As is known, the human health record data is highly private and sensitive, and thus FL can be naturally fit for the cooperation in medical machine learning tasks across multiple institutions [77]. Fu et al. [46] utilize human mobility data to identify underlying COVID-19 infection cases, where a federated graph machine learning algorithm is designed to mine the contact information of crowd. Liu et al. [105] compare the performance of federated algorithms with different models to detect COVID-19 based on chest X-ray images, of which the models include MobileNet, ResNet18, and COVID-Net. Brisimi et al. [21] focus on utilizing the sparse Support Vector Machine to predict hospitalization based on FL. Specifically, they propose an iterative **cluster Primal Dual Splitting** (**cPDS**) algorithm to solve this problem. Yang et al. [187] propose an FL algorithm to automatically diagnose COVID-19 and protect patient privacy simultaneously. Specifically, they propose an algorithm named **Federated Learning on medical datasets using partial networks** (**FLOP**), which shares only a partial model between the server and clients to protect user privacy. Silva et al. [151] focus on investigating brain structural relationships with diseases based on FL techniques. Specifically, they propose a framework for data standardization, confounding factors correction, and multivariate analysis based on ADMM and **federated principal component analysis** (**fPCA**).

Smart Transportation. With the large-scale urbanization, it is urgently needed to intelligentize the **vehicular networks** (**VANET**) so as to mitigate the traffic congestion, and FL has become a promising technique due to its natural fit for the network topology [36, 134]. Liu et al. [111] mitigate the privacy and security issues of the existing data gathering fashion by combining **FL with gated recurrent unit** (**FedGRU**) neural network for traffic flow prediction. For further mining the spatio-temporal information for traffic flow prediction, Yuan et al. [194] first propose to transform the traffic trajectory into a graph representation, and then combines the FL paradigm, mitigating the privacy issue in centralized paradigm. Meng et al. [119] propose a cross-node FL combining with GNN approach, while it fully exploits the spatial and temporal dynamics of the server and devices, it allows the data generating on each node

7 Challenges and Opportunities

We have reviewed the major topics in FL. Next, we will provide the future research directions of FL, with the challenges and opportunities. They can be summarized as follows:

— **FL for semi-supervised learning:** FL typically assumes all data to be labeled, which is often impractical as many data may not have corresponding labels. Therefore, it is necessary to

Table 8. List of FL Applications

Ref.	Applications	Idea
Comparison [64]	Recommendation	Comparison between FL and gossip learning.
CNFGNN [118]	Traffic flow prediction	Modeling spatial dynamics based on federated graph neural network
FedGNN [177]	Recommendation	Generating pseudo interacted items, FE-based neighboring users matching
FedNewsRec [137]	Recommendation	Gradient clipping, adding Laplace noise
FedCT [107]	Recommendation	Cross-domain transfer learning, intermediate transfer agents on user devices.
FED-MNMF [44]	Recommendation	Collaborating between the FL server, item server, and clients
FCF [13]	Recommendation	Federated collaborative filtering.
FLOP [187]	Automatic diagnosis	Sharing a partial model between the server and clients.
FedMatch [25]	Question answer	Partition the model into the shared backbone and the private patch.
FedE [28]	Knowledge graph completion	Optimization-based model fusion procedure at clients.
FKGE [132]	Knowledge graph embedding	PATE-GAN based translation module between different knowledge graphs.
Falcon [46]	COVID-19 contact tracing	Incorporating FL and Graph Neural Networks to mine crowd contact information.
Experiments [105]	COVID-19 detection	Compare different models.
SegCaps [85]	COVID-19 detection	Data normalization, capsule network, blockchain.
cPDS [21]	Hospitalization prediction	SVM, iterative cluster primal dual splitting algorithm.
fPCA [151]	Brain structure analysis	ADMM, data standardization, confounding factors correction, and multivariate analysis.
CCMF [48]	Recommendation	Laplace perturbation mechanism using both semantics and location information.
DPLCF [47]	Recommendation	Asymmetric flipping perturbation mechanism.

combine FL with semi-supervised learning, that is, **federated semi-supervised learning** (**FSSL**). Existing methods mainly consider two application scenarios: the first is the Labels-at-Client Scenario, where labeled and unlabeled data are stored on local clients, and local clients perform standard semi-supervised learning training. The second is the Labels-at-Server Scenario, where labeled data are stored on the server side, while unlabeled data are stored on local clients. The process of supervised learning with labeled data and the process of unsupervised learning with unlabeled data will be carried out separately on the server and client sides. Although some existing works [69, 112] have proposed to handle these two situations well, the model needs to communicate frequently between the server and clients, leading to low communication efficiency. In addition, inherent problems in FL may become prominent in FSSL, including the data heterogeneity affecting the model performance and generalization ability, and frequent information transmission also brings high privacy and security challenges, making it easier for malicious attackers to obtain user information.

- **FL with knowledge distillation:** The knowledge distillation technology has a strong ability to fuse a large number of deep neural network models into one single efficient model by transferring knowledge between them. A number of studies have sought to combine the knowledge distillation technology to design more efficient FL method [61, 102]. Based on the knowledge distillation technology, we are able to develop more efficient model aggregation mechanism at the server, or design more efficient transmitted parameters between the participants and server, which are able to further fuse with the privacy-preserving mechanisms to better protect the privacy of the participants.
- **FL for digital twin paradigm:** The rising paradigm of digital twin requires accurately simulating physical objects. However, building the simulating model requires real-world data of the target physical objects, which leads to privacy concerns. The FL techniques provide a promising solution to this problem. Specifically, we can train the simulation models for physical objects at end devices or edge devices close to the data collector of the physical objects. Then, the FL methods enable us to accurately train the model based on the massive data collected from multiple physical objects without leaking private data of each physical object, which paves the way for constructing an accurate and effective digital twin system.
- **High performance theoretical approach:** The massive clients normally generate the data in their own fashion and the data is generally non-IID. Moreover, in most cases, since only a small portion of clients participate in the update, this will lead to model divergence problem.

SCAFFOLD and Mime have been proposed for handling this problem, both with the main idea of estimating the global gradient [74, 75]. Therefore, we can develop a more compact estimate of the global gradient, where the global gradient estimate is aggregated on the server and also updated on the clients, this will reach a performance improvement.

- **Decentralized FL over network:** As is known, the most popular FL architecture is centralized. However, in many real world applications, the clients are decentralized massively, they frequently communicate with each other without a server coordinating them, forming a complicated network. Hence, we are in urgent need of semi-decentralized or decentralized algorithms to cope with the increasingly complex networks. Very recently, researchers started to study decentralized FL in theoretical aspects [82, 168, 191]. However, decentralized FL includes the asynchronous local training, and the network dynamics make the clients participating and quitting the training frequently. These technical obstacles will bring significant challenges to the decentralized algorithm development.

8 Conclusion

In this survey, we have conducted a comprehensive review of FL. Firstly, we have conducted an in-depth analysis of the architecture of federated learning after the extensive review. In addition to the existing categorization, we have proposed a novel categorization, namely centralized, decentralized, and clustered FL, which we believe will facilitate the abstraction of network topology structures according to real-world problems, and then quickly adopt FL algorithms with corresponding structures. Furthermore, after sorting out most of the existing FL applications, we have concluded the basic paradigm, which is the combination of machine learning models with FL for distributed training. In addition, for some common theoretical models, such as non-smooth regularization, MTL, and matrix decomposition, we have also proposed FL solutions. We believe this will promote the application of FL in practice. For federated optimization, its essence is distributed optimization combined with specific constraints of FL settings. This survey only considers HFL due to its popularity and conducts a comprehensive review of existing federated optimization algorithms, concluding two frameworks: local acceleration framework and global acceleration framework. Thus, we can obtain new federated optimization algorithms by instantiating one of the two frameworks, thereby accelerating algorithm development. In addition, this survey also reviewed and classified the privacy and security issues of FL. Finally, we summarize the existing applications of FL from the aspects of technical and practical applications. With the increasing requirements for privacy and security, FL is still in rapid development, and has to face some challenges. These challenges include the application issues in emerging technologies such as digital twins; how to design decentralized algorithms to improve the ability of federated learning to handle complex tasks; and in the real complex environment, the performance of FL faces huge challenges. Therefore, how to use new technologies under the framework of FL to solve the real application problems remains the open issues that need to be considered in the future.

References

[1] 2021. F.748.13 : Technical framework for the shared machine learning system. *ITU-T* (2021).

[2] 2021. IEEE guide for architectural framework and application of federated machine learning. *IEEE Std 3652.1-2020* (2021), 1–69.

[3] M. S. H. Abad, E. Ozfatura, D. GUndUz, and O. Ercetin. 2020. Hierarchical federated learning across heterogeneous cellular networks. In *Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 8866–8870.

[4] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems* 24 (2011), 2312–2320.

[5] Abrar H. Abdulnabi, Gang Wang, Jiwen Lu, and Kui Jia. 2015. Multi-task CNN model for attribute prediction. *IEEE Transactions on Multimedia* 17, 11 (2015), 1949–1959.

[6] Artem Agafonov, Brahim Erraji, and Martin Takác. 2022. FLECS-CGD: A federated learning second-order framework via compression and sketching with compressed gradient differences. In *Proceedings of the NeurIPS 2022*.

[7] Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. 2014. A reliable effective terascale linear learning system. *Journal of Machine Learning Research* 15, 1 (2014), 1111–1133.

[8] Amr Ahmed, Mohamed Aly, Abhimanyu Das, Alexander J. Smola, and Tasos Anastasakos. 2012. Web-scale multi-task feature selection for behavioral targeting. ACM, 1737–1741.

[9] Amr Ahmed, Abhimanyu Das, and Alexander J. Smola. 2014. Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. 153–162.

[10] Alham Fikri Aji and Kenneth Heafield. 2017. Sparse communication for distributed gradient descent. arXiv:1704.05021. Retrieved from https://arxiv.org/abs/1704.05021

[11] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Proceedings of the Advances in Neural Information Processing Systems 30, December 4-9, 2017, Long Beach, CA, USA*. 1709–1720.

[12] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cedric Renggli. 2018. The convergence of sparsified gradient methods. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[13] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. arXiv preprint arXiv:1901.09888 (2019).

[14] Corey W. Arnold and William Speier. 2012. A topic model of clinical reports. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*. ACM, 1031–1032.

[15] Ozan Aygün, Mohammad Kazemi, Deniz Gündüz, and Tolga M. Duman. 2022. Hierarchical over-the-air federated edge learning. In *Proceedings of the ICC 2022—IEEE International Conference on Communications*. 3376–3381.

[16] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SIGNSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 559–568.

[17] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *Proceedings of the International Conference on Machine Learning*. 634–643.

[18] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurISP)*. 118–128.

[19] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *SIAM Review* 60, 2 (2018), 223–311.

[20] Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

[21] Theodora S. Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. 2018. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics* 112 (2018), 59–67.

[22] Dongqi Cai, Shangguang Wang, Yaozong Wu, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Federated few-shot learning for mobile NLP. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'23)*. Association for Computing Machinery, New York, NY, USA, Article 63, 1–17.

[23] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2021. Secure federated matrix factorization. *IEEE Intelligent Systems* 36, 5 (2021), 11–20.

[24] Chunjiang Che, Xiaoli Li, Chuan Chen, Xiaoyu He, and Zibin Zheng. 2022. A decentralized federated learning framework via committee mechanism with convergence guarantee. *IEEE Transactions on Parallel and Distributed Systems* 33, 12 (2022), 4783–4800.

[25] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2021. FedMatch: Federated learning over heterogeneous question answering data. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM'21)*. Association for Computing Machinery, New York, NY, USA, 181–190.

[26] Lei Chen, Jie Cao, Weichao Liang, Jia Wu, and Qiaolin Ye. 2022. Keywords-enhanced deep reinforcement learning model for travel recommendation. *ACM Transactions on the Web* 17, 1, Article 5 (2022), 21 pages.

[27] Mingyang Chen, Wen Zhang, Zhen Yao, Xiangnan Chen, Mengxiao Ding, Fei Huang, and Huajun Chen. 2022. Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 1966–1972.

[28] Mingyang Chen, Wen Zhang, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2022. FedE: Embedding knowledge graphs in federated setting. In *Proceedings of the 10th International Joint Conference on Knowledge Graphs (IJCKG'21)*. Association for Computing Machinery, New York, NY, USA, 80–88.

[29] Patrick Chen, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2021. DRONE: Data-aware low-rank compression for large NLP models. In *Proceedings of the Advances in Neural Information Processing Systems*. 29321–29334.

[30] Yiqiang Chen, Wang Lu, Xin Qin, Jindong Wang, and Xing Xie. 2023. MetaFed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare. *IEEE Transactions on Neural Networks and Learning Systems* (2023), 1–12.

[31] CHINA. 2017. The cybersecurity LAW of the people's republic of China. (2017).

[32] Rixon Crane and Fred Roosta. 2019. DINGO: Distributed newton-type method for gradient-norm optimization. In *Proceedings of the Advances in Neural Information Processing Systems*.

[33] Floriano De Rango, Antonio Guerrieri, Pierfrancesco Raimondo, and Giandomenico Spezzano. 2021. A novel edge-based multi-layer hierarchical architecture for federated learning. In *Proceedings of the 2021 IEEE DASC/PiCom/CBDCom/CyberSciTech*. 221–225.

[34] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. 2006. R1-PCA: Rotational invariant l1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*. New York, NY, USA, 281–288.

[35] Abhimanyu Dubey and AlexSandy Pentland. 2020. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems* 33 (2020), 6003–6014.

[36] Ahmet M. Elbir, Burak Soner, Sinem Coleri, Deniz Gündüz, and Mehdi Bennis. 2022. Federated learning in vehicular networks. In *Proceedings of the IEEE International Mediterranean Conference on Communications and Networking, MeditCom 2022, Athens, Greece, September 5-8, 2022*. IEEE, 72–77.

[37] Jianyu Wang et al. 2021. A Field guide to federated optimization. *CoRR* abs/2107.06917 (2021).

[38] Peter Kairouz et al. 2021. Advances and open problems in federated learning. *Now Foundations and Trends* 14, 1–2 (2021), 1–210.

[39] E.U. 2016. Regulation (EU) 2016/679 of the European Parliament and of the council on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT. Access Date: 2016.

[40] V. Eyert. 1996. A comparative study on methods for convergence acceleration of iterative vector sequences. *Journal of Computational Physics* 124, 2 (1996), 271–285.

[41] Dian Fan, Xiaojun Yuan, and Ying-Jun Angela Zhang. 2021. Temporal-structure-assisted gradient aggregation for over-the-air federated edge learning. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3757–3771.

[42] Haw-ren Fang and Yousef Saad. 2009. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications* 16, 3 (2009), 197–221.

[43] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. 2020. PMF: A privacy-preserving human mobility prediction framework via federated learning. *The Web Conference* (2020).

[44] Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan Eeik Tan, Suleiman A. Khan, and Muhammad Ammad-Ud-Din. 2020. Federated multi-view matrix factorization for personalized recommendations. *arXiv e-prints* (2020), arXiv–2004.

[45] Seide Frank, Fu Hao, Droppo Jasha, Li Gang, and Yu Dong. 2014. 1-Bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*.

[46] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2024. Privacy-preserving individual-level covid-19 infection prediction via federated graph learning. *ACM Transactions on Information Systems* 42, 3 (2024), 1–29.

[47] Chen Gao, Chao Huang, Dongsheng Lin, Depeng Jin, and Yong Li. 2020. DPLCF: Differentially private local collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 961–970.

[48] Chen Gao, Chao Huang, Yue Yu, Huandong Wang, Yong Li, and Depeng Jin. 2019. Privacy-preserving cross-domain location recommendation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–21.

[49] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

[50] Anousheh Gholami, Nariman Torkzaban, and John S. Baras. 2022. Trusted decentralized federated learning. In *Proceedings of the 2022 IEEE 19th Annual Consumer Communications and Networking Conference (CCNC).* 1–6.

[51] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. In *Proceedings of the Advances in Neural Information Processing Systems.* 19586–19597.

[52] Avishek Ghosh, Raj Kumar Maity, and Arya Mazumdar. 2020. Distributed newton can communicate less and resist byzantine workers. In *Proceedings of the NeurIPS.*

[53] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard G. Baraniuk. 2014. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences* 7, 3 (2014), 1588–1623.

[54] Biyao Gong, Tianzhang Xing, Zhidan Liu, Wei Xi, and Xiaojiang Chen. 2022. Adaptive client clustering for efficient federated learning over non-IID and imbalanced data. *IEEE Transactions on Big Data* (2022), 1–1.

[55] Yan Gou, Ruiyu Wang, Zongyao Li, Muhammad Ali Imran, and Lei Zhang. 2022. Clustered hierarchical distributed federated learning. In *Proceedings of the ICC 2022-IEEE International Conference on Communications.* 177–182.

[56] A. Gouissem, K. Abualsaud, E. Yaacoub, T. Khattab, and M. Guizani. 2022. Robust decentralized federated learning using collaborative decisions. In *Proceedings of the 2022 International Wireless Communications and Mobile Computing.* 254–258.

[57] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A simple and practical algorithm for differentially private data release. In *Proceedings of the NIPS.*

[58] Morgan Harvey, Fabio Crestani, and Mark James Carman. 2013. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management.* Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.), ACM, 2309–2314.

[59] Toshihiro Hasegawa, Hitomi Wakatsuki, Hui Ju, Shalika Vyas, Gerald C. Nelson, Aidan Farrell, Delphine Deryng, Francisco Meza, and David Makowski. 2022. A global dataset for the projected impacts of climate change on four major crops. *Scientific Data* 9, 1 (2022), 58.

[60] Abolfazl Hashemi, Anish Acharya, Rudrajit Das, Haris Vikalo, Sujay Sanghavi, and Inderjit Dhillon. 2022. On the benefits of multiple gossip steps in communication-constrained decentralized federated learning. *IEEE Transactions on Parallel and Distributed Systems* 33, 11 (2022), 2727–2739.

[61] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group knowledge transfer: Federated learning of large CNNs at the edge. In *Proceedings of the Advances in Neural Information Processing Systems.*

[62] Dan He, David Kuhn, and Laxmi Parida. 2016. Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction. *Bioinformatics* 32, 12 (2016), 37–43.

[63] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. 2022. Byzantine-robust decentralized learning via self-centered clipping. *CoRR* abs/2202.01545 (2022).

[64] István Hegedűs, Gábor Danner, and Márk Jelasity. 2019. Decentralized recommendation based on matrix factorization: A comparison of gossip and federated learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD).* 317–332.

[65] M. R. Hestenes. 1969. Multiplier and gradient methods. *Journal of Optimization Theory and Applications* 4 (1969), 303–320.

[66] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir braverman, Ion Stoica, and Raman Arora. 2019. Communication-efficient distributed SGD with sketching. In *Proceedings of the Advances in Neural Information Processing Systems.* Curran Associates, Inc.

[67] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. 2018. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18).* 6346–6357.

[68] 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. *CoRR* abs/1811.11479 (2018). arXiv:1811.11479.

[69] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency and disjoint learning. In *Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.*

[70] Di Jiang, Yuanfeng Song, Yongxin Tong, Xueyang Wu, Weiwei Zhao, Qian Xu, and Qiang Yang. 2019. Federated topic modeling. In *Proceedings of the CIKM 2019, Beijing, China, November 3-7, 2019.* ACM, 1071–1080.

[71] Di Jiang, Yongxin Tong, Yuanfeng Song, Xueyang Wu, Weiwei Zhao, Jinhua Peng, Rongzhong Lian, Qian Xu, and Qiang Yang. 2021. Industrial federated topic modeling. *ACM Transactions on Intelligent Systems and Technology* 12, 1 (2021), 2:1–2:22.

[72] Shusen Jing and Chengshan Xiao. 2022. Federated learning via over-the-air computation with statistical channel state information. *IEEE Transactions on Wireless Communications* (2022), 1–1.

[73] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proceedings of the International Conference on Learning Representations*.

[74] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2021. Mime: mimicking centralized stochastic algorithms in federated learning. *Advances in Neural Information Processing Systems (NeurIPS 2021)*, 34 (2021), 28663–28676.

[75] Sai Praneeth Reddy Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Jakkam Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the ICML*.

[76] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2020. Blockchained on-device federated learning. *IEEE Communications Letters* 24, 6 (2020), 1279–1283.

[77] Junu Kim, Kyunghoon Hur, Seongjun Yang, and Edward Choi. 2022. Universal EHR federated learning framework. *CoRR* abs/2211.07300 (2022). arXiv:2211.07300.

[78] Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, Junbum Shin, and Sungwook Kim. 2018. Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption. *ACM Transactions on Privacy and Security* 21, 4 (2018), 17:1–17:30.

[79] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

[80] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: distributed machine learning for on-device intelligence. *CoRR* abs/1610.02527 (2016). arXiv:1610.02527.

[81] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: strategies for improving communication eiciency. *CoRR* abs/1610.05492 (2016).

[82] George P. Kontoudis and Daniel J. Stilwell. 2022. Fully decentralized, Scalable gaussian processes for multi-agent federated learning. *CoRR* abs/2203.02865 (2022). https://doi.org/10.48550/arXiv.2203.02865

[83] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[84] Caner Korkmaz, Halil Eralp Kocas, Ahmet Uysal, Ahmed Masry, Oznur Ozkasap, and Baris Akgun. 2020. Chain FL: Decentralized federated machine learning via blockchain. In *Proceedings of the 2020 2nd International Conference on Blockchain Computing and Applications (BCCA)*. 140–146.

[85] Rajesh Kumar, Abdullah Aman Khan, Jay Kumar, A Zakria, Noorbakhsh Amiri Golilarz, Simin Zhang, Yang Ting, Chengyu Zheng, and WenYong Wang. 2021. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. *IEEE Sensors Journal* (2021).

[86] Hanjiang Lai, Yan Pan, Cong Liu, Liang Lin, and Jie Wu. 2013. Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers* 62, 6 (2013), 1221–1233.

[87] Maksim Lapin, Bernt Schiele, and Matthias Hein. 2014. Scalable multitask representation learning for scene classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1434–1441.

[88] Chengxi Li, Gang Li, and Pramod K. Varshney. 2022. Decentralized federated learning via mutual knowledge transfer. *IEEE Internet of Things Journal* 9, 2 (2022), 1136–1147.

[89] Chengxi Li, Gang Li, and Pramod K. Varshney. 2022. Federated learning with soft clustering. *IEEE Internet of Things Journal* 9, 10 (2022), 7773–7782.

[90] J. Li, W. Abbas, and X. Koutsoukos. 2020. Byzantine resilient distributed multi-task learning. *Advances in Neural Information Processing Systems* (2020).

[91] Jun Li, Yumeng Shao, Kang Wei, Ming Ding, Chuan Ma, Long Shi, Zhu Han, and H. Vincent Poor. 2022. Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation. *.IEEE Transactions on Parallel and Distributed Systems* 33, 10 (2022), 2401–2415.

[92] Lei Li, Deborah Chang, Lei Han, Xiaojian Zhang, Joseph Zaia, and Xiu-Feng Wan. 2020. Multi-task learning sparse group lasso: A method for quantifying antigenicity of influenza A(H1N1) virus using mutations and variations in glycosylation of Hemagglutinin. In *Proceedings of the BMC Bioinformatics Volume*.

[93] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 661–670.

[94] Tian Li, Anit Kumar Sahu, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *Proceedings of the 3rd MLSys Conference, Austin, TX, USA*.

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

[96] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the convergence of FedAvg on non-IID data. In *Proceedings of the ICLR*.

[97] Yan Li, Jie Wang, Jieping Ye, and Chandan K. Reddy. 2016. A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. New York, NY, USA, 1715–1724.

[98] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia. 2024. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 35, 1 (2024), 5–22.

[99] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtarik. 2020. Acceleration for compressed gradient descent in distributed and federated optimization. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 5895–5904.

[100] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. 2019. Variance reduced local SGD with lower communication complexity. *CoRR* abs/1912.12844 (2019).

[101] Frank Po-Chen Lin, Seyyedali Hosseinalipour, Sheikh Shams Azam, Christopher G. Brinton, and Nicolò Michelusi. 2021. Semi-decentralized federated learning with cooperative D2D local model aggregations. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3851–3869.

[102] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. In *Proceedings of the Advances in Neural Information Processing Systems*.

[103] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. *CoRR* abs/1712.01887 (2017).

[104] Qing Ling, Wei Shi, Gang Wu, and Alejandro Ribeiro. 2015. DLM: Decentralized linearized alternating direction method of multipliers. *IEEE Transactions on Signal Processing* 63, 15 (2015), 4051–4064.

[105] Bingjie Yan et al. 2021. Experiments of federated learning for COVID-19 chest X-ray images. *Advances in Artificial Intelligence and Security: 7th International Conference (ICAIS'21)*, Dublin, Ireland.

[106] Lumin Liu, Jun Zhang, S. H. Song, and Khaled B. Letaief. 2020. Client-edge-cloud hierarchical federated learning. In *Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC)*. 1–6.

[107] Shuchang Liu, Shuyuan Xu, Wenhui Yu, Zuohui Fu, Yongfeng Zhang, and Amelie Marian. 2021. FedCT: Federated collaborative transfer for recommendation. In *Proceedings of the SIGIR 21*. 716–725.

[108] Shengli Liu, Guanding Yu, Xianfu Chen, and Mehdi Bennis. 2022. Joint user association and resource allocation for wireless hierarchical federated learning with IID and Non-IID data. *IEEE Transactions on Wireless Communications* 21, 10 (2022), 7852–7866.

[109] Wei Liu, Li Chen, Yunfei Chen, and Wenyi Zhang. 2020. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems* 31, 8 (2020), 1754–1766.

[110] Wei Liu, Li Chen, and Wenyi Zhang. 2022. Decentralized federated learning: Balancing communication and computing costs. *IEEE Transactions on Signal and Information Processing over Networks* 8 (2022), 131–143.

[111] Yi Liu, James J. Q. Yu, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. 2020. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal* 7, 8 (2020), 7751–7763.

[112] Zewei Long, Liwei Che, Yaqing Wang, Muchao Ye, Junyu Luo, Jinze Wu, Houping Xiao, and Fenglong Ma. 2020. FedSiam: Towards adaptive federated semi-supervised learning. arXiv preprint arXiv:2012.03292 (2020).

[113] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task Sequence to Sequence Learning. In *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[114] Xin Ma, Renyi Bao, Jinpeng Jiang, Yang Liu, Arthur Jiang, Jun Yan, Xin Liu, and Zhisong Pan. 2022. FedSSO: A Federated server-side second-order optimization algorithm. *CoRR* abs/2206.09576 (2022). https://doi.org/10.48550/arXiv.2206.09576

[115] Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*. USA, 456–464.

[116] Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon Mann. 2009. Efficient large-scale distributed training of conditional maximum entropy models. In *Proceedings of the Advances in Neural Information Processing Systems*.

[117] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y. Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th AISTATS*.

[118] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. 2021. Cross-node federated graph neural network for spatio-temporal data modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

[119] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. 2021. Cross-node federated graph neural network for spatio-temporal data modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*. 1202–1211.

[120] Konstantin Mishchenko, Eduard A. Gorbunov, Martin Takác, and Peter Richtárik. 2019. Distributed learning with compressed gradient diferences. *CoRR* abs/1901.09269 (2019).

[121] B. Mishra, G. Meyer, and R. Sepulchre. 2011. Low-rank optimization for distance matrix completion. In *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference.* 4455–4460.

[122] Arup Mondal, Harpreet Virk, and Debayan Gupta. 2022. BEAS: Blockchain enabled asynchronous and secure federated machine learning. *CoRR* abs/2202.02817 (2022).

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

[124] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP).* 739–753.

[125] Yurii Nesterov. 1983. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences* 269 (1983), 543–547.

[126] Tri Gia Nguyen, Trung V. Phan, Dinh Thai Hoang, Tu N. Nguyen, and Chakchai So-In. 2021. Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks. *IEEE Transactions on Cognitive Communications and Networking* 7, 4 (2021), 1048–1065.

[127] 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.

[128] U.S. SELECT COMMITTEE ON ARTIFICIAL INTELLIGENCE of the NATIONAL SCIENCE AND TECHNOLOGY COUNCIL. 2019. The national artificial intelligence research and development strategic plan: 2019 Update. Retrieved from https://www.nitrd.gov/pubs/National-AI-RD-Strategy-2019.pdf. Access Date: 2019.

[129] P. Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques.* 223–238.

[130] Chao Pan, Jin Sima, Saurav Prakash, Vishal Rana, and Olgica Milenkovic. 2023. Machine unlearning of federated clusters. In *Proceedings of the 11th International Conference on Learning Representations.* Retrieved from https://openreview.net/forum?id=VzwfoFyYDga

[131] Dimitris Papailiopoulos, Alexandros Dimakis, and Stavros Korokythakis. 2013. Sparse PCA through Low-rank Approximations. In *Proceedings of the 30th International Conference on Machine Learning.* PMLR, Atlanta, Georgia, USA, 747–755.

[132] H. Peng, H. Li, Y. Song, V. Zheng, and J. Li. 2021. Differentially private federated knowledge graphs embedding. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.*

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

[134] Jason Posner, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh. 2021. Federated learning in vehicular networks: Opportunities and solutions. *IEEE Network* 35, 2 (2021), 152–159.

[135] M. J. D. Powell. 1978. Algorithms for nonlinear constraints that use lagrangian functions. *Mathematical Programming* 14, 1 (1978), 224–248.

[136] Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng. 2021. Federated reinforcement learning: Techniques, applications, and open challenges. *CoRR* abs/2108.11887 (2021). arXiv:2108.11887.

[137] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings.* 1423–1432.

[138] Zhe Qu, Rui Duan, Lixing Chen, Jie Xu, Zhuo Lu, and Yao Liu. 2022. Context-aware online client selection for hierarchical federated learning. *IEEE Transactions on Parallel and Distributed Systems* 33, 12 (2022), 4353–4367.

[139] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. 2021. Adaptive federated optimization. In *Proceedings of the 9th International Conference on Learning Representations, Virtual Event, Austria, May 3-7.*

[140] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics.* PMLR, Online, 2021–2031.

[141] Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. 2012. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *Journal of Privacy and Confidentiality* 4, 1 (2012), 65–100.

[142] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. 2022. FedNL: Making newton-type methods applicable to federated learning. In *Proceedings of the International Conference on Machine Learning, Baltimore, Maryland, USA.* 18959–19010.

[143] Kanthi Sarpatwar, Karthikeyan Shanmugam, Venkata Sitaramagiridharganesh Ganapavarapu, Ashish Jagmohan, and Roman Vaculin. 2019. Differentially private distributed data summarization under covariate shift. *Advances in Neural Information Processing Systems* 32 (2019), 14459–14469.

[144] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2021. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems* 32, 8 (2021), 3710–3722.

[145] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2021. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems* 32, 8 (2021), 3710–3722.

[146] Ohad Shamir, Nati Srebro, and Tong Zhang. 2014. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31st International Conference on Machine Learning*. PMLR, Bejing, China, 1000–1008.

[147] Roshan Shariff and Or Sheffet. 2018. Differentially private contextual linear bandits. *Advances in Neural Information Processing Systems* 31 (2018).

[148] Elaine Shi, T.-H. Hubert Chan, Eleanor Gilbert Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011.*

[149] Lu Shi, Jiangang Shu, Weizhe Zhang, and Yang Liu. 2021. HFL-DP: Hierarchical federated learning with differential privacy. In *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM).* 1–7.

[150] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. 2014. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing* 62, 7 (2014), 1750–1761.

[151] Santiago Silva, Boris A. Gutman, Eduardo Romero, Paul M. Thompson, Andre Altmann, and Marco Lorenzi. 2019. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019).* IEEE, 270–274.

[152] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S. Talwalkar. 2017. Federated multi-task learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA.* 4424–4434.

[153] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. Sparsified SGD with memory. In *Proceedings of the Advances in Neural Information Processing Systems.* Curran Associates, Inc.

[154] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry Steven Davis, and Wen Gao. 2018. Multi-task learning with low rank attribute embedding for multi-camera person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 5 (2018), 1167–1181.

[155] Zhou Su, Yuntao Wang, Tom H. Luan, Ning Zhang, Feng Li, Tao Chen, and Hui Cao. 2022. Secure and efficient federated learning for smart grid with edge-cloud collaboration. *IEEE Transactions on Industrial Informatics* 18, 2 (2022), 1333–1344.

[156] Chih-Che Sun, Adam Hahn, and Chen-Ching Liu. 2018. Cyber security of a power grid: State-of-the-art. *International Journal of Electrical Power and Energy Systems* 99 (2018), 45–56.

[157] Jiankai Sun, Yuanshun Yao, Weihao Gao, Junyuan Xie, and Chong Wang. 2021. Defending against reconstruction attack in vertical federated learning. *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with (ICML'21).*

[158] Yuchang Sun, Jiawei Shao, Yuyi Mao, Jessie Hui Wang, and Jun Zhang. 2022. Semi-decentralized federated edge learning for fast convergence on non-IID data. In *Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC).* 1898–1903.

[159] Zehua Sun et al. 2024. A survey on federated recommendation systems. *IEEE Transactions on Neural Networks and Learning Systems.* 1–15.

[160] Akihito Taya, Takayuki Nishio, Masahiro Morikura, and Koji Yamamoto. 2022. Decentralized and model-free federated learning: Consensus-based distillation in function space. *IEEE Transactions on Signal and Information Processing over Networks* (2022), 799–814.

[161] Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit B. Patel, and Tom Goldstein. 2016. Training neural networks without gradients: A scalable ADMM approach. In *Proceedings of the 33nd International Conference on Machine Learning, New York City, NY, USA.* 2722–2731.

[162] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.* 8011–8021.

[163] Sheng-Po Tseng, Jan-Yue Lin, Wei-Chien Cheng, Lo-Yao Yeh, and Chih-Ya Shen. 2022. Decentralized federated learning with enhanced privacy preservation. In *Proceedings of the 2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW).* 1–1.

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

[165] Homer F. Walker and Peng Ni. 2011. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis* 49, 4 (2011), 1715–1735.

[166] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. IEEE, 707–723.

[167] Huandong Wang, Changzheng Gao, Yuchen Wu, Depeng Jin, Lina Yao, and Yong Li. 2023. PateGail: A privacy-preserving mobility trajectory generator with imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 14539–14547.

[168] Huiyuan Wang, Xuyang Zhao, and Wei Lin. 2022. Heterogeneous federated learning on a graph. *CoRR* abs/2209.08737 (2022).

[169] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah M'hamed. 2016. Sparse mobile crowdsensing: Challenges and opportunities. *IEEE Communications Magazine* 54, 7 (2016), 161–167.

[170] Shusen Wang, Fred Roosta, Peng Xu, and Michael W. Mahoney. 2018. GIANT: Globally improved approximate newton method for distributed optimization. In *Proceedings of the Advances in Neural Information Processing Systems*.

[171] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221.

[172] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. 2019. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network* 33, 5 (2019), 156–165.

[173] Xing Wang and Yijun Wang. 2022. Asynchronous hierarchical federated learning. *CoRR* abs/2206.00054 (2022).

[174] Yuntao Wang, Zhou Su, Yanghe Pan, Tom H. Luan, Ruidong Li, and Shui Yu. 2022. Social-aware clustered federated learning with customized privacy preservation. arXiv:2212.13992 [cs.CR].

[175] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient sparsification for communication-efficient distributed optimization. In *Proceedings of the Advances in Neural Information Processing Systems*.

[176] Ybl Wei, N. C. Luong, D. T. Hoang, Y. Jiao, and C. Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials* PP, 99 (2020), 1–1.

[177] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie. 2021. FedGNN: Federated graph neural network for privacy-preserving recommendation. *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction With (ICML'21)*.

[178] 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.

[179] Fangzhao Wu and Yongfeng Huang. 2015. Collaborative multi-domain sentiment classification. In *Proceedings of the 2015 IEEE International Conference on Data Mining*. 459–468.

[180] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. 2021. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing* 1, 1 (2021).

[181] Yue Xiao, Yu Ye, Shaocheng Huang, Li Hao, Zheng Ma, Ming Xiao, Shahid Mumtaz, and Octavia A. Dobre. 2021. Fully decentralized federated learning-based on-board mission for UAV swarm system. *IEEE Communications Letters* 25, 10 (2021), 3296–3300.

[182] Bo Xu, Wenchao Xia, Wanli Wen, Pei Liu, Haitao Zhao, and Hongbo Zhu. 2022. Adaptive hierarchical federated learning over wireless networks. *IEEE Transactions on Vehicular Technology* 71, 2 (2022), 2070–2083.

[183] Chunmei Xu, Shengheng Liu, Zhaohui Yang, Yongming Huang, and Kai-Kit Wong. 2021. Learning rate optimization for federated learning exploiting over-the-air computation. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3742–3756.

[184] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. 2020. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications* 19, 3 (2020), 2022–2035.

[185] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2, Article 12 (2019), 19 pages.

[186] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 1–19.

[187] Qian Yang, Jianyi Zhang, Weituo Hao, Gregory P. Spell, and Lawrence Carin. 2021. Flop: Federated learning on medical datasets using partial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3845–3853.

[188] Yuzhi Yang, Zhaoyang Zhang, and Qianqian Yang. 2021. Communication-efficient federated learning with binary neural networks. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3836–3850.

[189] Andrew Chi-Chih Yao. 1982. Protocols for secure computations (extended abstract). In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*. IEEE Computer Society, 160–164.

[190] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the International Conference on Machine Learning (ICML)*. 5650–5659.

[191] Guangsheng Yu, Xu Wang, Caijun Sun, Qin Wang, Ping Yu, Wei Ni, and Ren Ping Liu. 2023. Ironforge: An open, secure, fair, decentralized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*. 1–15.

[192] H. Yu, S. Shen, S. Yang, and S. Zhu. 2019. Parallel restarted SGD with faster convergence and less communication: demystifying why model averaging works for deep learning. In *Proceedings of the AAAI*.

[193] Honglin Yuan, Manzil Zaheer, and Sashank J. Reddi. 2021. Federated composite optimization. In *Proceedings of the 38th International Conference on Machine Learning*. Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 12253–12266.

[194] Xiaoming Yuan, Jiahui Chen, Jiayu Yang, Ning Zhang, Tingting Yang, Tao Han, and Amir Taherkordi. 2022. FedSTN: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* (2022), 1–11.

[195] Juyong Zhang, Yue Peng, Wenqing Ouyang, and Bailin Deng. 2019. Accelerating ADMM for efficient simulation and optimization. *ACM Transactions on Graphics* 38, 6, Article 163 (2019), 21 pages.

[196] Ruiliang Zhang and James Kwok. 2014. Asynchronous distributed ADMM for consensus optimization. In *Proceedings of the 31st International Conference on Machine Learning*. PMLR, Bejing, China, 1701–1709.

[197] Sai Qian Zhang, Jieyu Lin, and Qi Zhang. A multi-agent reinforcement learning approach for efficient client selection in federated learning. In *Proceedings of the AAAI 2022 Virtual Event, February 22 - March 1, 2022*. 9091–9099.

[198] Xinran Zhang, Mugen Peng, Shi Yan, and Yaohua Sun. 2020. Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. *IEEE Internet of Things Journal* 7, 7 (2020), 6380–6391.

[199] Yu Zhang and Qiang Yang. 2018. An overview of multi-task learning. *National Science Review* 5, 1 (2018), 30–43.

[200] Yu Zhang and Qiang Yang. 2022. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2022), 5586–5609.

[201] Zehui Zhang, Ningxin He, Dongyu Li, Hang Gao, Tiegang Gao, and Chuan Zhou. 2022. Federated transfer learning for disaster classification in social computing networks. *Journal of Safety Science and Resilience* 3, 1 (2022), 15–23.

[202] Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2017. Feature constrained multi-task learning models for spatiotemporal event forecasting. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1059–1072.

[203] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with Non-IID data. *CoRR* abs/1806.00582 (2018).

[204] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the World Wide Web Conference*. Republic and Canton of Geneva, CHE, 167–176.

[205] Haifeng Zheng, Min Gao, Zhizhang Chen, and Xinxin Feng. 2021. A distributed hierarchical deep computation model for federated learning in edge computing. *IEEE Transactions on Industrial Informatics* 17, 12 (2021), 7946–7956.

[206] Shuai Zheng, Ziyue Huang, and James Kwok. 2019. Communication-efficient distributed blockwise momentum SGD with error-feedback. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[207] Xiangyu Zhong, Xiaojun Yuan, Huiyuan Yang, and Chenxi Zhong. 2022. UAV-assisted hierarchical aggregation for over-the-air federated learning. In *Proceedings of the IEEE Global Communications Conference, GLOBECOM 2022, Rio de Janeiro, Brazil, December 4-8, 2022*. IEEE, 807–812.

[208] Zhengyi Zhong, Weidong Bao, Ji Wang, Xiaomin Zhu, and Xiongtao Zhang. 2022. FLEE: A hierarchical federated learning framework for distributed deep neural network over cloud, edge, and end device. *ACM Transactions on Intelligent Systems and Technology* 13, 5, Article 71 (2022), 24 pages.

[209] Sicong Zhou, Huawei Huang, Wuhui Chen, Pan Zhou, Zibin Zheng, and Song Guo. 2020. PIRATE: A blockchain-based secure framework of distributed machine learning in 5G networks. *IEEE Network* 34, 6 (2020), 84–91.

[210] Shenglong Zhou and Geofrey Ye Li. 2021. Communication-Efficient ADMM-based federated learning. *CoRR* abs/2110.15318 (2021).

[211] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex Smola. 2010. Parallelized stochastic gradient descent. In *Proceedings of the Advances in Neural Information Processing Systems*.

Received 5 March 2023; revised 28 February 2024; accepted 6 April 2024