Advancements in Federated Learning: Models, Methods, and Privacy

HUIMING CHEN*, Department of Electronic Engineering, Tsinghua University, China
HUANDONG WANG*, Department of Electronic Engineering, Tsinghua University, China
DEPENG JIN, Department of Electronic Engineering, Tsinghua University, China
YONG LI†, Department of Electronic Engineering, Tsinghua University, China

Federated learning (FL) is a promising technique for addressing the rising privacy and security issues. Its main ingredient is to cooperatively learn the model among the distributed clients without uploading any sensitive data. In this paper, we conducted a thorough review of the related works, following the development context and deeply mining the key technologies behind FL from both theoretical and practical perspectives. 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. Based on these frameworks, we have instantiated FedOpt algorithms. As privacy and security is the fundamental requirement 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

---

*The first two authors have equal contributions.

Authors' addresses: Huiming Chen, chenhuiming@mail.tsinghua.edu.cn, Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084; Huandong Wang, Department of Electronic Engineering, Tsinghua University, Beijing, China, wanghuandong@tsinghua.edu.cn; Depeng Jin, Department of Electronic Engineering, Tsinghua University, Beijing, China, jindp@tsinghua.edu.cn; Yong Li, Department of Electronic Engineering, Tsinghua University, Beijing, China, liyong07@tsinghua.edu.cn.

---

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 [35] and legislative efforts in data security by the United States [112] and China [27], 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. As a new paradigm of machine learning, it is essentially a distributed machine learning framework that collaboratively solves a learning task by the federation of participating 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 [104]. It has been generally categorized into the following architectures [155]:

- Horizontal FL (HFL): HFL is the most common scenario in FL, where the data owners share the same feature space but different samples. HFL has been widely investigated since this scenario is widely popular in real applications. For example, with increasing health consciousness, a large amount of people wear smartwatch 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 over clients have different feature spaces but share the same sample ID space. It was first considered for two-party collaboration and recently has been extended to multiple-party scenario. VFL matches the real world applications very well. For example, a patient can have the medical record in different hospitals and each may have distinct features. With many samples, these hospitals will have more complete features and is 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 ID spaces. For example, in social computing network [165], it requires highly expensive and time consuming data labeling process. FTL techniques can be applied to alleviate the computation burden and provide the solutions for the entire sample and feature space under a federation.

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 processing power to train the model, preventing the leakage of sensitive data without data transmission [150]. Moreover, reinforcement learning has been incorporated for optimizing the resource allocations over both clients and servers [133]. It has reached the incentive mechanism to encourage the clients' participation.

Currently, the field of FL has witnessed many outstanding research achievements, and these include performance [67, 88, 125] and security enhancements[31, 44, 134], etc. 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 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 security defense 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, and 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 abstract problems from a specified actual application and propose solutions for FL?
- How to deploy multiple (semi-) decentralized parameter servers to establish a (semi-) decentralized FL system?
- How to build an efficient, secure, trustworthy, and scalable FL system?

Paper	Description
[155]	Concepts and applications of FL in terms of its definition, privacy techniques, and categories, including horizontal FL, vertical FL, and federated transfer learning.
[111]	Federated machine learning in wireless networks in terms of its applications and challenges.
[86]	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.
[148]	Survey on FL implementing in mobile edge networks in terms of several challenges, including communication costs, resource allocation, and privacy and security.
[34]	Survey on comprehensive FL.

Table 1. Selected surveys of FL.

Differences between existing surveys. There have been a number of existing surveys about FL techniques. Next, we provide the introduction of the representative surveys. As the pioneer FL survey, Yang et al. [155] introduce the concepts and applications of FL in terms of its definition, privacy techniques, and categories, including horizontal FL, vertical FL, and federated transfer learning. Niknam et al. [111] investigate federated machine learning in wireless networks in terms of its applications and challenges. Li et al. [86] focus on 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. Wei et al. [148] focus on FL implementing in mobile edge networks in terms of several challenges, including communication costs, resource allocation, and privacy and security. Kairouz et al. [34] focus on the privacy and
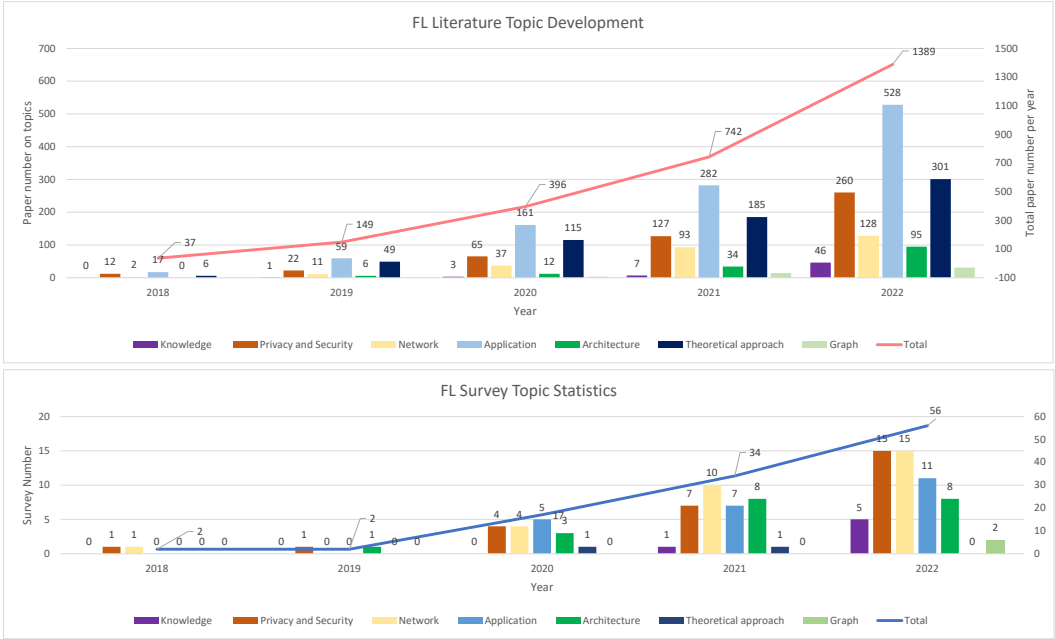
Fig. 1. The statistics on FL related topic papers from years 2018 to 2022. It can be seen FL has been under fast development and can be expected to maintain quick growth in near future. Specifically, the topics of privacy and security, theoretical approaches (optimization in majority) and applications dominates this field, the surveys on privacy and security and applications follows this pattern while this area significantly lacks the optimization related surveys.

security issues in the FL system, and they mainly summarize existing studies in terms of attack methods and defence methods.

Different from existing surveys, we have a thoughtful summary and discussion of existing literature, 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 [33], 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 provide the general frameworks including local and global acceleration strategies. In addition, based on the existing surveys on FL applications, we have provided a generalized federated base models and offered our solutions to facilitate larger-scale applications.

Our contributions. In this survey, we provide a comprehensive and motivative review for targeting at developing 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. 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 papers and surveys from the year 2018 to 2022, **including 2,713 research papers and 93 surveys**, and the results are shown in Fig. 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 papers released in 2022, which is nearly the sum of works in previous years. Especially, with the development of smart city 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 2 surveys on the theoretical approach topic.
- **Federated base model abstraction and our proposed solutions:** From Fig. 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, and matrix factorization, in the 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 for algorithm development. **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 reviewed and we also provide the instantiations in the supplementary material. Section 5 studies the privacy and security issues. Section 6 presents both the technical and real applications. Section 7 illustrate the challenges and future directions.

2 ARCHITECTURE

FL architecture plays the fundamental role in FL [140], building FL architectures can be effective and practical for improving the performance. Based on the distribution characteristics of the data, three types of FL techniques have been identified: horizontal FL, vertical FL, and federated transfer learning [155]. 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 and a number of clients, with the central server coordinating clients for cooperative learning [86, 140], the decentralization [137] or hierarchy [119] have become increasingly popular
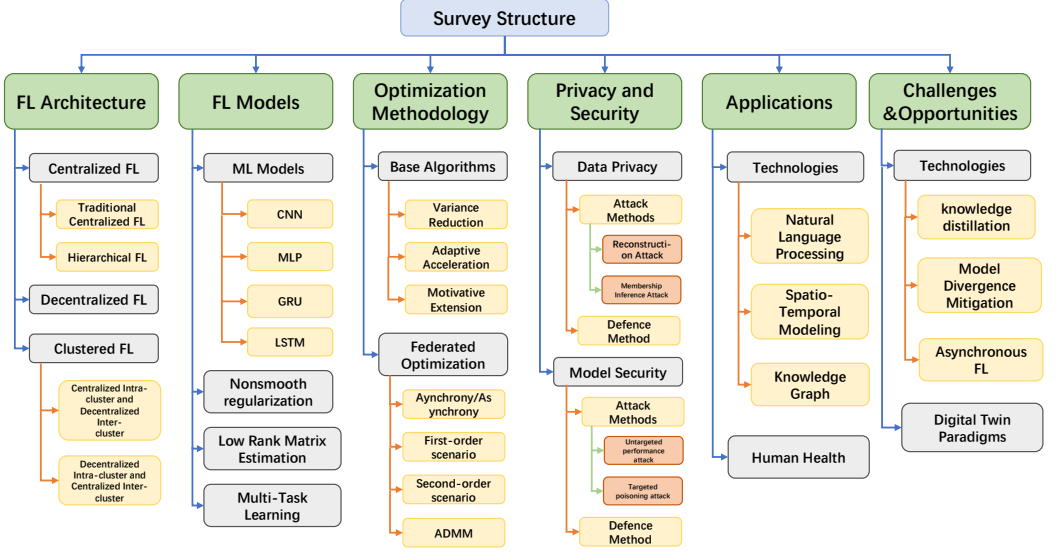
Fig. 2. Organization of this survey.

recently. Therefore, we will categorize the FL architectures into three scenarios: centralized FL, decentralized FL and clustered FL.

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 [104], where each client trains the local model with the its data and communicates with the server for aggregation [157].

Traditional FL. The traditional centralized FL has been among the most popular choice in real applications. Its major bottleneck is the communication overhead. For mitigating this issue, Yang et al. [154] combine device selection and beamforming design, and investigate the superposition property of a wireless MAC for achieving the fast global model aggregation. By considering the characteristics of the wireless channel, Xu et al. [153] 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. [37] first characterize the intrinsic temporal structure of the model aggregation series via a Markovian probability model and develop a message passing based algorithm with low complexity and near-optimal performance. For further communication efficiency, Jing et al. [64] adopt statistical channel state information. However, traditional centralized FL still exists other challenges including statistical heterogeneity, computation overhead, security and privacy [140].

Hierarchical FL. This scenario generally exists in the well-known cloud-edge-client system and is organized in a tree structure [128], the major goal is to alleviate the above issues in traditional FL [95]. Specifically, the cloud layer coordinates the edge layer, while the edge layer cooperates with the terminal layer. Since the edge layer has taken on the communication overhead from the cloud layer and the computation from the terminal layer [171], this architecture has been shown the capability of high communication efficiency and low latency in the asynchronous and heterogeneous system [146, 168]. For further improving the system robustness, wireless channel

has been considered for hierarchical over-the-air FL (HOTAFL) [12]. Liu et al. [97] and Xu et al. [152] investigate the issues of joint user association and wireless resource allocation 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. [82] propose a decentralized framework with the integration of the blockchain to delegate the responsibility of storing the model to the nodes on the network, it also optimizes the computing resource allocation. Liu et al. [99] suggest a generic decentralized FL framework , which implements both multiple local updates and multiple inter-node communications with the compression technique periodically. Hashemi et al. [55] propose DeLi-CoCo and demonstrate how the decentralized FL benefiting from additional gossip stages between gradient iterations for faster convergence. For further improving the performance, Li et al. [79] propose a Def-KT algorithm to fuse models among clients by transferring their learned knowledge to each other and has alleviated the client-drift issue. Xiao et al. [151] develop a decentralized FL framework with an inexact stochastic parallel random walk, which can be partially immune to the time-varying dynamic network.

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. [57] propose a Self-Centered Clipping (SCClip) algorithm for Byzantine-robust consensus and optimization. Che et al. [21] adopt the committee mechanism to track the uploaded local gradients, which has shown both performance and security improvement.

Categories	Topological structure	Feature	Ref.
Centralized FL	Traditional Centralized FL	One centralized server aggregates all model parameters updated by clients.	[104], [157], [154], [153], [37]
	Hierarchical FL	The client-edge-cloud structure: One cloud server is connected to edge servers in the middle layer. Every edge server connects to partial clients in the bottom layer.	[95], [128], [146], [168], [3], [12], [97], [152], . [29], [170]
Decentralized FL	Decentralized FL	Without central server. All clients communicate with each other in a distributed way.	[68], [76], [139], [99], [55], [45], [51], [21], . [79], [151]
Clustered FL	Centralized Intra-cluster, Decentralized Inter-cluster	Some client are connected with one edge server. The edge servers are decentraliedly connected with each other.	[50], [135].
	Decentralized Intra-cluster, Centralized Inter-cluster	Each client is decentraliedly connected with each other. Global aggregations is aggregated by a central server.	[46],[90], [80].

Table 2. Existing architectures in FL.

2.3 Clustered FL

In this scenario, users are distributed and partitioned into clusters. It exploits geometric properties of the FL loss surface to group the client population into clusters with jointly trainable data distributions [46, 125]. 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 communicates with the service provider (SP) for the aggregation. Then, the SPs are deployed in decentralized manner and they communicate with their neighboring SPs for information exchange. Here, communication
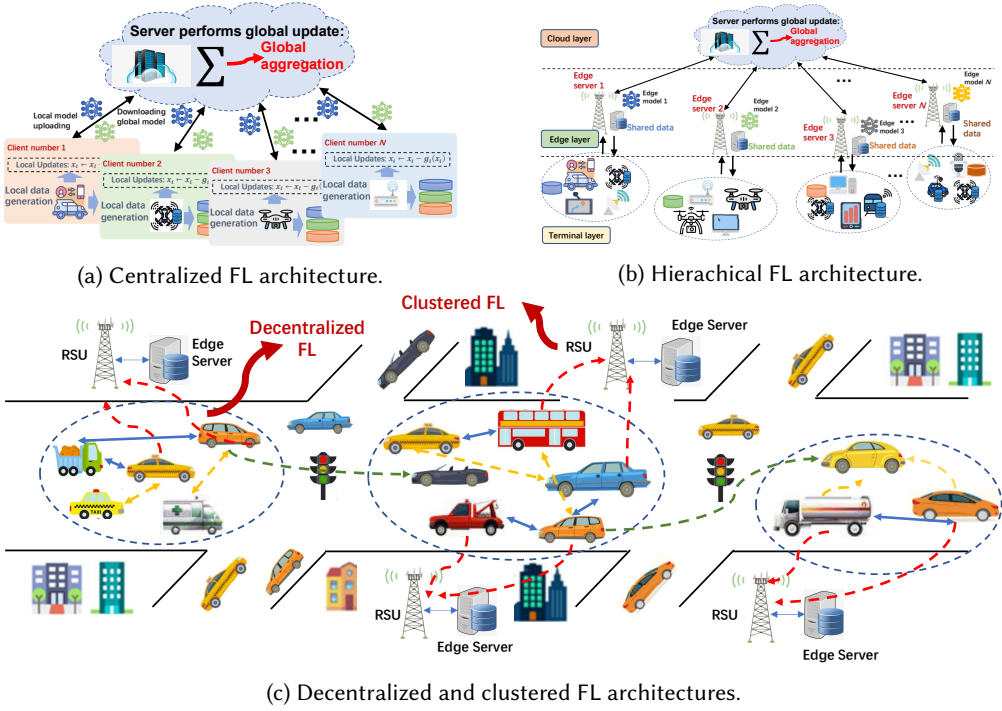
(a) Centralized FL architecture.

(b) Hierachical FL architecture.

(c) Decentralized and clustered FL architectures.

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

strategy plays the crucial role in the performance. Gou et al. [50] 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. [90] 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 [49].

2.4   Summarization

Centralized FL is fundamental in this field, but due to the simple and fixed pattern, it is not flexible and scalable, which will significantly limit the applications in large-scale networks. Decentralized FL can reach the learning tasks by only requiring the clients communication with their neighbors. We can expect it will have great potential for applications in large-scale networks. However, it still demands high performance algorithms and insurance in 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., 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 multi-task learning under FL solution. Next follows matrix factorization, 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.

Model	Federated training procedure
MLP	
CNN	The client performs multiple SGD to obtain the local model,
LSTM	and then the server performs aggregation to update the global model.
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.

Table 3. Training models via FL paradigm.

3.1 Machine Learning Models

Many machine learning models can be collaboratively trained via FL framework, e.g., as the representatives 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) [104]. Another popular model is the long short-term memory (LSTM), which has been utilized in FL for the language modeling [104]. Gated recurrent unit neural network (GRU) based on FedAvg solution has been applyed for the traffic flow prediction [100]. 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 [11], and web search log mining [53]. It has been extended to FL settings due to the increasing concerns on the data privacy issue [62, 63]. Specifically, Jiang et al. [62] propose a federeted 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 the 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 [78, 83]. Moreover, the biomedical application data is high 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 formulated in the following:

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

where $w \in \mathbb{R}^d$ is the model, $f_i(w) : \mathbb{R}^d \to \mathbb{R}$ is the local loss function, and $\Psi : \mathbb{R}^d \to \mathbb{R}$ is the nonsmooth convex regularizer. When $f_i(w) = \frac{1}{n_i} \sum_{(x_i, y_i) \in \mathcal{D}_i} \|w^T x_i - y_i\|^2$, 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 [161]. Specifically, we introduce the Bregman distance. Let $h : \mathbb{R}^d \to \mathbb{R}$ 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 $g^*(z) = \sup_{w \in \mathbb{R}^d} \{\langle z, w \rangle - g(w)\}$. 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

Low rank matrix estimation (LRME) is an efficient technique in dealing with large-scale problems with high-dimensional data and thus rising the significant interests in scientific computing. Exemplar applications include the sparse principle component analysis [113], data compression for natural language processing [26] and distance matrix completion [108]. 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 algoritm ADMM is a convenient and efficient tool for solving nonsmooth optimization problems [17]. For the adaptation to FedOpt, we first formulated the federated LRME problem. Assume there are N distributed clients, and denote $D_j \in \mathbb{R}^{d \times d}$ as the data sample. Then the local loss function of LRME on client i is defined as $f_i(X) = \sum_{j=1}^{n_i} (\langle X, D_j \rangle - y_j)^2$, where y_j is the noisy observation. All clients target at solving the following problem via L-FedADMM:

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

Here, we use the nuclear norm $\|\cdot\|_*$ to constrain the low rank of the matrix. The problem (2) targets at recovering the low rank matrix $X \in \mathbb{R}^{d \times d}$ from the noisy observations y_j. For L-FedADMM, the

---

Algorithm 1 L-FedAdmm

1: **server input:** initial Z^0, η_g.
2: **client i's input:** initial X_i^0, π_i^0 and η_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 π_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**

---

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. Then, the augmented Lagrangian function can be further derived [17]:

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

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 [17] 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 $f_i(X_i^+) \approx f_i(X_i) + \langle \nabla f_i(X_i), X_i^+ - X_i \rangle + \frac{1}{2\eta_l} \|X_i^+ - X_i\|_2^2$, 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 for the nuclear norm iteratively is employed:

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

for $i = 0, \ldots, I - 1$, then we can obtain $Z^+ \leftarrow Z^M$. Next, we mimic the adaptation of SGD to FedAvg for L-FedADMM [104]. Precisely, the participated clients $i \in \mathcal{S}$ performs the local updates for
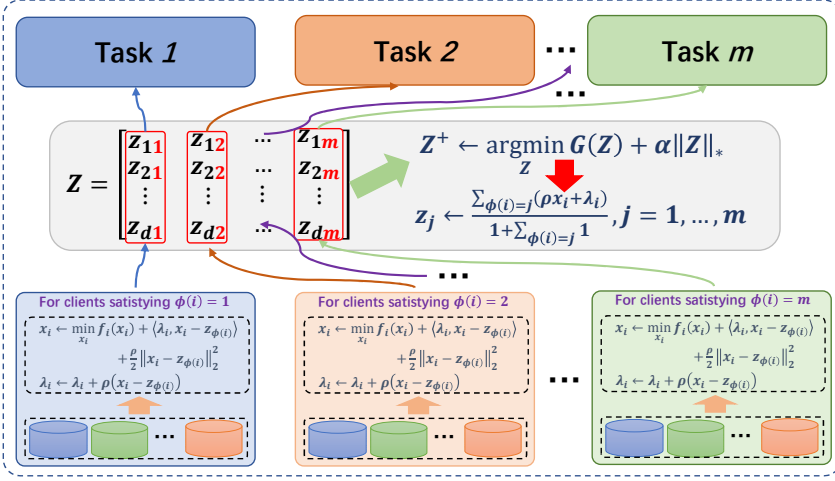
Fig. 4. Multi-task learning via FL with ADMM strategy, the basic idea is each task corresponding to multiple clients, namely, for task j, $j = 1 \ldots, m$, its corresponding clients satisfy $\psi(i) = j$, where $i = 1, \ldots, N$.

X_i (via (5)) and π_i for multiple times (say T local iterations), and then the quantity $\pi_i^+ + \rho X_i^+$ is transmitted to the server for performing the global update (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 Multi-Task Learning

In real-world applications, there is an urgent need solving the problem of multiple learning tasks simultaneously, while exploiting commonalities and differences across tasks. 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 multi-task learning (MTL), where some clients have the shared task, while other clients have different ones. Since MTL learns separate models for each task among the clients, FL is naturally suited to MTL [131]. 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 $\psi(i) \in \{1, \ldots, m\}$. 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 to learn the joint model $Z = \{z_1, \ldots, z_m\}$ 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 [131]. 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.

Decentralized 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 $\{x_i, i = 1, \ldots, N\}$. Subsequently, we formulate the federated consensus optimization problem over the N clients as

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

Then, we will use the distributed ADMM discussed in Section 4 to solve the above federated problem. For the local model x_i update, it can be obtained via standard subproblem minimization in ADMM:

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

where λ_i is the Lagrangian multiplier. In addition to local model update, the client also needs to update λ_i: $\lambda_i \leftarrow \lambda_i + \rho(x_i - z_{\phi(i)})$. 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:

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

  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 $Z^+ \leftarrow U \cdot \operatorname{diag}\{\operatorname{prox}_{\frac{\alpha}{\rho}\|\cdot\|_1}(\sigma)\} \cdot V^T$, 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:

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

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

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

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., $\nabla_{z_j} G(Z)$, $j = 1, \cdots, m$ 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 iteration $i = 0, I - 1$

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

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:

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

Algorithm 2 Decentralized ADMM for MTL

1: **server input:** The communication round R and iteration number I, initial Z, α and ρ.
2: **client i's input:** The local iteration number T, initial η, x_i, and $y_{i,j}$.
3: **for** $r = 1, \ldots, R$ **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 λ_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**

3.5 Matrix Factorization

Matrix factorization (MF) is a prominent machine learning technique which has widely applied in various domains including recommendation system [70, 75] and environment monitoring [143]. While MF has been developed with security guarantee in centralized scenarios [70, 115], its application in FL scenarios is still under investigation. Chai et al. [20] 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:

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

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 [20] to make the our approach exploit well in the item dependency and more robust
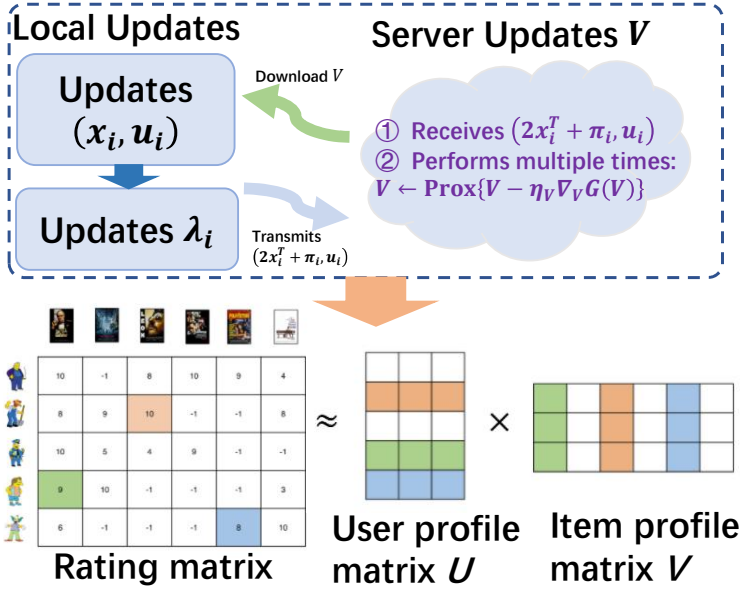
Fig. 5. Federated matrix factorization. Its main ingredient is: first, each user holds there rating vector $R_{i,:}$, moreover, it initializes a user profile vector u_i, $(R_{i,:}, u_i)$ are kept private. The item matrix V can be shared through users. They collaborate with each other to learn (U, V).

to the data error [30]. For the FedOpt solution, we propose a three-blocked distributed ADMM based framework. First of all, we incorporate a concensus 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):

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

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

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

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

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

where $\eta_V > 0$ is the step size. After the server finished 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:

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

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: $\pi_i \leftarrow \pi_i + \rho(x_i^T - V \cdot u_i^T)$.

As is shown, the above ADMM procedure can conveniently solve the MF problem in recommendation system under the federated settings, it should be noted that the steps 2) and 3) can be carried out in parallel on each client.

Algorithm 3 3-Block Decentralized ADMM for MF

1: **server input:** The communication round R and iteration number I, initial U and V, α and ρ.
2: **user i's input:** The local iteration number T, initial η, 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 $i = 0, \cdots, I - 1$.
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 π_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**

---

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 [72, 73, 104, 155]. 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 participated 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 [73, 85, 87, 104].

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

(a) Local acceleration scheme.
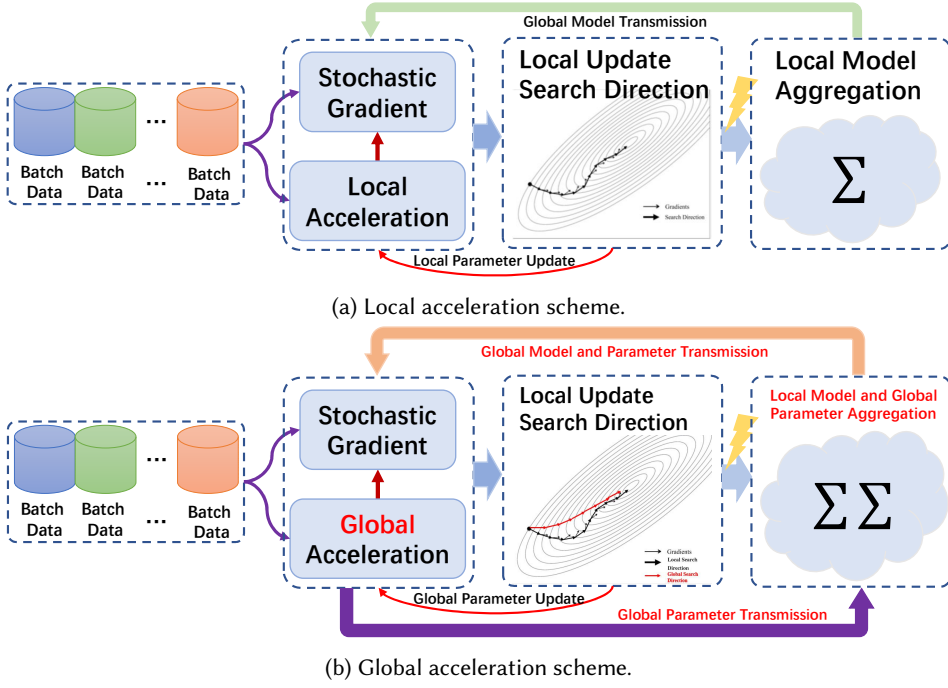

(b) Global acceleration scheme.

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.

choices. **We extensively review these works and find out their basic paradigms of 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 $f : \mathbb{R}^d \to \mathbb{R}$, $f_i : \mathbb{R}^d \to \mathbb{R}$.

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 (e.g., gradients and variables, etc..) to the central server, which subsequently performs aggregation [6, 102, 103, 173]. 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 [164]).

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. We also provide the instantiation of these two scenarios in the supplementary material.

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 etc., they are also employed in the distributed SGD to speed up the convergence. Specifically, mimicking the adaptation of SGD to FedAvg, the momentum gradient descent method which is a variant of SGD has been extended to the local update step, then the global performs aggregating all the local models and the momentum terms, where the resultant algorithm is called MFL [98]. Similarly, FedAdam [120] extend Adam [71] to FedOpt. However, these methods still suffer from the model divergence problem [87, 145, 160, 166]. This can be due to the reason of the multiple local update steps in FedOpt which will make the local model fast approach to the local optimum of the local loss function instead of the global one [67]. Moreover, since only a small portion of the clients participate in the update on each communication round, the aggregated model will be further biased from the optimal global one. 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.

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**

Global acceleration.   For improving the performance, the control variate for variance reduction strategy has been adopted, i.e., VRL-SGD, which has shown faster speed even with non-IID data [89]. However, VRL-SGD does not support the client sampling which is more practical in FL. Furthermore, with the control variate to include as more information from clients as possible, SCAFFOLD has shown the significant performance improvement for the data heterogeneity problem in FedOpt [67]. Its core notion is to estimate the full gradient for the local search direction. Most recently, a framework called Mime is proposed [66], which adapts popular centralized algorithms (e.g., SGD, Adam etc.) to FedOpt.

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

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**

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. In [121], a quantized version of FedAvg, known as FedPaq, is proposed for reducing the message overload. Another strategy, called QSGD [8], 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. Jeremy et al. [13] proposes 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 [169], which has demonstrated numerically reducing the communication by about $32\times$ 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. [9] and Stich et al. [132] study sparsifying gradients by magnitude with local error correction and selecting top K components. Another algorithm only transmits the large entries to the server [41], which shows the significant reduction in the communication overhead

[7]. While the sparsification can bring performance degradation, deep gradient compression [92] 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 to 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 [60]. 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 methods 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. [147] proposes 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 $\mathbb{E} \sum_{j=1}^{d} [Q(g_i^j)^2] = \sum_{j=1}^{d} (g_i^j)^2/p_j$. In addition, the expected sparsity $\mathbb{E} \|Q(g_i)\|_0$ can be computed by $\mathbb{E} \|Q(g_i)\|_0 = \sum_{j=1}^{d} p_j$. 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: $\min_p \sum_{j=1}^{d} p_j$, s.t. $\sum_{j=1}^{d} (g_i^j)^2/p_j \leq \epsilon$.

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 [107]. Another strategy ADIANA combines the acceleration/momentum and compression strategies [88].

4.1.3 Second-order Scenarios. The second-order scheme has attracted attentions 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 [16].

DANE minimizes the linear approximation of the objective function with Bregmen divergence measurement [126]. 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. [144] proposes a distributed Newton-type optimization method GIANT. Technically, GIANT realizes the local approximated Newton directions by the conjugate gradient descent method (CG), which involves only Hessian-vector products in CG iterations. In [28], 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. [123] proposes 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. [5] proposes the similar strategy FLECS-CGD, which also compresses the gradient information. However, they both suffer from high computation complexity, making them unpractical in mobile applications with low power

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	✓

Table 4. Representative federated optimization methods based on first-order gradient. Specifically, M.D. is acronym for model divergence, synchony and asynchrony are abbreviated as SC and AS respectively, the abbreviation of partial client participation is PCP.

consumption. To mitigate this issue, Ma et al. [101] proposes 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) [17], 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 [59, 117], which penalizes the constraint equality with a quadratic term, making it advantageous in numerical stability. 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 attracted a considerable amount of attention in both practical and theoretical aspects [17, 138]. With simple and straightforward modification, ADMM can be adapted to FedOpt.

For algorithmic development, Goldstein et al. [48] has incorporated Nesterov's acceleration strategy [110] 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 FedOpt. Another strategy is based on Anderson acceleration [163], it targets at speeding up the convergence of a fixed-point iteration [141] via the quasi-Newton method by adopting previous iterates by approximating its inverse Jacobian [36, 38]. 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 [129]. For further improving the computational efficiency in DADMM, Ling et al. [93] has proposed a linearized version of DADMM (DLM). Both

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 5. Representative federated optimization methods based on second-order gradient.

DADMM and DLM have been demonstrated the linear convergence rate under the assumption of strong convexity in the objective function. For supporting ADMM in FL, Zhou et al. [172] proposes CEADMM approach, its key idea is to modify the classical ADMM by update the local model and Lagrangian multiplier multiple times before transmitting to the server.

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	×

Table 6. Representative federated optimization methods based on ADMM.

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 Fig. 7. In general, privacy and security in FL can be divided into two categories, i.e., data privacy and model security. 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 and developing privacy-preserving mechanisms to prevent privacy leakage. On the other hand, in terms of the security issues, existing studies mainly focus on modeling the treat model of influencing the performance of the model learned based on FL against malicious participants. We will introduce each problem in terms of their corresponding attack models and defence methods, respectively.

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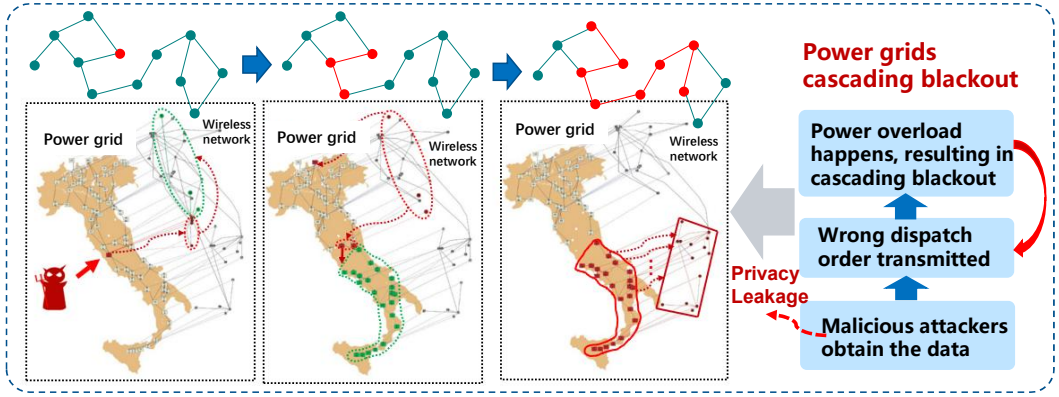
Fig. 7. Cascading blackout resulted by information leakage, directly impacting national security.

- **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 including 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 [44, 109]. Nasr et al. [109] 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. [44] 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 sill necessary, e.g., the provable differential privacy.

5.1 Data Privacy

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

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 keep 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 $\mathcal{M} : \mathcal{D} \rightarrow O$ 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 $Pr(\mathcal{M}(D_1) \in O) \leq e^\epsilon Pr(\mathcal{M}(D_2) \in O) + \delta$.*

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

Ref.	Privacy/Security	FL Tasks	Categories	Idea
[44]	Privacy	Information disclosure	Attack Method	Magnitude-invariant loss.
[124]	Privacy	Data Summarization	Defence Method	Random Hash
[61]	Privacy	Model optimization	Defence Method	SMC, Add perturbation
[31]	Privacy	Federated contextual bandit	Defence Method	Upper confidence bounds (UCB), differential private perturbations
[134]	Privacy	Model optimization	Defence Method	Adversarial reconstruction, noise regularization, distance correlation minimization
[14]	Security	Model optimization	Attack Method	Optimization based on indicators of stealth
[15]	Security	Model optimization	Defence Method	Distance-based aggregation rule
[158]	Security	Model optimization	Defence Method	Media and trimmed mean operations of the gradient
[81]	Security	Multi-task learning	Defence Method	Accumulated loss
[47]	Security	Second order optimization	Defence Method	Newton's method

Table 7. Privacy and security in FL.

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, homomorphic encryption is a technique that allows us to directly implement computing on the ciphertext and guarantee the correctness of the obtain results after decryption. On the other hand, security multi-party computation 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. [124] 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 [122] and MWEM method [52]. Jayaraman et al. [61] 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 secure multi-party computation (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. [31] 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, 84] and differential private perturbations [127]. Sun et al. [134] focus on defending the privacy leakage in the vertical FL 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 [14]. 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 defence method against this attack methods.

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 attacks 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. [14] focus on the targeted poisoning attack. Specifically, they propose two indicators of stealth to detect Byzantine participants, and their further incorporate these two indicators in the optimization object of the adversary to derive a stealthy poisoning strategy. Their experiments should that Byzantine-resilient aggregation strategies [15, 158], which are designed for defending the untargeted performance attack, are not robust to the proposed attacks.

Defence Method. The defence 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. [15] 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 [15] further propose a 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. [158] propose two distributed gradient descent algorithms based on media and trimmed mean operations, and they futher 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. [81] focus on federated multi-task learning 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. [47] propose a federated optimization algorithm based on Newton's method, which is communication-efficient as well as robust against Byzantine failures.

6 APPLICATION

As we have seen, the FL applications have witnessed the 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

Recommendation. Recommendation techniques have been widely used to model user interests and mitigate information overload problems in many real applications[23, 130, 167]. For further improving privacy and security, FL has been introduced in recommendation [136]. HegedHus et al. [58] 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. [149] 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 transimitted embedding vectors based on local differential privacy. In addition, high-order user-item interactions are modeled by matching neighboring users based on the homomorphic encryption technique. Qi et al. [118] 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. [96] focus on utilizing the FL

techniques to solve collaborative transfer learning problem between different service providers. 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. [10] propose a federated collaborative filtering algorithm for privacy-preserving recommendation. Flanagan et al. [40] propose a federated multi-view matrix factorization method to avoid collecting raw user data centralizedly by collaborating between the FL server, item server, and clients. Gao et al. [43] design a privacy-preserving cross-domain location recommendation by proposing a Laplace perturbation mechanism using both semantics and location information. Gao et al. [42] propose a differential private local CF algorithm based on the asymmetric flipping perturbation mechanism.

Natural Language Processing.   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 [19]. Chen et al. [22] 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 real world, particularly in climate and environmental problems, we often have measurements at a number of locations in the specific time, and will often need to make predictions of the output at unmeasured locations in the next several time steps [54]. We can approach the spatial problem by constructing spatio-temporal modeling via FL. Meng et al. [105] 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. Feng et al. [39] 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.   Knowledge graphs (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. For further enhance decentralized privacy-preserved applications of KGs, FL has been incorporated [24]. Chen et al. [25] propose a FL algorithm to complete multiple knowledge graph without sharing their private triples, where they propose an optimization-based model fusion procedure at clients. Peng et al. [114] propose a differential private federated algorithm to learn the knowledge graph embeddding, which utilizes a module named privacy-preserving adversarial translation (PPAT) to project embeddings of aligned entities and relations of different knowledge graph to capture their relationship. Specifically, this PPAT module is designed based on the technique of PATE-GAN [65] 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 [69]. Liu et al. [94] 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. [18] 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 sovle this problem. Yang et al. [156] propose a FL algorithm automatic diagnosis of COVID-19 and protect patient privacy simultaneous. 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. [130] 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 [32, 116]. Liu et al. [100] mitigates the privacy and security issues of the existing data gathering fashion by combining FL with gated recurrent unit neural network (FedGRU) for traffic flow prediction. For further mining the spatio-temporal information for traffic flow prediction, Yuan et al. [162] first proposes 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. [106] 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

Ref.	Applications	Idea
Comparison [58]	Recommendation	Comparison between FL and gossip learning.
CNFGNN [105]	Traffic flow prediction	Modeling spatial dynamics based on federated graph neural network
FedGNN [149]	Recommendation	Generating pseudo interacted items, FE-based neighboring users matching
FedNewsRec [118]	Recommendation	Gradient clipping, adding Laplace noise
FedCT [96]	Recommendation	Cross-domain transfer learning, intermediate transfer agents on user devices.
FED-MNMF [40]	Recommendation	Collaborating between the FL server, item server, and clients
FCF [10]	Recommendation	Federated collaborative filtering.
FLOP [156]	Automatic diagnosis	Sharing a partial model between the server and clients.
FedMatch [22]	Question answer	Partition the model into the shared backbone and the private patch.
FedE [25]	Knowledge graph completion	Optimization-based model fusion procedure at clients.
FKGE [114]	Knowledge graph embedding	PATE-GAN based translation module between different knowledge graphs.
Experiments [94]	COVID-19 detection	Compare different models.
SegCaps [77]	COVID-19 detection	Data normalization, capsule network, blockchain.
cPDS [18]	Hospitalization prediction	SVM, iterative cluster primal dual splitting algorithm.
fPCA [130]	Brain structure analysis	ADMM, data standardization, confounding factors correction, and multivariate analysis.
CCMF [43]	Recommendation	Laplace perturbation mechanism using both semantics and location information.
DPLCF [42]	Recommendation	Asymmetric flipping perturbation mechanism.

Table 8. List of FL applications.

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 summarize as follows:

- **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 [56, 91]. 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 fused 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 on 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 handing this problem, both with the main idea of estimating the global gradient [66, 67]. 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 communicates 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 start to study decentralized FL in theoretical aspects [74, 142, 159]. 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.

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 *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] 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 *NeurIPS 2022*.

[6] Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. 2014. A Reliable Effective Terascale Linear Learning System. *J. Mach. Learn. Res.* 15, 1 (jan 2014), 1111–1133.

[7] Alham Fikri Aji and Kenneth Heafield. 2017. Sparse Communication for Distributed Gradient Descent. *CoRR* abs/1704.05021 (2017).

[8] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *Advances in Neural Information Processing Systems 30, December 4-9, 2017, Long Beach, CA, USA*. 1709–1720.

[9] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cedric Renggli. 2018. The Convergence of Sparsified Gradient Methods. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.

[10] 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).

[11] Corey W. Arnold and William Speier. 2012. A topic model of clinical reports. In *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.

[12] Ozan Aygün, Mohammad Kazemi, Deniz Gündüz, and Tolga M. Duman. 2022. Hierarchical Over-the-Air Federated Edge Learning. In *ICC 2022 - IEEE International Conference on Communications*. 3376–3381.

[13] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. [n. d.]. 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*, Vol. 80. 559–568.

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

[15] 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.

[16] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. 2018. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* 60, 2 (2018), 223–311.

[17] 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. *Found. Trends Mach. Learn.* 3, 1 (2011), 1–122.

[18] 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.

[19] Dongqi Cai, Shangguang Wang, Yaozong Wu, Felix Xiaozhu Lin, and Mengwei Xu. 2022. Federated NLP in Few-shot Scenarios. *CoRR abs/2212.05974 (2022)*. arXiv:2212.05974

[20] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2021. Secure Federated Matrix Factorization. *IEEE Intelligent Systems* 36, 5 (2021), 11–20.

[21] 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.

[22] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2021. FedMatch: Federated Learning Over Heterogeneous Question Answering Data. *arXiv preprint arXiv:2108.05069* (2021).

[23] Lei Chen, Jie Cao, Weichao Liang, Jia Wu, and Qiaolin Ye. 2022. Keywords-Enhanced Deep Reinforcement Learning Model for Travel Recommendation. *ACM Trans. Web* 17, 1, Article 5 (dec 2022), 21 pages.

[24] 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 Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 1966–1972.

[25] Mingyang Chen, Wen Zhang, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2020. Fede: Embedding knowledge graphs in federated setting. *arXiv preprint arXiv:2010.12882* (2020).

[26] Patrick Chen, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2021. DRONE: Data-aware Low-rank Compression for Large NLP Models. In *Advances in Neural Information Processing Systems*, Vol. 34. 29321–29334.

[27] CHINA. 2017. THE CYBERSECURITY LAW OF THE PEOPLE'S REPUBLIC OF CHINA. (2017).

[28] Rixon Crane and Fred Roosta. 2019. DINGO: Distributed Newton-Type Method for Gradient-Norm Optimization. In *Advances in Neural Information Processing Systems*, Vol. 32.

[29] Floriano De Rango, Antonio Guerrieri, Pierfrancesco Raimondo, and Giandomenico Spezzano. 2021. A Novel Edge-based Multi-Layer Hierarchical Architecture for Federated Learning. In *2021 IEEE DASC/PiCom/CBDCom/CyberSciTech*. 221–225.

[30] 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* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. New York, NY, USA, 281–288.

[31] Abhimanyu Dubey and AlexSandy' Pentland. 2020. Differentially-Private Federated Linear Bandits. *Advances in Neural Information Processing Systems* 33 (2020).

[32] Ahmet M. Elbir, Burak Soner, Sinem Coleri, Deniz Gündüz, and Mehdi Bennis. 2022. Federated Learning in Vehicular Networks. In *IEEE International Mediterranean Conference on Communications and Networking, MeditCom 2022, Athens, Greece, September 5-8, 2022*. IEEE, 72–77.

[33] Jianyu Wang et al. 2021. A Field Guide to Federated Optimization. *CoRR* abs/2107.06917 (2021).

[34] Peter Kairouz et al. 2019. Advances and Open Problems in Federated Learning. *CoRR* abs/1912.04977 (2019). arXiv:1912.04977

[35] 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). *Available at: https://eur-lex. europa. eu/legal-content/EN/TXT* (2016).

[36] V. Eyert. 1996. A Comparative Study on Methods for Convergence Acceleration of Iterative Vector Sequences. *J. Comput. Phys.* 124, 2 (1996), 271–285.

[37] 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.

[38] 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.

[39] 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).

[40] 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.

[41] 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 *15th Annual Conference of the International Speech Communication Association*.

[42] 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.

[43] 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 (IMWUT)* 3, 1 (2019), 1–21.

[44] 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* (2020).

[45] Anousheh Gholami, Nariman Torkzaban, and John S. Baras. 2022. Trusted Decentralized Federated Learning. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. 1–6.

[46] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An Efficient Framework for Clustered Federated Learning. In *Advances in Neural Information Processing Systems*, Vol. 33. 19586–19597.

[47] Avishek Ghosh, Raj Kumar Maity, and Arya Mazumdar. 2020. Distributed Newton Can Communicate Less and Resist Byzantine Workers. In *NeurIPS*.

[48] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard G. Baraniuk. 2014. Fast Alternating Direction Optimization Methods. *SIAM J. Imaging Sci.* 7, 3 (2014), 1588–1623.

[49] 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.

[50] Yan Gou, Ruiyu Wang, Zongyao Li, Muhammad Ali Imran, and Lei Zhang. 2022. Clustered Hierarchical Distributed Federated Learning. In *ICC 2022-IEEE International Conference on Communications*. 177–182.

[51] A. Gouissem, K. Abualsaud, E. Yaacoub, T. Khattab, and M. Guizani. 2022. Robust Decentralized Federated Learning Using Collaborative Decisions. In *2022 International Wireless Communications and Mobile Computing (IWCMC)*. 254–258.

[52] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A Simple and Practical Algorithm for Differentially Private Data Release. In *NIPS*.

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

[54] Toshihiro et al Hasegawa. 2022. A global dataset for the projected impacts of climate change on four major crops. *Scientific data* 9, 1 (2022), 58.

[55] 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.

[56] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. In *Advances in Neural Information Processing Systems*.

[57] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. 2022. Byzantine-Robust Decentralized Learning via Self-Centered Clipping. *CoRR* abs/2202.01545 (2022).

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

[59] M.R Hestenes. 1969. Multiplier and gradient methods. *J. Optim. Theory Appl.* 4 (1969), 303–320.

[60] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir braverman, Ion Stoica, and Raman Arora. 2019. Communication-efficient Distributed SGD with Sketching. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[61] 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.

[62] Di Jiang, Yuanfeng Song, Yongxin Tong, Xueyang Wu, Weiwei Zhao, Qian Xu, and Qiang Yang. 2019. Federated Topic Modeling. In *CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 1071–1080.

[63] 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 Trans. Intell. Syst. Technol.* 12, 1 (2021), 2:1–2:22.

[64] 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.

[65] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*.

[66] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2020. Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning. arXiv:2008.03606 [cs.LG]

[67] 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 *ICML*.

[68] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2020. Blockchained On-Device Federated Learning. *IEEE Communications Letters* 24, 6 (2020), 1279–1283.

[69] Junu Kim, Kyunghoon Hur, Seongjun Yang, and Edward Choi. 2022. Universal EHR Federated Learning Framework. *CoRR* abs/2211.07300 (2022). arXiv:2211.07300

[70] 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 Trans. Priv. Secur.* 21, 4 (2018), 17:1–17:30.

[71] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[72] 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

[73] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR* abs/1610.05492 (2016).

[74] 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

[75] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[76] Caner Korkmaz, Halil Eralp Kocas, Ahmet Uysal, Ahmed Masry, Oznur Ozkasap, and Baris Akgun. 2020. Chain FL: Decentralized Federated Machine Learning via Blockchain. In *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. 140–146.

[77] 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).

[78] Hanjiang Lai, Yan Pan, Cong Liu, Liang Lin, and Jie Wu. 2013. Sparse Learning-to-Rank via an Efficient Primal-Dual Algorithm. *IEEE Trans. Comput.* 62, 6 (2013), 1221–1233.

[79] 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.

[80] Chengxi Li, Gang Li, and Pramod K. Varshney. 2022. Federated Learning With Soft Clustering. *IEEE Internet of Things Journal* 9, 10 (2022), 7773–7782.

[81] J. Li, W. Abbas, and X. Koutsoukos. 2020. Byzantine Resilient Distributed Multi-Task Learning. *Advances in Neural Information Processing Systems* (2020).

[82] 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 Trans. Parallel Distributed Syst.* 33, 10 (2022), 2401–2415.

[83] 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 *BMC Bioinformatics volume*, Vol. 182.

[84] 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.

[85] Tian Li, Anit Kumar Sahu, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. *Proceedings of the 3rd MLSys Conference, Austin, TX, USA* (2020).

[86] 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.

[87] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In *ICLR*.

[88] 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*, Vol. 119. PMLR, 5895–5904.

[89] 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).

[90] 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.

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

[92] 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).

[93] Qing Ling, Wei Shi, Gang Wu, and Alejandro Ribeiro. 2015. DLM: Decentralized Linearized Alternating Direction Method of Multipliers. *IEEE Trans. Signal Process.* 63, 15 (2015), 4051–4064.

[94] Boyi Liu, Bingjie Yan, Yize Zhou, Yifan Yang, and Yixian Zhang. 2020. Experiments of federated learning for covid-19 chest x-ray images. *arXiv preprint arXiv:2007.05592* (2020).

[95] Lumin Liu, Jun Zhang, S.H. Song, and Khaled B. Letaief. 2020. Client-Edge-Cloud Hierarchical Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 1–6.

[96] Shuchang Liu, Shuyuan Xu, Wenhui Yu, Zuohui Fu, Yongfeng Zhang, and Amelie Marian. 2021. FedCT: Federated Collaborative Transfer for Recommendation. In *SIGIR 21*. 716–725.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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 arXiv:2206.09576

[102] Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed Training Strategies for the Structured Perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Los Angeles, California) *(HLT '10)*. USA, 456–464.

[103] Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon Mann. 2009. Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models. In *Advances in Neural Information Processing Systems*, Vol. 22.

[104] 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*.

[105] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. 2021. Cross-Node Federated Graph Neural Network for Spatio-Temporal Data Modeling. *arXiv preprint arXiv:2106.05223* (2021).

[106] 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* (Virtual Event, Singapore) *(KDD '21)*. 1202–1211.

[107] Konstantin Mishchenko, Eduard A. Gorbunov, Martin Takác, and Peter Richtárik. 2019. Distributed Learning with Compressed Gradient Differences. *CoRR* abs/1901.09269 (2019).

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

[109] 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 *2019 IEEE symposium on security and privacy (SP)*. 739–753.

[110] 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.

[111] 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.

[112] U.S. SELECT COMMITTEE ON ARTIFICIAL INTELLIGENCE of the NATIONAL SCIENCE & TECHNOLOGY COUNCIL. 2019. THE NATIONAL ARTIFICIAL INTELLIGENCE RESEARCH AND DEVELOPMENT STRATEGIC PLAN: 2019 UPDATE. *Available at: https://www.nitrd.gov/pubs/National-AI-RD-Strategy-2019.pdf* (2019).

[113] Dimitris Papailiopoulos, Alexandros Dimakis, and Stavros Korokythakis. 2013. Sparse PCA through Low-rank Approximations. In *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28. PMLR, Atlanta, Georgia, USA, 747–755.

[114] H. Peng, H. Li, Y. Song, V Zheng, and J. Li. 2021. Differentially Private Federated Knowledge Graphs Embedding. *Proc. ACM CIKM* (2021).

[115] 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.

[116] 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.

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

[118] 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.

[119] 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.

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

[121] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization *(Proceedings of Machine Learning Research, Vol. 108)*. PMLR, Online, 2021–2031.

[122] Benjamin IP 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.

[123] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. 2022. FedNL: Making Newton-Type Methods Applicable to Federated Learning. In *International Conference on Machine Learning, Baltimore, Maryland, USA*, Vol. 162. 18959–19010.

[124] 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.

[125] 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.

[126] 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*, Vol. 32. PMLR, Bejing, China, 1000–1008.

[127] Roshan Shariff and Or Sheffet. 2018. Differentially private contextual linear bandits. *arXiv preprint arXiv:1810.00068* (2018).

[128] Lu Shi, Jiangang Shu, Weizhe Zhang, and Yang Liu. 2021. HFL-DP: Hierarchical Federated Learning with Differential Privacy. In *2021 IEEE Global Communications Conference (GLOBECOM)*. 1–7.

[129] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. 2014. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Trans. Signal Process.* 62, 7 (2014), 1750–1761.

[130] 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 *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE, 270–274.

[131] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S. Talwalkar. 2017. Federated Multi-Task Learning. In *Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA*. 4424–4434.

[132] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. Sparsified SGD with Memory. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.

[133] 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.

[134] Jiankai Sun, Yuanshun Yao, Weihao Gao, Junyuan Xie, and Chong Wang. 2021. Defending against Reconstruction Attack in Vertical Federated Learning. *CoRR* abs/2107.09898 (2021).

[135] 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 *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. 1898–1903.

[136] Zehua Sun, Yonghui Xu, Yong Liu, Wei He, Yali Jiang, Fangzhao Wu, and Lizhen Cui. 2023. A Survey on Federated Recommendation Systems. *CoRR* abs/2301.00767 (2023). arXiv:2301.00767

[137] 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.

[138] 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*, Vol. 48. 2722–2731.

[139] 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 *2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. 1–1.

[140] Omar Abdel Wahab, Azzam Mourad, Hadi Otrok, and Tarik Taleb. 2021. Federated Machine Learning: Survey, Multi-Level Classification, Desirable Criteria and Future Directions in Communication and Networking Systems. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1342–1397.

[141] Homer F. Walker and Peng Ni. 2011. Anderson Acceleration for Fixed-Point Iterations. *SIAM J. Numer. Anal.* 49, 4 (2011), 1715–1735.

[142] Huiyuan Wang, Xuyang Zhao, and Wei Lin. 2022. Heterogeneous Federated Learning on a Graph. *CoRR* abs/2209.08737 (2022).

[143] 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.

[144] Shusen Wang, Fred Roosta, Peng Xu, and Michael W Mahoney. 2018. GIANT: Globally Improved Approximate Newton Method for Distributed Optimization. In *Advances in Neural Information Processing Systems*, Vol. 31.

[145] 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.

[146] Xing Wang and Yijun Wang. 2022. Asynchronous Hierarchical Federated Learning. *CoRR* abs/2206.00054 (2022).

[147] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient Sparsification for Communication-Efficient Distributed Optimization. In *Advances in Neural Information Processing Systems*, Vol. 31.

[148] 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.

[149] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie. 2021. FedGNN: Federated Graph Neural Network for Privacy-Preserving Recommendation. *Proc. ACM KDD* (2021).

[150] 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).

[151] 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.

[152] 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.

[153] 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.

[154] 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.

[155] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12 (March 2019), 19 pages.

[156] Qian Yang, Jianyi Zhang, Weituo Hao, Gregory Spell, and Lawrence Carin. 2021. FLOP: Federated Learning on Medical Datasets using Partial Networks. *Proc. KDD* abs/2102.05218 (2021).

[157] 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.

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

[159] Guangsheng Yu, Xu Wang, Caijun Sun, Qin Wang, Ping Yu, Wei Ni, Ren Ping Liu, and Xiwei Xu. 2023. IronForge: An Open, Secure, Fair, Decentralized Federated Learning. *CoRR* abs/2301.04006 (2023). arXiv:2301.04006

[160] 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 *AAAI*.

[161] 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.

[162] 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.

[163] Juyong Zhang, Yue Peng, Wenqing Ouyang, and Bailin Deng. 2019. Accelerating ADMM for Efficient Simulation and Optimization. *ACM Trans. Graph.* 38, 6, Article 163 (nov 2019), 21 pages.

[164] Ruiliang Zhang and James Kwok. 2014. Asynchronous Distributed ADMM for Consensus Optimization. In *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32. PMLR, Bejing, China, 1701–1709.

[165] 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.

[166] 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).

[167] 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* (Lyon, France). Republic and Canton of Geneva, CHE, 167–176.

[168] 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.

[169] Shuai Zheng, Ziyue Huang, and James Kwok. 2019. Communication-Efficient Distributed Blockwise Momentum SGD with Error-Feedback. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[170] Xiangyu Zhong, Xiaojun Yuan, Huiyuan Yang, and Chenxi Zhong. 2022. UAV-Assisted Hierarchical Aggregation for Over-the-Air Federated Learning. In *IEEE Global Communications Conference, GLOBECOM 2022, Rio de Janeiro, Brazil, December 4-8, 2022*. IEEE, 807–812.

[171] 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 Trans. Intell. Syst. Technol.* 13, 5, Article 71 (oct 2022), 24 pages.

[172] Shenglong Zhou and Geoffrey Ye Li. 2021. Communication-Efficient ADMM-based Federated Learning. *CoRR* abs/2110.15318 (2021).

[173] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex Smola. 2010. Parallelized Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems*, Vol. 23.