Front. Comput. Sci., 2024, 18(1): 181336
https://doi.org/10.1007/s11704-023-3282-7

REVIEW ARTICLE

A survey on federated learning: a perspective from multi-party computation

Fengxia LIU[1,2,3], Zhiming ZHENG[1,2,3,4], Yexuan SHI (✉)[2], Yongxin TONG[2], Yi ZHANG[5]

1 Institute of Artificial Intelligence and Key Laboratory of Mathematics Informatics Behavioral Semantics, Beihang University, Beijing 100191, China

2 State Key Laboratory of Software Development Environment and Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China

3 Pengcheng Laboratory, Shenzhen 518055, China

4 Zhongguancun Laboratory, Beijing 100190, China

5 Institute for Mathematical Sciences and Engineering Research Center of Financial Computing and Digital Engineering, Renmin University of China, Beijing 100872, China

Abstract Federated learning is a promising learning paradigm that allows collaborative training of models across multiple data owners without sharing their raw datasets. To enhance privacy in federated learning, multi-party computation can be leveraged for secure communication and computation during model training. This survey provides a comprehensive review on how to integrate mainstream multi-party computation techniques into diverse federated learning setups for guaranteed privacy, as well as the corresponding optimization techniques to improve model accuracy and training efficiency. We also pinpoint future directions to deploy federated learning to a wider range of applications.

Keywords federated learning, multi-party computation, privacy-preserving data mining, distributed learning

1 Introduction

Federated learning (FL) has emerged as a popular machine learning paradigm which allows multiple data owners to train models collaboratively without sharing their raw datasets [1–4]. It holds potential for a wide spectrum of analytics applications on sensitive data. For example, federated learning has been applied on medical big data analysis such as disease prediction and diagnosis without revealing the patients' private medical information to third-party services [5]. It has also been exploited by banks and insurance companies to train an accurate machine learning model for risk assessment or customer recommendation [6,7].

Federated learning enables collaborative model training without sharing raw datasets among data owners by decomposing the training procedure into local training and model aggregation. Each data owner performs local training on its own data partition and only communicates intermediate results e.g., gradients for model aggregation at either a centralized server or other data owners. Federated learning with a central server to coordinate the model aggregation is called centralized FL [8,9], while model aggregation in a peer-to-peer manner is known as decentralized FL [2,10]. Centralized FL imposes high computation workload to the server, whereas decentralized FL involves excessive communication among peers. Consequently, semi-centralized FL [11–13] is recently proposed to balance the computation and communication cost by conducting clustered or hierarchical model aggregation.

We focus on federated learning with privacy guarantees. Note that exchanging intermediate results e.g., gradients rather than raw datasets may still leak privacy [14,15]. Accordingly, extra techniques are compulsory for secure communication and computation during federated learning. Of our particular interest is multi-party computation, a generic and fundamental category of techniques that takes multi-party private inputs for aggregated computation without revealing the private data of each party [16–18]. Common multi-party computation techniques include garbled circuit, secret sharing, homomorphic encryption, differential privacy, and so on. Recent years have witnessed a surge to enhance the privacy of federated learning via multi-party computation [19–25].

This survey aims at a comprehensive overview on federated learning with privacy guarantees in the lens of multi-party computation. We review which multi-party computation schemes are suited for privacy protection in centralized, decentralized, and semi-centralized federated learning. We also discuss how to improve the accuracy and efficiency of federated learning when adopting diverse multi-party computation techniques.

The rest of this survey is organized as follows. We first introduce the basics and taxonomy of federated learning and

Received April 9, 2023; accepted July 23, 2023

E-mail: skyxuan@buaa.edu.cn

multi-party computation in Section 2. We then explain the appropriate multi-party computation techniques as well as representative optimizations on the accuracy and efficiency for centralized (Section 3), decentralized (Section 4), and semi-centralized federated learning (Section 5), respectively. Finally, we summarize the future directions in Section 6 and conclude in Section 7.

2 Basic concepts and taxonomy

This section presents an overview of the basic concepts of federated learning and multi-party computation. Furthermore, we introduce a taxonomy of federated learning, which is grounded in the perspective of multi-party computation.

2.1 Federated learning

Federated learning is a paradigm that aims to enable multiple data owners F_i to collaboratively train a machine learning model M using their respective datasets D_i, while ensuring the data kept locally for each data owners. Traditional approaches often require the integration of all data to create a universal dataset $D = D_1 \cup \cdots \cup D_n$, on which the model M_{sum} is trained. However, this method can lead to privacy breaches and can potentially violate relevant regulations. On the contrast, federated learning enables the training of a federated model M_{fed} using secure transmission of intermediate results, without requiring data owners to directly provide their data D_i. With federated learning, the performance V_{fed} of the federated model M_{fed} can approach that of the direct training model M_{sum}, denoted by V_{sum}. Formally,

$$|V_{fed} - V_{sum}| \leqslant \Delta, \tag{1}$$

where Δ is a small positive number used to measure the precision loss of model in federated learning. In contrast to general distributed learning, which typically involves manually adjusting the data distribution of each party to achieve faster convergence, federated learning leverages the local data of each participating party to train the model. In federated learning, each data owner trains the model on its own local data, and only the updated model weights are transmitted to a central server for aggregation.

The two fundamental dimensions of data used in federated learning are records and their associated features, denoted by U and X, respectively. To illustrate, consider a federated learning scenario where two data owners participate, each with their own set of records denoted as U_1 and U_2, and their respective feature sets denoted as X_1 and X_2. Given that the feature sets and records may differ between data owners, federated learning can be classified into horizontal federated learning [8], vertical federated learning [26], and federated transfer learning [27] based on the overlap of features and records. The current literature on federated learning is typically categorized based on the aforementioned horizontal, vertical, and transfer perspectives. Thus, we omit the details of this classification and refer a comprehensive survey [2] for more details.

2.2 Multi-party computation

The multi-party computation is a fundamental techniques that can take multi-party private inputs and perform aggregated output without revealing the private data of each party. It can be described by the mathematical model that:

$$y = f(x_1, x_2, \ldots, x_n), \tag{2}$$

where $x_1, x_2, \ldots, x_n$ are the private input of each data owner, y is the aggregated output, and f is the agreed function with all data owners. During the computation of y, all private data x_i should be kept in each data owner locally. Federated learning typically utilizes multi-party computation techniques to enable efficient and secure data aggregation across multiple data owners. These techniques can be categorized into two types based on whether a centralized server is used for the aggregation: centralized multi-party computation and decentralized multi-party computation.

2.2.1 Centralized multi-party computation

Centralized multi-party computation can securely aggregate data from multiple data owners at a centralized server. To preserve data privacy, data owners should obfuscate their local models before uploading them to the server. Popular techniques in this category include (central) differential privacy and local differential privacy [28]. Due to the obfuscation, however, more bias may be introduced into the federated model training process [29]. As a result, researchers working with centralized multi-party computation often focus on developing optimization techniques that can improve model accuracy while accounting for this source of bias [30].

Central differential privacy Central differential privacy (CDP) [31] works by adding random noise to uploaded parameters to prevent individual data from being identified. The idea is to make it difficult for an adversary to determine whether a particular individual's data is included in a dataset, while still allowing for accurate statistical analysis. It is defined on two adjacent datasets D and D', i.e., two datasets differing from one record. Specifically, a randomized algorithm $\mathcal{M}$ is called (ϵ,δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all adjacent datasets D and D',

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leqslant \exp(\epsilon)\Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta, \tag{3}$$

where $\epsilon > 0$ is the private budget. As described above, the definition of differential privacy guarantees privacy theoretically, but implementation requires perturbing the data by adding noise.

Local differential privacy Local differential privacy (LDP) [16] is a privacy-preserving technique that operates directly on individual data points, rather than on aggregated data or query results. In local differential privacy, each data point is perturbed with random noise before it is shared, rather than perturbing the output of a query or aggregation. This helps to prevent individual data points from being linked to specific users, while still allowing for accurate analysis of the data. It is defined on two arbitray data x and x':

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leqslant \exp(\epsilon)\Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta. \tag{4}$$

The main difference between local differential privacy and differential privacy is the level at which the privacy protection is applied. Differential privacy perturbs the output of queries or aggregations on data, while local differential privacy perturbs the individual data points themselves. As a result, local differential privacy can provide stronger privacy guarantees than differential privacy in some cases, but may also be more computationally expensive, as it requires adding noise to each individual data point. Thus, shuffler-based solutions are gradually used to improve the utility of local differential privacy [32].

2.2.2 Decentralized multi-party computation

Decentralized multi-party computation enables secure aggregation without the need for a third-party server. This approach can be traced back to the millionaires' problem first proposed by Yao in 1982 [33], which allowed two millionaires to determine who was richer without revealing their actual wealth. In decentralized multi-party computation, parties communicate with each other to aggregate the final results, resulting in lower efficiency compared to centralized computation techniques. However, decentralized multi-party computation does not introduce additional bias to the results and can provide better accuracy for federated models. Representative techniques of decentralized multi-party computation include garbled circuit, secret sharing, homomorphic encryption, etc. [34]

Garbled circuit Garbled circuit (GC) is first introduced by Yao in 1986 [35], wherein a Boolean circuit is constructed to enable secure computation between two parties. The development of GC technology has been focused on two key aspects: performance and security enhancements. The security aspect of the GC is mainly reflected in its ability to provide protection against both semi-honest and malicious adversaries. Meanwhile, research on enhancing the performance of GC schemes while maintaining a similar level of security remains an active area of investigation.

The increasing demand for multi-party computation applications in real-world scenarios has led to a surge in research on Garbled circuit schemes involving multiple parties. Furthermore, several compilers [36,37] for secure two-and multi-party computation that employ Garbled circuit have been developed.

Secret sharing Secret sharing (SS) is a cryptographic technique used to distribute a secret among a group of participants in such a way that no single participant has access to the complete secret. Instead, the secret is divided into shares, and each participant is given a share of the secret [38]. Only when a sufficient number of shares are combined (known as the "threshold") can the original secret be reconstructed.

By dividing the data into shares, SS can ensure that no single participant has access to the complete data. This makes it possible to train the machine learning model securely without disclosing the raw data to any participant [39]. This can be done by dividing the model into shares and distributing them among the participants, so that the model can only be reconstructed when a sufficient number of shares are combined (i.e., the threshold).

Homomorphic encryption Homomorphic encryption is a type of encryption technique that allows computations to be performed on encrypted data without requiring decryption [40]. This means that the data remains encrypted throughout the computation process, protecting the privacy and confidentiality of the data. In other words, homomorphic encryption enables data to be securely processed and manipulated while it is still in an encrypted state. It can help to protect the privacy of the data during the model training process. By using homomorphic encryption, data owners can perform computations on their encrypted data and share the encrypted results with each other without revealing any information about the underlying data.

2.3 Taxonomy

According to the underline multi-party computation techniques, federated learning can be classified into three categories: centralized FL, decentralized FL, and semi-centralized FL. Figure 1 shows the relationship between the federated learning and multi-party computation.

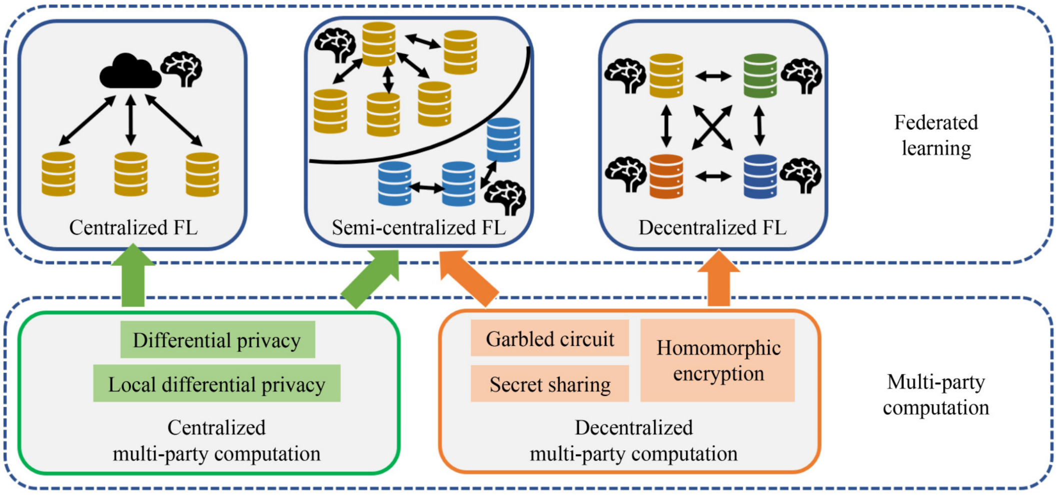

Fig. 1 The taxonomy of federated learning from a perspective of multi-party computation

Centralized FL The prevailing federated learning models assume the presence of a central server for coordinating the model aggregation process [8,41,42]. For example, the FedAvg algorithm [8] follows the centralized federated learning process. First, the central server initializes the global federated model. Next, a randomly selected subset U of data owners download the global model from the central server, followed by training and updating the model based on their local data. These data owners then upload their local models to the central server. Finally, the central server aggregates the uploaded local models from all participants. This process repeats until the global federated model satisfies the convergence condition.

Decentralized FL Decentralized FL eliminates the need for coordinators in federated learning. Instead, they are based on a peer-to-peer network that supports model aggregation, with participants requiring predefined permissions to access the federated learning process [43–45]. For example, Kim et al. [45] designed a decentralized federated learning framework based on blockchain technology that leverages the verifiability and incentivization properties of blockchain. By updating models asynchronously through a distributed ledger mechanism, they avoid the waiting problem that arises from synchronous updates. The verifiability of block-chain also supports the validation of local model training results and extends the scope of federated learning to untrusted public network environments.

Semi-centralized FL Since the centralized FL usually involves high computation overhead for server, and the decentralized FL requires high peer-to-peer communication cost, a new federated learning called semi-centralized FL has been proposed to trade off the communication cost and computation overhead. There are two types of semi-centralized FL methods: clustered federated learning [12] and hierarchical federated learning [13]. In clustered federated learning, the data owners are clustered based on the inference of their task types, and multiple cluster centers are united to aggregate the global models. On the other hand, hierarchical federated learning aims to reduce the communication delay in wireless networks. Local model aggregation is carried out at the intermediate nodes, thereby reducing the communication costs of the data owners who need to communicate with the remote central server over long distances.

3 Centralized FL

This section introduces the related work about centralized FL from three aspects: privacy preservation, accuracy optimization, and efficiency improvement.

3.1 Privacy preservation

To preserve the data privacy of each data owner, both the data owners themselves and the central server must implement privacy protection techniques. One such technique that has proven successful in federated learning is differential privacy, which operates on a probabilistic basis. Research into differential privacy techniques in the context of federated learning can be classified into two categories: federated learning using central differential privacy (CDP) and local differential privacy (LDP).

CDP based solutions Since federated learning is vulnerable to differential attacks [19], there has been a growing interest in exploring federated learning methods that incorporate central differential privacy. The conventional approach to preventing differential attacks involves introducing noise into aggregated results through a third-party server, which, unfortunately, results in a loss of accuracy. To address this issue, Agarwal et al. [46] suggested using the binomial mechanism along with a stochastic k-level quantization method and randomized rotation method. Canonne et al. [47] presented a discrete Gaussian mechanism as an alternative approach. Agarwal et al. [48] further proposed a new multi-dimensional Skellam mechanism to enhance privacy protection. Meanwhile, Jiang et al. [49] focused on the problem of client dropout in distributed differential privacy and developed a secure federated learning framework. There are some other studies about the centralized FL based on DP, such as [50,51]. Triastcyn et al. [20] proposed the use of Bayesian differential privacy, which facilitates more precise communication. In the same vein, Wei et al. [21] proposed a novel federated learning method known as NbAFL, which involves adding manual noise to client-side parameters before aggregation. NbAFL meets central differential privacy standards at various levels of preservation. Furthermore, Zhang et al. [22] proposed a clipping-enabled FedAvg approach that incorporates the clipping technique into federated learning and central differential privacy. To minimize the loss of accuracy, Hu et al. [52] proposed a new scheme known as Fed-SMP, which guarantees differential privacy at the data owner level. And the differential privacy based federated topic modeling has been studied in [23,53,54]. Truex et al. [55] further proposed a hybrid privacy-preserving federated learning framework by combining differential privacy and homomorphic encryption.

LDP based solutions Kasiviswanathan et al. [56] were the first to propose federated learning with local differential privacy. Later, Erlingsson et al. [57] introduced a privacy-preserving mechanism called Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR), which enables the collection of statistics on the population of client-side strings while offering strong privacy guarantees for each client. Truex et al. [9] proposed LDP-Fed, which provides formal differential privacy guarantees for the collection of parameters in federated neural networks. Additionally, Girgis et al. [58] proposed a novel shuffle privacy mechanism where each party randomizes its response, and the server only receives a random shuffle of the clients' responses. Numerical results indicate significant improvements in privacy guarantees. Wang et al. [59] studied the federated topic modeling based on local differential privacy. And a three-plane approach was proposed by [60] that applies local differential privacy to user data before it is uploaded.

3.2 Accuracy optimization

The essence of federated learning is to generate a machine learning model with strong generalization ability on specific tasks through multi-client cooperation. It can enable all data owners to train the machine learning model under the premise of ensuring data security, so that data owners can share data value without sharing the data itself. In the actual application scenario of federated learning, data heterogeneity challenges the generalization performance of the algorithm. In order to reduce the impact of data heterogeneity on model performance, Mcmahan et al. [8] of Google Research Institute first proposed FedAvg algorithm. This algorithm trains the federated model on a mixed distribution $\hat{D} = \sum_{i=1}^{N} w_i D_i$, where the weight $w_i = |D_i|/|D|$ is the proportion of data size for each client, and $|D|$ is the total data size. However, the resulting model will be biased towards clients with large amounts of data, thus affecting the generalization performance of the model. Mohri et al. [41] proposed an agnostic federated learning framework to analyze the generalization performance of federated learning in heterogeneous data scenarios. For the distribution of multi-clients $\{D_1, \ldots, D_N\}$, the agnostic federated learning train the federated models on all possible test distribution $\hat{D} = \sum_{i=1}^{N} \lambda_i D_i$, where the weight λ belongs to a n-dimension single pure Δ_N. Based on the skewness of weight λ, sample ratio $|D_i|/|D|$, and Rademacher complexity [61], it gives the generalized error bound of the unknowable federated learning.

Agnostic federated learning can produce a global model with robust performance on any target distribution by solving the corresponding minmax optimization problem. However, for clients with large differences between distribution and global distribution, the performance of a single global model on this client is usually poor. In order to remedy the limitations of a single model, Mansour et al. [62] proposed three methods that can customize personalized models for clients, namely client clustering, data interpolation and model interpolation, and gave the corresponding generalization error bounds; Deng et al. [63] also improved the model interpolation of the client, and proposed a learning method of adaptive generalization error bound.

3.3 Efficiency improvement

Federated learning is affected by the heterogeneity of participants' equipment and the limited network bandwidth, which leads to the biggest challenge that hinders its implementation. The main factor that affects the efficiency of the federated learning algorithm is the communication cost of transferring parameters between the client and the central service, especially in a scenario with a large number of clients.

In the scenario with a large number of clients, the federated learning algorithm needs to communicate with each client, resulting in a low efficiency of the algorithm. The existing research aims at this problem by selecting a certain number of clients from many clients and training them as representatives to reduce communication costs and optimize the efficiency of the algorithm. According to whether the online status of the client changes dynamically during the federated learning process, the client selection algorithm can be divided into two categories: static client selection and dynamic client selection.

In the static client selection, there is no downtime and sudden exit of the federated learning client. The client selection only need to perform once during the federated learning process. The FedAvg algorithm proposed by Mcmahan et al. [8] adopted a simple random sampling strategy, which can also achieve good training results when the data distribution satisfies the assumption of independent identical distribution. Wei et al. [64] and Song et al. [65] introduced the concept of Sharpley value into federated learning, and the data quality was evaluated by efficiently calculating the Sharpley value of data owners in federated learning, so as to effectively select clients. Chai et al. [66] proposed a tier-based federated learning (TiFL) system. The participants of federated learning are layered based on the training performance, and the participants are selected according to the level of participants in the training, which improves the convergence speed of federated learning in heterogeneous scenarios.

In dynamic client selection, the state of the client is dynamic, and each data owner in federated learning may be offline due to network, hardware, etc. Huang et al. [67] dynamically selected clients in each round based on multi-arm bandit machines. Lai et al. [68] further implemented a federated learning client selection algorithm based on exploration-utilization strategy. Zhang et al. [69] proposed a submodular based solution for client selection. And Wang et al. [70] evaluated the value of clients in each round by training a deep reinforcement learning model, and selected K participants with the highest value to conduct federated learning training. However, this method requires additional deep model training on the basis of the federated model, which puts forward higher requirements for computing resources.

4 Decentralized FL

In this section, we introduce the optimization techniques of decentralized FL.

4.1 Privacy preservation

In decentalized FL, the model is aggregated without relying on a central server. Instead, data owners communicate directly with each other to perform the model aggregation, typically using decentralized multi-party computation techniques to preserve individual data privacy. Techniques commonly used in decentralized FL include garbled circuits, secret sharing, and homomorphic encryption. These techniques enable secure multi-party computation without revealing any sensitive information, facilitating effective collaboration while maintaining individual data privacy.

It is worth noting that blockchain-based solutions [44,45] can also be considered a form of decentralized FL. While blockchain-based solutions can offer benefits such as verifiability and transparency, they must also address data privacy concerns to ensure that individual data remains protected.

GC and SS based solutions The secret sharing involves participants calculating the share of different secret data and obtaining the share of calculation results, which are combined to recover the calculation results [71]. SecureML [10] is the first federated learning system that uses secret-sharing and Yao's Garbled Circuit [33] for encryption to ensure security.

And Google [1] proposed an efficient and secure aggregation method that utilizes secret sharing technology, enabling servers to securely aggregate gradients without leaking the gradient of a single user.

The advantage of adding MPC protocols to FL based on secret sharing is its ability to extend to a large number of users with relatively low computational cost. However, the weakness of generic MPC protocols based on secret sharing is the huge communication cost. Chen et al. [72] and Ziller et al. [73] implemented the federated learning via a general decentralized multi-party computation framework, SPDZ [74]. And Shamir's t-out-of-n Secret Sharing was used in the protocol of [17] under the assumption of an honest-but-curious setting, which allows a user to split a secret into shares. Liu [24] proposed a PFK-Means profile that combines secret sharing and federated learning, by transmitting secret shared gradients instead of uploading encrypted data directly. Jeon et al. [75] proposed a secret-sharing based model aggregation m-ethod called Alternating Direction Method of Multiplier (ADMM), which can control the communication pattern among data owners.

HE based solutions Homomorphic encryption (HE) enables computation on ciphertext, but it is expensive and involves modular computations, leading to high computational and communication overhead. CryptoNets [76] used leveled homomorphic encryption for encrypted prediction on the server side, but it has limitations due to the degree of polynomial approximation of non-linear activation functions. Chen et al. [77] utilized the additive homomorphic encryption to boosting model, and proposed a lossless federated boosting framework. Liu et al. [24] further combined HE and secret sharing for neural networks in two-party computation with almost lossless accuracy. Zhang et al. studied the federated skyline analysis over vertical data federation [78], and proposed an homomorhpic encryption based private set emptiness protocol to accelerate the efficiency of skyline analysis.

4.2 Accuracy optimization

In decentralized FL, the heterogenous data can significantly affect the training efficiency of the model, leading to slow convergence rates. This is because, in each round of model training, the global model needs to aggregate parameters from all data owners, and the distributed offset between the data owners can have a significant impact on the convergence rate of the federated model. The research of Li et al. [79] showed that when there is a large deviation between the data distribution of participants and the average distribution, the convergence rate of existing federated learning methods will decrease significantly.

In order to solve the above problems, Karimireddy et al. [80] proposed SCAFFOLD, which can correct the client offset phenomenon when processing non-IID data with the help of control variables. They further proved theoretically that the proposed stochastic controlled averaging for federated learning method has a higher convergence rate. However, the SCAFFOLD method only considers reducing the communication cost, and the accuracy of the model cannot be well guaranteed. Hamer et al. [81] proposed an efficient FedBoost (Federated Boosting) method based on the idea of ensemble learning, and theoretically analyzed the generalized error bound of FedBoost for the specific task of density estimation. Although the work of Rothchild et al. [82] and Hamer et al. [81] has improved the model performance and training efficiency. These methods increase the complexity of the model to a certain extent and bring difficulties to the actual deployment of the methods.

4.3 Efficiency improvement

In decentralized FL, the huge communication cost brought by the transmission of the deep learning model has become the bottleneck of federated learning. Model compression can reduce the communication overhead caused by model transmission and improve the efficiency of the federated learning algorithm at the expense of certain model performance. Suresh et al. [83] first proposed a communication coding algorithm based on random rotation in a distributed scenario, proved that the minimum mean square error can be achieved without making any assumptions about the data characteristics, and applied it to the distributed Lloyd algorithm. The experimental results show that the proposed algorithm can greatly reduce the communication cost while maintaining the model's accuracy. On the basis of compressing the model, Caldas et al. [84] proposed the federated dropout to select the subset of the global model, so as to update the parameters. Compared with the existing work, the communication cost is reduced to 1/14. Xu et al. [85] aimed at the problem that a large number of redundant parameters need to be updated in the federated learning algorithm, proposed a federated trained ternary quantization (FTTQ) algorithm to optimize the learning model in the client through self-learning and proved the convergence of the proposed algorithm. Hadadpour et al. [86] proposed periodic compression algorithms for homogeneous and heterogeneous federated learning, FedCOM and FedCOMGATE, respectively. They further gave the convergence bounds of these algorithms under different assumptions such as non-con-

Table 1 Summary of federated learning from a perspective of multi-party computation

Federated learning	Multi-party computation	Literature
Centrailized FL	Central differential privacy	[19], [20], [21], [22], [23], [46], [47] [48], [49], [50], [51], [52], [55]
	Local differential privacy	[9], [56], [58], [59], [60]
Decentralized FL	Garbled circuit and secret sharing	[10], [17], [24], [72], [73], [75]
	Homomorphic Encryption	[24], [76], [77], [78]
Semi-centralized FL	/	[11], [12], [13], [88]

vexity and strong convexity. On the basis of model compression, Cui et al. [87] implemented the blockchain-based federated learning algorithm, a compressed algorithm of federated learning for content caching (CREAT), which further protected the security of the client node data.

5 Semi-centralized FL

Semi-centralized FL is a novel way to organize the data owners in federated learning. Since the centralized FL may involve too many communication costs of server and the decentralized FL suffers from the inefficiency of multi-party computations, semi-centralized FL is proposed to mitigate these two forms for better training performance. In semi-centralized FL, several data owners are selected as agents for a set of data owners. Each data owner communicates with its assigned agent in a centralized manner. The agents then communicate with each other using decentralized multi-party computation techniques to aggregate the model updates and obtain the global model. Table 1 lists a summary of centralized, decentralized and semi-centralized federated learning.

Cluster based solutions Ghosh et al. [11] has earlier studied the federated learning algorithm in the multi-center group structure scenario, which assumes that the model of all N federated participants applies to K tasks, that is, there are K cluster centers $C_1, C_2, \ldots, C_K$. Cluster partitioning can avoid mutual interference between unrelated federal participants and improve the accuracy of the jointly constructed group model. The federated learning framework based on clustering includes two parts. The first is the centralized federated learning module, which aggregates the models on K cluster centers $C_1, C_2, \ldots, C_K$ according to the gradient parameters uploaded by each federated participant to obtain the global model $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$. The second is the cluster iterative division module. First, calculate the gradient of each global model about the objective function, then calculate each participant's similarity, update the cluster division according to each participant's distance, and repeat the above process until convergence.

Sattler et al. [12] proposed a similar federal learning framework, which does not explicitly calculate the distance (similarity) between participants, but determines which category each federal participant belongs to by evaluating the accuracy of the local model improved by different clusters. Ouyang et al. [88] proposed a clustering-based federated learning system that aims to improve the efficiency and accuracy of human activity recognition applications. They utilize a dynamic clustering algorithm to federated learning, which can drop nodes that converge slower or have little correlations with others in each cluster. This approach can speed up convergence and reduce communication overhead, leading to improved efficiency and accuracy.

Hierarchy based solutions In mobile networks, the high communication cost resulting from direct data transmission between federated participants and the central server due to varying link distances can pose a challenge. To address this issue, Abad et al. [13] proposed a hierarchical framework for federated learning to reduce communication latency. The participants in hierarchical federated learning are divided into three levels based on their geographical locations and latency size: (1) data nodes (workers), which are federated learning participants with local data; (2) intermediate nodes (cluster heads), which are responsible for aggregating the training results of some data nodes; and (3) model nodes (model owners), which are responsible for aggregating the local models of intermediate nodes and obtaining the final federated model.

The training process of the hierarchical partition-based federated learning model is similar to that of centralized federated learning. Firstly, the local model M_i is trained by the data nodes at the bottom layer. Then, the intermediate nodes aggregate the local models of their responsible regions to obtain a new intermediate-level model M_{mid}, which is distributed to all data nodes for further training. After T rounds of iteration, the intermediate-level models M_{mid} are uploaded to the model node for aggregation, and the new model is distributed to all intermediate nodes. This iteration continues until the converged federated learning model M_{fed} is obtained.

The hierarchical partition-based federated learning framework effectively reduces communication costs in mobile networks by partitioning levels and selecting regional centers to lower communication latency.

6 Future directions

So far, some challenges still remain in the study of Federated Learning, we've identified the new trends as three things:

Incentive mechanism One of the main challenges in federated learning is incentivizing data owners to participate in the training process. Incentive mechanisms aim to motivate data owners to contribute their data and computing resources to the federated learning model by providing them with some form of reward or benefit. This is particularly important because in federated learning, data owners are essentially providing their data and computing resources to the model without direct compensation [65]. The development of incentive mechanisms could encourage more data owners to participate in Federated Learning, which would in turn improve the quality and diversity of the data used for model training.

Personalization Personalization is a promising direction for the development of federated learning. In federated learning, the data held by multiple data owners is usually non-IID. And these data owners may have different objectives for the learning model. Thus, a personalization of federated learning can dedicate to the objective of each data owner and thus can lead to significant improvements in model performance and user experience [89].

Legitimacy Another important challenge of federated learning is ensuring the legitimacy of the models generated by the training process. In the context of federated learning, computational legitimacy can be used to ensure the legitimacy of the models generated by the federated learning process.

This involves the use of algorithms to verify that the models generated by the participating devices are consistent with legal and ethical requirements, such as data privacy regulations and fairness constraints [90].

7 Conclusion

This paper provides an overview of federated learning from the perspective of multi-party computation. According to whether there is a centralized server to perform model aggregation during multi-party training, we classify federated learning into three categories: centralized, decentralized, and semi-centralized. We explained how representative studies trade-off among privacy, accuracy, and efficiency for each category and pointed out a few future directions of federated learning. We envision federated learning as a versatile enabler for large-scale data sharing and circulation.

Acknowledgements We are grateful to anonymous reviewers for their constructive comments. This work was partially supported by the National Natural Science Foundation of China (NSFC) (Grant Nos. U21A20516, 62076017, and 62141605), the Funding of Advanced Innovation Center for Future Blockchain and Privacy Computing (No. ZF226G2201), the Beihang University Basic Research Funding (No. YWF-22-L-531), the Funding (No. 22-TQ23-14-ZD-01-001) and WeBank Scholars Program.

Competing interests The authors declare that they have no competing interests or financial conflicts to disclose.

References

1. Konečný J, McMahan H B, Yu F X, Richtárik P, Suresh A T, Bacon D Federated learning: strategies for improving communication efficiency. 2016, arXiv preprint arXiv: 1610.05492
2. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 12
3. Tong Y, Zeng Y, Zhou Z, Liu B, Shi Y, Li S, Xu K, Lv W. Federated computing: query, learning, and beyond. IEEE Data Engineering Bulletin, 2023, 46(1): 9–26
4. Zhang K, Song X, Zhang C, Yu S. Challenges and future directions of secure federated learning: a survey. Frontiers of Computer Science, 2022, 16(5): 165817
5. Chen Y, Qin X, Wang J, Yu C, Gao W. FedHealth: a federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems, 2020, 35(4): 83–93
6. Byrd D, Polychroniadou A. Differentially private secure multi-party computation for federated learning in financial applications. In: Proceedings of the 1st ACM International Conference on AI in Finance. 2020, 16
7. Liu S, Xu S, Yu W, Fu Z, Zhang Y, Marian A. FedCT: federated collaborative transfer for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 716−725
8. McMahan B, Moore E, Ramage D, Hampson S, Arcas B A Y. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. 2017, 1273−1282
9. Truex S, Liu L, Chow K H, Gursoy M E, Wei W. LDP-Fed: federated learning with local differential privacy. In: Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking. 2020, 61−66
10. Mohassel P, Zhang Y. SecureML: a system for scalable privacy-preserving machine learning. In: Proceedings of 2017 IEEE Symposium on Security and Privacy. 2017, 19−38
11. Ghosh A, Chung J, Yin D, Ramchandran K. An efficient framework for clustered federated learning. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1643
12. Sattler F, Müller K R, Samek W. Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(8): 3710–3722
13. Abad M S H, Ozfatura E, GUndUz D, Ercetin O. Hierarchical federated learning ACROSS heterogeneous cellular networks. In: Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing. 2020, 8866−8870
14. Wang Z, Song M, Zhang Z, Song Y, Wang Q, Qi H. Beyond inferring class representatives: User-level privacy leakage from federated learning. In: Proceedings of 2019-IEEE Conference on Computer Communications. 2019, 2512−2520
15. Wei W, Liu L, Loper M, Chow K H, Gursoy M E, Truex S, Wu Y A framework for evaluating gradient leakage attacks in federated learning. 2020, arXiv preprint arXiv: 2004.10397
16. Cormode G, Jha S, Kulkarni T, Li N, Srivastava D, Wang T. Privacy at scale: local differential privacy in practice. In: Proceedings of 2018 International Conference on Management of Data. 2018, 1655−1658
17. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H B, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, 1175−1191
18. Tong Y, Pan X, Zeng Y, Shi Y, Xue C, Zhou Z, Zhang X, Chen L, Xu Y, Xu K, Lv W. Hu-Fu: efficient and secure spatial queries over data federation. Proceedings of the VLDB Endowment, 2022, 15(6): 1159–1172
19. Geyer R C, Klein T, Nabi M. Differentially private federated learning: a client level perspective. 2017, arXiv preprint arXiv: 1712.07557
20. Triastcyn A, Faltings B. Federated learning with Bayesian differential privacy. In: Proceedings of 2019 IEEE International Conference on Big Data. 2019, 2587−2596
21. Wei K, Li J, Ding M, Ma C, Su H, Zhang B, Poor H V. User-level privacy-preserving federated learning: analysis and performance optimization. IEEE Transactions on Mobile Computing, 2022, 21(9): 3388–3401
22. Zhang X, Chen X, Hong M, Wu S, Yi J. Understanding clipping for federated learning: Convergence and client-level differential privacy. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 26048−26067
23. Shi Y, Tong Y, Su Z, Jiang D, Zhou Z, Zhang W. Federated topic discovery: a semantic consistent approach. IEEE Intelligent Systems, 2021, 36(5): 96–103

24. Liu Y, Ma Z, Yan Z, Wang Z, Liu X, Ma J. Privacy-preserving federated k-means for proactive caching in next generation cellular networks. Information Sciences, 2020, 521: 14–31
25. Wang Y, Tong Y, Zhou Z, Ren Z, Xu Y, Wu G, Lv W. Fed-LTD: towards cross-platform ride hailing via federated learning to dispatch. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 4079−4089
26. Fu F, Shao Y, Yu L, Jiang J, Xue H, Tao Y, Cui B. VF^{2}Boost: very fast vertical federated gradient boosting for cross-enterprise learning. In: Proceedings of 2021 International Conference on Management of Data. 2021, 563−576
27. Liu Y, Kang Y, Xing C, Chen T, Yang Q. A secure federated transfer learning framework. IEEE Intelligent Systems, 2020, 35(4): 70–82
28. Zhang Y, Lu Y, Liu F. A systematic survey for differential privacy techniques in federated learning. Journal of Information Security, 2023, 14(2): 111–135
29. Ning B, Li X, Yang F, Sun Y, Li G, Yuan G Y. Group relational privacy protection on time-constrained point of interests. Frontiers of Computer Science, 2023, 17(3): 173607
30. Wang H, Xu Z, Zhang X, Peng X, Li K. An optimal differentially private data release mechanism with constrained error. Frontiers of Computer Science, 2022, 16(1): 161608
31. Dwork C, Kenthapadi K, McSherry F, Mironov I, Naor M. Our data, ourselves: privacy via distributed noise generation. In: Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2006, 486−503
32. Wang N, Zheng W, Wang Z, Wei Z, Gu Y, Tang P, Yu G. Collecting and analyzing key-value data under shuffled differential privacy. Frontiers of Computer Science, 2023, 17(2): 172606
33. Yao A C. Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. 1982, 160−164
34. Bayatbabolghani F, Blanton M. Secure multi-party computation. In: Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018, 2157−2159
35. Yao A C. How to generate and exchange secrets. In: Proceedings of the 27th Annual Symposium on Foundations of Computer Science. 1986, 162−167
36. Liu C, Wang X S, Nayak K, Huang Y, Shi E. ObliVM: a programming framework for secure computation. In: Proceedings of 2015 IEEE Symposium on Security and Privacy. 2015, 359−376
37. Zahur S, Rosulek M, Evans D. Two halves make a whole - reducing data transfer in garbled circuits using half gates. In: Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2015, 220−250
38. Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613
39. Zhang K, Tong Y, Shi Y, Zeng Y, Xu Y, Chen L, Zhou Z, Xu K, Lv W, Zheng Z. Approximate *k*-nearest neighbor query over spatial data federation. In: Proceedings of the 28th International Conference on Database Systems for Advanced Applications. 2023, 351−368
40. Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. 2009, 169−178
41. Mohri M, Sivek G, Suresh A T. Agnostic federated learning. In: Proceedings of the 36th International Conference on Machine Learning. 2019, 4615−4625
42. Shi Y, Tong Y, Zeng Y, Zhou Z, Ding B, Chen L. Efficient approximate range aggregation over large-scale spatial data federation. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(1): 418–430
43. Lim W Y B, Ng J S, Xiong Z, Jin J, Zhang Y, Niyato D, Leung C, Miao C. Decentralized edge intelligence: a dynamic resource allocation framework for hierarchical federated learning. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(3): 536–550
44. Warnat-Herresthal S, Schultze H, Shastry K L, Manamohan S, Mukherjee S, Garg V, Sarveswara R, Händler K, Pickkers P, Aziz N A, Ktena S, Tran F, Bitzer M, Ossowski S, Casadei N, Herr C, Petersheim D, Behrends U, Kern F, Fehlmann T, Schommers P, Lehmann C, Augustin M, Rybniker J, Altmüller J, Mishra N, Bernardes J P, Krämer B, Bonaguro L, Schulte-Schrepping J, De Domenico E, Siever C, Kraut M, Desai M, Monnet B, Saridaki M, Siegel C M, Drews A, Nuesch-Germano M, Theis H, Heyckendorf J, Schreiber S, Kim-Hellmuth S, COVID-19 Aachen Study (COVAS), Nattermann J, Skowasch D, Kurth I, Keller A, Bals R, Nürnberg P, Rieß O, Rosenstiel P, Netea M G, Theis F, Mukherjee S, Backes M, Aschenbrenner A C, Ulas T, Deutsche COVID-19 Omics Initiative (DeCOI), Breteler M M B, Giamarellos-Bourboulis E J, Kox M, Becker M, Cheran S, Woodacre M S, Goh E L, Schultze J L. Swarm learning for decentralized and confidential clinical machine learning. Nature, 2021, 594(7862): 265–270
45. Kim H, Park J, Bennis M, Kim S L. Blockchained on-device federated learning. IEEE Communications Letters, 2020, 24(6): 1279–1283
46. Agarwal N, Suresh A T, Yu F, Kumar S, McMahan H B. cpSGD: communication-efficient and differentially-private distributed SGD. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018, 7575−7586
47. Canonne C L, Kamath G, Steinke T. The discrete Gaussian for differential privacy. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1315
48. Agarwal N, Kairouz P, Liu Z. The skellam mechanism for differentially private federated learning. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2021, 5052−5064
49. Jiang L, Wang Y, Zheng W, Jin C, Li Z, Teo S G LSTMSPLIT: effective SPLIT learning based LSTM on sequential time-series data. 2022, arXiv preprint arXiv: 2203.04305
50. Cheu A, Smith A, Ullman J, Zeber D, Zhilyaev M. Distributed differential privacy via shuffling. In: Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2019, 375−403
51. Wang Y, Tong Y, Shi D, Xu K. An efficient approach for cross-silo federated learning to rank. In: Proceedings of the 37th International Conference on Data Engineering. 2021, 1128−1139
52. Hu R, Gong Y, Guo Y. Federated learning with sparsified model perturbation: improving accuracy under client-level differential privacy. 2022, arXiv preprint arXiv: 2202.07178
53. Jiang D, Song Y, Tong Y, Wu X, Zhao W, Xu Q, Yang Q. Federated topic modeling. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019, 1071−1080
54. Jiang D, Tong Y, Song Y, Wu X, Zhao W, Peng J, Lian R, Xu Q, Yang Q. Industrial federated topic modeling. ACM Transactions on Intelligent Systems and Technology, 2021, 12(1): 2
55. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y. A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. 2019, 1−11
56. Kasiviswanathan S P, Lee H K, Nissim K, Raskhodnikova S, Smith A. What can we learn privately? SIAM Journal on Computing, 2011, 40(3): 793−826
57. Erlingsson Ú, Pihur V, Korolova A. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of

2014 ACM SIGSAC Conference on Computer and Communications Security. 2014, 1054−1067

58. Girgis A M, Data D, Diggavi S N. Renyi differential privacy of the subsampled shuffle model in distributed learning. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2021, 29181−29192
59. Wang Y, Tong Y, Shi D. Federated latent dirichlet allocation: a local differential privacy based framework. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence. 2020, 6283−6290
60. Wang C, Wu X, Liu G, Deng T, Peng K, Wan S. Safeguarding cross-silo federated learning with local differential privacy. Digital Communications and Networks, 2022, 8(4): 446–454
61. Mohri M, Rostamizadeh A. Rademacher complexity bounds for non-I.I.D. processes. In: Proceedings of the 21st International Conference on Neural Information Processing Systems. 2008, 1097−1104
62. Mansour Y, Mohri M, Ro J, Suresh A T. Three approaches for personalization with applications to federated learning. 2020, arXiv preprint arXiv: 2002.10619
63. Deng Y, Kamani M M, Mahdavi M. Adaptive personalized federated learning. 2020, arXiv preprint arXiv: 2003.13461
64. Wei S, Tong Y, Zhou Z, Song T. Efficient and fair data valuation for horizontal federated learning. In: Yang Q, Fan L, Yu H, eds. Federated Learning. Cham: Springer, 2020, 139−152
65. Song T, Tong Y, Wei S. Profit allocation for federated learning. In: Proceedings of 2019 IEEE International Conference on Big Data. 2019, 2577−2586
66. Chai Z, Ali A, Zawad S, Truex S, Anwar A, Baracaldo N, Zhou Y, Ludwig H, Yan F, Cheng Y. TiFL: a tier-based federated learning system. In: Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing. 2020, 125−136
67. Huang T, Lin W, Wu W, He L, Li K, Zomaya A Y. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(7): 1552–1564
68. Lai F, Zhu X, Madhyastha H V, Chowdhury M. Oort: efficient federated learning via guided participant selection. 2020, arXiv preprint arXiv: 2010.06081
69. Zhang R, Wang Y, Zhou Z, Ren Z, Tong Y, Xu K. Data source selection in federated learning: a submodular optimization approach. In: Proceedings of the 27th International Conference on Database Systems for Advanced Applications. 2022, 606−614
70. Wang H, Kaplan Z, Niu D, Li B. Optimizing federated learning on non-IID data with reinforcement learning. In: Proceedings of the IEEE Conference on Computer Communications. 2020, 1698−1707
71. Pan X, Tong Y, Xue C, Zhou Z, Du J, Zeng Y, Shi Y, Zhang X, Chen L, Xu Y, Xu K, Lv W. Hu-fu: a data federation system for secure spatial queries. Proceedings of the VLDB Endowment, 2022, 15(12): 3582–3585
72. Chen V, Pastro V, Raykova M. Secure computation for machine learning with SPDZ. 2019, arXiv preprint arXiv: 1901.00329
73. Ziller A, Trask A, Lopardo A, Szymkow B, Wagner B, Bluemke E, Nounahon J M, Passerat-Palmbach J, Prakash K, Rose N, Ryffel T, Reza Z N, Kaissis G. PySyft: a library for easy federated learning. In: Rehman M H U, Gaber M M, eds. Federated Learning Systems: Towards Next-Generation AI. Cham: Springer, 2021, 111−139
74. Damgård I, Pastro V, Smart N, Zakarias S. Multiparty computation from somewhat homomorphic encryption. In: Proceedings of the 32nd Annual Cryptology Conference. 2012, 643−662
75. Jeon B, Ferdous S M, Rahman M R, Walid A. Privacy-preserving decentralized aggregation for federated learning. In: Proceedings of 2021 IEEE Conference on Computer Communications Workshops. 2021, 1−6
76. Dowlin N, Gilad-Bachrach R, Laine K, Lauter K E, Naehrig M, Wernsing J. CryptoNets: applying neural networks to encrypted data with high throughput and accuracy. In: Proceedings of the 33rd International Conference on Machine Learning. 2016, 201−210
77. Cheng K, Fan T, Jin Y, Liu Y, Chen T, Papadopoulos D, Yang Q. SecureBoost: a lossless federated learning framework. IEEE Intelligent Systems, 2021, 36(6): 87–98
78. Zhang Y, Shi Y, Zhou Z, Xue C, Xu Y, Xu K, Du J. Efficient and secure skyline queries over vertical data federation. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(9): 9269 − 9280
79. Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. In: Proceedings of the Machine Learning and Systems 2020. 2020
80. Karimireddy S P, Kale S, Mohri M, Reddi S J, Stich S U, Suresh A T. SCAFFOLD: stochastic controlled averaging for federated learning. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 476
81. Hamer J, Mohri M, Suresh A T. FedBoost: communication-efficient algorithms for federated learning. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 372
82. Rothchild D, Panda A, Ullah E, Ivkin N, Stoica I, Braverman V, Gonzalez J, Arora R. FetchSGD: communication-efficient federated learning with sketching. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 764
83. Suresh A T, Yu F X, Kumar S, McMahan H B. Distributed mean estimation with limited communication. In: Proceedings of the 34th International Conference on Machine Learning. 2017, 3329−3337
84. Caldas S, Konečný J, McMahan B, Talwalkar A. Expanding the reach of federated learning by reducing client resource requirements. In: Proceedings of the ICLR 2019. 2019
85. Xu J, Du W, Jin Y, He W, Cheng R. Ternary compression for communication-efficient federated learning. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(3): 1162–1176
86. Haddadpour F, Kamani M M, Mokhtari A, Mahdavi M. Federated learning with compression: unified analysis and sharp guarantees. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. 2021, 2350−2358
87. Cui L, Su X, Ming Z, Chen Z, Yang S, Zhou Y, Xiao W. CREAT: blockchain-assisted compression algorithm of federated learning for content caching in edge computing. IEEE Internet of Things Journal, 2022, 9(16): 14151–14161
88. Ouyang X, Xie Z, Zhou J, Xing G, Huang J. ClusterFL: a clustering-based federated learning system for human activity recognition. ACM Transactions on Sensor Networks, 2023, 19(1): 17
89. Tan A Z, Yu H, Cui L, Yang Q. Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems, 2022
90. Jiang D, Tan C, Peng J, Chen C, Wu X, Zhao W, Song Y, Tong Y, Liu C, Xu Q, Yang Q, Deng L. A GDPR-compliant ecosystem for speech recognition with transfer, federated, and evolutionary learning. ACM Transactions on Intelligent Systems and Technology, 2021, 12(3): 30

Fengxia Liu received the PhD degree in mathematics from the China Academy of Engineering Physics, China in 2021. Her research interests include the complexity of privacy analysis, federated learning, and graph neural networks.

Zhiming Zheng recieved the PhD degree in mathematics from Peking University, China in 1987. He has been engaged in network security, artificial intelligence and blockchain research for a long time, and has achieved a series of original research results. For example, in the aspect of network security research, he has established the theory and method of dynamic cryptographic-based cryptoanalysis based on the integration of algebra and dynamics and the related network security system, breaking through the key technical bottlenecks of space and space information security.

Yexuan Shi received the BE and PhD degrees in computer science and technology from Beihang University, China in 2017 and 2022, respectively. He is currently a post-doctoral researcher in the School of Computer Science and Engineering, Beihang University, China. His research interests include big spatio-temporal data analytics, federated learning, and privacy-preserving data analytics.

Yongxin Tong received the PhD degree in computer science and engineering from The Hong Kong University of Science and Technology, China in 2014. He is currently a professor in the School of Computer Science and Engineering, Beihang University, China. His research interests include big spatio-temporal data analytics, federated learning, crowdsourcing, privacy-preserving data analytics, and uncertain data management.

Yi Zhang received the PhD degree in probability theory and mathematical statistics from the Renmin University of China, China in 2020. He is currently a postdoc with Institute for mathematical science, Renmin University of China, China. His research interests include federated learning, supply chain finance, and optimization under uncertainty.