Federated Learning for Healthcare Informatics

JIE XU and FEI WANG, Weil Cornell Medical College, USA

Recent rapid development of medical informatization and the corresponding advances of automated data collection in clinical sciences generate large volume of healthcare data. Proper use of these big data is closely related to the perfection of the whole health system, and is of great significance to drug development, health management and public health services. However, in addition to the heterogeneous and highly dimensional data characteristics caused by a spectrum of complex data types ranging from free-text clinical notes to various medical images, the fragmented data sources and privacy concerns of healthcare data are also huge obstacles to multi-institutional healthcare informatics research. Federated learning, a mechanism of training a shared global model with a central server while keeping all the sensitive data in local institutions where the data belong, is a new attempt to connect the scattered healthcare data sources without ignoring the privacy of data. This survey focuses on reviewing the current progress on federated learning including, but not limited to, healthcare informatics. We summarize the general solutions to the statistical challenges, system challenges and privacy issues in federated learning research for reference. By doing the survey, we hope to provide a useful resource for health informatics and computational research on current progress of how to perform machine learning techniques on heterogeneous data scattered in a large volume of institutions while considering the privacy concerns on sharing data.

CCS Concepts: • **Applied computing → Health informatics**; • **Computing methodologies → Machine learning**; *Learning settings*; • **Security and privacy**;

Additional Key Words and Phrases: federated learning, healthcare, heterogeneous, multi-institutional

ACM Reference Format:
Jie Xu and Fei Wang. 2019. Federated Learning for Healthcare Informatics. *ACM Trans. Comput. Healthcare* 1, 1 (November 2019), 25 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the fast improvements in the informatization level of medical institutions, massive data have been produced during the processing of medical services, health care and health management, including electronic medical records, medical insurance information, health log, genetic inheritance, medical experimental results, scientific research data, *etc* [47, 96]. Analyzing these big data which stored in multiple institutions by machine learning techniques plays a great role in various aspects, *e.g.*, effectively integrating medical information resources, sharing diagnosis and treatment technology, accelerating drug research and development, assisting doctors in accurate judgement, reducing medical costs, predicting treatment plans and curative effects [29, 33]. Watson, one of the most famous applications of artificial intelligence in the medical field, focusing on the diagnosis of various cancer diseases and providing medical advice. A recent document revealed that Watson had mistakenly prescribed a drug that could have killed a patient during

Authors' address: Jie Xu, jix4002@med.cornell.edu; Fei Wang, few2001@med.cornell.edu, Weil Cornell Medical College, New York, NY, USA.

a simulation [1]. The misdiagnosis is largely due to data sources are far from enough. Sufficient medical data is key to accelerating and promoting medical research through the application of AI technology. However, the special properties of medical data, *e.g.*, heterogeneous, sensitive and poor accessibility, are huge obstacles not only to healthcare sciences, but also to computational research.

Personal medical data involve individual privacy, while medical experimental data or scientific research data are not only related to the privacy of data subjects, industry development, and even related to national security. The development of genomics and the change of the rules of research activities make the disclosure of privacy almost inevitable. Therefore, there have been regulatory policies or protection mechanisms for the privacy of data subjects being set to restrict the data access. The Standards for Privacy of Individually Identifiable Health Information, commonly known as the HIPAA (Health Insurance Portability and Accountability Act) Privacy Rule[2], establishes the first national standards in the United States to protect patients' personal or protected health information (PHI). On May 25, 2018, the General Data Protection Regulation (GDPR) issued by the European Union set strict rules on data security and privacy protection, emphasizing that the collection of user Data must be open and transparent [119]. After GDPR being enforceable, the California Consumer Privacy Act (CCPA) of 2018, the China Internet Security Law and some other laws have also strengthened their attention to data security. In this environment where governments have their own patient privacy protection mechanisms, analyzing medical big data may be subject to different, even conflicting regulations. When data come from a variety of sources, medical data analysts must abide by the provisions of multiple privacy regulation laws, increasing the difficulty of research. The balance between medical data analysis and patient privacy protection has indeed become a difficult and urgent problem to be solved.

Federated learning is a new attempt to solve the data dilemma faced by traditional machine learning methods. It enables training a shared global model with a central server while keeping all the sensitive data in local institutions where the data belong. In advance of involving much machine learning algorithms, the concept of "federated" has been well applied in learning community [57, 66], distributed data management and retrieval [9, 90, 98]. In 1976, Patrick Hill, a philosophy professor, first developed the Federated Learning Community (FLC) to bring people together to learn from each other, and helped students overcome the anonymity and isolation of large research universities [57]. After that, to support the discovery and access of learning content from diverse collection of content repositories, there are several efforts aimed at building federations of learning content and content repositories [9, 90, 98]. In 2005, Rehak *et al.* [98] developed a reference model that described how to establish an interoperable repository infrastructure by creating federations of repositories, where the metadata are collected from the contributing repositories into a central registry provided with a single point of discovery and access. The ultima goal of this model is to enable learning content from diverse content repositories to be found, retrieved and reused. Anyway, the practice of federated learning community or federated search service more or less provide references for the development of federated learning algorithms.

Before the term "federated learning" was formally introduced to describe the distributed-style learning technique of existing machine learning algorithms [71, 72, 87], there have been several work studied the analogous settings. In 2012, Balcan *et al.* [8] considered the problem of PAC-learning from distributed data and analyzed the fundamental communication questions, followed by general upper and lower bounds on the amount of communication required to obtain good outcomes. Richtárik *et al.* [100] developed a distributed coordinate descent method called Hydra for solving loss minimization problems with big data, where computations are done locally on each node, with minimum communication overhead. They also gave bounds on communication rounds sufficient to approximately solve the

[1]https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/
[2]https://www.hhs.gov/hipaa/for-professionals/privacy/index.html

strongly convex problem with high probability, and showed how it depended on the data and partitioning. Later on, Fercoq *et al.* [39] extended Hydra and proposed Hydra^2 for minimizing regularized non-strongly convex loss functions. They implemented the method on the largest supercomputer on UK and showed the method is capable of dealing with a LASSO problem with 50 billion variables. Following the development of big data analysis and big deep learning models, federated learning as an efficient distributed-style learning technique is getting more and more attention.

When Google first proposed federated learning concept in 2016, the application scenario is Gboard - a virtual keyboard of Google for touchscreen mobile devices with support for more than 600 language varieties [21, 53, 88, 97, 128]. At present, to implement the practical application of federated learning, the WeBank AI team has already committed to promoting the standardization of federal learning. In October 2018, they submitted to IEEE standards institute a proposal on establishing a federal Learning standard – "Guide for Architectural Framework and Application of Federated Machine Learning" (Federated Learning infrastructure and Application standard), which was approved in December 2018. Later, under the guidance of Prof. Qiang Yang, IEEE P3652.1 [6] (federated learning infrastructure and applications) standards working group was established. As this dialogue language supported by the international legal system between enterprises being established, the expansion of federal learning ecological can be further promoted.

As an innovative mechanism that could train global model from multiple parties with privacy-preserving property, federated learning has many other promising applications besides healthcare, *e.g.,* virtual keyboard prediction [21, 53, 88, 97, 128], smart retail [131], financial, vehicle-to-vehicle communication [104] and so on. Therefore, we want to summarizes the current progress on federated learning including, but not limited to, healthcare informatics. We hope to provide a useful resource for health informatics and computational research for reference.

There have been some related summative works on federated learning [28, 127]. Dai *et al.* [28] provide an overview of the architecture and optimization approach for federated data analysis, where Newton-Ralphson method and alternating direction multiplier (ADMM) framework are used for distributed computation. Yang *et al.* [127] provide definitions, architectures and applications for the federated learning framework, and introduce a general privacy-preserving techniques that can be applied to federated learning. They also categorize federated learning based on the distribution characteristics of the data. Different from these works, this paper mainly summarizes the current progress on federated learning. We discuss the general solutions to the statistical challenges, system challenges and privacy issues in federated learning research. By doing the survey, we hope to provide a useful resource for health informatics and computational research on current progress of how to perform machine learning techniques on heterogeneous data scattered in a large volume of institutions while considering the privacy concerns on sharing data.

The rest of survey is organized as follows. In Sec. 2, we give a general overview of federated learning and define some notations in Tab. 1 which will be used later. Then, we summarize the challenges of federated learning and introduce the current progress on studying these issues in the next three Sections 3,4,5. After that, we briefly summarize the federated optimization algorithms in Sec. 6. In Sec. 7, we introduce some other applications and the popular platforms or federated learning research and hope to provide a useful resource for the beginners. Finally, we conclude the paper and discuss some other probably encountered questions when the federated learning is applied in healthcare area in Sec. 9.

2 FEDERATED LEARNING PROBLEM SETTING

Federated learning is a problem of training a high-quality shared global model with a central server from decentralized data scattered among extremely large number of different clients.

Table 1. List of Important notations

Symbol	Description
K	Number of activated clients
n	Total number of data points participated in collaboratively training
$\bar{\mathcal{D}}$	Target data distribution for the learning model
n_k	Number of data points stored on client k
$\mathcal{D}_k$	Data distribution associated to client k

Formally, assume there are K activated clients (a client could be a mobile, a wearable device or a medical institution, *etc*). Let $\mathcal{D}_k$ denote the data distribution associated to client k and n_k the number of samples available from that client. $n = \sum_{k=1}^{K} n_k$ is the total sample size. Federated machine learning problem boils down to solving a empirical risk minimization problem of the form [70, 71, 86]:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}) \quad \text{where} \quad F_k(\mathbf{w}) := \frac{1}{n_k} \sum_{i\in\mathcal{D}_k} f_i(\mathbf{w}). \tag{1}$$

The objective $F(\mathbf{w})$ in problem (1) can be rephrased as a linear combination of the local empirical objectives $F_k(\mathbf{w})$. In particular, algorithms for federated learning face with following challenges [17, 110]:

- **Statistical:** The data distribution among all clients differ greatly, *i.e.,* $\forall k \neq \tilde{k}$, we have $\mathbb{E}_{\mathbf{x}_i\sim\mathcal{D}_k}[f_i(\mathbf{w};\mathbf{x}_i)] \neq \mathbb{E}_{\mathbf{x}_i\sim\mathcal{D}_{\tilde{k}}}[f_i(\mathbf{w};\mathbf{x}_i)]$. It is such that any data points available locally are far from being a representative sample of the overall distribution, *i.e.,* $\mathbb{E}_{\mathbf{x}_i\sim\mathcal{D}_k}[f_i(\mathbf{w};\mathbf{x}_i)] \neq F(\mathbf{w})$.
- **Communication:** The number of clients K is large and can be much bigger than the average number of training sample stored in the activated clients, *i.e.,* $K \gg (n/K)$.
- **Privacy and Security:** Additional privacy protections are needed for unreliable participating clients. It is impossible to ensure none of the millions of clients are malicious.

Next, we will detailedly survey existing federated learning related works on handling with these challenges.

3 STATISTICAL CHALLENGES OF FEDERATED LEARNING

The naive way to solve the federated learning problem is through Federated Averaging (*FedAvg*) [86]. It is demonstrated can work with certain non-IID data by requiring all the clients to share the same model. However, *FedAvg* does not address the statistical challenge of strongly skewed data distributions. The performance of convolutional neural networks trained with *FedAvg* algorithm can reduce significantly due to the weight divergence [132]. We roughly organize existing research on dealing with the statistical challenge of federated learning into two groups, *i.e.,* consensus solution and pluralistic solution. We will detailedly discuss them in the following.

3.1 Consensus Solution

Most centralized models are trained on the aggregate training sample obtained from the samples drawn from the local clients [110, 132]. Intrinsically, the centralized model is trained to minimize the loss with respect to the uniform distribution [89]:

$$\bar{\mathcal{D}} = \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{D}_k, \tag{2}$$

where $\bar{\mathcal{D}}$ is the target data distribution for the learning model. However, this specific uniform distribution is not an adequate solution in most scenarios.

To address this issue, the recent proposed solution is to model the target distribution or force the data adapt to the uniform distribution [89, 132]. Specifically, Mohri *et al.* [89] proposed a minimax optimization scheme, *i.e.,* agnostic federated learning (AFL), where the centralized model is optimized for any possible target distribution formed by a mixture of the client distributions. This method has only been applied at small scales. Compared to AFL, Li *et al.* [79] proposed q-Fair Federated Learning (*q-FFL*), assigning higher weight to devices with poor performance, so that the distribution of accuracy in the network reduces in variance. They empirically demonstrate the improved flexibility and scalability of *q-FFL* compared to AFL. Duan *et al.* [34] built a self-balancing federated learning framework called Astraea, rebalancing the training by performing data augmentation to minority classes and rescheduling clients to achieve a partial equilibrium.

Another commonly used method is sharing a small portion of data. Zhao *et al.* [132] proposed a data-sharing strategy to improve FedAvg with non-IID data by creating a small subset of data which is globally shared between all the clients. The shared subset is required containing a uniform distribution over classes from the central server to the clients. Similarly, Yoshida *et al.* [129] presents a protocol called *Hybird-FL*, extending *FedCS* [92] to mitigate the non-IID data problem. The main idea of *Hybird-FL* is to construct an approximately IID dataset on the server by gathering data from a limited number of clients, and the model updated by the approximately IID data is aggregated with other models updated by other clients. In addition to handle non-IID issue, Han *et al.* [52] proposed to identify training bugs (*i.e.,* local data corruption) by sharing information of a small portion of trusted instances and noise patterns among. The trusted instances guide the local agents to select compact training subset, while the agents learn to add changes to selected data samples, in order to improve the test performance of the global model.

Besides the above methods, there are some work solving the statistical challenge by incorporating some special strategies in the optimization process [23, 45]. Chen *et al.* [23] analyzed *signSGD* and *medianSGD* in distributed settings with heterogeneous data by providing a gradient correction mechanism. After incorporating the perturbation mechanism, both algorithms are able to converge with provable rate. Ghosh *et al.* [45] proposed a modular algorithm for robust federated learning in a heterogeneous environment. After each client sends the local update to the server, the server runs outlier-robust clustering algorithm on these local parameters. After clustering, they run an outlier-robust distributed algorithm on each cluster, where each cluster can be thought of an instance of homogeneous distributed learning problem with possibly Byzantine machines. Different from the previous federated optimization problem, the server will do some relatively complicated task in this case.

The skewed distribution of data across different clients lead to very different learning rates for different clients, making tuning difficult without adaptive algorithms. To address these problems, Koskela *et al.* [73] propose a rigorous adaptive method for finding a good learning rate for *SGD*, and apply to differential privacy (DP) and federated learning settings. These works provide a good reference for solving the heterogeneous data problem in federated learning.

3.2 Pluralistic Solution

It is difficult to find a consensus solution $\mathbf{w}$ that is good for all components $\mathcal{D}_i$. Instead of wastefully insisting on a consensus solution, many researchers choose to embracing this heterogeneity.

Multi-task learning is a natural way to deal with the data drawn from different distributions. It directly captures relationships amongst non-IID and unbalanced data by leveraging the relatedness between them in comparison to learn a single global model. In order to do this, it is necessary to target a particular way in which tasks are related, *e.g.* sharing sparsity, sharing low-rank structure, graph-based relatedness and so forth. Recently, Smith *et al.* [110] empirically demonstrated this point on real-world federated datasets and proposed a novel method *MOCHA* to solve a general convex MTL problem with handling the system challenges at the same time. Later, Corinzia *et al.* [27] introduced *VIRTUAL*, an algorithm for federated multi-task learning with non-convex models. They consider the federation of central server and clients as a Bayesian network and perform training using approximated variational inference. This work bridges the frameworks of federated and transfer/continuous learning.

The success of multi-task learning rests on whether the chosen relatedness assumptions hold. Compared to this, pluralism can be a critical tool for dealing with heterogeneous data without any additional or even low-order terms that depend on the relatedness as in MTL [37]. Eichner *et al.* [37] considered training in the presence of block-cyclic data, and showed that a remarkably simple pluralistic approach can entirely resolve the source of data heterogeneity. When the component distributions are actually different, pluralism can outperform the "ideal" i.i.d. baseline.

Besides, different special cases of machine learning, *e.g.*, transfer learning, active learning, meta learning, are combined with federated learning principle to inherit their own advantages. Transfer learning is naturally introduced to solve the data heterogeneity problem, expands the scale of the available data and further improves the performance of the global model [85]. Active learning and meta learning are applied on local clients to deal with insufficient labeled data [20, 67, 95].

4 COMMUNICATION EFFICIENCY OF FEDERATED LEARNING

In federated learning setting, training data remain distributed over a large number of clients each with unreliable and relatively slow network connections. Generally, for synchronous algorithms in federated learning [72, 110], let $\mathbf{w}^0$ be the initial value, a typical round t consists of the following steps:

- A subset of existing clients is selected, each of which downloads the current model $\mathbf{w}^t$.
- Each client in the subset computes an updated model $\mathbf{w}_k^{t+1}$ based on their local data.
- The model updates $\mathbf{w}_k^{t+1}, k = 1, \dots, K$ are sent from the selected clients to the sever.
- The server aggregates these models (typically by averaging) to construct an improved global model, *i.e.*

$$\mathbf{w}^{t+1} := \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{w}_k^{t+1}. \tag{3}$$

Naively for the above protocol, the total number of bits that required during uplink (clinets $\longrightarrow$ server) and downlink (server $\longrightarrow$ clients) communication by each of the K clients during training are given by

$$\mathcal{B}^{up/down} \in \mathcal{O}(U \times \underbrace{|\mathbf{w}| \times (H(\triangle \mathbf{w}^{up/down}) + \beta)}_{\text{update size}}) \tag{4}$$

where U is the total number of updates performed by each client, $|\mathbf{w}|$ is the size of the model and $H(\triangle \mathbf{w}^{up/down})$ is the entropy of the weight updates exchanged during transmitting process. β is the difference between the true update size and the minimal update size (which is given by the entropy) [105]. Apparently, we can consider three ways to reduce the communication cost: a) reduce the number of clients K, b) reduce the update size and c) reduce the number of updates U. Starting at these three points, we can organize existing research on communication-efficient federated learning into four groups, *i.e.,* model compression, clients selection, updates reducing and peer-to-peer learning. We will detailedly discuss them in the following.

4.1 Client Selection

The most natural and rough way is to restrict the participated clients or choose a fraction of parameters to be updated at each round. Shokri *et al.* [107] use the selective stochastic gradient descent protocol, where the selection can be completely random or only the parameters whose current values are farther away from their local optima are selected, *i.e.*, those that have a larger gradient. Bui *et al.* [15] improved federated learning for Bayesian Neural Networks using Partitioned Variational Inference (PVI), where the client can decide to upload the parameters back to the central server after multiple passes through its data, after one local epoch, or after just one mini-batch. Wang [120] calculated Shapley value for each feature to explain the prediction of the model, and help us further quantify the contribution function from the clients without needing to know detailed values of data. This leaves room for participants to choose a learning schedule that meets the communication constraints.

Nishio *et al.* [92] propose a new protocol referred to as *FedCS*, where the central server manage the resources of heterogeneous clients and determine which clients should participate the current training task by analyzing the resource information of each client, such as wireless channel states, computational capacities and the size of data resources relevant to the current task. The server should decide how much data, energy and CPU resources used by the mobile devices such that the energy consumption, training latency, and bandwidth cost are minimized while meeting requirements of the training tasks. Anh [5] thus propose to use the Deep Q-Learning (DQL) [117] technique that enables the server to find the optimal data and energy management for the mobile devices participating in the Mobile Crowd-Machine Learning (MCML) through federated learning without any prior knowledge of network dynamics.

The limited communication bandwidth becomes the main bottleneck for aggregating the locally computed updates. Yang *et al.* [126] thus propose a novel over-the-air computation based approach for fast global model aggregation via exploring the superposition property of a wireless multiple-access channel. During federated model training process, the clients suffer from considerable overhead in communication and computation. Without well-designed incentives, self-interested mobile devices will be reluctant to participate in federal learning tasks, which will hinder the adoption of federated learning [65]. For this reason, Kim *et al.* [68] introduced the reward mechanism which is proportional to the training sample sizes into the proposed blockchained federated learning architecture. This measure promotes the federation of more clients with more training samples. Feng *et al.* [38] adopted the service pricing scheme to encourage the clients to participate the federated learning, where the price is also related to the training data size. They presented the Stackelberg game model to analyze the transmission strategy, training data pricing strategy of the self-organized mobile device and model owner's learning service subscription in the cooperative federated learning system. They focused on the interactions among mobile devices and considered the impact of the interference costs on the profits of mobile devices. Kang *et al.* [65] designed an incentive mechanism based on contract theory to motivate data owners with high-accuracy local training data to participate in the learning process, so as to achieve efficient federated learning.

4.2 Model Compression

The goal of reducing uplink communication cost is to compress the server-to-client exchanges. The first way is through structured updates, where the update is directly learned from a restricted space parameterized using a smaller number of variables, *e.g.* sparse or low-rank [72]. The second way is lossy compression, where a full model update is first learned and then compressed using a combination of quantization, random rotations, and subsampling before sending it to the server [3, 72]. Then the server decodes the updates before doing the aggregation. For deep neural networks, Chen *et al.* [24] categorize the multiple layers into shallow and deep layers and update the parameters of the deep layers less frequently than those of the shallow layers. Also, a temporally weighted aggregation is adopted, where the most recently updated model have higher weight in the aggregation. Sattler *et al.* [105] propose Sparse Ternary Compression (STC), a new compression framework that is specifically designed to meet the requirements of the Federated Learning environment. STC extends the existing compression technique of top-k gradient sparsification with a novel mechanism to enable downstream compression as well as ternarization and optimal Golomb encoding of the weight updates.

Most traditional distributed learning works focus on reducing the uplink communication cost and neglect that downloading a large model can still be considerable burden for users. For example, deep models which require significant computational resources both for training and inference are not easily downloaded and trained on edge devices. Due to this fact, many alternatives are proposed to compress the modes before deploying them on-device, *e.g.* pruning the least useful connections in a network [50, 51], weight quantization [30, 61, 82], and model distillation [58]. However, many of these approaches are not applicable for the federated learning problem, as they are either ingrained in the training procedure or are mostly optimized for inference [16]. Moreover, federated learning aims to deal with a large number of clients, thus communicating the global model may even become a bottleneck for the server [16].

Federated dropout, in which each client, instead of locally training an update to the whole global model, trains an update to a smaller sub-model [16]. These sub-models are subsets of the global model and, as such, the computed local updates have a natural interpretation as updates to the larger global model. It is noted that federated dropout not only reduces the downlink communication but also reduces the size of uplink updates. Moreover, the local computational costs is correspondingly reduced since the local training procedure dealing with parameters with smaller dimensions. Zhu *et al.* [133] proposes a multi-objective federated learning to simultaneously maximize the learning performance and minimize the communication cost using a multi-objective evolutionary algorithm. To improve the scalability in evolving large neural networks, a modified sparse evolutionary algorithm method is used to indirectly encode the connectivity of the neural network which effectively reduce the number of the connections of neural networks by encoding only two hyper parameters.

4.3 Updates Reducing

Kamp *et al.* [64] proposed to average models dynamically depending on the utility of the communication, which leads to a reduction of communication by an order of magnitude compared to periodically communicating state-of-the-art approaches. This is well suited for massively distributed systems with limited communication infrastructure. Guha [48] focus on techniques for one-shot federated learning, in which they learn a global model from data in the network using only a single round of communication between the devices and the central server. Besides above works, Ren *et al.* [99] theoretically analyze the detailed expression of the learning efficiency in the CPU scenario and formulate a training acceleration problem under both communication and learning resource budget. This work provides an important step towards the implementation of AI in wireless communication systems. Besides, reinforcement learning and round robin learning are used to manage the communication and computation resources [5, 62, 83, 121, 134].

4.4 Peer-to-Peer Learning

In federated learning, a central server is required to coordinate the training process of the global model. However, the communication cost to the central server may be not affordable since a large number of clients are usually involved. Also, many practical peer-to-peer networks are usually dynamic, and it is not possible to regularly access a fixed central server. Moreover, because of the dependence on central server, all clients are required to agree on one trusted central body, and whose failure would interrupt the training process for all clients. Therefore, some researches began to study fully decentralized framework where the central server is not required [74, 75, 101, 106].

Towards medical applications, Roy *et al.* [101] proposed *BrainTorrent*, where all clients directly interact with each other without depending on a central body. Lalitha *et al.* [74, 75] introduce a posterior distribution over a parameter space for each client to characterize the unknown global space. The local clients are distributed over the graph/network where they only communicate with their one-hop neighbors. Each client updates its local belief based on own data, then aggregates information from the one-hop neighbors. Shayan *et al.* [106] proposed a fully decentralized peer-to-peer approach called Biscotti, which uses crypto primitives and blockchain to coordinate a privacy-preserving multi-party ML process between local clients.

Although the advantages of a decentralized architecture have been proved as superior to its centralized counterpart when the nodes number is relatively large under a poor network condition [81], it generally operates only on a network where two nodes (or users) can exchange their local models only if they trust each other. However, in the case where node A may trust node B, but they still cannot communicate if node B does not trust node A. To solve this problem, He *et al.* [56] propose a central server free federated learning algorithm, named Online Push-Sum (OPS) method, to handle a generic scenario where the social network is unidirectional or of single-sided trust.

5 PRIVACY AND SECURITY

In federated learning, we usually assume the number of participated clients (*e.g.*, phones, cars, ...) is large and maybe reach to thousands or millions. It is impossible to ensure none of the clients are malicious. The setting of federated learning, where the model is trained locally without revealing the input data or the model's output to any clients, prevents the direct leakage while training or using the model. However, the clients may infer some information about another client's private dataset given the execution of $f(\mathbf{w})$, or over the shared predictive model $\mathbf{w}$ [114]. Yang *et al.* [127] introduce a comprehensive secure federated learning framework, which emphasize on general privacy-preserving techniques that can be applied to federated learning. In this section, we only focus on the federated learning scenario. We first surveying the attack related works, followed by the researches dealing with privacy issues.

5.1 Attack (Honest-but-Curious and Adversary Setting)

Apparently, neither data poisoning defense nor anomaly detection can be used in federated learning, since they require access to participated clients' training data or their uploaded model updates, respectively. The aggregation server cannot observe training data or model updates based on it without compromising participants' privacy. All of these problems could make federated learning be vulnerable to backdoors and other model-poisoning attacks [7].

5.1.1 Data Poisoning.

The naive approach is that the attacker can simply train its model on label-flipping or backdoor inputs. Also, the attacker can maximize the overfitting to the backdoor data by changing the local learning rate and the number of local epochs. This naive approach does not hinder federal learning. Aggregation offsets most of the contributions of the backdoored model, and the federation model soon forgets about backdoors. The attacker needs to be selected frequently, and even then, poisoning is slow [7].

5.1.2 Model Poisoning.

Inference attack aims to learn if a particular individual participated in training or the attributes of the records in training set [91]. In native federated learning setting where additional privacy preserving techniques are not included, that is, the parameters are visible to local clients even curious adversaries. The adversary can actively exploit SGD which is widely used in training deep neural networks, to leak more information about the participated local clients' training data. Nasr *et al.* [91] adopted the privacy vulnerabilities of the SGD algorithm and designed an active white-box attack that performs gradient ascent on a set of target data samples before uploading the parameters. This gradient ascent attacker forces the target model to show great differences between target members and non-member instances, which makes the membership inference attack easier. And the accuracy of the central attacker can be further improved by isolating participant during parameter update.

Another method is using model replacement to introduce backdoor functionality into the global model [7]. In this approach, the attacker makes an ambitious attempt to replace the new global model $\mathbf{w}^{t+1}$ with a malicious model $\mathbf{v}$ in Eq. (3):

$$\mathbf{v} := \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{w}_k^{t+1} = \mathbf{w}^t + \sum_{k=1}^{K} \frac{n_k}{n} (\mathbf{w}_k^{t+1} - \mathbf{w}^t). \tag{5}$$

Because the data distribution among all clients differ greatly, each local model may be far from the current global model. As the global model converges, these deviations begin to cancel out, *i.e.,* $\sum_{k=1}^{K-1} \frac{n_k}{n} (\mathbf{w}_k^{t+1} - \mathbf{w}^t) \approx 0$. Accordingly, the attacker can change the submitted model as below:

$$\tilde{\mathbf{w}}_k^{t+1} = \frac{n}{n_K} \mathbf{v} - (\frac{n}{n_K} - 1) \mathbf{w}^t - \sum_{k=1}^{K-1} (\mathbf{w}_k^{t+1} - \mathbf{w}^t) \approx \frac{n}{n_K} (\mathbf{v} - \mathbf{w}^t) + \mathbf{w}^t. \tag{6}$$

This attack expands the weight of the backdoored model $\mathbf{v}$ to ensure that the attack's contribution remains after averaging and transfers to the global model.

Bagdasaryan *et al.* [7] evaluated the above attack for standard federated learning tasks under different assumptions, and showed that model replacement is much better than training data poisoning. What's more, due to the success of the deep neural networks based machine learning models, most federated learning related papers also use deep networks. The phenomenon that deep networks tend to memorize training data makes them susceptible to various inference attacks [91]. Bhagoji *et al.* [10] also explored the threat of model poisoning attacks on federated learning and indicated the vulnerability of the federated learning setting. Besides, due to the differences in the number of samples used in training for different participants, the disparate vulnerability (*i.e.,* certain subgroups can be significantly more vulnerable than others) to privacy attacks on machine learning models should also be considered [125]. Thus there is an urgent need to develop effective defense strategies.

5.2 Defense (Honest-but-Curious Setting)

In this part, all users follow the protocol honestly, but the server may attempt to learn extra information in different ways [13]. The most direct way to alleviate this problem is reducing the shared parameters or gradients of each client. Shokri *et al.* [107] showed that in modern deep learning, even sharing as few as 1% gradients still results in significantly better accuracy than learning just on local data. Obviously such an approach does not solve the underlying potential threats to data privacy. To this end, there have been many efforts focus on privacy either from an individual point of view or multiparty views, especially in social media field which significantly exacerbated multiparty privacy (MP) conflicts [111, 113].

5.2.1 Secure Multi-Party Computation.

Secure multi-party computation (SMC) is a natural way to be applied to federated learning scenario, where each individual use a combination of cryptographic techniques and oblivious transfer to jointly compute a function of their private data [94]. Bonawitz *et al.* [13] design a secure Multi-Party Computation protocol for secure aggregation of high-dimensional data, where encryption technology is used to make the updates of a single device undetectable by the server and the sum is revealed only after receiving sufficient number of updates. This technique well dealt with one of the threats we talked before, *i.e.*, any participant cannot inferring anything about another participant's private data during local training process [7].

Homomorphic encryption, due to its success in Cloud Computing, comes naturally into our sight. It has certainly been used in many federated learning researches [18, 55, 85]. Homomorphic encryption is a public key system, where any party can encrypt its data with a known public key and perform calculations with data encrypted by others with the same public key [40]. Liu *et al.* [85] introduce Federated Transfer Learning (FTL) framework in a privacy-preserving setting and provide a novel approach for adapting additively homomorphic encryption to multi-party computation (MPC) with neural networks such that the accuracy is almost lossless and only minimal modifications to the neural networks is required. Chai *et al.* [18] propose a secure matrix factorization framework under the federated learning setting, where the distributed matrix factorization framework is enhanced with homomorphic encryption.

In addition to an additively homomorphic encryption scheme, Hardy*et al.* [55] also described a three-party end-to-end solution in privacy-preserving entity resolution. They provide a formal analysis of the impact of entity resolution's mistake on learning, which brings a clear and strong support for federated learning. Specifically, they proved that, under reasonable assumptions on the number and magnitude of entity resolution's mistakes, federated learning is of great value in the setting where each peer's data significantly improves the other.

Although SMC guarantee that none of the parties share anything with each other or with any third party, it can not prevent an adversary from learning some individual information, *e.g.*, whose absence might change the decision boundary of a classifier, etc. Moreover, SMC protocols are usually computationally expensive even for the simplest problems, requiring iterated encryption/decryption and repeated communication between participants about some of the encrypted results [94].

5.2.2 *Differential Privacy.*

Differential privacy (DP) is an alternative theoretical model for protecting the privacy of individual data, which has been widely applied to many areas, not only traditional algorithms, *e.g.* boosting [36], principal component analysis [19], support vector machine [102], but also deep learning research [2, 88]. Abadi *et al.* [2] firstly demonstrate the training of deep neural networks with differential privacy, incurring a modest total privacy loss, computed over entire models with many parameters. Formally, it says:

Definition 5.1 ((ϵ, δ)-Differential Privacy [35]). A randomized algorithm $\mathcal{A} : \mathcal{D} \rightarrow \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent datasets $D_1, D_2 \in \mathcal{D}$ that differ in at most one entry, and for any subset of outputs $S \subseteq \mathcal{R}$,

$$Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D_2) \in S] + \delta. \tag{7}$$

The parameter ϵ balances the accuracy of the differentially private $\mathcal{A}$ and how much it leaks [107]. The presence of a non-zero δ allows us to relax the strict relative shift in unlikely events [35]. In DP, a stochastic component (typically by additional noise) is usually added to or removed from the locally trained model. For instance, the Gaussian mechanism is defined by:

$$\mathcal{A} \triangleq f(d) + \mathcal{N}(0, \sigma^2 S_f^2), \tag{8}$$

where $\mathcal{N}(0, \sigma^2 S_f^2)$ is the Gaussian distribution with mean 0 and standard deviation σS_f.

Differential privacy ensures that the addition or removal does not substantially affect the outcome of any analysis, thus is also widely studied in federated learning research to prevent the indirect leakage. Besides reducing the shared parameters by selecting a small subset of gradients using sparse vector technique, Shokri *et al.* [107] choose to share perturbed values of the selected gradients under a consistent differentially private framework. They use the Laplacian mechanism to add noise which depends on the privacy budget as well as the sensitivity of the gradient for each parameter, and the (global) sensitivity of a function f is defined as:

$$\triangle f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|. \tag{9}$$

The global sensitivity estimates are expected significantly reduced, resulting in higher accuracy by ensuring the norm of all gradients is bounded for each update - either globally, or locally [107].

Afterwards, McMahan *et al.* [88] add client-level privacy protection to the federated averaging algorithm [86] relied heavily on privacy accounting for stochastic gradient descent [2]. But opposed to Abadi's work [2] which aims to protecting a single data point's contribution, client-level privacy means the learnt model does not reveal whether a client participated during training. Almost at the same time, Geyer *et al.* [43] propose a similar procedure for client level-DP, dynamically adapting the DP-preserving mechanism during decentralized training. Chen *et al.* [22] propose a differentially private autoencoder-based generative model (DP-AuGM) and a differentially private variational autoencoder-based generative model (DP-VaeGM). They conjectured that differential privacy is targeted to protect membership privacy while the key to defend against model inversion and GAN-based attacks is the perturbation of training data. Training global model with user-level DP usually adopts *FedSGD* and *FedAvg* with noised updates, and compute a DP guarantee using the Moments Accountant. All these processes rely on selecting a norm bound for each user's update to the model, which requires careful parameter tuning. Happily, Thakkar *et al.* [112] removed the need for extensive parameter tuning by adaptively setting the clipping norm applied to each user's update.

Instead of using Gaussian mechanism, Agarwal *et al.* [3] improve previous analysis of the Binomial mechanism showing that it achieves nearly the same utility as the Gaussian mechanism, while requiring fewer representation bits. Traditionally used local differential privacy may prove too strict in practical applications. Consequently, Bhowmick *et al.* [11] revisit the types of disclosures and adversaries against which they provide protections, and design new (minimax) optimal locally differentially private mechanisms for statistical learning problems for all privacy levels, where large privacy parameters in local differential privacy are allowed.

However, DP only protect users from data leakage to a certain extent, and may reduce performance in prediction accuracy because it is a lossy method [7, 25]. Thus, Cheng *et al.* [25] propose a lossless privacy-preserving tree-boosting framework known as SecureBoost in a federated learning setting. This framework allows learning processes to be performed jointly over multiple parties with partially common user samples but different feature sets, corresponding to a vertically partitioned virtual data set. In addition to this, Truex *et al.* [114] combines DP with SMC to reduce the growth of noise injection as the number of parties increases without sacrificing privacy while preserving provable privacy guarantees, protecting against extraction attacks and collusion threats. Besides, combining the scalability of local DP with the high utility and MP, Ghazi *et al.* [44] provides further evidence that the shuffled model of differential privacy is a fertile "middle ground" between local differential privacy and general multi-party computations.

Table 2. Summary of Papers based on Relatively More Emphasized Problem

Problems	Paper Indices
Statistical	[110] [132] [89] [79] [129] [37] [27] [34] [52] [23] [45]
Communication	[64] [72] [92] [107] [87] [72] [16] [24] [3] [99] [15] [133] [48] [5] [105] [117] [126]
Privacy & Security	[114] [3] [127] [22] [114] [42] [41] [13] [25] [107] [88] [43] [11] [44] [85] [18] [94] [55] [7] [91] [10]
Optimization	[70] [71] [123] [86] [80] [88] [79] [78] [124]
Others	[106] [101] [75] [74] [112] [73] [65] [68] [38] [120] [38] [12]

5.2.3 *Others.*

The current utility protocols for secure aggregation work in an honest-but-curious environment That is, if the server is honest and follows the protocol, then a curious adversary cannot learn any private information while observing all communication with the server. Unlike this protocol, a more robust and scalable primitive for privacy-preserving protocol is to shuffle user data to hide the origin of each data [26]. Based on it, Ghazi *et al.* [44] put forward a simple and more efficient protocol for aggregation in the shuffled model, where communication as well as error increases only polylogarithmically in the the number of users.

Fung *et al.* [42] considered that honest clients can be separated from sybils by the diversity of gradient updates. Thus they proposed FoolsGold to defense the sybil-based poisoning attacks where the learning rate of clients that provide unique gradient updates is maintained. At the same time, the learning rate of clients that repeatedly contribute similar-looking gradient updates should be reduced. Besides, Fung *et al.* [41] claimed they proposed a novel setting called brokered learning, where a short-lived, honest-but-curious broker is introduced to break the direct link between global center and local clients. This is essentially the same thing with previous federated learning works in honest-but-curious setting.

6 FEDERATED OPTIMIZATION

Many popular machine learning models have been studied in federated learning scenario, *e.g.* tensor factorization [18, 69], Bayesian [27, 74, 75, 130], Generative Adversarial Networks (GAN) [1, 54, 122]. Recall Eq. (1) and suppose we have a set of data samples $\{\mathbf{x}_i, y_j\}_{i=1}^N$, then simple examples of local machine learning models include:

- Linear regression: $f_i(\mathbf{w}) = \frac{1}{2}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2, y_i \in \mathbb{R}$
- Logistic regression: $f_i(\mathbf{w}) = -\log(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{w})), y_i \in \{-1, 1\}$
- Support vector machines: $f_i(\mathbf{w}) = \max\{0, 1 - y_i \mathbf{x}_i^\top \mathbf{w}\}, y_i \in \{-1, 1\}$

A more complex non-convex problems arise in the context of neural networks, which predict through the non-convex function of the feature vector $\mathbf{x}_i$ instead of the mapping $\mathbf{x}_i^\top \mathbf{w}$. However, the resulting loss can still be written as $f_i(\mathbf{w})$, and the gradients can be effectively calculated using back-propagation [71]. This section briefly summarize the federated optimization algorithms, and list the baseline algorithm with/without privacy concern.

6.1 Baseline Algorithms

Instead of learning separate parameters to the data for each client as multi-task learning did [110], we mainly focus on summarizing the progress on training a single global model which corresponds to the consensus solution summarized in the previous Section 3.1.

6.1.1 *Federated Averaging (FedAvg).*

Algorithm 1 Federated Averaging. The K activated clients are indexed by k, B is the local minibatch size, and η is the learning rate

[Server Executes]:
initialize $\mathbf{w}^0$
for <$t = 0, 1, \ldots, T-1$> **do**
  $Z^t \leftarrow$ random set of K clients (each device k is chosen with probability p_k);
  Server sends $\mathbf{w}^t$ to all chosen devices;
  for each client $k \in Z^t$ **in parallel do**
    $\mathbf{w}_k^{t+1} \leftarrow$ **[Client Update** $(k, \mathbf{w}^t)$**]**
  $\mathbf{w}^{t+1} \leftarrow \frac{1}{K}\sum_{k \in Z^t} \mathbf{w}_k^{t+1}$.

[Client Update $(k, \mathbf{w}^t)$**]:**
$\mathcal{B}_k \leftarrow$ (split $\mathcal{D}_k$ into patches of size B)
for <$i = 1, 2, \ldots$> **do**
  for $b \in \mathcal{B}_k$ **do**
    $\mathbf{w}_k^{t+1} \leftarrow \mathbf{w}^t - \eta \triangledown F_k(\mathbf{w}^t; b)$
return $\mathbf{w}_k^{t+1}$ to server

The naive way to solve the federated learning problem without privacy is through Federated Averaging (*FedAvg*) Algorithm [86], as shown in Alg. 1. *FedAvg* is expected to be a baseline, but it ended up working well enough. It trains high-quality models using relatively few rounds of communication. Later, Li *et al.* [80] established a convergence analysis of *FedAvg* for strongly convex and smooth problems without assuming the data are i.i.d and all the devices are active.

In particular, if the full local dataset is treated as a single mini-batch, *i.e.,* $B = \infty$ and only one epoch performed in **[Client Update]** step, the *FedAvg* algorithm degenerates into *FedSGD* [86]. Alternative, if SVRG is used as the local solver, we could further derive Federated SVRG (*FSVRG*) [70, 71].

6.1.2 *Differentially Private Version (DP-FedAvg).*

Some researchers [43, 88] further derived privacy-preserving versions of federated averaging [86]. We list the main procedures in the Algorithm 2.

6.1.3 *Variants.*

In addition to these intuitive inferences, Li *et al.* [79] developed an scalable method *q-FedAvg* inspired by fair resource allocation strategies in wireless networks, which encourages more fair accuracy distributions in federated learning. In **[Client Update]** step of *q-FedAvg*, besides the local epochs of SGD, each selected client k should also computes:

$$\triangle \mathbf{w}_k^t = \mathbf{w}^t - \mathbf{w}_k^{t+1}, \quad \triangle_k^t = F_k^q(\mathbf{w}^t)\triangle \mathbf{w}_k^t, \quad h_t^k = qF_k^{q-1}(\mathbf{w}^t)\|\triangle \mathbf{w}_k^t\|^2 + LF_k^q(\mathbf{w}^t), \tag{10}$$

where q can be tuned based on the desired amount of fairness (with larger q inducing more fairness). Then, the server aggregation correspondingly changes to:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \frac{\sum_{k \in S^t} \triangle_k^t}{\sum_{k \in S^t} h_k^t}. \tag{11}$$

FedProx is proposed to tackle statistical heterogeneity [78]. It is similar to *FedAve* and can encompass *FedAvg* as a special case. In **[Client Update]** step of *FedProx*, instead of just minimizing the local function $F_k(\cdot)$ as in *FedAvg*, the

Algorithm 2 (Client-side) Differentially Private Federated Averaging. The K activated clients are indexed by k, B is the local minibatch size, and η is the learning rate. $\{\sigma\}_{t=0}^{T}$ is the set of variances for the Gaussian mechanism (GM). ϵ defines the DP we aim for. Q is the threshold for δ, the probability that ϵ-DP is broken.

[Server Execution]:
initialize $\mathbf{w}^0$, Accountant(ϵ, K)
for $<t = 0, 1, \ldots, T-1>$ **do**
  $\delta \leftarrow$Accountant(K, σ_t)
  if $\delta > Q$ **then** return $\mathbf{w}^t$
  $Z^t \leftarrow$ random set of K clients (each device k is chosen with probability p_k);
  Server sends $\mathbf{w}^t$ to all chosen devices;
  for each client $k \in Z^t$ **in parallel do**
    $\triangle\mathbf{w}_k^{t+1}, \zeta_k \leftarrow$ **[Client Update** $(k, \mathbf{w}^t)$**]**
  S =median$\{\zeta_k\}_{k\in Z^t}$
  $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \frac{1}{|Z_t|}(\sum_{k=1}^{K} \triangle\mathbf{w}_k^{t+1}/\max(1, \frac{\zeta_k}{S}) + \mathcal{N}(0, S^2 \cdot \sigma^2))$.

[Client Update $(k, \mathbf{w}^t)$**]:**
$\mathcal{B}_k \leftarrow$ (split $\mathcal{D}_k$ into patches of size B)
for $<i = 1, 2, \ldots>$ **do**
  for $b \in \mathcal{B}_k$ **do**
    $\mathbf{w}_k^{t+1} \leftarrow \mathbf{w}^t - \eta\triangledown F_k(\mathbf{w}^t; b)$
$\triangle\mathbf{w}_k^{t+1} = \mathbf{w}_k^{t+1} - \mathbf{w}^t$
$\zeta = \|\triangle\mathbf{w}_k^{t+1}\|_2$
return $\triangle\mathbf{w}_k^{t+1}, \zeta_k$ to server

k-th client uses its local solver of choice to approximately minimize the following surrogate objective h_k:

$$\min_{\mathbf{w}} h_k(\mathbf{w}; \mathbf{w}^t) = F_k(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{w} - \mathbf{w}^t\|^2. \tag{12}$$

The proximal term in Eq.(12) effectively limits the impact of local updates (by restricting them to be close to the initial model) without manaully adjusting the number of local epochs as in *FedAvg*. To further improve flexibility and scalability, Xie *et al.* [124] proposed a asynchronous federated optimization algorithm called *FedAsync* using similar surrogate objective. Huang *et al.* [60] devised a variant of FedAvg named LoAdaBoost FedAvg that was based on the median cross-entropy loss to adaptively boost the training process of clients who appear to be weak learners.

6.2 Theoretical Progress

In this section, we will roughly survey the current theoretical progress on federated learning problem. Generally, the quality of the federated learning predictions can be measured using the notion of *regret* [31], defined as

$$R_F = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}_k) - F(\mathbf{w}^*) \tag{13}$$

where $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[F(\mathbf{w}; \mathbf{x})]$. $\mathcal{D}$ denotes the overall data distribution. R_F measures the difference between the cumulative loss of the predictions in federated environment and the cumulative loss of the fixed predictor $\mathbf{w}^*$, which is optimal with respect to the overall distribution $\mathcal{D}$.

Most theoretical papers on federated optimization usually focus on bounding the expected regret $\mathbb{E}[R_F]$. The current theoretical progress on federated learning problem is summarised in Table 3.

Table 3. Summary of Federated Optimization Algorithms

Method	Convexity	Smoothness	Assumptions	Convergence
FedAvg [86]	–	–	–	–
FedAvg [80]	Strongly Convex	Lipschitz Smooth	1, 2	$O(1/T)$
q-FedAvg [79]	–	Lipschitz Smooth	–	–
Pluralistic Averaging [37]	Convex	Lipschitz Smooth	Semi-cyclic Samples	√
Pluralistic Hedging [37]	Convex	Lipschitz Smooth	Semi-cyclic Samples	√
MOCHA [110]	Convex	Lipschitz Smooth/Continuous	Ref [110]	√
FedProx [78]	Nonconvex	Lipschitz Smooth	1, 2 and Ref [78]	√
FedAsync [124]	Weakly Convex	Lipschitz Smooth	1, 2	√
Modular [45]	Strongly Convex	Lipschitz Smooth	1, 2 and Ref [45]	√

7 APPLICATIONS

As an collaborative modeling mechanism that could carry out efficient machine learning under the premise of ensuring data privacy and legal compliance between multiple parties or multiple computing nodes, federated learning has attracted broad attention of all circles. Besides healthcare, federated learning has many other promising applications in various areas, *e.g.*, virtual keyboard prediction [21, 53, 88, 97, 128], smart retail [131], financial, vehicle-to-vehicle communication [104] and so on. In the following, we first summary some federated learning works in healthcare, then we roughly introduce federated learning works in other applications for reference.

7.1 Healthcare

Federated learning is a good way to connect all the medical institutions and makes them share their experiences with privacy guarantee. In this case, the performance of machine learning model will be significantly improved by the formed large medical data set. There have been some tasks were studied in federated learning setting in healthcare, *e.g.*, patient similarity learning [76], patient representation learning, phenotyping [69, 84], predicting future hospitalizations [14], predicting mortality and ICU stay time [59], *etc.*

Lee *et al.* [76] presented a privacy-preserving platform in a federated setting for patient similarity learning across institutions. Their model can find similar patients from one hospital to another without sharing patient-level information. Kim *et al.* [69] used tensor factorization models to convert massive electronic health records into meaningful phenotypes for data analysis in federated learning setting. Vepakomma *et al.* [118] built several configurations upon a distributed deep learning method called SplitNN [49] to facilitate the health entities collaboratively training deep learning models without sharing sensitive raw data or model details. Silva *et al.* [108] illustrated their federated learning framework by investigating brain structural relationships across diseases and clinical cohorts. Huang *et al.* [59] sought to tackle the challenge of non-IID ICU patient data that complicated decentralized learning, by clustering patients into clinically meaningful communities and optimizing performance of predicting mortality and ICU stay time. Brisimi *et al.* [14] aimed at predicting future hospitalizations for patients with heart-related diseases using EHR data spread among various data sources/agents by solving the l_1-regularized sparse Support Vector Machine classifier in federated learning environment. Liu *et al.* [84] conducted both patient representation learning and obesity comorbidity phenotyping in a federated manner and got good results.

7.2 Others

An important application of federated learning is natural language processing task. When Google first proposed federated learning concept in 2016, the application scenario is Gboard - a virtual keyboard of Google for touchscreen mobile devices with support for more than 600 language varieties [21, 53, 88, 97, 128]. Indeed, as users increasingly turn to mobile devices, fast mobile input methods with auto-correction, word completion, and next-word prediction features are becoming more and more important. For these natural language processing tasks, especially for next-word prediction, the data typed in mobile apps are usually better than the data from scanned books or transcribed utterances on aiding typing on a mobile keyboard. However, the language data are often with sensitive information, *e.g.*, the text typed on a mobile phone might including passwords, search queries or text messages. Typically, language data may identify the speaker by name or some rare phrases, and then link the speaker to confidential or sensitive information [88]. Therefore, as an innovative mechanism that could train global model from multiple parties with privacy-preserving property, federated learning has a promising application in natural language task like virtual keyboard prediction [21, 53, 88, 97, 128]. Hard *et al.* [53] trained a recurrent neural network language model in federated learning environment for the purpose of next-word prediction in a virtual keyboard for smartphones, which demonstrates the feasibility and benefits of training production-quality models for natural language understanding tasks while keeping users' data on their devices.

Besides the next word prediction on Gboard, other user cases also include search query suggestions [128], emoji prediction in a mobile keyboard [97], and learning out-of-vocabulary (OOV) words for the purpose of expanding the vocabulary of a virtual keyboard for smartphones [21]. Except for the text data, Leroy [77] investigated the use of federated learning on crowd-sourced speech data, to solve out-of-domain issues such as wake word detection. Additionally, they open source the *Hey Snips* wake word dataset to further foster transparent research in the application of federated learning to speech data. Bonawitz *et al.* [12] built a scalable production system based on TensorFlow for federated learning in the domain of mobile devices. They addresses numerous practical issues and describe the resulting high-level design.

Other applications include smart retail [131], financial, vehicle-to-vehicle communication [104] and so on. Smart retail aims to use machine learning technology to provide personalized services to customers based on some data like user purchasing power and product characteristics, including product recommendation and sales services. Zhao *et al.* [131] designed a smart system to help Internet-of-Things (IoT) device manufacturers leverage customers' data and built a machine learning model to predict customers' requirements and possible consumption behaviours in federated learning (FL) environment. They also add differential privacy to protect the privacy of customers' data. For financial applications, one example is that WeBank use federated learning principle to detect multiparty borrowing which is the pain point of financial institutions. Under the federated learning mechanism, there is no need to set up a central database, which not only protects the privacy and data integrity of existing users in various financial institutions, but also completes the inquiry of multiparty borrowing.

8 PLATFORMS

With the growth and development of federated learning, there are many companies or research teams carried out kinds of federated learning research oriented to scientific research and product development. In addition to Google's TensorFlow, another one of the most popular deep learning frameworks in the world, *i.e.,* PyTorch from Facebook, has also started to adopt the federated learning approach to achieve privacy protection. Facebook's AI research team launched a free two-month Udacity course at the same time [3]. It specifically mentions how to use federated learning in PyTorch. Particularly, the popular platforms or tools for federated learning research include:

- **PySyft**. PySyft is an open source project of OpenMined, which is mainly designed to protect the privacy of deep learning [103]. It decouples private data from model training using federated learning, DP and MPC within PyTorch. Currently, TensorFlow bindings for PySyft is also available [93].
- **TFF**. TensorFlow Federated (TFF) is also an open source framework for machine learning and other calculations on distributed data [46]. It is designed based on their experience in developing federated learning technologies at Google, and Google supports machine models for mobile keyboard retrieval and in-device search. With TFF, TensorFlow provides users with a more flexible and open framework through which they can simulate distributed computing locally.
- **FATE**. Federated AI Technology Enabler (FATE) is an open source project initiated by Webank's AI division [4]. It aims to provide a secure computing framework to support the Federated AI ecosystem, where a secure computing protocol is implemented based on homomorphic encryption and MPC. FATE supports federated learning architectures and secure computing of various machine learning algorithms, including logistic regression, tree-based algorithms, transfer learning and deep learning. Recently, Webank upgraded FATE again and launched the first visual federated learning tool - FATEBoard, as well as federated learning modeling pipeline scheduling and life cycle management tool - FATEFlow. The new version of FATE also includes partial multi-party support. In future versions, Webank's AI team will further enhance the multi-party support.
- **Tensor/IO**. Tensor/IO is a lightweight cross-platform library for on-device machine learning, bringing the power of TensorFlow and TensorFlow Lite to iOS, Android, and React native applications [32]. Tensor /IO itself does not implement any machine learning algorithms, but works with underlying libraries such as TensorFlow to simplify the process of deploying and using models on mobile phones. It runs on iOS and Android phones, with bridging for React Native. The library will interact with the specific backend you selected in the language of your choice (objective-c, Swift, Java, Kotlin, or JavaScript).
- **Functional Federated Learning in Erlang (ffl-erl).** ffl-erl is the first open-source implementation of a framework for federated learning in Erlang [115]. Erlang is a structured, dynamically typed programming language with built-in parallel computing support, which is well suited for building distributed, real-time soft parallel computing systems. The ffl-erl project has influenced an ongoing work to develop a real-world system for distributed data analysis for the automotive industry [116].

9 CONCLUSIONS AND OPEN QUESTIONS

In this survey, we have reviewed the current progress on federated learning including, but not limited to healthcare informatics. We summary the general solutions to the various challenges in federated learning. We briefly summarized the federated optimization algorithms and list the baseline algorithm with/without privacy concern. We also introduced existing federated learning platforms and hope to provide a useful resource for researchers to refer. Besides the summarized general issues in federated learning setting, we list some probably encountered directions or open questions when federated learning is applied in healthcare area in the following.

- **Data Quality**. Federated learning has the potential to connect all the isolated medical institutions, hospitals or devices to make them share their experiences with privacy guarantee. However, most health systems suffer from

[3]https://www.udacity.com/course/secure-and-private-ai–ud185

data clutter and efficiency problems. The quality of data collected from multiple sources is uneven and there is no uniform data standard. The analyzed results are apparently worthless when dirty data are accidentally used as samples. The ability to strategically leverage medical data is critical. Therefore, how to clean, correct and complete data and accordingly ensure data quality is a key to improve the machine learning model weather we are dealing with federated learning scenario or not.

- **Incorporating Expert Knowledge**. In 2016, IBM introduced Watson for Oncology, a tool that uses the natural language processing system to summarize patients' electronic health records and search the powerful database behind it to advise doctors on treatments. Unfortunately, some oncologists say they trust their judgment more than Watson tells them what needs to be done [4]. Therefore, hopefully doctors will be involved in the training process. Since every data set collected here cannot be of high quality, so it will be very helpful if the standards of evidence-based machine is introduced, doctors will also see the diagnostic criteria of artificial intelligence. If wrong, doctors will give further guidance to artificial intelligence to improve the accuracy of machine learning model during training process."
- **Incentive Mechanisms**. With the internet of things and the variety of third party portals, a growing number of smartphone healthcare apps are compatible with wearable devices. In addition to data accumulated in hospitals or medical centers, another type of data that is of great value is coming from wearable devices not only to the researchers, but more importantly for the owners. However, during federated model training process, the clients suffer from considerable overhead in communication and computation. Without well-designed incentives, self-interested mobile or other wearable devices will be reluctant to participate in federal learning tasks, which will hinder the adoption of federated learning [65]. How to design an efficient incentive mechanism to attract devices with high-quality data to join federated learning is another important problem.
- **Personalization**. Wearable devices are more focus on public health, which means helping people who are already healthy to improve their health, such as helping them exercise, practice meditation and improve their sleep quality. How to assist patients to carry out scientifically designed personalized health management, correct the functional pathological state by examining indicators, and interrupt the pathological change process are very important. Reasonable chronic disease management can avoid emergency visits and hospitalization and reduce the number of visits. Cost and labor savings. Although there are some general work about federated learning personalization [63, 109], for healthcare informatics, how to combining the medical domain knowledge and make the global model be personalized for every medical institutions or wearable devices is another open question.
- **Model Precision**. Federated tries to make isolated institutions or devices share their experiences, and the performance of machine learning model will be significantly improved by the formed large medical dataset. However, the prediction task is currently restricted and relatively simple. Medical treatment itself is a very professional and accurate field. Medical devices in hospitals have incomparable advantages over wearable devices. And the models of Doc.ai could predict the phenome collection of one's biometric data based on its selfie, such as height, weight, age, sex and BMI[5]. How to improve the prediction model to predict future health conditions is definitely worth exploring.

[4]http://news.moore.ren/industry/158978.htm
[5]https://doc.ai/blog/do-you-know-how-valuable-your-medical-da/

REFERENCES

[1] Rajagopal A. and Nirmala V. Federated AI lets a team imagine together: Federated learning of gans. *CoRR*, abs/1906.03595, 2019.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[3] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[4] Webank's AI. Federated ai technology enabler. https://www.fedai.org/cn/, 2019.

[5] Tran The Anh, Nguyen Cong Luong, Dusit Niyato, Dong In Kim, and Li-Chun Wang. Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach. *IEEE Wireless Communications Letters*, 2019.

[6] IEEE Standard Association. P3652.1 - guide for architectural framework and application of federated machine learning. https://standards.ieee.org/project/3652_1.html, 2018.

[7] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.

[8] Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, pages 26–1, 2012.

[9] Carla Barcelos, João Gluz, and Rosa Vicari. An agent-based federated learning object search service. *Interdisciplinary journal of e-learning and learning objects*, 7(1):37–54, 2011.

[10] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. *arXiv preprint arXiv:1811.12470*, 2018.

[11] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

[12] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[13] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. ACM, 2017.

[14] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

[15] Thang D Bui, Cuong V Nguyen, Siddharth Swaroop, and Richard E Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018.

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

[17] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[18] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. 06 2019.

[19] Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *The Journal of Machine Learning Research*, 14(1):2905–2943, 2013.

[20] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876*, 2018.

[21] Mingqing Chen, Rajiv Mathews, Tom Ouyang, and Francoise Beaufays. Federated learning of out-of-vocabulary words. *arXiv preprint arXiv:1903.10635*, 2019.

[22] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models. *arXiv preprint arXiv:1812.02274*, 2018.

[23] Xiangyi Chen, Tiancong Chen, Haoran Sun, Zhiwei Steven Wu, and Mingyi Hong. Distributed training with heterogeneous data: Bridging median and mean based algorithms. *arXiv preprint arXiv:1906.01736*, 2019.

[24] Yang Chen, Xiaoyan Sun, and Yaochu Jin. Communication-efficient federated deep learning with asynchronous model update and temporally weighted aggregation. *arXiv preprint arXiv:1903.07424*, 2019.

[25] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. Secureboost: A lossless federated learning framework. *arXiv preprint arXiv:1901.08755*, 2019.

[26] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

[27] Luca Corinzia and Joachim M Buhmann. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2019.

[28] Wenrui Dai, Shuang Wang, Hongkai Xiong, and Xiaoqian Jiang. Privacy preserving federated big data analysis. In *Guide to Big Data Applications*, pages 49–82. Springer, 2018.

[29] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*, 6(1):54, 2019.

[30] Christopher De Sa, Megan Leszczynski, Jian Zhang, Alana Marzoev, Christopher R Aberger, Kunle Olukotun, and Christopher Ré. High-accuracy low-precision training. *arXiv preprint arXiv:1803.03383*, 2018.

[31] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

[32] doc.ai. Declarative, on-device machine learning for ios, android, and react native. https://github.com/doc-ai/tensorio, 2019.

[33] Sumeet Dua, U Rajendra Acharya, and Prerna Dua. *Machine learning in healthcare informatics*, volume 56. Springer, 2014.

[34] Moming Duan. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. *arXiv preprint arXiv:1907.01132*, 2019.

[35] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[36] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[37] Hubert Eichner, Tomer Koren, H Brendan McMahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic stochastic gradient descent. *arXiv preprint arXiv:1904.10120*, 2019.

[38] Shaohan Feng, Dusit Niyato, Ping Wang, Dong In Kim, and Ying-Chang Liang. Joint service pricing and cooperative relay communication for federated learning. *arXiv preprint arXiv:1811.12082*, 2018.

[39] Olivier Fercoq, Zheng Qu, Peter Richtárik, and Martin Takáč. Fast distributed coordinate descent for non-strongly convex losses. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2014.

[40] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007:15, 2007.

[41] Clement Fung, Jamie Koerner, Stewart Grant, and Ivan Beschastnikh. Dancing in the dark: private multi-party machine learning in an untrusted setting. *arXiv preprint arXiv:1811.09712*, 2018.

[42] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

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

[44] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *arXiv preprint arXiv:1906.08320*, 2019.

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

[46] Google. Tensorflow federated. https://www.tensorflow.org/federated, 2019.

[47] Peter Groves, Basel Kayyali, David Knott, and Steve Van Kuiken. The'big data'revolution in healthcare: Accelerating value and innovation. 2016.

[48] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[49] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[50] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[51] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[52] Yufei Han and Xiangliang Zhang. Robust federated training via collaborative machine teaching using trusted instances. *arXiv preprint arXiv:1905.02941*, 2019.

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

[54] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. *arXiv preprint arXiv:1811.03850*, 2018.

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

[56] Chaoyang He, Conghui Tan, Hanlin Tang, Shuang Qiu, and Ji Liu. Central server free federated learning over single-sided trust social networks. *arXiv preprint arXiv:1910.04956*, 2019.

[57] Patrick Hill. The rationale for learning communities and learning community models. 1985.

[58] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[59] Li Huang and Dianbo Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *arXiv preprint arXiv:1903.09296*, 2019.

[60] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. Loadaboost: Loss-based adaboost federated machine learning on medical data. *arXiv preprint arXiv:1811.12629*, 2018.

[61] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in neural information processing systems*, pages 4107–4115, 2016.

[62] Selim Ickin, Konstantinos Vandikas, and Markus Fiedler. Privacy preserving qoe modeling using collaborative learning. *arXiv preprint arXiv:1906.09248*, 2019.

[63] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488v1*, 2019.

[64] Michael Kamp, Linara Adilova, Joachim Sicking, Fabian Hüger, Peter Schlicht, Tim Wirtz, and Stefan Wrobel. Efficient decentralized deep learning by dynamic model averaging. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 393–409. Springer, 2018.

[65] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. Incentive design for efficient federated learning in mobile networks: A contract theory approach. *arXiv preprint arXiv:1905.07479*, 2019.

[66] Karen Kellogg. Learning communities. eric digest. 1999.

[67] Mikhail Khodak, Maria Florina-Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

[68] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, 2019.

[69] Yejin Kim, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Federated tensor factorization for computational phenotyping. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 887–895. ACM, 2017.

[70] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[71] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[72] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[73] Antti Koskela and Antti Honkela. Learning rate adaptation for federated and differentially private learning. *arXiv preprint arXiv:1809.03832v3*, 2019.

[74] Anusha Lalitha, Osman Cihan Kilinc, Tara Javidi, and Farinaz Koushanfar. Peer-to-peer federated learning on graphs. *rXiv preprint arXiv:1901.11173*, 2019.

[75] Anusha Lalitha, Xinghan Wang, Osman Kilinc, Yongxi Lu, Tara Javidi, and Farinaz Koushanfar. Decentralized bayesian learning over graphs. page arXiv preprint arXiv:1905.10466, 2019.

[76] Junghye Lee, Jimeng Sun, Fei Wang, Shuang Wang, Chi-Hyuck Jun, and Xiaoqian Jiang. Privacy-preserving patient similarity learning in a federated environment: development and analysis. *JMIR medical informatics*, 6(2):e20, 2018.

[77] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6341–6345. IEEE, 2019.

[78] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith1. Federated optimization for heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2019.

[79] Tian Li, Maziar Sanjabi, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[80] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[81] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

[82] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *Advances in Neural Information Processing Systems*, pages 345–353, 2017.

[83] Boyi Liu, Lujia Wang, Ming Liu, and Chengzhong Xu. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. *arXiv preprint arXiv:1901.06455*, 2019.

[84] Dianbo Liu, Dmitriy Dligach, and Timothy Miller. Two-stage federated phenotyping and patient representation learning. *arXiv preprint arXiv:1908.05596*, 2019.

[85] Yang Liu, Tianjian Chen, and Qiang Yang. Secure federated transfer learning, 2018.

[86] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[87] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[88] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[89] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[90] Rajat Mukherjee and Howard Jaffe. System and method for dynamic context-sensitive federated search of multiple information repositories, July 7 2005. US Patent App. 10/743,196.

[91] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. *arXiv preprint arXiv:1812.00910*, 2018.

[92] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. *arXiv preprint arXiv:1804.08333*, 2018.

[93] OpenMined. Pysyft-tensorflow. https://github.com/OpenMined/PySyft-TensorFlow, 2019.

[94] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems*, pages 1876–1884, 2010.

[95] Jia Qian, Sayantan Sengupta, and Lars Kai Hansen. Active learning solution on distributed edge computing. *arXiv preprint arXiv:1906.10718*, 2019.

[96] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3, 2014.

[97] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

[98] Dan Rehak, Philip Dodds, and Larry Lannom. A model and infrastructure for federated learning content repositories. In *Interoperability of Web-Based Educational Systems Workshop*, volume 143. Citeseer, 2005.

[99] Jinke Ren, Guanding Yu, and Guangyao Ding. Accelerating dnn training in wireless federated edge learning system. *arXiv preprint arXiv:1905.09712*, 2019.

[100] Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *The Journal of Machine Learning Research*, 17(1):2657–2681, 2016.

[101] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731*, 2019.

[102] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *arXiv preprint arXiv:0911.5708*, 2009.

[103] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

[104] Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and Merouane Debbah. Federated learning for ultra-reliable low-latency v2v communications. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[105] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *arXiv preprint arXiv:1903.02891*, 2019.

[106] Muhammad Shayan, Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Biscotti: A ledger for private and secure peer-to-peer machine learning. *arXiv preprint arXiv:1811.09904*, 2018.

[107] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[108] Santiago Silva, Boris Gutman, Eduardo Romero, Paul M Thompson, Andre Altmann, and Marco Lorenzi. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. *arXiv preprint arXiv:1810.08553*, 2018.

[109] Khe Chai Sim, Petr Zadrazil, and Françoise Beaufays. An investigation into on-device personalization of end-to-end automatic speech recognition models. *arXiv preprint arXiv:1909.06678*, 2019.

[110] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[111] Jose M Such and Natalia Criado. Multiparty privacy in social media. *Commun. ACM*, 61(8):74–81, 2018.

[112] Om Thakkar, Galen Andrew, and H Brendan McMahan. Differentially private learning with adaptive clipping. *arXiv preprint arXiv:1905.03871*, 2019.

[113] Kurt Thomas, Chris Grier, and David M Nicol. unfriendly: Multi-party privacy risks in social networks. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 236–252. Springer, 2010.

[114] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, and Rui Zhang. A hybrid approach to privacy-preserving federated learning. *arXiv preprint arXiv:1812.03224*, 2018.

[115] Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. Functional federated learning in erlang (ffl-erl). In *International Workshop on Functional and Constraint Logic Programming*, pages 162–178. Springer, 2018.

[116] Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. Oodida: On-board/off-board distributed data analytics for connected vehicles. *arXiv preprint arXiv:1902.00319*, 2019.

[117] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[118] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[119] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[120] Guan Wang. Interpret federated learning with shapley values. *arXiv preprint arXiv:1905.04519*, 2019.

[121] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *arXiv preprint arXiv:1809.07857*, 2018.

[122] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

[123] Blake E Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 8496–8506, 2018.

[124] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv::1903.03934*, 2019.

[125] Mohammad Yaghini, Bogdan Kulynych, and Carmela Troncoso. Disparate vulnerability: on the unfairness of privacy attacks against machine learning. *arXiv preprint arXiv:1906.00389*, 2019.

[126] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. Federated learning via over-the-air computation. *arXiv preprint arXiv:1812.11750*, 2018.

[127] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, January 2019.

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

[129] Naoya Yoshida, Takayuki Nishio, Masahiro Morikura, Koji Yamamoto, and Ryo Yonetani. Hybrid-fl: Cooperative learning mechanism using non-iid data in wireless networks. *arXiv preprint arXiv:1905.07210*, 2019.

[130] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. *arXiv preprint arXiv:1905.12022*, 2019.

[131] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, and Dusit Niyato. Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system. *arXiv preprint arXiv:1906.10893*, 2019.

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

[133] Hangyu Zhu and Yaochu Jin. Multi-objective evolutionary federated learning. *IEEE transactions on neural networks and learning systems*, 2019.

[134] Hankz Hankui Zhuo, Wenfeng Feng, Qian Xu, Qiang Yang, and Yufeng Lin. Federated reinforcement learning. *rXiv preprint arXiv:1901.08277*, 2019.

A PRELIMINARIES

We recall some standard definitions and assumptions for stochastic optimization, with some specific assumptions adopted in individual federated learning studies.

DEFINITION 1. *(L-Lipschitz) A function f is L-Lipschitz if for any $\mathbf{x}, \mathbf{y}$ in its domain,*

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|. \tag{14}$$

REMARK 1. *If a function is L-Lipschitz then its dual will be L-bounded, i.e., for any $\mathbf{w}$ such that $\|\mathbf{w}\|_2 > L$, then $f^*(\mathbf{w}) = +\infty$*

DEFINITION 2. *(ρ-smooth) A differentiable function f is ρ-smooth if for $\forall \mathbf{x}, \mathbf{y}$,*

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \triangledown f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\rho}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{15}$$

DEFINITION 3. *(Convex) A differentiable function f is convex if for $\forall \mathbf{x}, \mathbf{y}$,*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \triangledown f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle. \tag{16}$$

DEFINITION 4. *(μ-strongly convex) A differentiable function f is μ-strongly convex with positive coefficient μ if for $\forall \mathbf{x}, \mathbf{y}$,*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \triangledown f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{17}$$

DEFINITION 5. *(μ-weakly convex) A differentiable function f is μ-weakly convex if the function g with $g(\mathbf{x}) = f(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{x}\|^2$ is convex, where $\mu \geq 0$.*

REMARK 2. *Note that when f is μ-weakly convex, then f is convex if $\mu = 0$, and potentially non-convex if $\mu > 0$.*

Assumption 1. *(Bounded Second Moment)*

$$\mathbb{E}_k[\|\nabla F_k(\mathbf{w})\|^2] \leq G^2, \quad \mathbb{E}_k[\|\nabla f_i(\mathbf{w})\|^2] \leq B^2, \quad \forall \mathbf{w}, k, i. \tag{18}$$

Assumption 2. *(Bounded Gradient Variance)*

$$\mathbb{E}_k[\|\nabla F_k(\mathbf{w}) - \nabla f(\mathbf{w})\|^2] \leq \sigma^2, \forall \mathbf{w}, k. \tag{19}$$

Assumption 3. *(Bounded Dissimilarity [78]). For some $\epsilon > 0$, where exists a B_ϵ such that for all the points $\mathbf{w} \in \mathcal{S}_\epsilon = \{\mathbf{w} | \|\nabla f(\mathbf{w})\|^2 > \epsilon$, we have*

$$B(\mathbf{w}) \leq B_\epsilon, \tag{20}$$

where $B(\mathbf{w}) = \sqrt{\frac{\mathbb{E}_k[\|\nabla F_k(\mathbf{w})\|^2]}{\|\nabla f(\mathbf{w})\|^2 >}}$ for $\|\nabla f(\mathbf{w})\| \neq 0$.

Assumption 4. *In [45], central server cluster $\{\mathbf{w}_k\}_{k=1}^K$ to obtain $C_1, ..., C_m$. $\{\mathbf{w}^*_{(i)}\}_{i=1}^m$ are separated:*

$$\min_{i \neq j} \|\mathbf{w}^*_{(i)} - \mathbf{w}^*_{(j)}\| \geq R \text{ and } n \geq \frac{L^2 G \log m}{\lambda^3}. \tag{21}$$

where L and λ represent that $f(\mathbf{w})$ is L Lipschitz and λ-strongly convex.