# A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection

Qinbin Li[1], Zeyi Wen[2], Zhaomin Wu[1], Sixu Hu[1], Naibo Wang[3], Bingsheng He[1]

[1,3]National University of Singapore
[2]The University of Western Australia
[1]{qinbin, zhaomin, sixuhu, hebs}@comp.nus.edu.sg,
[2]zeyi.wen@uwa.edu.au,
[3]naibowang.vip@gmail.com



**Abstract**

Federated learning has been a hot research topic in enabling the collaborative training of machine learning models among different organizations under the privacy restrictions. As researchers try to support more machine learning models with different privacy-preserving approaches, there is a requirement in developing systems and infrastructures to ease the development of various federated learning algorithms. Similar to deep learning systems such as PyTorch and TensorFlow that boost the development of deep learning, federated learning systems (FLSs) are equivalently important, and face challenges from various aspects such as effectiveness, efficiency, and privacy. In this survey, we conduct a comprehensive review on federated learning systems. To achieve smooth flow and guide future research, we introduce the definition of federated learning systems and analyze the system components. Moreover, we provide a thorough categorization for federated learning systems according to six different aspects, including data distribution, machine learning model, privacy mechanism, communication architecture, scale of federation and motivation of federation. The categorization can help the design of federated learning systems as shown in our case studies. By systematically summarizing the existing federated learning systems, we present the design factors, case studies, and future research opportunities.

## 1 Introduction

Many machine learning algorithms are data hungry, and in reality, data are dispersed over different organizations under the protection of privacy restrictions. Due to these factors, federated learning (FL) [159] has become a hot research topic in machine learning. For example, data of different hospitals are isolated and become "data islands". Since each data island has limitations in size and approximating real distributions, a single hospital may not be able to train a high quality model that has a good predictive accuracy for a specific task. Ideally, hospitals can benefit more if they can collaboratively train a machine learning model with the union of their data. However, the data cannot simply be shared among the hospitals due to various policies and regulations. Such phenomena on "data islands" are commonly seen in many areas such as finance, government, and supply chains. Policies such as General Data Protection Regulation (GDPR) [11] stipulate rules on data sharing among different organizations. Thus, it is challenging to develop a federated learning system which has a good predictive accuracy while obeying policies and regulations to protect privacy.

Many efforts have recently been devoted to implementing federated learning algorithms to support effective machine learning models. Specifically, researchers try to support more machine learning models with different privacy-preserving approaches, including deep neural networks (NNs) [114, 205, 25, 148, 122], gradient boosted decision trees (GBDTs) [208, 44, 103], logistics regression [134, 41] and support vector machines (SVMs) [162]. For instance, Nikolaenko et al. [134] and Chen et al. [41] propose

approaches to conduct FL based on linear regression. Hardy et al. [77] implement an FL framework to train a logistic regression model. Since GBDTs have become very successful in recent years [38, 190], the corresponding Federated Learning Systems (FLSs) have also been proposed by Zhao et al. [208], Cheng et al. [44], Li et al. [103]. Another popular ensemble method of decision trees, i.e., random forests, has also been extended to support privacy-preserving [144], which is an important step towards supporting FL. Moreover, there are many neural network based FLSs. Google proposes a scalable production system which enables tens of millions of devices to train a deep neural network [25]. Yurochkin et al. [205] develop a probabilistic FL framework for neural networks by applying Bayesian nonparametric machinery. Several methods try to combine FL with machine learning techniques such as multi-task learning and transfer learning. Smith et al. [162] combine FL with multi-task learning to allow multiple parties to complete separate tasks. To address the scenario where the label information only exists in one party, Yang et al. [197] adopt transfer learning to collaboratively learn a model.

Among the studies on customizing machine learning algorithms under the federated context, we have identified a few commonly used methods and approaches. Take the methods to provide privacy guarantees as an example. One common method is to use cryptographic techniques [24] such as secure multi-party computation [126] and homomorphic encryption [77]. The other popular method is differential privacy [208], which adds noises to the model parameters to protect the individual record. For example, Google's FLS [24] adopts both secure aggregation and differential privacy to enhance privacy protection.

As there are common methods and building blocks for building FL algorithms, it makes sense to develop systems and infrastructures to ease the development of various FL algorithms. Systems and infrastructures allow algorithm developers to reuse the common building blocks, and avoid building algorithms every time from scratch. Similar to deep learning systems such as PyTorch [140, 141] and TensorFlow [8] that boost the development of deep learning algorithms, FLSs are equivalently important for the success of FL. However, building a successful FLS is challenging, which needs to consider multiple aspects such as effectiveness, efficiency, privacy, and autonomy.

In this paper, we take a survey on the existing FLSs from a system view. First, we show the definition of FLSs, and compare it with conventional federated systems. Second, we analyze the system components of FLSs, including the parties, the manager, and the computation-communication framework. Third, we categorize FLSs based on six different aspects: data distribution, machine learning model, privacy mechanism, communication architecture, scale of federation, and motivation of federation. These aspects can direct the design of an FLS as common building blocks and system abstractions. Fourth, based on these aspects, we systematically summarize the existing studies, which can be used to direct the design of FLSs. Last, to make FL more practical and powerful, we present future research directions to work on. We believe that systems and infrastructures are essential for the success of FL. More work has to be carried out to address the system research issues in effectiveness, efficiency, privacy, and autonomy.

## 1.1 Related Surveys

There have been several surveys on FL. A seminal survey written by Yang et al. [197] introduces the basics and concepts in FL, and further proposes a comprehensive secure FL framework. Later, WeBank [187] has published a white paper in introducing the background and related work in FL and most importantly presented a development roadmap including establishing local and global standards, building use cases and forming industrial data alliance. The paper mainly target at a relatively small number of parties which are typically enterprise data owners. Lim et al. [109] conduct a survey of FL specific to mobile edge computing. Li et al. [105] summarize challenges and future directions of FL in massive networks of mobile and edge devices. Recently, Kairouz et al. [85] has a comprehensive description about the characteristics and challenges on FL from different research topics. However, they mainly focus on cross-device FL, where the participants are a very large number of mobile or IoT devices.

### 1.2 Our Contribution

To the best of our knowledge, there lacks a survey on reviewing existing systems and infrastructure of FLSs and on boosting the attention of creating systems for FL (Similar to prosperous system research in deep learning). In comparison with the previous surveys, the main contributions of this paper are as follows. (1) Our survey is the first one to provide a comprehensive analysis on FL from a system's point of view, including system components, taxonomy, summary, design, and vision. (2) We provide a comprehensive taxonomy against FLSs on six different aspects, including data distribution, machine learning model, privacy mechanism, communication architecture, scale of federation, and motivation of federation, which can be as common building blocks and system abstractions of FLSs. (3) We summarize existing typical and state-of-the-art studies according to their domains, which is convenient for researchers and developers to refer to. (4) We present the design factors for a successful FLS and comprehensively review solutions for each scenario. (5) We propose interesting research directions and challenges for future generations of FLSs.

The rest of the paper is organized as follows. In Section 2, we introduce the concept of the FLS and compare it with conventional federated systems. In Section 3, we present the system components of FLSs. In Section 4, we propose six aspects to classify FLSs. In Section 5, we summary existing studies and systems on FL. We then present the design factors and solutions for an FLS in Section 6. Next, in Section 7, we show two case studies directed by our system characteristics. Last, we propose possible future directions on FL in Section 8 and conclude our paper in Section 9.

## 2 An Overview of Federated Learning Systems

### 2.1 Background

In the recent years, data breaches has seriously threatened users' data privacy. In a major breach in 2019, over 540 million records about Facebook users are exposed on Amazon's cloud [3]. In 2019, U.S. customs and border protection announced a data leakage in which dozens of thousands of photos of travelers were compromised [6].

As the data breach becomes a major concern, more and more governments establish regulations to protect users' data, such as GDPR in European Union [178], PDPA in Singapore [45], and CCPA [1] in the US. The cost of breaching these policies is pretty high for companies. In a breach of 600,000 drivers' personal information in 2016, Uber had to pay $148 million to settle the investigation [2]. SingHealth was fined $750,000 by the Singapore government for a breach of PDPA [5]. Google was fined $57 million for a breach of GDPR [4], which is the largest penalty as of March 18, 2020 under the European Union privacy law.

### 2.2 Definition

Under the above circumstances, federated learning, a collaborative learning without exchanging users' original data, has drawn increasingly attention nowadays. While machine learning, especially deep learning, has attracted many attentions again recently, the combination of federation and machine learning is emerging as a new and hot research topic. FL enables multiple parties jointly train a machine learning model without exchanging the local data. It covers the techniques from multiple research areas such as distributed system, machine learning, and privacy. Here we give a formal definition of FLSs.

We assume that there are $N$ different parties, and each party is denoted by $T_i$, where $i \in [1, N]$. We use $D_i$ to denote the data of $T_i$. For the non-federated setting, each party $T_i$ uses only its local data $D_i$ to train a machine learning model $M_i$. The predictive accuracy of $M_i$ is denoted as $P_i$. For the federated setting, all the parties jointly train a model $\hat{M}_f$ while each party $T_i$ protects its data $D_i$ according to its specific privacy restrictions. The predictive accuracy of $\hat{M}_f$ is denoted as $\hat{P}_f$. Then, for a valid FLS, there exists $i \in [1, N]$ such that $\hat{P}_f > P_i$.

Note that, in the above definition, we only require that there exists any party that can achieve a higher model quality from FL. Even though some parties may not get a better model from FL, they may still join the federation and make an agreement with the other parties to ask for the other kinds of incentives (e.g., money).

## 2.3 Compare with Conventional Federated Systems

The concept of federation can be found with its counterparts in the real world such as business and sports. The main characteristic of federation is cooperation. Federation not only commonly appears in society, but also plays an important role in computing. In computer science, federated computing systems have been an attractive area of research under different contexts.

Around 1990, there were many studies on federated database systems (FDBSs) [158]. An FDBS is a collection of autonomous databases cooperating for mutual benefits. As pointed out in a previous study [158], three important components of an FDBS are autonomy, heterogeneity, and distribution.

- *Autonomy*. A database system (DBS) that participates in an FDBS is autonomous, which means it is under separate and independent control. The parties can still manage the data without the FDBS.
- *Heterogeneiry*. The database management systems can be different inside an FDBS. For example, the difference can lie in the data structures, query languages, system software requirements, and communication capabilities.
- *Distribution*. Due to the existence of multiple DBSs before an FDBS is built, the data distribution may differ in different DBSs. A data record can be horizontally or vertically partitioned into different DBSs, and can also be duplicated in multiple DBSs to increase the reliability.

More recently, with the development of cloud computing, many studies have been done for federated cloud computing [97]. A federated cloud (FC) is the deployment and management of multiple external and internal cloud computing services. The concept of cloud federation enables further reduction of costs due to partial outsourcing to more cost-efficient regions. Resource migration and resource redundancy are two basic features of federated clouds [97]. First, resources may be transferred from one cloud provider to another. Migration enables the relocation of resources. Second, redundancy allows concurrent usage of similar service features in different domains. For example, the data can be partitioned and processed at different providers following the same computation logic. Overall, the scheduling of different resources is a key factor in the design of a federated cloud system.

**Observations on existing federated systems.** There are some similarities and differences between FLSs and conventional federated systems. On the one hand, the concept of federation still applies. The common and basic idea is about the cooperation of multiple independent parties. Therefore, the perspective of considering heterogeneity and autonomy among the parties can still be applied to FLSs. Furthermore, some factors in the design of distributed systems are still important for FLSs. For example, how the data are shared between the parties can influence the efficiency of the systems. On the other hand, these federated systems have different emphasis on collaboration and constraints. While FDBSs focus on the management of distributed data and FCs focus on the scheduling of the resources, FLSs care more about the secure computation among multiple parties. FLSs induce new challenges such as the algorithm designs of the distributed training and the data protection under the privacy restrictions.

Figure 1 shows the number of papers in each year for these three research areas. Here we count the papers by searching keywords “federated database”, “federated cloud”, and “federated learning” in Google Scholar[1]. Although federated database was proposed 30 years ago, there are still about 400 papers that mentioned it in recent years. The popularity of federated cloud grows more quickly than federated database at the beginning, while it appears to decrease in recent years probably because cloud computing becomes more mature and the incentives of federation diminish. For FL, the number of related papers is increasing rapidly and has achieved 1,200 last year. Nowadays, the “data island” phenomena are common

[1] https://scholar.google.com/

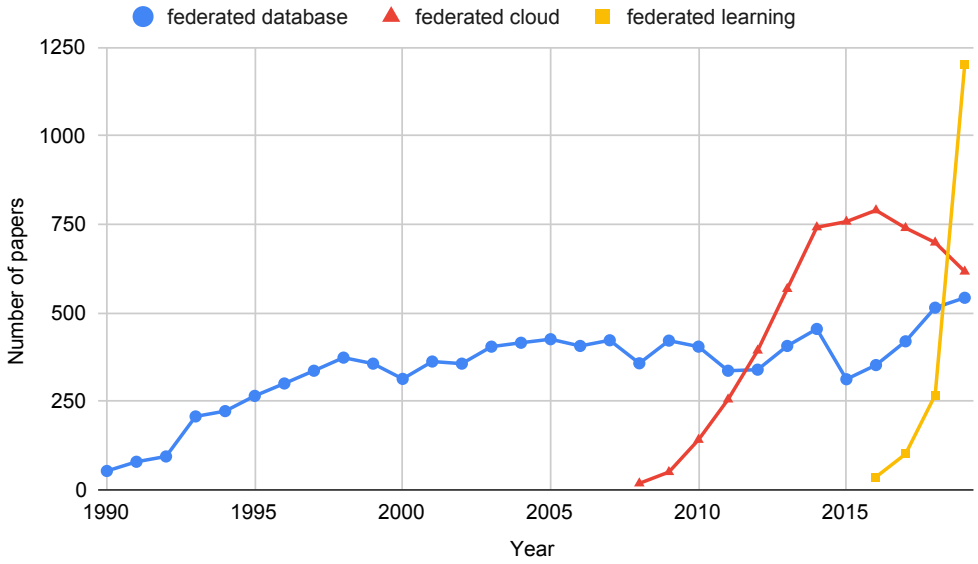



Figure 1: The number of related papers on "federated database", "federated cloud", and "federated learning"

and have increasingly become an important issue in machine learning. Also, there is a increasing privacy concern and social awareness from the general public. Thus, we expect the popularity of FL will keep increasing for at least five years until there may be mature FLSs.

# 3 System Components

There are three major components in an FLS: parties (e.g., clients), the manager (e.g., server), and the communication-computation framework to train the machine learning model.

## 3.1 Parties

In FLSs, the parties are the data owners and also the beneficiaries of FL. They can be organizations or mobile devices, named cross-silo or cross-device settings [85], respectively. We can consider the following properties of the parties that may influence the design of FLSs.

First, what is the hardware capacity of the parties? The hardware capacity includes the computation power and the storage. If the parties are mobile phones, the capacity is weak and the parties cannot perform much computation and train a huge model. For example, Wang et al. [184] consider a resource constrained setting in FL. They design an objective to include the resource budgets and proposed an algorithm to determine the rounds of local updates.

Second, what is the scale and stability of the parties? For organizations, the scale is relative small compared with the mobile devices. Also, the stability of the cross-silo setting is better than the cross-device setting. Thus, in the cross-silo setting, we can except that every party can continuously conduct computation and communication tasks in the entire federated process, which is a common setting in many studies [103, 44, 162]. If the parties are mobile devices, the system has to handle possible issues such as connection lost [25]. Moreover, since the number of devices can be huge (e.g., millions), it is unpractical to assume all the devices to participate every round in FL. The widely used setting is to choose a fraction of devices to perform computation in each round [122, 25].

Last, what is the data distribution among the parties? Usually, no matter cross-device or cross-silo setting, the non-IID (identically and independently distributed) data distribution is considered to be a practical and challenging setting in federated learning [85], which is evaluated in the experiments of recent work [103, 205, 108, 182]. Such non-IID data distribution may be more obvious among the organizations.

For example, a bank and an insurance company can conduct FL to improve their predictions (e.g., whether a person can repay the loan and whether the person will buy the insurance products), while even the features can vary a lot in these organizations. Techniques in transfer learning [139], meta-learning [59], and multi-task learning [147] may be useful to combine the knowledge of various kinds of parties.

### 3.2 Manager

In the cross-device setting, the manager is usually a powerful central server. It conducts the training of the global machine learning model and manages the communication between the parties and the server. The stability and reliability of the server are quite important. Once the server fails to provide the accurate computation results, the FLS may produce a bad model. To address these potential issues, blockchain [168] may be a possible technique to offer a decentralized solution in order to increase the system reliability. For example, Kim et al. [93] leverage blockchain in lieu of the central server in their system, where the blockchain enables exchanging the devices' updates and providing rewards to them.

In the cross-silo setting, since the organizations are expected to have powerful machines, the manager can also be one of the organizations who dominates the FL process. This is particularly used in the vertical FL [197], which we will introduce in Section 4.1 in detail. In a vertical FL setting by Liu et al. [114], the features of data are vertically partitioned across the parties and only one party have the labels. The party that owns the labels is naturally considered as the FL manager.

One problem can be that it is hard to find a trusted server or party as the manager, especially in the cross-silo setting. Then, a fully-decentralized setting can be a good choice, where the parties communicate with each other directly and almost equally contribute to the global machine learning model training. Here the manager is actually all the parties. These parties jointly set a FL task and deploy the FLS. Li et al. [103] proposed a federated gradient boosting decision trees framework, where each party trains decision trees sequentially and the final model is the combination of all the trees. It is challenging to design a fully-decentralized FLS with reasonable communication overhead.

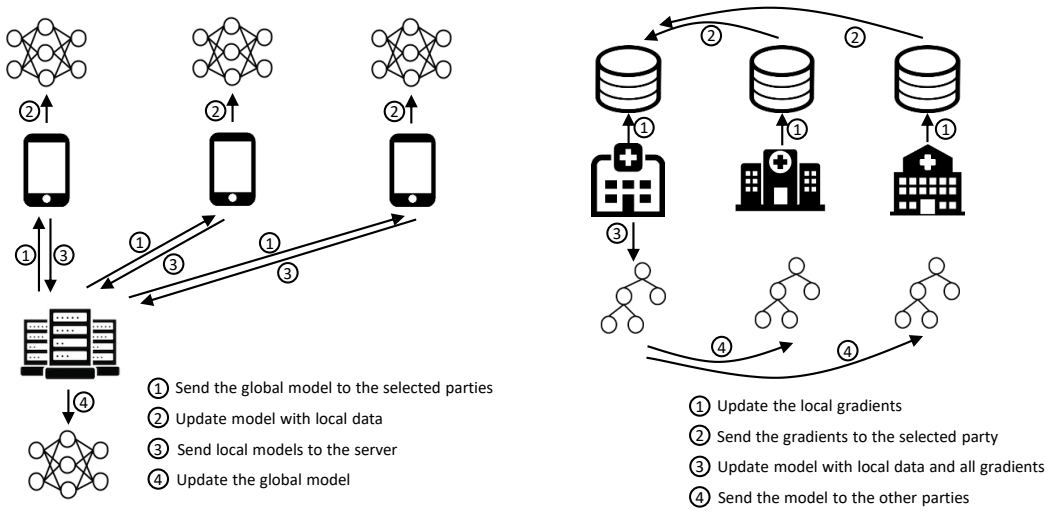
### 3.3 Communication-Computation Framework

In FLSs, the computation happens on the parties and the manager, while the communication happens between the parties and the manager. Usually, the aim of the computation is for the model training and the aim of the communication is for exchanging the model parameters.

A basic and widely used framework is Federated Averaging (FedAvg) [122] proposed in 2016, as shown in Figure 2a. In each iteration, the server first sends the current global model to the selected parties. Then, the selected parties update the global model with their local data. Next, the updated models are sent back to the server. Last, the server averages all the received local models to get a new global model. FedAvg repeats the above process until reaches the specified number of iterations. The global model of the server is the final output.

While FedAvg is a centralized FL framework, SimFL, proposed by Li et al. [105], represents a decentralized FL framework. In SimFL, no trusted server is needed. In each iteration, the parties first update the gradients of their local data. Then, the gradients are sent to a selected party. Next, the selected party use its local data and the gradients to update the model. Last, the model is sent to all the other parties. To ensure fairness and utilize the data from different parties, every party is selected for updating the model for about the same number of rounds. SimFL repeats a specified number of iterations and outputs the final model.

A challenging and important direction is to study the trade-off between the computation and communication cost. Specifically, one may want to know the relationship among the convergence rate, the local computation iterations between two communication rounds, and the total communication rounds. Recently, Li et al. [108] has done a good study on the convergence of FedAvg on non-IID data distribution. Their theories show that the convergence rate is in inverse ratio with the total number of local iterations (i.e., the local computation iterations between two communication rounds times the total communication rounds).




(a) FedAvg (b) SimFL

Figure 2: Federated learning frameworks

Another notable aspect is that one may need more information to compute and communicate besides the model parameters to satisfy privacy guarantees. Model parameters are vulnerable to inference attacks and may expose sensitive information about the training data [160, 130, 60]. A possible solution is secure multi-party computation [110, 68], which enables parties jointly computing a function over their inputs while keeping those inputs private. However, computation overhead of encryption and communication overhead of sending keys are significant and can be the bottleneck of the whole FL process. Thus, efficiency is an important metric in FLSs and many people have been working on reducing the overheads, especially communication size [95, 122, 156, 26].

# 4 Taxonomy

Considering the common system abstractions and building blocks for different FLSs, we classify FLSs by six aspects: data partitioning, machine learning model, privacy mechanism, communication architecture, scale of federation, and motivation of federation. These aspects include common factors (e.g., data partitioning, communication architecture) in previous FLSs [158, 97] and unique consideration (e.g., machine learning model and privacy mechanism) for FLSs. Furthermore, these aspects can be used to guide the design of FLSs. Figure 3 shows the summary of the taxonomy of FLSs.

Let us explain the six aspects with an intuitive example. The hospitals in different regions want to conduct FL to improve the performance of prediction task on lung cancer. Then, the six aspects have to be considered to design such an FLS.

- *Data partitioning*. We should study how the patient records are distributed among hospitals. While the hospitals may have different patients, they may also have different knowledge for a common patient. Thus, we have to utilize both the non-overlapped instances and features in FL.
- *Machine learning model*. We should figure out which machine learning model should be adopted for such a task. For example, if we want to perform a classification task on the diagnostic images, we may want to train a convolutional neural network in FL.
- *Privacy mechanism*. We have to decide what techniques to use for privacy protection. Since the patient records are quite private, we may have to ensure that they cannot be inferred by the exchanged gradients and models. Differential privacy is an option to achieve the privacy guarantee.

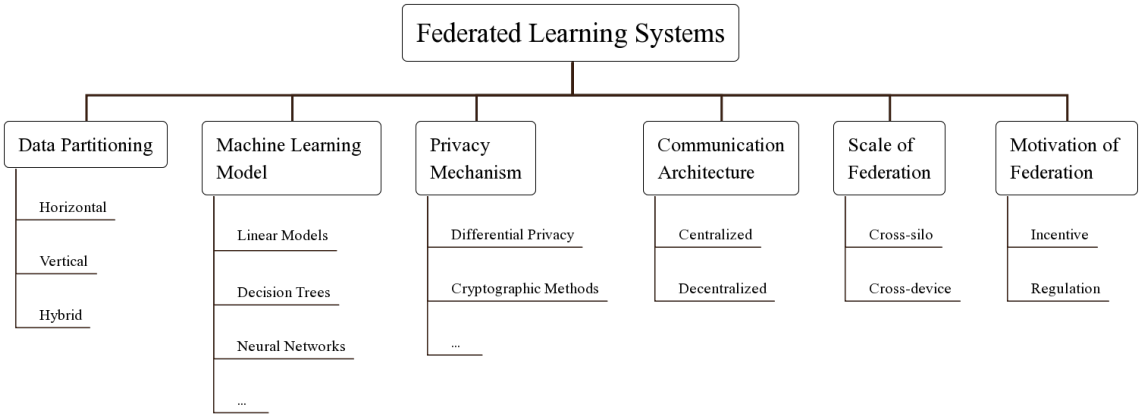



Figure 3: Taxonomy of federated learning systems

- *Communication architecture*. We have to determine the communication architecture. If there is a trusted server, then it can be the manager in FL. Otherwise we have to adopt a decentralized setting.
- *Scale of federation*. Unlike FL on mobile devices, we have a relatively small scale and well stability of federation in this scenario. Also, each party has a relative large computation power, which means we can tolerate more computation operations in the FL process.
- *Motivation of federation*. We should consider the incentive for each party to encourage them to participate in FL. A clear and straightforward motivation for the hospitals is to increase the accuracy of lung cancer prediction. Then, FL should achieve a model with a higher accuracy than the local training for every party.

## 4.1 Data Partitioning

Based on how data are distributed over the sample and feature spaces, FLSs can be typically categorized in horizontal, vertical, and hybrid FLSs [197].

In horizontal FL, the datasets of different organizations have the same feature space but little intersection on the sample space. This is a natural data partitioning especially for the cross-device setting, where different users try to improve their model performance on the same task using FL. Also, the majority of FL studies adopt the horizontal partitioning. Since the local data are with the same feature space, the parties can train the local models using their local data using the same model architecture. The key challenge is how to aggregate the information of different parties in the local or global training. FedAvg, as shown in Figure 2, directly averaging all the local models, which is simple and effective. Wake-word recognition [98], such as 'Hey Siri' and 'OK Google', is a typical application of horizontal partition because each user speaks the same sentence with a different voice.

In vertical FL, the datasets of different organizations have the same or similar sample space but differ in the feature space. Vaidya *et al.* propose multiple secure models on vertically partitioned data, including association rule mining [173], k-means [174], naive bayes classifier [175] and decision tree [176]. For the vertical FLS, it usually adopts *entity alignment* techniques [196, 47] to collect the overlapped samples of the organizations. Then the overlapped data are used to train the machine learning model using encryption methods. Cheng et al. [44] propose a lossless vertical FLS to enable parties to collaboratively train gradient boosting decision trees. They use privacy-preserving entity alignment to find common users among two parties, whose gradients are used to jointly train the decision trees. Cooperation among government agencies can be treated as a situation of vertical partition. Suppose the department of taxation

requires the housing data of residents, which are stored in the department of housing, to formulate tax policies. Meanwhile, the department of housing also needs the tax information of residents, which is kept by the department of taxation, to adapt their housing policies. These two departments share the same sample space (i.e. all the residents in the country) but each of them only has one part of features (e.g. housing or tax related personal data).

In many other applications, while existing FLSs mostly focus on one kind of partition, the partition of data among the parties may be a hybrid of horizontal partition and vertical partition. Let us take cancer diagnosis system as an example. A group of hospitals wants to build an FLS for cancer diagnosis but each hospital has different patients as well as different kinds of medical examination results. Transfer learning [139] is a possible solution for such scenarios. Liu et al. [114] propose a secure federated transfer learning system which can learn a representation among the features of parties using common instances.

### 4.2 Machine Learning Models

Since FL is used to solve machine learning problems, the parties usually want to train a state-of-the-art machine learning model. There have been many efforts in developing new models or reinventing current models to the federated setting. Here, we consider the widely-used models nowadays. The most popular machine learning model now is neural network (NN), which achieves good results in many tasks such as image classification and word prediction [96, 167]. There are many studies on federated stochastic gradient descent [122, 182, 25], which can be used to train NNs.

Another widely used model is decision tree, which is highly efficient to train compared with NNs. A tree based FLS is designed for the training for a single or multiple decision trees (e.g., gradient boosting decision trees (GBDTs) and random forests). GBDTs are especially popular recently and it has a very good performance in many classification and regression tasks. Li et al. [103] and Cheng et al. [44] propose FLSs for GBDTs on horizontally and vertically partitioned data, respectively.

Besides NNs and trees, linear models (e.g., linear regression, logistic regression, support vector machines) are classic and easy to use models. There are some well developed systems for linear regression and logistic regression [134, 77]. These linear models are basically easy to learn compared with other complex models (e.g., NNs).

Currently, many FL framework are proposed based on stochastic gradient descent [122, 94, 150, 184, 182], which is a typical optimization algorithm for many models including neural networks and linear regression. However, to increase the effectiveness of the model, we may have to exploit the model architecture to improve the FL framework. Since FL is at a early stage, there is still a gap for FLSs to better support the state-of-the-art models.

### 4.3 Privacy Mechanisms

Although the local data is not exposed in FL, it is sometimes not safe enough since the model parameters are exchanged, which may leak sensitive information about the data. There have been many attacks against machine learning models [180, 60, 30, 160, 130, 124], such as model inversion attack [60] and membership inference attack [160], which can potentially infer the raw data by accessing to the model. Also, there are many privacy mechanisms nowadays such as differential privacy [54] and $k$-anonymity [56], which provide different privacy guarantees. The characteristics of existing privacy mechanisms are summarized in the survey [179]. Here we introduce two major approaches that are adopted in current FLSs for data protection: cryptographic methods and differential privacy.

Cryptographic methods such as homomorphic encryption [15, 77, 27, 33, 74, 145, 146, 204, 206, 112], and secure multi-party computation (SMC) [157, 36, 23, 51, 24, 101, 17, 62, 91, 181, 39, 65] are widely used in privacy-preserving machine learning algorithms. Basically, the parties have to encrypt their messages before sending, operate on the encrypted messages, and decrypt the encrypted output to get the final result. Applying the above methods, the user privacy of FLSs can usually be well protected [88, 201, 89, 136, 202]. For example, SMC [68] guarantees that all the parties cannot learn

anything except the output. However, SMC is vulnerable to the inference attack. Also, due to the additional encryption and decryption operations, such systems suffer from the extremely high computation overhead.

Differential privacy [54, 55] guarantees that one single record does not influence much on the output of a function. Many studies adopt differential privacy [35, 18, 9, 192, 208, 78, 104, 171] for data privacy protection, where the parties cannot know whether an individual record participates in the learning or not. By adding random noises to the data or the model parameters [9, 104, 163], differential privacy provides statistical privacy guarantees for individual records and protection against the inference attack on the model. Due to the noises in the learning process, such systems tend to produce less accurate models.

Note that the above methods are independent of each other, and an FLS can adopt multiple methods to enhance the privacy guarantees [69, 195]. There are also other approaches to protect the user privacy. An interesting hardware-based approach is to use trusted execution environment (TEE) such as Intel SGX processors [149, 137], which can guarantee that code and data loaded inside are protected. Such environment can be used inside the central server to increase its credibility.

While most of the existing FLSs adopt cryptographic techniques or differential privacy to achieve well privacy guarantee, the limitations of these approaches seem hard to overcome currently. While trying to minimize the side effects brought by these methods, it may also be a good choice to look for novel approaches to protect data privacy and flexible privacy requirements. For example, Liu et al. [114] adopts a weaker security model [52], which can make the system more practical.

Related to privacy level, the threat models also vary in FLSs [119]. The attacks can come from any stage of the process of FL, including inputs, the learning process, and the learnt model.

- *Inputs* The malicious parties can conduct data poisoning attacks [40, 99, 12] on FL. For example, the parties can modify the label of a specific class of samples before learning, so that the learnt model perform badly on this class.

- *Learning process* During the learning process, the parties can perform model poisoning attacks [16, 193] to upload designed model parameters. Like data poisoning attacks, the global model can have a very low accuracy due to the poisoned local updating. Besides model poisoning attacks, the Byzantine fault [32, 22, 43, 166] is also a common issue in distributed learning, where the parties may behave arbitrarily badly and upload random updates.

- *The learnt model*. If the learnt model is published, the inference attacks [60, 160, 124, 130] can be conducted on it. The server can infer sensitive information about the training data from the exchanged model parameters. For example, membership inference attacks [160, 130] can infer whether a specific data record is used in the training. Note that the inference attacks may also be conducted in the learning process by the FL manager, who has access to the local updates of the parties.

### 4.4 Communication Architecture

There are two major ways of communications in FLSs: centralized design and decentralized design. In the centralized design, the data flow is often asymmetric, which means the manager aggregates the information (e.g., gradients or model parameters) from the other parties and send back training results [25]. The parameter updates on the global model are always done in this manager. The communication between the manager and the local parties can be synchronous [122] or asynchronous [194, 164]. In a decentralized design, the communications are performed among the parties [208, 103] and every party is able to update the global parameters directly.

Google Keyboard [76] is a case of centralized architecture. The server collects local model updates from users' devices and train a global model, which is sent back to the users for inference. As shown in Figure 2a, a centralized architecture is usually simple and effective. The scalability and stability are two important factors in the system design of the centralized FL. While the centralized design is widely used in existing studies, the decentralized design is preferred at some aspects since concentrating information on one server may bring potential risks or unfairness. Recently, blockchain [210] is a popular decentralized

platform for consideration. It is still challenging to design a decentralized system for FL while each party is treated nearly equally in terms of communication during the learning process and no trusted server is needed. The decentralized cancer diagnosis system among hospitals is an example of decentralized architecture. Each hospital shares the model trained with data from their patients and gets the global model for diagnosis [28]. In the decentralized design, the major challenge is that it is hard to design a protocol that treats every member almost fairly with reasonable communication overhead. As there is no central server and the training is conducted in the parties, the party may have to collect information from all the other parties, and the communication overhead of each party can be proportional to the number of parties naturally.

### 4.5 Scale of Federation

The FLSs can be categorized into two typical types by the scale of federation: cross-silo FLS and cross-device FLS [85]. The differences between them lie on the number of parties and the amount of data stored in each party.

In cross-silo FLS, the parties are organizations or data centers. There are usually a relatively small number of parties and each of them has a relatively large amount of data as well as computational power. For example, Amazon wants to recommend items for users by training the shopping data collected from hundreds of data centers around the world. Each data center possesses a huge amount of data as well as sufficient computational resources. One challenge that private FLS faces is how to efficiently distribute computation to data centers under the constraint of privacy models [211].

In cross-device FLS, on the contrary, the number of parties is relatively large and each party has a relatively small amount of data as well as computational power [183]. The parties are usually mobile devices. Google Keyboard [198] is a good example for public FLS. Google tries to improve the query suggestions of Google Keyboard with the help of FL. There are millions of Android devices and each device only has the data of its user. Meanwhile, due to the energy consumption concern, the devices cannot be asked to conduct complex training tasks. Under this occasion, the system should be powerful enough to manage a large number of parties and deal with possible issues such as the unstable connection between the device and the server.

### 4.6 Motivation of Federation

In real-world applications of FL, individual parties need the motivation to get involved in the FLS. The motivation can be regulations or incentives. FL inside a company or an organization is usually motivated by regulations. But in many cooperations, parties cannot be forced to provide their data by regulations. Taking Google Keyboard [198] as an example, Google cannot prevent users who do not provide data from using Google Keyboard. But those who agree to upload input data may enjoy a higher accuracy of word prediction. This kind of incentives can encourage every user providing their data to improve the performance of the overall model. However, how to design such a reasonable protocol remains challenging.

Incentive mechanism design can be very important for the success of an FLS. There have been some successful cases for incentive designs in blockchain [213, 57]. The parties inside the system can be collaborators as well as competitors. Other incentive designs like [87, 86] are proposed to attract participants with high-quality data for FL. We expect different game theory models [155, 84, 129] and their equilibrium designs should be revisited under the FLSs. Even in the case of Google Keyboard, the users need to be motivated to participate this collaborative learning process.

# 5 Summary of Existing Studies

In this section[2], we summarize and compare the existing studies on FLSs according to the aspects considered in Section 4.

## 5.1 Methodology

To discover the existing studies on FL, we search keyword "Federated Learning" in Google Scholar and arXiv[3]. Here we only consider the published studies in computer science community.

Since the scale of federation and the motivation of federation are problem dependent, we do not compare the existing studies by these two aspects. For ease of presentation, we use "NN", "DT" and "LM" to denote neural networks, decision trees and linear models, respectively. Also, we use "CM" and "DP" to denote cryptographic methods and differential privacy, respectively. Note that the algorithms (e.g., federated stochastic gradient descent) in some studies can be used to learn many machine learning models (e.g., logistic regression and neural networks). Thus, in the "model implementation" column, we present the models that are already implemented in the corresponding papers. Moreover, in the "main area" column, we indicate the major area that the papers study on.

## 5.2 Individual Studies

We summarize existing typical and the state-of-the-art research work, as shown in Table 1. From Table 1, we have the following four key findings.

First, most of the existing studies consider a horizontal data partitioning. We conjecture a part of the reason is that the experimental studies and benchmarking in horizontal data partitioning is relatively ready than vertical data partitioning. However, vertical FL is also common in real world, especially between different organizations. Vertical FL can enable more collaboration between diverse parties. Thus, more efforts should be paid to vertical FL in order to fill the gap.

Second, most studies consider exchanging the raw model parameters without any privacy guarantees. This may not be right if more powerful attacks on machine learning models are discovered in the future. Currently, the mainstream methods to provide privacy guarantees are differential privacy and cryptographic methods (e.g., secure multi-party computation and homomorphic encryption). Differential privacy may influence the final model quality a lot. Moreover, the cryptographic methods bring much computation and communication overhead and may be the bottleneck of FLSs. We look forward to a cheap way with reasonable privacy guarantees to satisfy the regulations.

Third, the centralized design is the mainstream of current implementations. A trusted server is needed in their settings. However, it may be hard to find a trusted server especially in the cross-silo setting. One naive approach to remove the central server is that the parties share the model parameters to all the other parties and each party also maintains the same global model locally. This method bring more communication and computation cost compared with the centralized setting. More studies should be done for practical FL with the decentralized architecture.

Last, the main research directions (also the challenging) of FL are to improve the effectiveness, efficiency, and privacy, which are also three important metrics to evaluate an FLS. Meanwhile, there are many other research topics on FL such as fairness and incentive mechanisms. Since FL is related to many research areas, we believe that FL will attract more researchers and we can see more interesting studies in the near future.

[2]Last updated on April 2, 2020. We will periodically update this section to include the state-of-the-art and valuable FL studies. Please check out our latest version at this URL: https://arxiv.org/abs/1907.09693. Also, if you have any reference that you want to add into this survey, kindly drop Dr. Bingsheng He an email (hebs@comp.nus.edu.sg).

[3]https://arxiv.org/

Table 1: Comparison among existing published studies. LM denotes Linear Models. DM denotes Decision Trees. NN denotes Neural Networks. CM denotes Cryptographic Methods. DP denotes Differential Privacy.

| FL Studies | main area | data partitioning | model implementation | privacy mechanism | communication architecture | remark |
|---|---|---|---|---|---|---|
| FedAvg [122] | Effective Algorithms | horizontal | NN | ⟍ | centralized | SGD-based |
| FedSVRG [94] | | | LM | | | |
| FedProx [150] | | | ⟍ | | | |
| Agnostic FL [127] | | | LM, NN | | | |
| FedBCD [115] | | vertical | NN | | | NN-specialized |
| PNFM [205] | | horizontal | | | | |
| FedMA [182] | | | | | | |
| Tree-based FL [208] | | | DT | DP | distributed | DT-specialized |
| SimFL [103] | | | | hashing | | |
| FedXGB [117] | | | | CM | centralized | |
| FedForest [116] | | | | | | |
| SecureBoost [44] | | vertical | | | | |
| Ridge Regression FL [134] | | horizontal | LM | | | LM-specialized |
| PPRR [41] | | | | | | |
| Linear Regression FL [153] | | vertical | | | | |
| Logistic Regression FL [77] | | horizontal | | | | |
| Federated MTL [162] | | | | ⟍ | | multi-task learning |
| Federated Meta-Learning [37] | | | NN | | | meta-learning |
| Personalized FedAvg [81] | | | | | | |
| LFRL [111] | | | | | | reinforcement learning |
| Structure Updates [95] | Practicality Enhancement | | | | | efficiency improvement |
| Multi-Objective FL [212] | | | | | | |
| On-Device ML [79] | | | | | | |
| Sparse Ternary Compression [156] | | | | | | |
| Client-Level DP FL [64] | | | | DP | | privacy guarantees |
| FL-LSTM [123] | | | | | | |
| Local DP FL [21] | | | LM, NN | | | |
| Secure Aggregation FL [24] | | | NN | CM | | |
| Hybrid FL [172] | | | LM, DT, NN | CM, DP | | |
| Backdoor FL [16] | | | NN | ⟍ | | attacks |
| Adversarial Lens [20] | | | | | | |
| Distributed Backdoor [193] | | | | | | |
| $q$-FedAvg [106] | | | LM, NN | | | fairness |
| BlockFL [93] | | | LM | | | incentives |
| Reputation FL [86] | | | | | | |
| FedCS [135] | Applications | | NN | | | edge computing |
| DRL-MEC [185] | | | | | | |
| Resource-Constrained MEC [184] | | | LM, NN | | | |
| FedCF [14] | | | LM | | | collaborative filter |
| FedMF [34] | | | | | | matrix factorization |
| FL Keyboard [76] | | | NN | | | natural language processing |
| LEAF [29] | Benchmark | | ⟍ | | ⟍ | benchmark |

### 5.2.1 Effective Algorithms

While some algorithms are based on SGD, the other algorithms are specially designed for one or several kinds of model architectures. Thus, we classify them into SGD-based algorithms and model specialized algorithms accordingly.

#### SGD-based

If we look the local data on a party as a single batch, SGD can be easily implemented in a federated setting by performing a single batch gradient calculation each round. However, such method may require a large number of communication rounds to converge. To reduce the number of communication rounds, FedAvg [122], as introduced in Section 3.3 and Figure 2a, is now a typical and practical FL framework based on SGD. In FedAvg, each party conducts multiple training rounds with SGD on its local model. Then, the weights of the global model are updated as the mean of weights of the local models. The global model is sent back to the parties to finish a global iteration. By averaging the weights, the local parties can take multiple steps of gradient descent on their local models, so that the number of communication rounds can be reduced compared with the naive federated SGD.

Konečnỳ et al. [94] propose federated SVRG (FSVRG). The major difference between federated SVRG and federated averaging is the way to update parameters of local model and global model (i.e., step 2 and step 4). The formulas to update the model weights are based on stochastic variance reduced gradient (SVRG) [82] and distributed approximate newton algorithm (DANE) in federated SVRG. They compare their algorithm with the other baselines like CoCoA+ [120] and simple distributed gradient descent. Their method can achieve better accuracy with the same communication rounds for the logistic regression model. There is no comparison between federated averaging and federated SVRG.

Some studies are based on FedAvg with the change of the objective function. Sahu et al. [150] propose FedProx, where a proximal term is added to the local objective loss to limit the amout of local changes. They provide theoretical analysis on the convergence of FedProx. Mohri et al. [127] propose a new framework named agnostic FL. Instead of minimizing the loss with respect to the uniform distribution, which is an average distribution among the data distributions from local clients, they try to train a centralized model optimized for any possible target distribution formed by a mixture of the client distributions.

Recently, [115] propose the Federated Stochastic Block Coordinate Descent (FedBCD) for vertical FL. Like FedAvg, each party updates its local parameter for multiple rounds before communicating the intermediate results. They also provide convergence analysis for FedBCD.

#### Neural Networks

Although neural networks can be trained using the SGD optimizer, we can potentially increase the model utility if the model architecture can also be exploited. Yurochkin et al. [205] develop probabilistic federated neural matching (PFNM) for multi-layer perceptrons by applying Bayesian nonparametric machinery [63]. They use an Beta-Bernoulli process informed matching procedure to combine the local models into a federated global model. The experiments show that their approach can outperform FedAvg on both IID and non-IID data partitioning.

Wang et al. [182] show how to apply PFNM to CNNs (convolutional neural networks) and LSTMs (long short-term memory networks). Moreover, they propose Federated Matched Averaging (FedMA) with a layer-wise matching scheme by exploting the model architecture. Specifically, they use matched averaging to update a layer of the global model each time, which also reduces the communication size. The experiments show that FedMA has a good performance on CNNs and LSTMs than FedAvg and FedProx [150].

### Trees

Besides neural networks, decision trees are also widely used in the academic and industry [38, 90, 58, 104]. Compared with NNs, the training and inference of trees are highly efficient. However, the tree parameters cannot be directly optimized by SGD, which means that SGD-based FL frameworks are not applicable to learn trees. We need specialized frameworks for trees. Among the tree models, the Gradient Boosting Decision Tree (GBDT) model [38] is quite popular. There are several studies on federated GBDT.

There are some studies on horizontal federated GBDTs. Zhao et al. [208] propose the first FLS for GBDTs. In their framework, each decision tree is trained locally without the communications between parties. The trees trained in a party are sent to the next party to continuous train a number of trees. Differential privacy is used to protect the decision trees. Li et al. [103] exploit similarity information in the building of federated GBDTs by using locality-sensitive hashing [50]. They utilize the data distribution of local parties by aggregating gradients of similar instances. Within a weaker privacy model compared with secure multi-party computation, their approach is effective and efficient. Liu et al. [117] propose a federated extreme boosting learning framework for mobile crowdsensing. They adopted secret sharing to achieve privacy-preserving learning of GBDTs.

Liu et al. [116] propose Federated Forest, which enables training random forests on vertical FL setting. In the building of each node, the party with the corresponding split feature is responsible for splitting the samples and share the results. They encrypt the communicated data to protect privacy. Their approach is as accurate as the non-federated version.

Cheng et al. [44] propose SecureBoost, a framework for GBDTs on vertical FL setting. In their assumptions, only one party has the label information. They used the entity alignment technique to get the common data and then build the decision trees. Additively homomorphic encryption is used to protect the gradients.

### Linear/Logistic Regression

Linear/logistic regression can be achieved using SGD. Here we show the studies that are not SGD-based and specially designed for linear/logistic regression.

In horizontal FL setting, Nikolaenko et al. [134] propose a system for privacy-preserving ridge regression. Their approaches combine both homomorphic encryption and Yao's garbled circuit to achieve privacy requirements. An extra evaluator is needed to run the algorithm. Chen et al. [41] propose a system for privacy-preserving ridge regression. Their approaches combine both secure summation and homomorphic encryption to achieve privacy requirements. They provided a complete communication and computation overhead comparison among their approach and the previous state-of-the-art approaches.

In vertical FL setting, Sanil et al. [153] present a secure regression model. They focus on the linear regression model and secret sharing is applied to ensure privacy in their solution. Hardy et al. [77] present a solution for two-party vertical federated logistic regression. They use entity resolution and additively homomorphic encryption. They also study the impact of entity resolution errors on learning.

### Others

There are many studies that combine FL with the other machine learning techniques such as multi-task learning [147], meta-learning [59], reinforcement learning [125], and transfer learning [139].

Smith et al. [162] combine FL with multi-task learning [31, 207]. Their method considers the issues of high communication cost, stragglers, and fault tolerance for MTL in the federated environment. Corinzia and Buhmann [49] propose a federated MTL method with non-convex models. They treated the central server and the local parties as a Bayesian network and the inference is performed using variational methods.

Chen et al. [37] adopt meta-learning in the learning process of FedAvg. Instead of training the local NNs and exchanging the model parameters, the parties adopt the Model-Agnostic Meta-Learning (MAML) [59] algorithm in the local training and exchange the gradients of MAML. Jiang et al. [81] interpret FedAvg in the light of existing MAML algorithms. Furthermore, they apply Reptile algorithm [132] to fine-tune the global model trained by FedAvg. Their experiment show that the meta-learning algorithm can improve the effectiveness of the global model.

Liu et al. [111] propose a lifelong federated reinforcement learning framework. Adopting transfer learning techniques, a global model is trained to effectively remember what the robots have learned.

**Summary**

We summarize the work above as follows.

- As the SGD-based framework has been widely studied and used, more studies are focus on model specialized FL recently. We expect to achieve better model accuracy by using model specialized methods. Also, we encourage researchers to study on federated decision trees models (e.g., GBDTs). The tree models have a small model size and are easy to train compared with neural networks, which can result in a low communication and computation overhead in FL.

- The studies on FL is still on a early stage. Although the simple neural networks are well investigated in the federated setting, few studies have been done for apply FL to train the state-of-the-art neural networks such as ResNeXt [121] and EfficientNet [169]. How to design an effective and practical algorithm on a complex machine learning task is still challenging and a on-going research direction.

- While most studies focus on horizontal FL, there is still no well developed algorithm for vertical FL. However, vertical federated setting is common in the real world applications where multiple organizations are involved.. We look forward to more studies on this promising area.

### 5.2.2 Practicality Enhancement

While Section 5.2.1 introduces effective algorithms, here we present the studies that aim to improve the frameworks on the other aspects such as efficiency, privacy, and fairness.

**Efficiency**

While the computation in FL can be accelerated using the modern hardware and techniques [118, 100, 102] in high performance computing community [188, 189], the FL studies mainly work on reducing the communication size during the FL process.

Konečnỳ et al. [95] propose two ways, structured updates and sketched updates, to reduce the communication costs in federated averaging. The first approach restricts the structure of local updates and transforms it to the multiplication of two smaller matrices. Only one small matrix is sent during the learning process. The second approach uses a lossy compression method to compress the updates. Their methods can reduce the communication cost by two orders of magnitude with a slight degradation in convergence speed. Zhu and Jin [212] design a multi-objective evolutionary algorithm to minimize the communication costs and the global model test errors simultaneously. Considering the minimization of the communication cost and the maximization of the global learning accuracy as two objectives, they formulated FL as a bi-objective optimization problem and solve it by the multi-objective evolutionary algorithm. Jeong et al. [79] propose a FL framework for devices with non-IID local data. They designed federated distillation, whose communication size depends on the output dimension but not on the model size. Also, they proposed a data augmentation scheme using a generative adversarial network (GAN) to make the training dataset become IID. Many other studies also design specialize approach for non-IID data [209, 108, 113, 200]. Sattler et al. [156] propose a new compression framework named sparse ternary

compression (STC). Specifically, STC compresses the communication using sparsification, ternarization, error accumulation and optimal Golomb encoding. Their method is robust to non-IID data and large numbers of parties.

### Privacy and Attacks

Although the original data is not exchanged in FL, the model parameters can also leak sensitive information about the training data [160, 130, 186]. Thus, it is important to provide privacy guarantees for the exchanged local updates.

Differential privacy is a popular method to provide privacy guarantees. Geyer et al. [64] apply differential privacy in federated averaging on a client level perspective. They use the Gaussian mechanism to distort the sum of updates of gradients to protect a whole client's dataset instead of a single data point. McMahan et al. [123] deploy federated averaging in the training of Long Short-Term Memory (LSTM) recurrent neural networks (RNNs). In addition, they use user-level differential privacy to protect the parameters. Bhowmick et al. [21] apply local differential privacy to protect the parameters in FL. To increase the model quality, they considered a practical threat model that wishes to decode individuals' data but has little prior information on them. Withing this assumption, they could get a much larger privacy budget.

Bonawitz et al. [24] apply secure multi-party computation to protect the local parameters on the basis of federated averaging. Specifically, they present a secure aggregation protocol to securely compute the sum of vectors based on secret sharing [157]. They also discuss how to combine differential privacy with secure aggregation.

Truex et al. [172] combine both secure multiparty computation and differential privacy for privacy-preserving FL. They use differential privacy to inject noises to the local updates. Then the noisy updates will be encrypted using the Paillier cryptosystem [138] before sent to the central server.

For the attacks on FL, current studies mostly focus on backdoor attacks, which aim to achieve a bad global learnt model by exchanging designed local updates.

Bagdasaryan et al. [16] conduct model poisoning attack on FL. The malicious parties commit the attack models to the server so that the global model may overfit to the backdoored data. The secure multi-party computation cannot prevent such attack since it aims to protect the confidentiality of the model parameters. Bhagoji et al. [20] also study the model poisoning attack on FL. Since the averaging step will reduce the effect of the malicious model, it adopts a explicit boosting way to increase the committed weight update. Xie et al. [193] propose a distributed backdoor attack on FL. They decompose the global trigger pattern into local patterns. Each adversarial party only employ one local pattern. The experiments show that their distributed backdoor attack outperforms the central backdoor attack.

### Fairness and Incentive Mechanisms

By taking fairness into consideration based on FedAvg, Li et al. [106] propose $q$-FedAvg. Specifically, they define the fairness according to the variance of the performance of the model on the parties. If such variance is smaller, then the model is more fair. Thus, they design a new objective inspired by $\alpha$-fairness [13]. Based on federated averaging, they propose $q$-FedAvg to solve their new objective. The major difference between $q$-FedAvg with FedAvg is in the formulas to update model parameters.

Kim et al. [93] combine blockchain architecture with FL. On the basis of federated averaging, they use a blockchain network to exchange the devices' local model updates, which is more stable than a central server and can provide the rewards for the devices. Kang et al. [86] designed a reputation-based worker selection scheme for reliable FL by using a multi-weight subjective logic model. They also leverage the blockchain to achieve secure reputation management for workers with non-repudiation and tamper-resistance properties in a decentralized manner.

**Summary**

According to the review above, we summarize the above studies as follows.

- Besides effectiveness, efficiency and privacy are the other two important factors of an FLS. Compared with these three areas, there are fewer studies on fairness and incentive mechanisms. We look forward to more studies on fairness and incentive mechanisms, which can encourage the usage of FL in real world.
- For the efficiency improvement of FLSs, the communication overhead is still the main challenge. Most studies [95, 79, 156] try to reduce the communication size of each iteration. How to reasonably set the number of communication rounds is also promising [212]. The trade-off between the computation and communication still needs to be further investigated.
- For the privacy guarantees, differential privacy and secure multi-party computation are two popular techniques. However, differential privacy may impact the model quality significantly and secure multi-party computation may be very time-consuming. It is still challenging to design a practical FLS with strong privacy guarantees. Also, the effective defense algorithms against poisoning attacks are not widely adopted yet.

### 5.2.3 Applications

One related area with FL is edge computing [133, 203, 143, 53], where the parties are the edge devices. Many studies try to integrate FL with the mobile edge systems. FL also shows promising results in recommender system [14, 34] and natural language processing [76].

**Edge Computing**

Nishio and Yonetani [135] implement federated averaging in practical mobile edge computing (MEC) frameworks. They use an operator of MEC framworks to manage the resources of heterogeneous clients. Wang et al. [185] adopt both distributed deep reinforcement learning (DRL) and federatd learning in mobile edge computing system. The usage of DRL and FL can effectively optimize the mobile edge computing, caching, and communication. Wang et al. [184] perform FL on resource-constrained MEC systems. They address the problem of how to efficiently utilize the limited computation and communication resources at the edge. Using federated averaging, they implement many machine learning algorithms including linear regression, SVM, and CNN.

**Recommender System**

Ammad-ud din et al. [14] formulate the first federated collaborative filter method. Based on a stochastic gradient approach, the item-factor matrix is trained in a global server by aggregating the local updates. They empirically show that the federated method has almost no accuracy loss compared with the centralized method. Chai et al. [34] design a federated matrix factorization framework. They use federated SGD to learn the matrices. Moreover, they adopt homomorphic encryption to protect the communicated gradients.

**Natural Language Processing**

Hard et al. [76] apply FL in mobile keyboard next-word prediction. They adopt the federated averaging method to learn a variant of LSTM called Coupled Input and Forget Gate (CIFG) [70]. The FL method can achieve better precision recall than the server-based training with logs data.

**Summary**

According to the above studies, we have the following summaries.

- Edge computing naturally fits the cross-device federated setting. A non-trivial issue of applying FL to edge computing is how to effectively utilize and manage the edge resources. The usage of FL can bring benefits to the users, especially for improving the mobile device services.
- FL can solve many traditional machine learning tasks such as image classification and work prediction. Due to the regulations and "data islands", federated setting may be a common setting in the next years. With the fast development of FL, we believe that there will be more applications in computer vision, natural language processing, and healthcare.

### 5.2.4 Benchmark

Benchmark is quite important to direct the development of FLSs. Currently, we can only find one open source benchmark, LEAF, proposed by [29]. LEAF includes public federated datasets, an array of statistical and systems metrics, and a set of reference implementations. However, it lacks metrics to evaluate the privacy and efficiency of FLSs. Also, the current experiments of LEAF are limited to several FL implementation, which is not comprehensive enough.

## 5.3 Open Source Systems

In this section, we introduce four open source FLSs: Federated AI Technology Enabler (FATE)[4], Google TensorFlow Federated (TFF)[5], OpenMined PySyft[6], and Baidu PaddleFl[7].

### 5.3.1 FATE

FATE is a industrial level FL framework, which aims to provide FL services between different organizations. The overall structure of FATE is shown in Figure 4. It has six major modules: EggRoll, FederatedML, FATE-Flow, FATE-Serving, FATE-Board, and KubeFATE. EggRoll manages the distributed computing and storage. It provides computing and storage AIPs for the other modules. FederatedML includes the federated algorithms and secure protocols. Currently, it supports training many kinds of machine learning models under both horizontal and vertical federated setting, including NNs, GBDTs, and logistic regression. Also, it integrates secure multi-party computation and homomorphic encryption to provide privacy guarantees. FATE-Flow is a platform for the users to define their pipeline of the FL process. The pipeline can include the data preprocessing, federated training, federated evaluation, model management, and model publishing. FATE-Serving provides the inference services for the users. It supports loading the FL models and conducting online inference on them. FATE-Board is a visualization tool for FATE. It provides a visual way to track the job execution and model performance. Last, KubeFATE helps deploy FATE on clusters by using Docker[8] or Kubernetes[9]. It provides customized deployment and cluster management services. In general, FATE is a powerful and easy-to-use FLS. Users can simply set the parameters to run a FL algorithm. Moreover, FATE provides detailed documents on its deployment and usage. However, since FATE provides algorithm-level interfaces, practitioners have to modify the source code of FATE to implement their own federated algorithms. This is not easy for non-expert users.

[4]https://github.com/FederatedAI/FATE
[5]https://github.com/tensorflow/federated
[6]https://github.com/OpenMined/PySyft
[7]https://github.com/PaddlePaddle/PaddleFL
[8]https://www.docker.com/
[9]https://kubernetes.io/

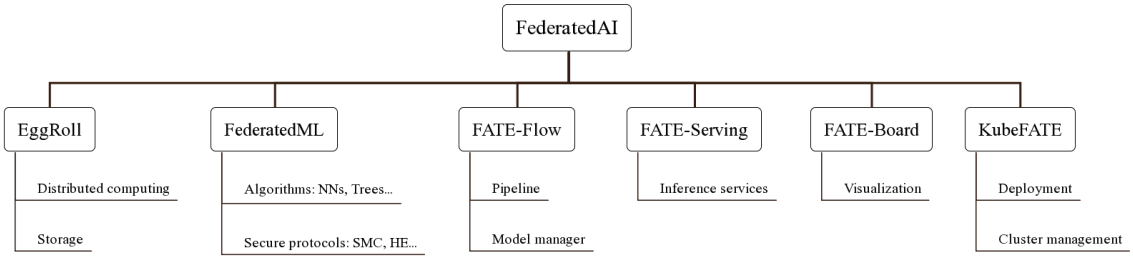



Figure 4: The FATE system structure

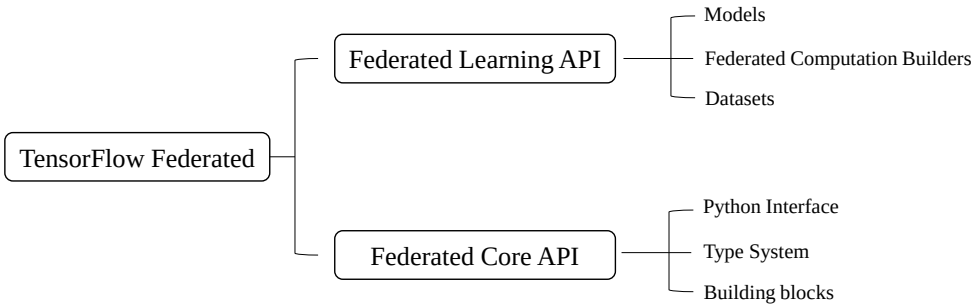



Figure 5: The TFF system structure

### 5.3.2 TFF

TFF provides the building blocks for FL based on TensorFlow. As shown in Figure 5, it provides two APIs of different layers: FL API[10] and Federated Core (FC) API[11]. On the one hand, FL API offers high-level interfaces. It includes three key parts, which are models, federated computation builders, and datasets. FL API allows users to define the models or simply load the Keras [72] model. The federated computation builders include the typical federated averaging algorithm. Also, FL API provides the simulated federated datasets and the functions to access and enumerate the local datasets for FL. On the other hand, FC API includes lower-level interfaces as the foundation of the FL process. Developers can implement their functions and interfaces inside the federated core. Specifically, as a Python package, FC provides Python interfaces and developers can use them and write new Python functions. To be easy-to-use especially for developers familiar with TensorFlow, it supports many types such as Tenser types, sequence types, tuple types, and function types. Finally, FC provides the building blocks for FL. It support multiple federated operators such as federated sum, federated reduce, and federated broadcast. Developers can define their own operators to implement the FL algorithm. Overall, TFF is a lightweight system for developers to design and implement new FL algorithms. Currently, TFF only supports FedAvg and does not provide privacy mechanisms. It can only deploy on a single machine now, where the federated setting is implemented by simulation.

### 5.3.3 PySyft

PySyft, first proposed by Ryffel et al. [148], is a python library that provides interfaces for developers to implement their training algorithm. While TFF is based on TensorFlow, PySyft can work well with both PyTorch and TensorFlow. Although PySyft only supports FedAvg algorithm, it provides multiple privacy mechanisms including secure multi-party computation and differential privacy. Also, it can be deployed

[10]https://github.com/tensorflow/federated/blob/master/docs/federated_learning.md
[11]https://github.com/tensorflow/federated/blob/master/docs/federated_core.md

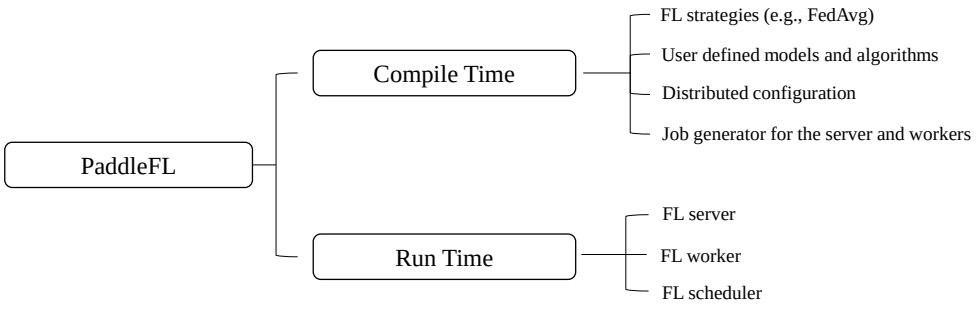



Figure 6: The PaddleFL system structure

on a single machine or multiple machines, where the communication between different clients is through the websocket API [161]. However, while PySyft provides a set of tutorials, there is no detailed document on its interfaces and system architecture.

### 5.3.4 PaddleFL

PaddleFL is a FLS based on PaddlePaddle [12], which is a deep learning platform developed by Baidu. The system structure of PaddleFL is shown in Figure 6. In the compile time, there are four components including FL strategies, user defined models and algorithms, distributed training configuration, and FL job generator. The FL strategies include the horizontal FL algorithms such as FedAvg. Vertical FL algorithms will be integrated in the future. Besides provided FL strategies, users can also define their own models and training algorithms. The distributed training configuration defines the training node information in the distributed setting. FL job generator generates the jobs for federated server and workers. In the run time, there are three components including FL server, FL worker, and FL scheduler. The server and worker are the manager and parties in FL, respectively. The scheduler selects the workers that participate in the training in each round. Currently, the development of PaddleFL is still in a early stage and the documents and examples are not clear enough.

### 5.3.5 Others

There are other closed source federated learning systems. NVIDIA Clara [13] has enabled FL. It adopts a centralized architecture and encrypted communication channel. The targeted users of Clara FL is hospitals and medical institutions [107]. Ping An Technology aims to build a federated learning system named Hive, which mainly target at the financial industries. While Clara FL provides APIs and documents, we cannot find the official documents of Hive.

### 5.3.6 Summary

Overall, FATE and PaddleFL try to provide algorithm level APIs for users to use directly, while TFF and PySyft try to provide more detailed building blocks so that the developers can easily implement their FL process. Table 2 shows the comparison between the open source systems. In algorithm level, FATE is the most comprehensive system that supports many machine learning models under both horizontal and vertical settings. TFF and PySyft only implement FedAvg, which is a basic framework in FL as shown in Section 5.2. PaddleFL supports several horizontal FL algorithms currently on NNs and logistic regression. Compared with FATE and TFF, PySyft and PaddleFL provides more privacy mechanisms. PySyft covers all the listed features that TFF supports, while TFF is based on TensorFlow and PySyft works better with PyTorch.

[12] https://github.com/PaddlePaddle/Paddle
[13] https://developer.nvidia.com/clara

Table 2: Comparison among some existing FLSs. The notations used in this table are the same as Table 1.

| Supported features | | FATE 1.3.0 | TFF 0.12.0 | PySyft 0.2.3 | PaddleFL 0.2.0 |
|---|---|---|---|---|---|
| Operation systems | Mac | ✓ | ✓ | ✓ | ✓ |
| | Linux | ✓ | ✓ | ✓ | ✓ |
| | Windows | ✗ | ✗ | ✓ | ✓ |
| | iOS | ✗ | ✗ | ✗ | ✗ |
| | Android | ✗ | ✗ | ✗ | ✗ |
| Data partitioning | horizontal | ✓ | ✓ | ✓ | ✓ |
| | vertical | ✓ | ✗ | ✗ | ✗ |
| Models | NN | ✓ | ✓ | ✓ | ✓ |
| | DT | ✓ | ✗ | ✗ | ✗ |
| | LM | ✓ | ✓ | ✓ | ✓ |
| Privacy Mechanisms | DP | ✗ | ✗ | ✓ | ✓ |
| | CM | ✓ | ✗ | ✓ | ✓ |
| Communication | simulated | ✓ | ✓ | ✓ | ✓ |
| | distributed | ✓ | ✗ | ✓ | ✓ |
| Hardwares | CPUs | ✓ | ✓ | ✓ | ✓ |
| | GPUs | ✗ | ✓ | ✗ | ✗ |

# 6 System Design

Figure 7 shows the factors that need to be considered in the design of an FLS, including effectiveness, efficiency, privacy, and autonomy. Next, we explain these factors and introduce the design guideline in detail.

## 6.1 Effectiveness

The core of an FLS is an (multiple) effective algorithm (algorithms). To determine the algorithm to be implemented from lots of existing studies as shown in Table 1, we should first check the data partitioning of the parties. If the parties have the same features but different samples, one can use FedAvg [122] for NNs and SimFL [103] for trees. If the parties have the same sample space but different features, one can use FedBCD [115] for NNs and SecureBoost [44] for trees.

## 6.2 Privacy

An important requirement of FLSs is to protect the user privacy. Here we analyze the reliability of the manager. If the manager is honest and not curious, then we do not need to adopt any additional technique, since the FL framework ensures that the raw data is not exchanged. If the manager is honest but curious, then we have to take possible inference attacks into consideration. The model parameters may also expose sensitive information about the training data. One can adopt differential privacy [64, 46, 123] to inject random noises into the parameters or use SMC [23, 77, 24] to exchanged encrypted parameters. If the manager cannot be trusted at all, then we can use trusted execution environments [42] to execute the code in the manager. Blockchain is also an option to play the role as a manager [93].

## 6.3 Efficiency

Efficiency plays a key role in making the system popular. To increase the efficiency, the most effective way is to deal with the bottleneck. If the bottleneck lies in the computation, we can use powerful hardware such as GPUs [48] and TPUs [83]. If the bottleneck lies in the communication, the compression techniques [19, 95, 156] can be applied to reduce the communication size.

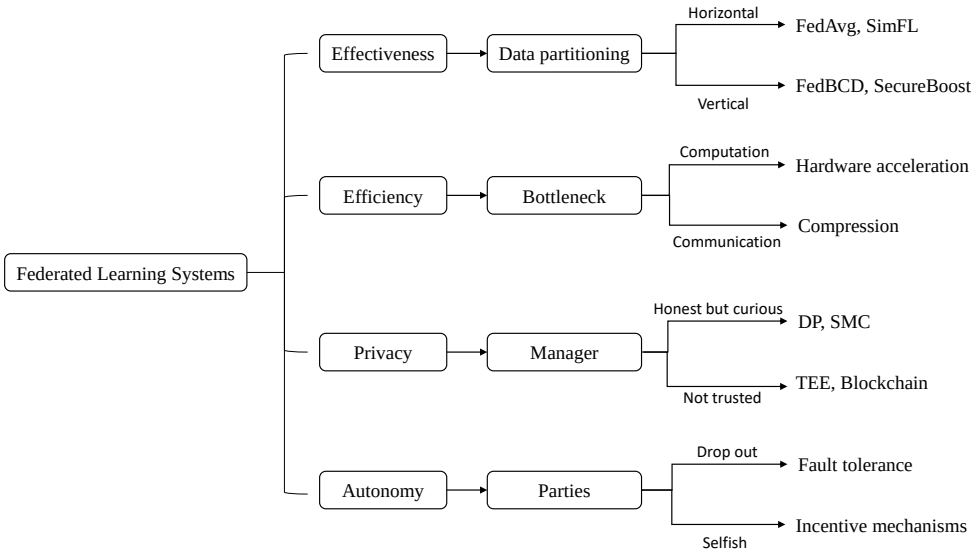



Figure 7: The design factors of FLSs

### 6.4 Autonomy

A practical FLS has to consider the autonomy of the parties. The parties may drop out during the FL process, especially in the cross-device setting. Thus, the system cannot reply too much on each single party. It should tolerate the failure of a small part of parties. Also, the parties may be selfish and are not willing to share the model with good quality. Incentive mechanisms [86, 87] can encourage the participation of the parties and improve the final model quality.

### 6.5 The Design Guideline

We derive a guideline for developing an FLS based on our taxonomy shown in Section 4 and the design factors.

The first step is to determine the FL algorithms by analyzing the system aspects from the actual scenario. The aspects include the data partitioning, the scale of federation, the communication architecture, and the machine learning model. The data partitioning and scale of federation are almost fixed once the participated entities are fixed. Suppose the users want to conduct FL to improve the Google keyboard prediction. Then in this case we have horizontal data partitioning and cross-device federated setting. Google can provide a server to take the role of manager of FL so we can adopt a centralized communication architecture. Thus, the algorithms such as FedAvg and FedMA can satisfy our requirements. Since LSTM performs well on the word prediction task, we can adopt FedMA [182] algorithm, which is specialized design for CNN and LSTM, to train a LSTM model.

The next step is to satisfy the privacy requirements. While current regulations do not put explicit restrictions on the transfer of model parameters, the FLS should protect the individual records from inference attacks if they are very sensitive. For example, the hospitals want to conduct FL on lung cancer prediction task. The patients' records should be well protected and we should adopt differential privacy or secure multi-party computation techniques in our FLS. an FLS without privacy guarantees can be very dangerous.

The last step is to consider the incentive mechanisms. The incentive mechanisms are not necessary in FLSs if the parties are regulation motivated to conduct FL. However, to design a lively FLS, the incentive mechanisms are important to encourage the parties to participate and contribute. The basic requirement of incentive mechanisms is to ensure that the parties can get more rewards if they provide more contribution. Blockchain is an option to provide stable and verifiable incentives [93, 86, 87].

Table 3: Requirements of the real-world federated systems

| System Aspect | Mobile Service | Healthcare | Financial |
|---|---|---|---|
| Data Partitioning | Horizontal Partitioning | Hybrid Partitioning | Vertical Partitioning |
| Machine Learning Model | No specific Models | No specific Models | No specific Models |
| Scale of Federations | Cross-device | Cross-silo | Cross-silo |
| Communication Architecture | Centralized | Distributed | Distributed |
| Privacy Mechanism | DP | DP/SMC | DP/SMC |
| Motivation of Federation | Incentive Motivated | Policy Motivated | Interest Motivated |

# 7 Case Study

In this section, we present several real-world applications of FL according to our taxonomy, as summarized in Table 3.

## 7.1 Mobile Service

There are many corporations providing predicting service to their mobile users, such as Google Keyboard [198], Apple's emoji suggestion and QuickType [170]. These services bring much convenience to the users. However, the training data come from users' edge devices, like smartphone. If the company collects data from all the users and trains a global model, it might potentially cause privacy leakage. On the other hand, the data of each single user are insufficient to train an accurate prediction model. FL enables these companies to train an accuracy prediction model without accessing users' original data, which means protecting users' privacy. In the framework of FLSs, the users calculate and send their local models instead of their original data. That means a Google Keyboard user can enjoy an accurate prediction for the next word while not sharing his/her input history. If FLS can be widely applied to such prediction services, there will be much less data leakage since data are always stored in the edge.

In such scenario, data are usually horizontally split on millions of devices. Hence, the limitation of single device computational resource and the bandwidth are two major problems. Besides, the robustness of the system should also be considered since a user could join or leave the system at anytime. In other words, a centralized, cross-device FLS on horizontal data should be designed for such prediction services.

Though the basic framework of an FLS can has somehow protected individuals' privacy, it may not be secure against inference attacks [160]. Some additional privacy mechanisms like differential privacy should be leveraged to ensure the indistinguishability of individuals. Here secure multi-party computation may not be appropriate since each device has a weak computation capacity and cannot afford expensive encryption operations. Apart from guaranteeing users' privacy, some incentive mechanisms should be developed to encourage users to contribute their data. In reality, these incentives could be vouchers or additional service.

## 7.2 Healthcare

Modern health systems require a cooperation among research institutes, hospitals, federal agencies in order to improve health care of the nation [61]. Moreover, a collaborative research among countries is vital when facing global health emergency, like COVID-19 [7]. These health systems mostly aim to train a model for diagnosis of a disease. These models for diagnosis should be as accurate as possible. However, the information of patients are not allowed to transfer under some regulations such as GDPR [11]. The privacy of data is even more concerned in international collaboration. Without solving the privacy issue, the collaborative research could be stagnated, threatening the public health. The data privacy in such collaboration is largely based on confidentiality agreement. But after all, this solution is based on "trust", which is not reliable. FL makes the cooperation possible because it can ensure the privacy theoretically, which is provable and reliable. In this way, every hospital or institute only has to share local models to get an accurate model for diagnosis.

In such a scenario, the health care data is partitioned both horizontally and vertically: each party contains health data of residents for a specific purpose (e.g. patient treatment), but the features used in each party are diverse. The number of parties is limited and each party usually has plenty of computational resource. In other words, a private FLS on hybrid partitioned data is required. One of the most challenging problems is how to train the hybrid partitioned data. The design of the FLS could be more complicated than a simple horizontal system. In a federation of healthcare, there is probably no central server. So, another challenging part is the design of a decentralized FLS, which should also be robust against some dishonest or malicious parties. Moreover, the privacy concern can be solved by additional mechanisms like secure multi-party computation and differential privacy. The collaboration is largely motivated by regulations.

### 7.3 Finance

A federation of financial consists of banks, insurance companies, etc. They often hope to cooperate in daily financial operations. For example, some 'bad' users might pack back loan in one back with the money borrowed from another bank. All the banks want to avoid such malicious behavior while not revealing other customers' information. Also, insurance companies also want to learn from the banks about the reputation of customers. However, a leakage of 'good' customers' information may cause loss of interest or some legal issues.

This kind of cooperation can happen if we have a trusted third party, like the government. But in many cases, the government is not involved in the federation or the government is not always trusted. So, an FLS with privacy mechanisms can be introduced. In the FLS, the privacy of each bank can be guaranteed by theoretical proved privacy mechanisms.

In such a scenario, financial data are often vertically partitioned, linked by user ID. Training a classifier in vertically partitioned data is quite challenging. Generally, the training process can be divided into two parts: privacy-preserving record linkage [177] and vertical federated training. The first part aims to find links between vertical partitioned data, and it has been well studied. The second part aims to train the linked data without sharing the original data of each party, which still remains a challenge. The cross-silo and decentralized setting are applied in this federation. Also, some privacy mechanisms should be adopted in this scenario and the participant can be motivated by interest.

## 8 Vision

In this section, we show interesting directions to work on in the future.

### 8.1 Heterogeneity

The heterogeneity of the parties is an important characteristic in FLSs. Basically, the parties can differ in the accessibility, privacy requirements, contribution to the federation, and reliability. Thus, it is important to consider such practical issues in FLSs.

**Dynamic scheduling** Due to the instability of the parties, the number of parties may not be fixed during the learning process. However, especially for the cross-silo setting, the number of parties is fixed in many existing studies and they do not consider the situations where there are entries of new parties or departures of the current parties. The system should support dynamic scheduling and have the ability to adjust its strategy when there is a change in the number of parties. There are some studies addressing this issue. For example, Google TensorFlow Federated [25] can tolerate the drop-outs of the devices. Also, the emergence of blockchain [210] can be an ideal and transparent platform for multi-party learning. More efforts need to be done in this direction.

**Diverse privacy restrictions** Little work has considered the privacy heterogeneity of FLSs, where the parties have different privacy requirements. The existing systems adopt techniques to protect the model parameters or gradients for all the parties on the same level. However, the privacy restrictions of the parties usually differ in reality. It would be interesting to design an FLS which treats the parties differently according to their privacy restrictions. The learned model should have a better performance if we can maximize the utilization of data of each party while not violating their privacy restrictions. The heterogeneous differential privacy [10] may be useful in such settings.

**Intelligent benefits** Intuitively, one party can gain more from the FLS if it contributes more information. A simple solution is to make agreements among the parties such that some parties pay for the other parties which contribute more information. Representative incentive mechanisms need to be developed.

**Robustness** While one can use differential privacy in FL to provide protection against potential inference attacks, there are other dangerous attacks such as data poisoning and backdoor attacks due to malicious parties. Along this line, Gu et al. [71] present a multi-party collaborative learning system to fulfill model accountability in trusted execution environment environments. Ghosh et al. [66] consider the model robustness upon Byzantine parties (or abnormal and adversarial parties). Another potential approach can be blockchain [142, 92]. Preuveneers et al. [142] propose a permissioned blockchain-based FL method to monitor the incremental updates to an anomaly detection machine learning model.

## 8.2 System Development

To boost the development of FLSs, besides the detailed algorithm design, we need to study from a high-level view.

**System architecture** Like the parameter server in deep learning which controls the parameter synchronization, some common system architectures are needed to be investigated for FL. Although FedAvg is a widely used framework, the applicable scenarios are still limited. For example, how to conduct federated learning in a unsupervised setting? Also, is there other aggregation methods besides the model averaging? We want a general system architecture, which provides many aggregation methods and learning algorithms for different settings.

**Model market** An possible variant of FL is that we maintain a model market for buying and selling. The party can buy the models to conduct model aggregation locally. Also, it can contribute its model to the market with additional information such as the target task. Such design introduce more flexibility to the federation and is more acceptable for the organizations, since the FL just like several transactions. A well evaluation of the models is important in such systems. The incentive mechanisms may be helpful [191, 86, 87].

**Benchmark** As more FLSs are being developed, a benchmark with representative data sets and workloads is quite important to evaluate the existing systems and direct future development. Caldas et al. [29] propose LEAF, which is a benchmark including federated datasets, an evaluation framework, and reference implementations. Hao et al. [75] present a computing testbed named Edge AIBench with FL support, and discussed four typical scenarios and six components for measurement included in the benchmark suite. Still, more applications and scenarios are the key to the success of FLSs.

**Data life cycles** Learning is simply one aspects of a federated system. A data life cycle consists of multiple stages including data creation, storage, use, share, archive and destroy. For the data security and privacy of the entire application, we need to invent new data life cycles under FL context. Although data sharing is clearly one of the focused stage, the design of FLSs also affects other stages. For example, data creation may help to prepare the data and features that are suitable for FL.

### 8.3 FL in Domains

**Internet-of-thing** Security and privacy issues have been a hot research area in fog computing and edge computing, due to the increasing deployment of Internet-of-thing applications. For more details, readers can refer to some recent surveys [165, 199, 128]. FL can be one potential approach in addressing the data privacy issues, while still offering reasonably good machine learning models [109, 131]. The additional key challenges come from the computation and energy constraints. The mechanisms of privacy and security introduces runtime overhead. For example, Jiang et al. [80] apply independent Gaussian random projection to improve the data privacy, and then the training of a deep network can be too costly. The authors need to develop new resource scheduling algorithm to move the workload to the nodes with more computation power. Similar issues happen on other environments such as vehicle-to-vehicle networks [151, 154].

**Regulations** While FL enables the collaborative learning without exposing the raw data, it is still not clear how FL comply with the existing regulations. For example, GDPR proposes limitations on the data transfer. Since the model and gradients are actually not safe enough, is such limitation still apply to the model or gradients? Also, the "right to explainability" is hard to execute since the global model is an averaging of the local models. The explainability of the FL models is an open problem Gunning [73], Samek et al. [152]. Moreover, if a user wants to delete its data, should the global model be retrained without the data [67]? There is still a gap between the FL techniques and the regulations in reality. We may expect the cooperation between the computer science community and the law community.

## 9 Conclusion

Many efforts have been devoted to developing federated learning systems (FLSs). A complete overview and summary for existing FLSs is important and meaningful. Inspired by the previous federated systems, we have shown that heterogeneity and autonomy are two important factors in the design of practical FLSs. Moreover, with six different aspects, we provide a comprehensive categorization for FLSs. Based on these aspects, we also present the comparison on features and designs among existing FLSs. More importantly, we have pointed out a number of opportunities, ranging from more benchmarks to integration of emerging platforms such as blockchain. FLSs will be an exciting research direction, which calls for the effort from machine learning, system and data privacy communities.

## Acknowledgement

This work is supported by a MoE AcRF Tier 1 grant (T1 251RES1824), an SenseTime Young Scholars Research Fund, and a MOE Tier 2 grant (MOE2017-T2-1-122) in Singapore.

## Acknowledgement

## References

[1] California Consumer Privacy Act Home Page. https://www.caprivacy.org/.

[2] Uber settles data breach investigation for $148 million, 2018. URL https://www.nytimes.com/2018/09/26/technology/uber-data-breach.html.

[3] Hundreds of millions of facebook user records were exposed on amazon cloud server, 2019. URL https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/.

[4] Google is fined $57 million under europe's data privacy law, 2019. URL https://www.nytimes.com/2019/01/21/technology/google-europe-gdpr-fine.html.

[5] 2019 is a 'fine' year: Pdpc has fined s'pore firms a record $1.29m for data breaches, 2019. URL https://vulcanpost.com/676006/pdpc-data-breach-singapore-2019/.

[6] U.s. customs and border protection says photos of travelers were taken in a data breach, 2019. URL https://www.washingtonpost.com/technology/2019/06/10/us-customs-border-protection-says-photos-travelers-into-out-country-were-recently-taken-data-breach/.

[7] Rolling updates on coronavirus disease (covid-19), 2020. URL https://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen.

[8] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[9] 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.

[10] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. *arXiv preprint arXiv:1504.06998*, 2015.

[11] Jan Philipp Albrecht. How the gdpr will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

[12] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[13] Eitan Altman, Konstantin Avrachenkov, and Andrey Garnaev. Generalized $\alpha$-fair resource allocation in wireless networks. In *2008 47th IEEE Conference on Decision and Control*, pages 2414–2419. IEEE, 2008.

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

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

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

[17] Raad Bahmani, Manuel Barbosa, Ferdinand Brasser, Bernardo Portela, Ahmad-Reza Sadeghi, Guillaume Scerri, and Bogdan Warinschi. Secure multiparty computation from sgx. In *International Conference on Financial Cryptography and Data Security*, pages 477–497. Springer, 2017.

[18] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[19] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

[20] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens, 2018.

[21] 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.

[22] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pages 119–129, 2017.

[23] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

[24] 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.

[25] 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.

[26] Keith Bonawitz, Fariborz Salehi, Jakub Konečnỳ, Brendan McMahan, and Marco Gruteser. Federated learning with autotuned communication-efficient secure aggregation. *arXiv preprint arXiv:1912.00131*, 2019.

[27] Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. Fast homomorphic evaluation of deep discretized neural networks. In *Annual International Cryptology Conference*, pages 483–512. Springer, 2018.

[28] 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.

[29] 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.

[30] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *arXiv preprint arXiv:1802.08232*, 2018.

[31] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[32] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[33] Hervé Chabanne, Amaury de Wargny, Jonathan Milgram, Constance Morel, and Emmanuel Prouff. Privacy-preserving classification on deep neural network. *IACR Cryptology ePrint Archive*, 2017: 35, 2017.

[34] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *arXiv preprint arXiv:1906.05108*, 2019.

[35] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[36] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

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

[38] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, pages 785–794. ACM, 2016.

[39] Valerie Chen, Valerio Pastro, and Mariana Raykova. Secure computation for machine learning with spdz. *arXiv preprint arXiv:1901.00329*, 2019.

[40] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[41] Yi-Ruei Chen, Amir Rezapour, and Wen-Guey Tzeng. Privacy-preserving ridge regression on distributed data. *Information Sciences*, 451:34–49, 2018.

[42] Yu Chen, Fang Luo, Tong Li, Tao Xiang, Zheli Liu, and Jin Li. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Information Sciences*, 522:69–79, 2020.

[43] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):44, 2017.

[44] 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.

[45] Warren B Chik. The singapore personal data protection act and an assessment of future trends in data privacy reform. *Computer Law & Security Review*, 29(5):554–575, 2013.

[46] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. Differential privacy-enabled federated learning for sensitive health data. *arXiv preprint arXiv:1910.02578*, 2019.

[47] Peter Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media, 2012.

[48] Shane Cook. *CUDA programming: a developer's guide to parallel computing with GPUs*. Newnes, 2012.

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

[50] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[51] Wenliang Du and Zhijun Zhan. Building decision tree classifier on private data. In *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*, pages 1–8. Australian Computer Society, Inc., 2002.

[52] Wenliang Du, Yunghsiang S Han, and Shigang Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *SDM*, pages 222–233. SIAM, 2004.

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

[54] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[55] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[56] Khaled El Emam and Fida Kamal Dankar. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15(5):627–637, 2008.

[57] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59, Santa Clara, CA, March 2016. USENIX Association. ISBN 978-1-931971-29-4. URL https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal.

[58] Ji Feng, Yang Yu, and Zhi-Hua Zhou. Multi-layered gradient boosting decision trees. In *Advances in neural information processing systems*, pages 3551–3561, 2018.

[59] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[60] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015.

[61] Charles P Friedman, Adam K Wong, and David Blumenthal. Achieving a nationwide learning health system. *Science translational medicine*, 2(57):57cm29–57cm29, 2010.

[62] Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans. Secure linear regression on vertically partitioned datasets. *IACR Cryptology ePrint Archive*, 2016:892, 2016.

[63] Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

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

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

[66] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment, 2019.

[67] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, pages 3513–3526, 2019.

[68] Oded Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 78, 1998.

[69] Slawomir Goryczka and Li Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE transactions on dependable and secure computing*, 14(5):463–477, 2015.

[70] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28 (10):2222–2232, 2016.

[71] Zhongshu Gu, Hani Jamjoom, Dong Su, Heqing Huang, Jialong Zhang, Tengfei Ma, Dimitrios Pendarakis, and Ian Molloy. Reaching data confidentiality and model accountability on the caltrain, 2018.

[72] Antonio Gulli and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[73] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2, 2017.

[74] Rob Hall, Stephen E Fienberg, and Yuval Nardi. Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27(4):669, 2011.

[75] Tianshu Hao, Yunyou Huang, Xu Wen, Wanling Gao, Fan Zhang, Chen Zheng, Lei Wang, Hainan Ye, Kai Hwang, Zujie Ren, et al. Edge aibench: Towards comprehensive end-to-end edge computing benchmarking. *arXiv preprint arXiv:1908.01924*, 2019.

[76] 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.

[77] 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.

[78] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *Towards Practical Differentially Private Convex Optimization*, page 0. IEEE, 2019.

[79] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[80] Linshan Jiang, Rui Tan, Xin Lou, and Guosheng Lin. On lightweight privacy-preserving collaborative learning for internet-of-things objects. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19, pages 70–81, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6283-2. doi: 10.1145/3302505.3310070. URL http://doi.acm.org/10.1145/3302505.3310070.

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

[82] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[83] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017.

[84] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *EEE International Conference on E-Commerce, 2003. CEC 2003.*, pages 285–292, June 2003. doi: 10.1109/COEC.2003.1210263.

[85] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[86] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 2019.

[87] 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.

[88] Murat Kantarcioglu and Chris Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge & Data Engineering*, (9): 1026–1037, 2004.

[89] Alan F Karr, Xiaodong Lin, Ashish P Sanil, and Jerome P Reiter. Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, 25(1):125, 2009.

[90] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, 2017.

[91] Niki Kilbertus, Adrià Gascón, Matt J Kusner, Michael Veale, Krishna P Gummadi, and Adrian Weller. Blind justice: Fairness with encrypted sensitive attributes. *arXiv preprint arXiv:1806.03281*, 2018.

[92] H. Kim, J. Park, M. Bennis, and S. Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, pages 1–1, 2019. doi: 10.1109/LCOMM.2019.2921755.

[93] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. On-device federated learning via blockchain and its latency analysis. *arXiv preprint arXiv:1808.03949*, 2018.

[94] 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.

[95] 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.

[96] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[97] Tobias Kurze, Markus Klems, David Bermbach, Alexander Lenk, Stefan Tai, and Marcel Kunze. Cloud federation. *Cloud Computing*, 2011:32–38, 2011.

[98] 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.

[99] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, pages 1885–1893, 2016.

[100] Peilong Li, Yan Luo, Ning Zhang, and Yu Cao. Heterospark: A heterogeneous cpu/gpu spark platform for machine learning algorithms. In *2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 347–348. IEEE, 2015.

[101] Ping Li, Jin Li, Zhengan Huang, Tong Li, Chong-Zhi Gao, Siu-Ming Yiu, and Kai Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74: 76–85, 2017.

[102] Qinbin Li, Zeyi Wen, and Bingsheng He. Adaptive kernel value caching for svm training. *IEEE transactions on neural networks and learning systems*, 2019.

[103] Qinbin Li, Zeyi Wen, and Bingsheng He. Practical federated gradient boosting decision trees. *arXiv preprint arXiv:1911.04206*, 2019.

[104] Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-preserving gradient boosting decision trees. *arXiv preprint arXiv:1911.04209*, 2019.

[105] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions, 2019.

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

[107] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 133–141. Springer, 2019.

[108] 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.

[109] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey, 2019.

[110] Yehida Lindell. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1005–1009. IGI Global, 2005.

[111] 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.

[112] Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–631. ACM, 2017.

[113] Lumin Liu, Jun Zhang, SH Song, and Khaled B Letaief. Edge-assisted hierarchical federated learning with non-iid data. *arXiv preprint arXiv:1905.06641*, 2019.

[114] Yang Liu, Tianjian Chen, and Qiang Yang. Secure federated transfer learning. *arXiv preprint arXiv:1812.03337*, 2018.

[115] Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient vertical federated learning framework. *arXiv preprint arXiv:1912.11187*, 2019.

[116] Yang Liu, Yingting Liu, Zhijie Liu, Junbo Zhang, Chuishi Meng, and Yu Zheng. Federated forest. *arXiv preprint arXiv:1905.10053*, 2019.

[117] Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, Surya Nepal, and Robert Deng. Boosting privately: Privacy-preserving federated extreme boosting for mobile crowdsensing. *arXiv preprint arXiv:1907.10218*, 2019.

[118] Noel Lopes and Bernardete Ribeiro. Gpumlib: An efficient open-source gpu machine learning library. *International Journal of Computer Information Systems and Industrial Management Applications*, 3:355–362, 2011.

[119] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[120] Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *optimization Methods and Software*, 32(4):813–848, 2017.

[121] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[122] 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.

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

[124] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[125] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[126] Payman Mohassel and Peter Rindal. Aby 3: a mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 35–52. ACM, 2018.

[127] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. *arXiv preprint arXiv:1902.00146*, 2019.

[128] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar. Security and privacy in fog computing: Challenges. *IEEE Access*, 5:19293–19304, 2017. doi: 10.1109/ACCESS.2017.2749422.

[129] Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, pages 129–139, New York, NY, USA, 1999. ACM. ISBN 1-58113-176-3. doi: 10.1145/336992.337028. URL http://doi.acm.org/10.1145/336992.337028.

[130] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning*, page 0. IEEE, 2019.

[131] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N. Asokan, and Ahmad-Reza Sadeghi. DÏot: A federated self-learning anomaly detection system for iot, 2018.

[132] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2:2, 2018.

[133] Solmaz Niknam, Harpreet S Dhillon, and Jeffery H Reed. Federated learning for wireless communications: Motivation, opportunities and challenges. *arXiv preprint arXiv:1908.06847*, 2019.

[134] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *2013 IEEE Symposium on Security and Privacy*, pages 334–348. IEEE, 2013.

[135] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[136] Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Entity resolution and federated learning get a federated resolution. *arXiv preprint arXiv:1803.04035*, 2018.

[137] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious multi-party machine learning on trusted processors. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 619–636, 2016.

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

[139] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[140] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[141] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[142] Davy Preuveneers, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth Ilie-Zudor. Chained anomaly detection models for federated learning: An intrusion detection case study. *Applied Sciences*, 8:2663, 12 2018. doi: 10.3390/app8122663.

[143] Yongfeng Qian, Long Hu, Jing Chen, Xin Guan, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi. Privacy-aware service placement for mobile edge computing via federated learning. *Information Sciences*, 505:562–570, 2019.

[144] Santu Rana, Sunil Kumar Gupta, and Svetha Venkatesh. Differentially private random forest with high utility. In *2015 IEEE International Conference on Data Mining*, pages 955–960. IEEE, 2015.

[145] M Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M Songhori, Thomas Schneider, and Farinaz Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 707–721. ACM, 2018.

[146] Bita Darvish Rouhani, M Sadegh Riazi, and Farinaz Koushanfar. Deepsecure: Scalable provably-secure deep learning. In *Proceedings of the 55th Annual Design Automation Conference*, page 2. ACM, 2018.

[147] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[148] 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.

[149] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. IEEE, 2015.

[150] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[151] Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and Merouane Debbah. Federated learning for ultra-reliable low-latency v2v communications, 2018.

[152] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[153] Ashish P Sanil, Alan F Karr, Xiaodong Lin, and Jerome P Reiter. Privacy preserving regression modelling via distributed computation. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 677–682. ACM, 2004.

[154] Yuris Mulya Saputra, Dinh Thai Hoang, Diep N. Nguyen, Eryk Dutkiewicz, Markus Dominik Mueck, and Srikathyayani Srikanteswara. Energy demand prediction with federated learning for electric vehicle networks, 2019.

[155] Yunus Sarikaya and Ozgur Ercetin. Motivating workers in federated learning: A stackelberg game perspective, 2019.

[156] 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.

[157] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[158] Amit P Sheth and James A Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)*, 22(3):183–236, 1990.

[159] Elaine Shi, T-H Hubert Chan, Eleanor Rieffel, and Dawn Song. Distributed private data analysis: Lower bounds and practical constructions. *ACM Transactions on Algorithms (TALG)*, 13(4):50, 2017.

[160] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[161] Dejan Skvorc, Matija Horvat, and Sinisa Srbljic. Performance evaluation of websocket protocol for implementation of full-duplex web streams. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1003–1008. IEEE, 2014.

[162] 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.

[163] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[164] Michael R Sprague, Amir Jalalirad, Marco Scavuzzo, Catalin Capota, Moritz Neun, Lyman Do, and Michael Kopp. Asynchronous federated learning for geospatial applications. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 21–28. Springer, 2018.

[165] Ivan Stojmenovic, Sheng Wen, Xinyi Huang, and Hao Luan. An overview of fog computing and its security issues. *Concurr. Comput. : Pract. Exper.*, 28(10):2991–3005, July 2016. ISSN 1532-0626. doi: 10.1002/cpe.3485. URL https://doi.org/10.1002/cpe.3485.

[166] Lili Su and Jiaming Xu. Securing distributed machine learning in high dimensions. *arXiv preprint arXiv:1804.10140*, 2018.

[167] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[168] Melanie Swan. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[169] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[170] ADP Team et al. Learning with privacy at scale. *Apple Machine Learning Journal*, 1(8), 2017.

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

[172] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11. ACM, 2019.

[173] Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 639–644. ACM, 2002.

[174] Jaideep Vaidya and Chris Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2003.

[175] Jaideep Vaidya and Chris Clifton. Privacy preserving naive bayes classifier for vertically partitioned data. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 522–526. SIAM, 2004.

[176] Jaideep Vaidya and Chris Clifton. Privacy-preserving decision trees over vertically partitioned data. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 139–152. Springer, 2005.

[177] Dinusha Vatsalan, Ziad Sehili, Peter Christen, and Erhard Rahm. Privacy-preserving record linkage for big data: Current approaches and research challenges. In *Handbook of Big Data Technologies*, pages 851–895. Springer, 2017.

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

[179] Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):57, 2018.

[180] Martin J Wainwright, Michael I Jordan, and John C Duchi. Privacy aware learning. In *Advances in Neural Information Processing Systems*, pages 1430–1438, 2012.

[181] Li Wan, Wee Keong Ng, Shuguo Han, and Vincent Lee. Privacy-preservation for gradient descent methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 775–783. ACM, 2007.

[182] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[183] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 63–71. IEEE, 2018.

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

[185] 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. *IEEE Network*, 2019.

[186] 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.

[187] China" "WeBank, Shenzhen. Federated learning white paper v1.0. In *https://www.fedai.org/static/flwp-en.pdf*, 2018.

[188] Zeyi Wen, Bingsheng He, Ramamohanarao Kotagiri, Shengliang Lu, and Jiashuai Shi. Efficient gradient boosted decision tree training on gpus. In *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 234–243. IEEE, 2018.

[189] Zeyi Wen, Jiashuai Shi, Bingsheng He, Jian Chen, Kotagiri Ramamohanarao, and Qinbin Li. Exploiting gpus for efficient gradient boosting decision tree training. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

[190] Zeyi Wen, Jiashuai Shi, Bingsheng He, Qinbin Li, and Jian Chen. ThunderGBM: Fast GBDTs and random forests on GPUs. *To appear in arXiv*, 2019.

[191] Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[192] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1307–1322. ACM, 2017.

[193] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

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

[195] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. Hybridalpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 13–23, 2019.

[196] Zhuang Yan, Li Guoliang, and Feng Jianhua. A survey on entity alignment of knowledge base. *Journal of Computer Research and Development*, 1:165–192, 2016.

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

[198] 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.

[199] Shanhe Yi, Zhengrui Qin, and Qun Li. Security and privacy issues of fog computing: A survey. In *WASA*, 2015.

[200] 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.

[201] Hwanjo Yu, Xiaoqian Jiang, and Jaideep Vaidya. Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 603–610. ACM, 2006.

[202] Hwanjo Yu, Jaideep Vaidya, and Xiaoqian Jiang. Privacy-preserving svm classification on vertically partitioned data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 647–656. Springer, 2006.

[203] Zhengxin Yu, Jia Hu, Geyong Min, Haochuan Lu, Zhiwei Zhao, Haozhe Wang, and Nektarios Georgalas. Federated learning based proactive content caching in edge computing. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[204] Jiawei Yuan and Shucheng Yu. Privacy preserving back-propagation neural network learning made practical with cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 25(1): 212–221, 2013.

[205] 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.

[206] Qingchen Zhang, Laurence T Yang, and Zhikui Chen. Privacy preserving deep computation model on cloud for big data feature learning. *IEEE Transactions on Computers*, 65(5):1351–1362, 2015.

[207] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[208] Lingchen Zhao, Lihao Ni, Shengshan Hu, Yaniiao Chen, Pan Zhou, Fu Xiao, and Libing Wu. Inprivate digging: Enabling tree-based distributed data mining with differential privacy. In *INFOCOM*, pages 2087–2095. IEEE, 2018.

[209] 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.

[210] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4): 352–375, 2018.

[211] Amelie Chi Zhou, Yao Xiao, Bingsheng He, Jidong Zhai, Rui Mao, et al. Privacy regulation aware process mapping in geo-distributed cloud data centers. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

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

[213] G. Zyskind, O. Nathan, and A. '. Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184, May 2015. doi: 10.1109/SPW.2015.27.