IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Challenges, Applications and Design Aspects of Federated Learning: A Survey

K M Jawadur Rahman[1], (Graduate Student Member, IEEE), Faisal Ahmed[1], Nazma Akhter[1], Mohammad Hasan[1], Ruhul Amin[1], Kazi Ehsan Aziz[1], A.K.M. Muzahidul Islam[2], (Senior Member, IEEE), Md Saddam Hossain Mukta[2], (Member, IEEE), and A.K.M. Najmul Islam[3]

[1] Graduate School of Science and Engineering, Master of Science in CSE, United International University, Dhaka, Bangladesh

[2] Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh

[3] LUT School of Engineering Science, Finland

Corresponding author: A.K.M. Muzahidul Islam (e-mail: muzahid@cse.uiu.ac.bd).

ABSTRACT Federated Learning (FL) is a new technology that has been a hot research topic. It enables training an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging them. There are many application domains where large amounts of properly labeled and complete data are not available in a centralized location, for example, doctors' diagnosis from medical image analysis. There are also growing concerns over data and user privacy as Artificial Intelligence is becoming ubiquitous in new application domains. As such, very recently, a lot of research has been conducted in several areas within the nascent field of FL. A variety of surveys on different subtopics exist in current literature, focusing on specific challenges, design aspects and application domains. In this paper, we review existing contemporary works in the related areas in order to understand the challenges and topics that are emphasized by each type of FL surveys. Furthermore, we categorize FL research in terms of challenges, design factors and applications, conducting a holistic review of each and outlining promising research directions.

INDEX TERMS Data privacy, Data security, Decentralized data, Distributed processing, Federated learning, Machine learning.

I. INTRODUCTION

In recent years, machine learning (ML) technologies have seen tremendous growth. The availability of large amounts of data is one of the reasons for this rapid growth of ML and Deep Learning (DL) based techniques/methods. However, not all application domains have large amounts of properly labeled and complete data available in a centralized location, for example, doctors' diagnosis from medical image analysis. Curating such large high-quality datasets can be time consuming and tedious and often requires domain experts. Efforts from individual organizations result in data silos with each one having high-quality but small datasets. In these application domains, very few organizations manage to gather high-quality, complete, fully-labeled and large enough datasets that are required for DL applications to be effective. Traditionally, data used to be gathered in a centralized location to build ML models. However, due to concerns over data ownership and data confidentiality, user privacy, and new laws over data management and data usage like General Data Protection Regulation (GDPR), there is a need for distributed model training in a private, secure, efficient and fair way.

Thus, instead of training on centralized data, separate models can be trained locally where the data resides in a distributed manner. Then, the respective local model updates can be communicated to obtain a global model. This is the idea behind Federated Learning (FL), where the process of communication is carefully designed such that the data of individual organization or device remain private. FL was first introduced by the researchers at Google to update language models [1], [2] in Google's keyboard system for word auto-completion. FL builds a joint model using data located at different sites, where each party contributes some data to train the model. It is important to note that the data belonging to each party does not leave their premises. The model is then encrypted and shared among the participants so that no participant can reverse engineer others' data. This resulting joint model's performance is an approximation of the ideal model trained with centralized data. In practice, this added security and privacy results in some accuracy loss, but it is often worth it for specific application domains. In addition to the privacy and security benefits, collaborative training in FL can result in better models compared to models trained by individual organizations or devices.

FL architecture can follow client-server model (see Fig. 1) or peer-to-peer model (see Fig. 2) at the fundamental level. In client-server model, there is a coordinator responsible for centrally aggregating model parameters using federated averaging.

First, an initial model is sent by the coordinator to each participating client. Each client then locally trains individual learning models using their own local datasets and send the model updates back to the coordinator for aggregation. After aggregation, the combined model updates are again sent back to local participating client. This process repeats until the model converges or preset number of iterations is reached. The client- server architecture incurs less communication overhead. On the other hand, the peer-to-peer architecture is even more secure since the participating clients communicate directly without a third-party coordinator. The trade-off, however, is that peer-to-peer architecture requires more computation for message encryption and decryption.

There are three fundamental categories of FL that depend on data partitioning among participants in feature and sample

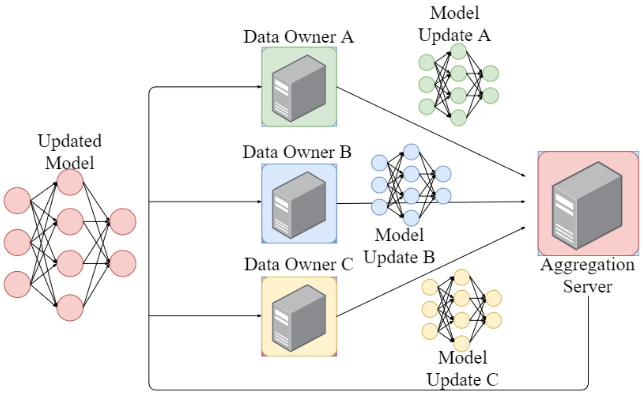

FIGURE 1. Client-server FL architecture

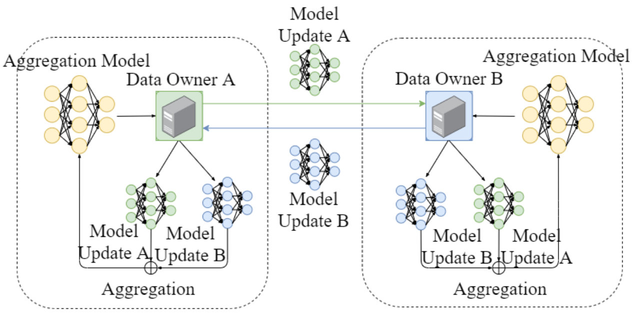

FIGURE 2. Peer-to-peer FL architecture

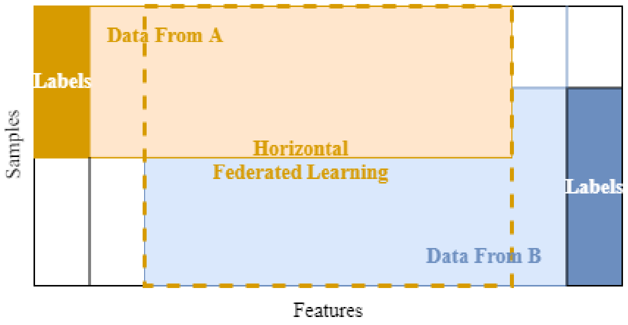

FIGURE 3. Horizontal FL architecture

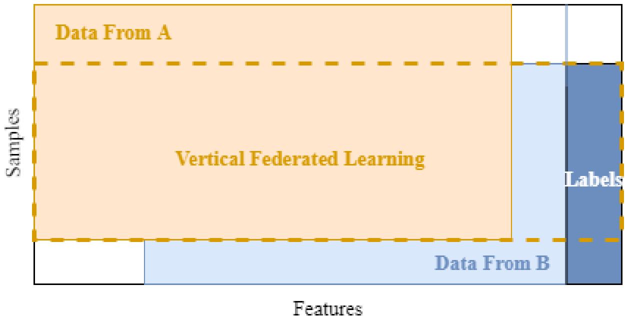

FIGURE 4. Vertical FL architecture

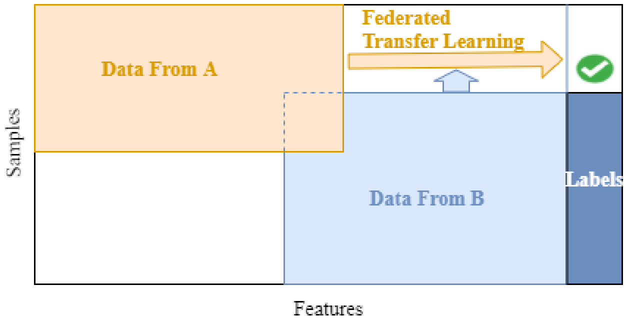

FIGURE 5. Federated Transfer Learning (FTL) architecture

spaces – Horizontal FL (see Fig. 3), Vertical FL (see Fig. 4), and Federated Transfer Learning (FTL) (see Fig. 5). For Horizontal FL, there is alignment in data features across participants, not in data samples. The exact opposite applies for Vertical FL where there is alignment in data samples, not in data features. Horizontal and Vertical FL can be ineffective when the data is highly heterogeneous. In such cases, FTL is an effective approach where it transfers knowledge learned from source domain to a target domain. FTL is inspired by transfer learning, where machine learning models that are trained on a dataset belonging to one domain are re-used and fine-tuned to solve a problem of a related domain.

The aforementioned architecture and FL categories are only the tip of the iceberg in the field of FL. There are numerous research thrusts such as novel architectures, data partitioning schemes and aggregation techniques. Moreover, current research efforts aim to mitigate the core challenging issues in FL like privacy and security, communication costs, system and statistical heterogeneity, personalization techniques, among others. Depending upon the application area where FL method gets utilized, unique application and domain-specific challenges and considerations also arise.

A lot of research has already been conducted in the field of FL in recent years. Consequently, numerous survey papers have also been written to summarize different focus areas. In this paper, we first conduct a review of existing surveys. The surveys cover a variety of domains and focus areas in FL research.

Several core challenges such as privacy, security, communication cost, system and statistical heterogeneity, architecture and aggregation algorithm designs, etc. vary by domain and specific use cases. The motivation of this paper is to review the current body of literature and summarize the current state-of-the-art approaches that have recently been developed to deal with these challenges. Our work also identifies the gaps in the reviewed FL surveys and fills them by surveying the latest developments in all aforementioned FL areas of research. We conduct a holistic review of the challenges, applications and design factors, and outline promising future research directions.

We have studied papers in the related areas and also have reviewed in depth most of the contemporary survey papers in these areas. We classify the topics in the FL survey papers according to the following categories: communication cost, statistical heterogeneity, systems heterogeneity, privacy/security as the core challenges; data partitioning, FL architectures, algorithms/aggregation techniques, personalization techniques as the implementation details; and applications of FL in different industries and domains.

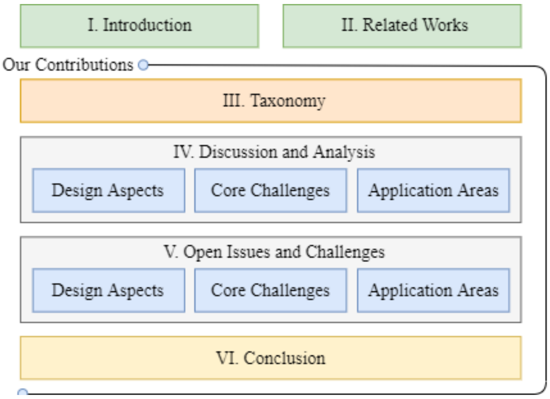

FIGURE 6. Organization of the paper

Our contributions in this paper are as follows:

1) Thoroughly investigated and analyzed contemporary FL survey papers.
2) Classified FL research into broad categories of design aspects, challenges, and application areas.
3) Conducted a holistic survey of the design aspects – data partitioning, FL architectures, aggregation techniques, personalization techniques; the core challenges – communication cost, systems heterogeneity, statistical heterogeneity, privacy/security; and, different application areas.
4) Discussed open issues and challenges in FL research.

The remainder of the paper is organized as shown in Fig. 6. In Section II, we discuss the Related Works. Section III illustrates the Taxonomy of the survey papers and discusses them in detail. Discussion and Analysis of all topics under each category are covered in Section IV. Section V discusses the Open Issues and Challenges in FL. Finally, Section VI concludes the paper.

II. RELATED WORKS

In this section, we investigate and analyze most of the contemporary survey papers. The reviewed papers are listed in Table I along with their summary and main focuses.

Li, Sahu *et al.* [3] discuss about how federated learning (FL) is different from standard distributed Machine Learning (ML). FL's unique characteristics and challenges are discussed, along with its current methods and future scopes. The paper does not focus on any specific domains and discusses approaches that deal with 4 core challenges, namely expensive communication, systems heterogeneity, statistical heterogeneity and privacy/security. Local updating [1], [4] is one approach to reduce the number of communication round. Compression schemes [5], on the other hand, reduce the message sizes at each rounds of communication. And, decentralized training [6], [7] decreases the burden on the central server in terms of communication. For systems heterogeneity challenges asynchronous communication [8]–[10] reduces stragglers, active sampling selects or influences participating devices based on system resources and overheads incurred, and fault tolerance [11]–[16] ignores failed devices utilizing algorithmic redundancy. Statistical heterogeneity issues are dealt by modeling heterogeneous data using methods like meta-learning, multi-task learning, adapting selection between global model and device-specific models, transfer learning for personalization. Studies also focus on convergence guarantees for non-independent and identically distributed data (non-IID) data [4], [10], [17], [18]. Lastly, this survey covers Secure Multi Party Computation (SMC) [19], [20] and Differential Privacy (DP) [21]–[24] approaches.

In [25], the focus area is Mobile Edge Networks. The core challenges in this survey include expensive communication, systems heterogeneity and privacy/security. Under communication cost challenges, approaches discussed include compression schemes such as model compression [26], [27], importance-based updating for selective gradients [28] or local model updates [29], and local updating [1], [30]–[32] focused on edge and end computation. Works mitigating systems heterogeneity include active sampling based on computation capabilities [33], data characteristics

[34], resource consumption [35] and allocation [36], [37]; joint radio and computation resource management by using superposition property of multiple-access channel [38]–[40]; asynchronous communication [41] for model aggregation; adaptive aggregation based on resource constraints [42], incentive mechanisms such as Stackelberg game [43]–[46], contract theoretic approach [47], [48], reputation mechanism [49] to encourage source contribution, and effective worker selection. For privacy/security challenges, information exploiting attacks are countered by DP [23], [50], selective participants [50], selective parameter sharing [51], secret sharing scheme [52], GAN model training [53]; data poisoning attacks are countered by distinguishing honest participants based on their gradient updates [54]; model poisoning attacks are countered by comparing updated models [55]; and free-riding attacks are countered by verifying local model updates [56].

The primary focus of [57] is privacy/security for Internet-of-Things (IoT). The approaches discussed in this survey are limiting the effects of individual client updates [57], [58], distinguishing honest participants [54], DP [23], [50], [51], SMC [19], [54], and Homomorphic Encryption (HE) [59].

Privacy/security is the main focus area of [60], particularly approaches like HE [61], [62], SMC [19], [63] and DP [64], [65] are covered in this survey. Data partitioning schemes, namely Horizontal FL [66], Vertical FL [67]–[70] and Hybrid FL [71], [72], as well as centralized [11], [73] and decentralized [74] design for communication architecture, and cross-silo [75] vs cross-device [76], [77] FL for scale of federation are the other challenges and approaches discussed here.

Work by Li *et al.* [78] is centered around applications, particularly in the domains of mobile devices, industrial engineering and healthcare. Applications of FL in mobile devices – predict user input [66], [79], [80], emoji [81], human trajectory [82], human behavior [83]; reduce network congestion [84]; detect physical hazards (smart-home IoT) [85]; industrial engineering – Environmental monitoring [86]; visual inspection [87]; malicious attack detection (Unmanned Aerial Vehicles); prevent energy congestion (charging stations); detect credit card fraud; spam filtering; sentiment analysis; healthcare – predict future

TABLE I
SUMMARY TABLE OF SURVEY PAPERS AND MAIN FOCUS

Survey Paper	Summary	Main Focus
Li, Sahu, *et al.* [3]	Discusses the unique characteristics and challenges of FL, provides details of current approaches, outlines directions of future work.	Challenges
Lim *et al.* [4]	Highlights challenges of FL implementation and existing solutions and presents applications of FL for mobile edge network optimization.	Mobile edge networks
Briggs *et al.* [5]	Focusing on IoT, covers works related to FL challenges and privacy preserving methods, identify the strengths and weaknesses of different methods applied to FL, and outlines future directions.	IoT, privacy/security
Li, Wen, *et al.* [6]	Categorizes FL systems according to six different aspects to facilitate and guide the design of FL systems, provides case studies and future research opportunities.	FL systems
Li, Fan, *et al.* [7]	Illustrates the evolution of FL and reviews existing applications of FL in industrial engineering, mobile devices and healthcare.	Applications
Kurupathi, Maass [8]	Highlights existing privacy techniques and proposes applications of FL in industries.	Privacy/security, applications
Yang *et al.* [9]	Introduces a secure FL framework, which includes horizontal FL, vertical FL and federated transfer learning, and proposes building data networks among organizations based on federated mechanisms.	Architecture, applications
Xu *et al.* [10]	Provides a review for FL technologies mainly for biomedicine, and discusses the challenges, issues and potential of FL in healthcare.	Healthcare
Kulkarni *et al.* [11]	Highlights the need for personalization in FL and surveys research on the topic.	Personalization
Lyu *et al.* [12]	Introduces taxonomy of threat models and major attacks on FL, highlighting intuitions, techniques and assumptions adopted by different attacks and discusses future research directions.	Threat models and attack types
Aledhari *et al.* [13]	Provides a thorough summary of relevant protocols, platforms, challenges and real-life uses cases of FL.	Platforms, protocols, applications
Mothukuri *et al.* [14]	Provides a detailed study of security and privacy, and presents current approaches, challenges and future directions in FL.	Privacy/security

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111118, IEEE Access

hospitalizations, mortality and hospital stay time, mortality over drug utilization data; similar patient matching.

Another privacy/security focused survey [88] elaborates on the current approaches such as SMC, DP, HE, Private Information Retrieval. [89] also covers privacy/security approaches, namely SMC [90], DP [23], HE [91]. Moreover, the paper also discusses approaches for data partitioning – Horizontal FL [92], Vertical FL, FTL.

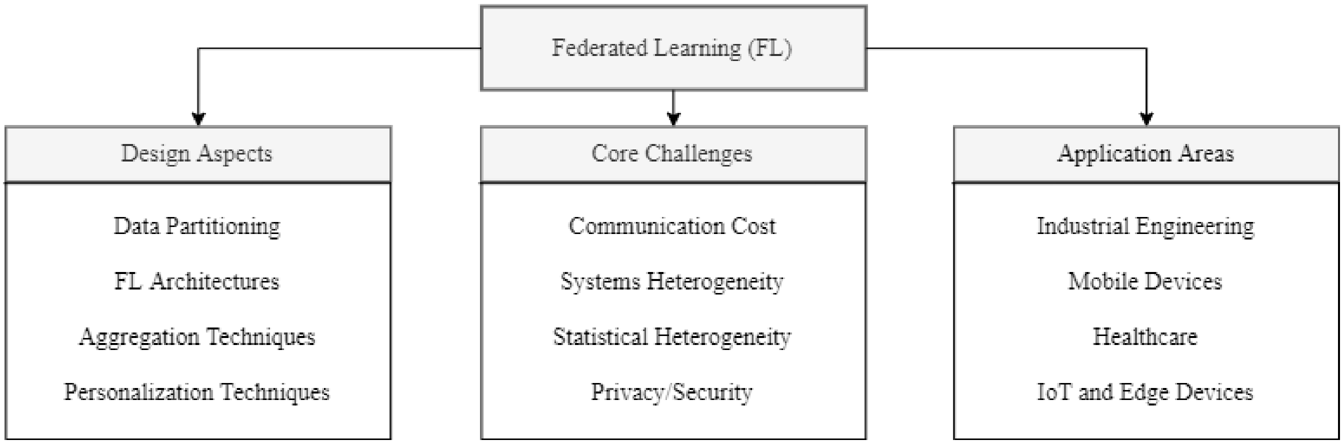

FIGURE 7. **Classification of the reviewed survey papers**

The challenges and approaches discussed in [93] is centered around healthcare domain. Consensus [37], [94] and pluralistic [95] solutions are mentioned to tackle statistical heterogeneity; client selection [33], compression schemes, updates reduction and peer-to-peer learning for expensive communications challenges; and SMC and DP for privacy/security challenges. The focus of [96] is on personalization techniques. The techniques discussed in this survey are adding user context [97], transfer learning [98], multi-task learning [99], meta-learning [100], knowledge distillation, base + personalization layers, mixture of global and local models. Article [101] is also based on privacy/security challenges. Specifically, it includes studies on threat models, different types of poisoning attacks and inference attacks.

Aledhari *et al.* [102] mainly focus on architecture options for FL based models – Horizontal FL [89], Vertical FL [89], MMVFL [103], FTL [71], FEDF [104], PerFit [105], FedHealth [106], FADL [107], Blockchain-FL [108], whereas a primary focus of [109] is on aggregation techniques – FedAvg [1], SMC-avg [19], FedProx [4],

TABLE II
COMPARISON OF TOPICS COVERED BY SURVEY PAPERS

Survey Paper	Data Partitioning	FL Architectures	Aggregation Techniques	Personalization Techniques	Communication Cost	Systems Heterogeneity	Statistical Heterogeneity	Privacy /Security	Application Areas
Li, Sahu, *et al.* [3]	✗	✗	✓	✓	✓	✓	✓	✓	✓
Lim *et al.* [4]	✗	✓	✓	✓	✓	✓	✓	✓	✓
Briggs *et al.* [5]	✗	✗	✓	✗	✓	✓	✓	✓	✓
Li, Wen, *et al.* [6]	✓	✓	✓	✗	✓	✓	✓	✓	✓
Li, Fan, *et al.* [7]	✓	✓	✓	✗	✓	✓	✓	✓	✓
Kurupathi, Maass [8]	✓	✗	✗	✗	✗	✗	✗	✓	✓
Yang *et al.* [9]	✓	✗	✗	✓	✓	✗	✓	✓	✓
Xu *et al.* [10]	✗	✗	✓	✓	✓	✗	✓	✓	✓
Kulkarni *et al.* [11]	✗	✗	✓	✓	✗	✗	✓	✗	✗
Lyu *et al.* [12]	✓	✗	✗	✗	✗	✗	✗	✓	✗
Aledhari *et al.* [13]	✓	✓	✓	✗	✓	✓	✓	✗	✓
Mothukuri *et al.* [14]	✓	✗	✓	✓	✓	✓	✗	✓	✓
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓

FedMA [110], Scaffold: Stochastic Controlled Averaging for FL [58], Tensor Factorization [111], FedBCD [31], Federated Distillation (FD) and Federated Augmentation (FAug) [18], Co-Op, LoAdaBoost [17], HybridFL [34], FedCS [112], PrivFL [113], VerifyNet [114].

The reviewed survey papers do not cover all the subtopics as highlighted in Table II. In particular, less than half of the surveys thoroughly reviewed FL architectures and personalization techniques. Our work classifies the topics as design aspects, core challenges, and application areas as shown in Fig. 7; and provides an in-depth discussion and analysis on all the subtopics.

III. TAXONOMY

The taxonomy of FL research, in terms of design aspects, core challenges and application areas, is portrayed in Fig. 7. Design aspects include data partitioning, FL architectures, aggregation techniques and personalization techniques. Communication cost, systems heterogeneity, statistical heterogeneity and privacy/security are among the core challenges. And, the reviewed survey papers mainly focus on the application areas of industrial engineering, mobile devices, healthcare, and IoT and edge devices. Comparison of topics covered by the survey papers is shown in Table II.

Data partitioning classifies FL as HFL, VFL or FTL as explained in the Introduction. Beyond these variants based on data partitioning, several specialized FL architectures have been developed to improve accuracy, training speed, efficiency, generalization, applicability, etc. for different areas like IoT, healthcare, Electronic Health Records (EHRs), privacy/security among others. Depending on the FL architecture used, aggregation techniques/algorithms are employed to integrate the local model updates from all participating clients during training to get the global model. Different aggregation techniques/algorithms have different priorities like increased privacy, optimal communication bandwidth, support of asynchronous updates, etc. Personalization is another design aspect that needs to be considered for certain scenarios, namely, device heterogeneity (storage, computation, and communication), data heterogeneity (i.e., non-IID data) and model heterogeneity (customized models depending on client's environment).

Expensive communication is a major challenge in FL systems. There can be a large number of devices in a federated network, which means that network communication is much slower than local computation. Therefore, a large body of work addresses communication efficiency. Moreover, there can be varying communication capabilities of devices in federated networks due to systems heterogeneity. The different devices may also have varying compute and storage capacities. Due to system and network constraints in a number of settings, only a few selected devices may participate during a training iteration, and some devices may even drop out during an iteration due to connectivity or power issues. Thus, FL techniques need to overcome such systems heterogeneity challenges. On the other hand, statistical heterogeneity issues arise due to violation of independent and identically distributed (IID) assumptions in distributed optimization. The violation occurs because different devices across the network often have non-identically distributed data. The number of data points across devices also vary. Therefore, FL approaches must handle statistical heterogeneity of data. Lastly, privacy/security issues are at the core of FL applications. Increased privacy/security using novel methods often comes at the cost of decreased system efficiency or model performance.

All these tradeoffs between the various application-specific challenges and design aspects need to be carefully considered and have to well-balanced to obtain effective privacy-preserving FL systems. These topics are discussed in greater detail in the following section.

IV. DISCUSSION AND ANALYSIS

In this section, we review and discuss the design aspects, core challenges and application areas to provide a comprehensive summary of subtopics – data partitioning, FL architectures, aggregation techniques, personalization techniques, communication cost, systems heterogeneity, statistical heterogeneity, privacy/security, and application areas.

A. DESIGN ASPECTS

Data partitioning: The data that are used for training FL is non-identical as the data is on various devices. The sample space of a dataset consists of all the dataset instances, while the feature space consists of the different dataset attributes. For instance, two hospitals may have records of different sets of patients (sample space), and they may also have different types of information stored about each patient in their electronic health records (feature space). Based on how the data is allocated over the sample and feature spaces across multiple participating devices in the FL process, FLSs can be typically categorized as horizontal, vertical and hybrid FL (aka FTL) [89].

1) *Horizontal Federated Learning (HFL)* is used in the scenarios in which the feature space of the datasets is same but the sample space differs. In HFL, the datasets belonging to different organizations have same feature space but the sample space are not that related. This type of is data partitioning is suitable for cross-device mode, where individual users use FL to try and enhance the performance of their model on a task. In FL, the horizontal partitioning is more common. Since the local data overlaps feature space, each individual user can train their local models using the duplicate model architecture. For example, two regional branches of an organization have different group of users but they have the same feature spaces as the

business is same. At present, the primary focus area of FLSs are smart devices and devices in the IoT. This work on FL from Google [78] falls into horizontal partitioning paradigm. In this framework, an individual user in android platform changes the model parameters locally and sends the updated parameters to the cloud server. This enables to train the centralized model along with other users. Furthermore, to deal with the issue of finite labeled entities, hierarchical heterogeneous HFL framework is proposed in [115]. Heterogeneous framework can address the shortage of label by adapting each user multiple times as target domain. The authors in [51] suggested a collaborative deep-learning framework where each user train independently and only share a subset of parameters for updating.Classified FL research into broad categories of design aspects, challenges, and application areas.

2) In *Vertical Federated Learning (VFL)*, the datasets across institutions share same or similar sample space but their feature spaces do not have much in common. In this setting, all the participants have homogeneous data which implies that they differ in feature space but have partial match on sample space. For example, two different organizations in a certain area want to train a machine learning model in collaboration. They have identical clients but the data of each organization are of distinct types. Due to privacy and security concerns, they cannot interchange their data. In scenario like this, VFL is suitable to train the model. VFL models aggregate these distinct features and calculates the model parameters in a privacy-preserving manner. Finally, it constructs a model by combining the data from both parties. An approach using linear regression was proposed by the authors in [116], [117] for data having vertical partitioning. Moreover, for such data, several secure models including k-means [70], association rule mining [67], decision tree [69] and naive bayes classifier [68] were proposed by Vaidya *et al*. Usually, VFL systems perform entity alignment [118], [119] to combine the common samples of different institutions. Then, employing encryption, the combined data are used for training the model. Cheng *et al.* [120] propose a lossless VFL system to enable joint training of gradient boosting decision trees. To recognize common users between two distinct parties, they make use of privacy-preserving entity alignment. Finally, those selected samples are used to train the decision trees collaboratively.

3) *Federated transfer learning (FTL)* is used in situations where two datasets differ in sample as well as feature space. FTL was first proposed in [71]. FTL enhances existing FL systems and can deal beyond the scope of existing FL algorithms. FTL gained enormous attention in various industries, especially in healthcare sector [121]. Various information related to treatment and diagnosis can be shared between hospitals to diagnose different diseases with the help of FTL. In general, transfer learning comprehends a common representation between the features of two different parties. Both parties still need to calculate the prediction results at the time of prediction. Hence, transfer learning [72] techniques can be adopted for the entire feature and sample space under a federated environment. To avoid the possibility of exposing the client data, FTL takes advantages of encryption and approximation to make sure the privacy is safeguarded. Hence, both the actual sensitive data and models are preserved locally [122]. *Sharma et. al.* work on improvement for FTL by integrating a secret sharing technology [123]. Authors of [124], [125] build a FedHealth model which collects data from different institutions via FL and provides customized services for healthcare by utilizing transfer learning.

There are advantages and benefits of using each of the afore-mentioned data partitioning paradigms. For example, two different clinics or hospitals can benefit from securely sharing data with each other based on either the number of instance or features that they need. One clinic could own millions of patient records, but it is possible that they only have very specific information about these patients based on their specialty, e.g., oncology. On the other hand, another clinic could be relatively new with a much smaller number of patient records in their possession. However, if this is a general clinic without a specialty, then it is likely that they own different types patient information. The first clinic would benefit from VFL, while the second one would benefit from HFL. Finally, via FTL, healthcare providers can provide more personalized care if they are given access to data from users' wearable devices for personal fitness.

FL architectures represents how different components are integrated to form an FL environment. Two common architectures of FL are client-server architecture and peer-to-peer architecture.

1) In *client server architecture*, as illustrated previously in Fig. 1, a central server initiates a global model that it shares with the clients to train on their local dataset. After local training, trained models from the clients that are involved in the FL environment, are collected by the server. The server then aggregates the models' parameter to build a global model, and shares it back with all clients. Client-server architecture is also known as centralized architecture for FL. Here the server coordinates the learning process which is continuous. In the conventional client-server architecture, the server hosts a model and trains the model on shared data. However, the server in the federated learning setting operates only on the local

models received from the clients synchronously or asynchronously. The main advantage of this architecture is it incurs less communication overhead. Google used this architecture to develop a virtual keyboard called Gboard for Android. Currently, almost all implementation of FL use client-server architecture.

2) As illustrated in Fig. 2, there is no concept of a central server *in peer-to-peer architecture* like in the client-server architecture for model aggregations. The role of central sever is replaced with algorithms to ensure security and reliability. Each participant in FL environment has its model. A participant improves its model by using the information from its neighbors [126]. In the adopted peer-to-peer topology, a protocol is established using a central authority. During training rounds, the network follows this protocol. Such architecture is more secure since the participating clients communicate directly without a third-party coordinator [127]. However, it requires more computation for message encryption and decryption.

The **aggregation algorithm** describes how the global model is formed by combining the local model updates from all the clients that participated in the train round. It plays a significant role in horizontal federated learning based on a centralized architecture. The most popular aggregation algorithms are compared in Table III and summarized below.

1) The Federated averaging (FedAvg) algorithm [1] proposed by Google is based on an Stochastic Gradient Descent (SGD) optimization algorithm. This aggregation algorithm is the best fit for HFL with a client-server architecture. In this algorithm, the server starts the training process by sharing the global model parameters with a group of clients selected randomly from a pool of clients. The clients then perform multiple epochs of SGD on their local dataset to train the global model and share the locally trained model with the server. The server next computes the weighted average of all the local models to generate a new global model. This process is repeated for several rounds. It is robust to unbalanced and non-IID data distribution. Although FedAvg has achieved great success, it has some convergence issues in some settings due to the factors like client drifting [58] and lack of adaptive learning rate [128].

2) Scaffold (Stochastic Controlled Averaging Federated Learning) [58] solves the problem of client drifting using variance reduction technique in its local update. It estimates the update direction of the server model and update direction of each client. From the difference it measures the client drifting which is then used to local update. This strategy helps to overcome the problem of client heterogeneity and reduce the communication round in model convergence.

3) Adaptive Federated Optimization [128] proposed by Google Research team introduces the adaptability in server optimization. The server optimization is more informed as the adaptive learning rates allow knowledge to be incorporated from previous iterations. In this optimization framework, a client optimizer minimizes the loss using local data over multiple training epochs. Then, to update its global model, the server performs gradient-based optimization on the average of the model updates of clients. FedAvg is the special case when SGD is used as both client and server optimizer with server learning rate being 1. Although it incorporates adaptive learning rates in the server optimization, it does not increase client storage or communication costs. Moreover, it is compatible with cross-device FL. However, it does not completely remove the effect of client heterogeneity. But for moderate, naturally arising heterogeneity, adaptive optimizer is quite effective, especially in cross-device setting.

4) FedBoost [129] is a communication-efficient algorithm for FL based on ensemble learning technique. In this approach an ensemble of pre-trained base predictors is trained via FL. It reduces the cost of both server-client and client-server communication without gradient

TABLE III
COMPARISON OF AGGREGATION ALGORITHMS

Aggregation algorithm	Overcome client drifting?	Adaptive learning rate?	Cross-device compatible?	Communica-tion-efficient?	Address client heterogeneity?	Ensure privacy?
FedAvg [1]	✗	✗	✓	✓	✓	✗
Scaffold [67]	✓	✗	✗	✓	✓	✗
Adaptive Federated Optimization [128]	✗	✓	✓	✓	✗	✗
FedBoost [129]	✓	✗	✗	✓	✓	✓
FedProx [15]	✓	✗	✗	✓	✓	✗
FedMA [110]	✗	✗	✗	✓	✓	✗
Secure Aggregation [30]	✗	✗	✓	✓	✗	✓

compression and model compression approach. In addition to communication efficiency, other advantages of this method include computational speedups, convergence guarantees, privacy and optimality of the solution for density estimation for which language modelling is a special case.

5) FedProx [4] addresses the two inherent challenges of FL. First one is system heterogeneity which means significant variable characteristics of system or device participating in FL. Second one is statistical heterogeneity that implies non-IID data across the network. It is a re-parametrized and generalized version of FedAvg. Specifically, FedProx is modified in two ways. First, it enables partial work to be tolerated. Based on availability of resources, a device can perform variable amounts of work locally, e.g., each device can run a variable number of local epochs. The partial solutions from the resource-constrained devices are accepted for aggregation. Secondly, a proximal term is introduced in a device's local solver objective to control the impact of the variable amounts of local updates.

6) Federated Matched Averaging (FedMA) [110] algorithm is proposed for introducing FL in modern network architectures for deep learning. Matching and averaging, based on similarity of features, is performed layer-wise across the channels of convolutional layers, across the hidden states of Long Short-Term Memory (LSTM) networks, and across fully-connected layer neurons to construct the shared global model at the server. FedMA can also handle client heterogeneity. Within a few rounds of training, it performs better than FedProx and FedAvg.

7) Secure Aggregation [19] algorithm is developed based on the principle of Secure Multiparty Computation (SMC) algorithm. It does not share any information with each other except the learnable parameters derived from aggregation and thus defends the privacy of each client's model. It is fault-tolerant up to 1/3rd of users, i.e., it works well even if 1/3rd of the clients fails to engage in the aggregation.

Personalization techniques: In FL, the goal is to train models with a central repository without changing their data samples. Personalization needs to adapt global model for individual client and permit users to acquire a richer model so that users' models are trained over a bigger set of data samples. Wu *et al.* [105] mention three major challenges handled by FL process during personalization. The challenges are: 1) device heterogeneity for communication capabilities, storage and computation, 2) data heterogeneity because of non-IID, and 3) model heterogeneity for different model at personalized situation.

TABLE IV
SUMMARY OF PERSONALIZATION TECHNIQUES

Research article	Personalization technique	Algorithm
McMahan *et al.* [133]	Adding user context	Context featurized
Mansour *et al.* [100]		Clustering
Wang *et al.* [134]	Transfer learning	
Smith *et al.* [102]	Multi-task learning	MOCHA
Finn *et al* [135]	Meta-learning	MAML
Fallah *et al.* [103]		Per-FedAvg
Khodak *et al.* [136]		ARUBA
Li *et al.* [137]	Knowledge distillation	FedMD
Arivazhagan *et al.* [138]	Base and personalization layers	FedPer
Hanzely *et al.* [139]	Mixture of global and local models	LLGD

Adding contextual features to datasets in a privacy-preserving manner can lead to predictions that are more personalized. Moreover, based on similarity of clients' data, different groups can be formed and a different model can be trained for each similar cluster [97]. Transfer learning can also be used in a federated setting for model personalization [130]. In transfer learning, the knowledge from a global model is transferred to local models. The local model parameters are then fine-tuned using local data. Other approaches like multi-task learning and meta-learning solve multiple tasks simultaneously. The joint learning in multi-task learning enables the model to make use of the differences and similarities across the tasks. Meta-learning produces models that are quite adaptive, and can solve new task with much less training data. Both meta-learning [99] and multi-task learning [100], [131], [132] algorithms have been proposed in the federated setting to achieve greater personalization. Knowledge distillation is another method where a student network mimics a larger teacher network. Using transfer learning and knowledge distillation, Li *et al.* [133] propose an FL framework that allows clients to design their own networks independently. Arivazhagan *et al.* [134] propose a neural network architecture, where global data is used to train only the base layers, while the personalization layers are trained on local data. A new gradient descent variant called Loopless Gradient Descent (LLGD) by Hanzely *et al.* [135] allow each device to learn a mixture of its own local model and the global model. We summarize the different techniques of personalization in Table IV.

B. CORE CHALLENGES

Communication is a basic bottleneck in federated networks, which coupled with security concerns over sending crude information, requires that information produced on

TABLE V
STRATEGIES AND APPROACHES TO REDUCE COMMUNICATION COSTS

Research article	Strategies	Contributions	Future concerns
Rothchild *et al.* [130]	Compression	A Count Sketch is used to compress the updates of model. Then, leverages sketch mergeability.	Explore effective ways to combine efficiency within a round and efficiency in number of rounds.
Reisizadeh *et al.* [131]	Local Updating	Method with models periodically averaged at the server and quantized uploads from edge devices.	Can experiment further with the trade-offs made between communication and computation.
Konecny *et al.* [36]	Compression	Structured updates from a restricted space; sketched updates using multiple techniques together like random rotations, subsampling and quantization.	Experimentation with selection of variables used to parametrize space.
McMahan *et al.* [1]	Local Updating	Locally computed SGD updates on each client is sent to a server, which then performs model averaging.	Mitigating the straggler problem.
Han *et al.* [132]	Compression	Pruned the network, quantized the weights and applied Huffman coding.	The quantized network with weight sharing needs to benchmarked on various hardware.

each device stay local. To overcome this, researchers have come up with several strategies some of which involve local updating, compression schemes, decentralized training, and importance-based updating.

Local-updating schemes tackle communication costs by performing additional work on the client that generates and consumes the machine learning model. As an extension of classical stochastic methods, mini-batch optimization methods have proven to be successful in many cases [136]. For both convex and non-convex objectives, distributed local-updating primal methods have also been applied with success [137]. Since the pivotal FedAvg algorithm proposed in [1], many directions have been taken that include quantizing uploads from edge devices [138].

Sketched and structures updates are among the compression schemes that enable the reduction of the model update size communicated to the FL server from the participating clients during each round [26], [139]. Subsampling, probabilistic quantization, and sparsification are also considered in [140]. The authors in [27] further extend the work of [26] to reduce the cost of communication from the server to participant, employing approaches such as federated dropout and lossy compression. The accumulation of error and momentum is handled by the central aggregator instead of the clients in [141].

Recent works like [6] have carried out decentralized training over heterogeneous data. Hierarchical communication patterns [142] is another approach that reduces the dependency on central server. First, updates from edge devices are aggregated on the edge servers. Then, from the edge servers, the updates are aggregated on the cloud servers.

Important-based updating utilizes the fact that most parameter values of a deep neural network model are sparsely distributed. Edge Stochastic Gradient Descent (eSGD) algorithm has been proposed in [28]. For updating parameters in each round of communication, only selected important gradients are sent to the server. Authors in [29] have come up with the Communication-Mitigated Federated Learning (CMFL) algorithm, which reduces the cost of communication by only uploading updates of the local model that are relevant. Global convergence is still guaranteed. A comparison is first made between the local update of a participant and the global update during every iteration, in

TABLE VI
STRATEGIES AND APPROACHES FOR MANAGING SYSTEMS HETEROGENEITY

Research article	Strategies	Contributions	Future concerns
Yang *et al.* [140]	Client Participation, concerned with number of clients	FLASH, an FL simulation platform for developers and researchers.	Experiments were conducted using geo-specific data, yet to try with more diverse data.
Nishio *et al.* [43]	Client Participation, concerned with number of clients	FedCS, Federated Learning with client selection.	Yet to train a more complex model with several million parameters.
Anelli *et al.* [141]	Client Participation, concerned with amount of data interaction by clients	Improved aggregation by measuring contribution of each device based on multiple criteria.	Identification of other local criteria, both general purpose and domain-specific.
Xu *et al.* [142]	Client Participation, concerned with amount of data interaction by clients	ELFISH, a “soft training” method for straggler acceleration, with corresponding aggregation scheme.	Needs further exploration with non-IID datasets.
Caldas *et al.* [37]	Client Participation, concerned with amount of data interaction by clients	Federated dropout, facilitates efficient local training by allowing users to train on subsets of the global model.	Studying the effect of adaptively using these strategies to prevent unfairly biased models.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111118, IEEE Access

order to assign a relevance score to the update. Strategies and approaches for reducing communication costs are summarized in Table V.

Systems heterogeneity: Due to having differences in connectivity of network, memory, CPU, battery power level, etc., participants in a federated network often have varying capacity in terms of communication, computation and storage. Straggler mitigation and other challenges are further compounded due to these system-level characteristics. Popular approaches include asynchronous communication, client participation, and fault tolerance.

Straggler mitigation in heterogeneous environments using asynchronous communication schemes [10] is a promising approach. When there is device variability, synchronous approaches are more susceptible to stragglers. However, asynchronous communication also suffers from bounded-delay assumptions made to control the measure of staleness.

Client participation schemes involve actively selecting participating devices based on systems resources like FedCS [33] and quality of data [47] at each round. The FedCS protocol is extended by the authors in [20]. Their Hybrid-FL protocol addresses the differences that exists in data distributions of participating clients. Deep Q-Learning [35] is also used to optimize allocation of resources needed for training models. Client participation is controlled on the aspect of the number of clients in [143] and on the aspect of the amount of data contributed or consumed by clients in [27], [144], [145].

Fault Tolerance [99] is used because learning over remote devices becomes more critical as some devices in the network often drop out even before an iteration is completed. Introducing algorithmic redundancy to tolerate device failures is another option known as coded computation. Authors in [15] have explored using codes to increase the speed of distributed training. Strategies and approaches for managing systems heterogeneity are summarized in Table VI.

Statistical heterogeneity refers to the existence of non-IID data across the network. The data that is generated and collected by the devices in the network are usually non-identically distributed. This causes complexity in terms of analysis, modeling, and evaluation. The usage patterns of different users are distinct. For some clients, the global shared model does not perform as well as local models that are trained locally. So, they are disincentivized to take part in the federated network. Moreover, there can be significant variance in terms of the amount of data per device. Also, possible presence of underlying structures can capture the relationship between the devices and their distributions.

In general, an FL system focuses on learning a single global model. There also exists other approaches such as learning distinct local parameters simultaneously via multi-task learning frameworks [99]. The authors of [146] have developed tools to measure statistical heterogeneity using metrics such as local dissimilarity. Although, calculating these metrics is quite difficult for a federated network before the training begins. The significance of these metrics influences future directions to the development of efficient algorithms to quickly quantify the heterogeneity in an FL system.

TABLE VII

SUMMARY OF FL THREAT MODELS

Research article	Attack type	Threat model	Attack strategies	Attack target
Shafahi *et al.* [148]	Poisoning Attack	Data Poisoning	"Watermarking" strategy (frog image)	Data
Gu *et al.* [149]			Backdoor attack	
Bhagoji *et al.* [65]		Model Poisoning	Stealth metrics, boosting of malicious agent's updates, parameter estimation for the benign agents' updates	Model
Fang *et al.* [150]			Manipulate global model via local model parameter manipulation on compromised devices	
Bagdasaryan *et al.* [151]			Backdoored image-classification model, backdoored word-prediction model	
Melis *et al.* [152]	Inference Attack	Membership inference, inferring properties	Gradient exchange	Data
Pyrgelis *et al.* [153]		Membership inference	ML classifier	
Zhu *et al.* [154]		Inferring training inputs (and labels; inconsistently)	DLG	
Zhao *et al.* [155]		Inferring training inputs and labels	iDLG	
Hitaj *et al.* [156]		Inferring class representative	GAN attack	

To tackle statistical heterogeneity, the authors in [134] utilizes the concepts from multitask learning. In FEDPER approach, the participants use a set of base layers pre-trained with the FedAvg [1] algorithm. Finally, each participant individually trains another set of layers using their local data. The authors empirically show that the FEDPER approach outperforms a pure FedAvg approach using the Flickr-AES dataset [134] considering the personalization layers are capable of representing the personal predilection of an FL user.

FL threat models: FL offers an emerging paradigm for facilitating multiple organization data collaboration without revealing their private data to each other. But recent research demonstrated that FL may not always provide sufficient privacy guarantee during model update. FL may face several vulnerabilities from both server and participants. As summarized in Table VII, per the threat models, two prominent forms of attacks that take place are:

1) *Poisoning Attacks* may be executed either on the training phase of the model or on the data. Two types of poisoning occur:
    a) *Data poisoning* take place for the period of collecting local data. Data poisoning attacks may occur in two ways such as clean-label attack (adversary can poison correct class of data sample) and dirty-label attack (adversary try to misclassify the target label of training dataset) [147], [148].
    b) *Model poisoning* take place for the period of model training. According to Bhagoji *et al.* [55], model poisoning is accomplished by an adversary controlling few malicious representatives with the aim to misclassify specific inputs with high confidence. Bagdasaryan *et al*. [149] introduced a new scope of vulnerability of FL, inserting backdoor to the joint model. FL models are more vulnerable for model poisoning attack than data poisoning attack. This form of attack can be used to cause misclassification in image and next word prediction problem.
2) *Inference Attacks*: Serious privacy leakage may occur in FL during update of the model. When exchanging gradient, private information of participant may be exposed to the adversary [62], [150]–[152]. Pyrgelis *et al.* [153] also conduct membership inference attack to identify the vulnerability at aggregate location. According to the survey of threat model by Lyu *et al*. [101], the inference attack falls into two categories – white-box attack and black-box attack. Deep Leakage from Gradients (DLG) [152] obtained the private training data at the inference phase. Another algorithm, iDLG also expose the label of training inputs [154]. Hitaj *et al*. [155] apply GAN attack which allows the adversarial party in the training process of FL to fabricate inferring class representative.

Privacy is one of the most critical parts of FL. In this section, we briefly review different types of privacy and security techniques for FL:

1) *Secure Multiparty Computation* (SMC) is one of privacy mechanisms used in FL. SMC model comprises of multiple parties and provide proper security. This model assures that every party only knows its inputs and outputs. Every party knows nothing about other parties. Bonawitz *et al*. designed a communication-efficient SMC protocol for high-dimensional data to protect the privacy of users' model gradients [19].
2) *Differential Privacy (DP)* is a privacy preserving mechanism which protects individual privacy by adding noise in the data. There are various types of DP:
    a) *Local DP*: Each individual data point is distorted with noise.
    b) *Global DP*: To protect privacy of individuals, the output of the dataset query is distorted with noise.
    c) *Hybrid DP*: Combine multiple trust model by partitioning users by their trust model preferences.

Geyer et al. [4] developed a method to obtain differential privacy at client level for FL. Wei *et al*. [156] proposed an aggregation algorithm called NbAFL where they add noise to client-side parameters before aggregation. Authors in [157] used both SMC and Differential privacy mechanism to avoid differential attack.

3) *Homomorphic Encryption (HE)* is another security mechanism in FL. It protects user data by changing parameter under the encryption method. HE is a cryptographic technique that perform mathematical operations on data as if it was unencrypted. Many researchers worked with Homomorphic encryption for preserving privacy [158], [159]. In order to guarantee the privacy of users' local gradients during FL, Xu *et al.* [114] proposed a double-masking protocol.

Applications: Although FL faces some limitations and severe challenges, it has been implemented successfully in several real-life applications:

1) *Applications in NLP:* FL has become a hot topic to researchers since the concept was first introduced by Google to predict next word in virtual keyboard for smartphones [160]. Further improvement in predicting next word using pretrained word embeddings was made by other researchers [80]. Wake word detection was also another contribution by David *et al.* [66]. Emoji prediction from text typed on a mobile keyboard was introduced by Swaroop *et al.* [81]. Besides, some researchers worked on learning out of vocabulary words in virtual keyboard for smartphones [79] whereas some tried to improve virtual keyboard search suggestion quality [161].

2) *Applications in healthcare:* Huang *et al.* [17] predicted the mortality rate of patients suffering from heart disease by using electronic medical records from multiple hospitals. Brisimi *et al.* [74] use electronic health record (EHR) to determine whether a heart disease patient would be hospitalized or not. Li *et al.* [162] also worked on mortality and hospital stay time. Using health records, Lee *et al.* [163] proposed a way to find similar patients from different hospitals while preserving the patients' privacy. They used a federated patient hashing framework.

3) *Applications in computer vision*: Another important application area of FL is computer vision. Shao *et al.* [164] proposed Federated Face Presentation Attack Detection method. Liu *et al.* [165] worked on smart city safety monitoring solutions based on computer vision.

4) *Applications in transportation*: Development of intelligent transportation systems using FL is explored by Elbir *et al.* [166]. Lim *et al.* [167] proposed an FL based approach in UAV enabled internet of vehicles (IoV) for developing applications like management of car parking occupancy, and traffic prediction.

V. OPEN ISSUES AND CHALLENGES

There are several open issues and challenges in FL [168]. Tradeoffs among accuracy, privacy, communication cost, level of personalization, etc. have to be carefully considered when designing an FL system. Often such considerations are dependent upon the specific use case or application area. In this section, we discuss some open issues related to design aspects, core challenges and application areas.

A. DESIGN ASPECTS

Data Partitioning and FL Architectures: In addition to the primary forms of data partitioning schemes and FL architectures discussed in this work, other variations in FL architectures have recently been developed. For instance, PerFit [105] is cloud-based and enables personalized FL approaches to be selected flexibly, thus making it suitable for IoT applications. Another architecture is FedHealth [106], which uses FTL framework for wearable healthcare to build personalized models, thus enabling personalized healthcare services. Future works can focus on developing FL architectures schemes that facilitate the specific requirements of different industries and application areas to be met.

Aggregation Techniques: Developers who wish to implement FL solutions can benefit from toolkits that would offer standardized and pre-configured aggregation algorithms that are suitable for their specific application areas and use cases. Much like AutoML solutions, such a toolkit for FL would lower the barrier of entry for non-specialist developers.

Personalization Techniques: Adding suitable user and context features to the shared global model is a possible alternative to having device-specific personalization. For example, the order of the filters in applications like Snapchat can be arranged according to certain features of the user like browsing history, age, sex, likes and dislikes, usage patterns, etc. Thus, developing architectures that can accommodate such user and context features effectively for different tasks is another open problem.

Moreover, it is observed from [169] that a gap exists between the accuracy of personalized and global models, making a case for personalization techniques to be one of the important research areas in FL. Nevertheless, no clear metrics have yet been formulated to measure the performance of personalization techniques. Wang *et al.* [130] evaluate conditions under which personalization yields desirable models. Further research is required to develop comprehensive metrics to assess the effective of personalization approaches.

B. CORE CHALLENGES

Communication: There is a tradeoff between communication cost and accuracy in FL. Benchmarks in machine learning do not usually set any restriction criteria. It is worth considering setting the communication budget as a restriction criterion in communication-focused FL benchmarks. For example, [170], [171] explores one-shot or few-shot communication schemes in FL, and [171] tries to maximize performance for fixed rounds of communication (i.e., single or few rounds). Additionally, these methods need to be thoroughly evaluated and analyzed for the FL setting where the networks can be highly heterogenous.

In cross-device FL, often only a few devices are active during an iteration. There is scope for in-depth analysis of the consequences of this asynchronous communication scheme, where the devices become active based on certain events.

Systems Heterogeneity: Various algorithms [33], [35] have been proposed to address systems heterogeneity. However, wireless connectivity might not be available consistently, and as such many participating devices may drop from the FL system during training. Future works can design new FL algorithms that are more robust even when a larger number of devices drop out from the network due to connectivity issues.

Statistical Heterogeneity: Eichner et al. [95] developed a pluralistic solution to alleviate a form of data heterogeneity where devices exhibited different characteristics during the day versus at night. Further research can be conducted to explore similar methods to tackle diurnal variations at more granular times of day (instead of only day versus night) or at different times of the week.

Privacy/Security: While device-specific local or global level privacy is well studied and understood, finer privacy requirements at the sample level is promising, ongoing research. Sample-specific privacy guarantee technique by Li et al. [172] trade off privacy for a greater accuracy. Hybrid methods need be explored that deal with both sample and device level privacy requirements.

C. APPLICATION AREAS

FL has mainly been applied to supervised learning problems. Future research could attempt to tackle the challenges that may arise when using FL in applications that calls for data exploration, unsupervised, semi-supervised, and reinforcement learning.

The challenges faced in the implementation of FL solutions for different application areas have not yet been thoroughly studied, with the focus of current studies being primarily on the training of FL models. In addition to the core challenges discussed in this paper, issues that are specific to the industry domain or application area also needs to be considered. For instance, there are application areas like Mobile Edge Networks that would require energy-efficient communication to be greatly emphasized.

VI. CONCLUSION

FL allows participating organizations to collaboratively train prediction models without having to share their data. There has been a growing interest in FL research in both industry and academia in recent years. FL enables certain industries like healthcare to overcome challenges related to data collection and privacy.

This growing interest in FL motivated us to review most of the contemporary survey papers in FL and to classify FL into several topics under design aspects, core challenges and application domains. We thoroughly investigated and analyzed the FL survey papers, and conducted a holistic review of each FL topic. Finally, we outlined promising future research directions. This work will hopefully help future researchers in FL and related areas to scope their work.

REFERENCES

[1] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” 2017.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Federated Learning of Deep Networks using Model Averaging,” *Arxiv*, 2016.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated Learning: Challenges, Methods, and Future Directions,” *IEEE Signal Process. Mag.*, 2020, doi: 10.1109/MSP.2020.2975749.

[4] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated Optimization in Heterogeneous Networks,” 2018, [Online]. Available: http://arxiv.org/abs/1812.06127.

[5] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, “Communication compression for decentralized training,” 2018.

[6] L. He, A. Bian, and M. Jaggi, “Cola: Decentralized linear learning,” 2018.

[7] A. Lalitha, X. Wang, O. Kilinc, Y. Lu, T. Javidi, and F. Koushanfar, “Decentralized bayesian learning over graphs,” *arXiv*. 2019.

[8] W. Dai, A. Kumar, J. Wei, Q. Ho, G. Gibson, and E. P. Xing, “High-performance distributed ML at scale through parameter server consistency models,” 2015.

[9] Q. Ho *et al.*, “More effective distributed ML via a stale synchronous parallel parameter server,” 2013.

[10] M. A. Zinkevich, M. Weimer, A. Smola, and L. Li, “Parallelized stochastic gradient descent,” 2010.

[11] K. Bonawitz *et al.*, “Towards federated learning at scale: System design,” *arXiv*. 2019.

[12] Z. Charles and D. Papailiopoulos, “Gradient Coding Using the Stochastic Block Model,” 2018, doi: 10.1109/ISIT.2018.8437887.

[13] Z. Charles, D. Papailiopoulos, and J. Ellenberg, “Approximate gradient coding via sparse random graphs,” *arXiv*. 2017.

[14] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, “Speeding Up Distributed Machine Learning Using Codes,” *IEEE Trans. Inf. Theory*, 2018, doi: 10.1109/TIT.2017.2736066.

[15] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, “Coded computation over heterogeneous clusters,” 2019, doi: 10.1109/TIT.2019.2904055.

[16] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, “Gradient coding: Avoiding stragglers in distributed learning,” 2017.

[17] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, “LoAdaBoost:Loss-Based AdaBoost Federated Machine Learning on medical Data,” *arXiv*, 2018.

[18] E. Jeong, S. Oh, H. Kim, S. L. Kim, J. Park, and M. Bennis, “Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data,” *arXiv*. 2018.

[19] K. Bonawitz *et al.*, “Practical secure aggregation for privacy-preserving machine learning,” 2017, doi: 10.1145/3133956.3133982.

[20] B. Ghazi, R. Pagh, and A. Velingker, “Scalable and differentially private distributed aggregation in the shuffled model,” *arXiv*. 2019.

[21] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. Brendan McMahan, “CPSGD: Communication-efficient and differentially-private distributed SGD,” 2018.

[22] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, “Protection Against Reconstruction and Its Applications in Private Federated Learning,” *arXiv*. 2018.

[23] R. C. Geyer, T. Klein, and M. Nabi, “Differentially private federated learning: A client level perspective,” *arXiv*. 2017.

[24] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, “Learning differentially private recurrent language models,” 2018.

[25] W. Y. B. Lim *et al.*, “Federated Learning in Mobile Edge Networks: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, 2020, doi: 10.1109/COMST.2020.2986024.

[26] KPMG LLP, “Federated Learning: Strategies for Improving Communication Efficiency,” *Iclr*. 2018.

[27] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, “EXPANDING THE REACH OF FEDERATED LEARNING BY REDUCING CLIENT RESOURCE REQUIREMENTS,” *arXiv*. 2018.

[28] Z. Tao and Q. Li, “ESGD: Commutation efficient distributed deep learning on the edge,” 2018.

[29] L. Wang, W. Wang, and B. Li, “CMFL: Mitigating communication overhead for federated learning,” 2019, doi: 10.1109/ICDCS.2019.00099.

[30] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, “Edge-assisted hierarchical federated learning with non-IID data,” *arXiv*. 2019.

[31] Y. Liu *et al.*, “A Communication Efficient Collaborative Learning Framework for Distributed Features,” 2019, vol. 2019, [Online]. Available: http://arxiv.org/abs/1912.11187.

[32] X. Yao, C. Huang, and L. Sun, “Two-Stream Federated Learning: Reduce the Communication Costs,” 2018, doi: 10.1109/VCIP.2018.8698609.

[33] T. Nishio and R. Yonetani, “Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge,” 2019, doi: 10.1109/ICC.2019.8761315.

[34] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL: Cooperative learning mechanism using non-iid data in wireless networks," *arXiv*. 2019.

[35] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L. C. Wang, "Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach," *IEEE Wirel. Commun. Lett.*, 2019, doi: 10.1109/LWC.2019.2917133.

[36] H. T. Nguyen, N. Cong Luong, J. Zhao, C. Yuen, and D. Niyato, "Resource Allocation in Mobility-Aware Federated Learning Networks: A Deep Reinforcement Learning Approach," 2020, doi: 10.1109/WF-IoT48130.2020.9221089.

[37] T. Li, M. Sanjabi, and V. Smith, "Fair resource allocation in federated learning," *arXiv*. 2019.

[38] G. Zhu, Y. Wang, and K. Huang, "Broadband Analog Aggregation for Low-Latency Federated Edge Learning," *IEEE Trans. Wirel. Commun.*, 2020, doi: 10.1109/TWC.2019.2946245.

[39] M. M. Amiri and D. Gündüz, "Federated Learning over Wireless Fading Channels," *IEEE Trans. Wirel. Commun.*, 2020, doi: 10.1109/TWC.2020.2974748.

[40] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wirel. Commun.*, 2020, doi: 10.1109/TWC.2019.2961673.

[41] M. R. Sprague *et al.*, "Asynchronous federated learning for geospatial applications," 2019, doi: 10.1007/978-3-030-14880-5_2.

[42] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE J. Sel. Areas Commun.*, 2019, doi: 10.1109/JSAC.2019.2904348.

[43] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y. C. Liang, "Joint service pricing and cooperative relay communication for federated learning," 2019, doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00148.

[44] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *arXiv*. 2019, doi: 10.1109/lnet.2019.2947144.

[45] L. U. Khan *et al.*, "Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism," *IEEE Commun. Mag.*, 2020, doi: 10.1109/MCOM.001.1900649.

[46] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A Learning-Based Incentive Mechanism for Federated Learning," *IEEE Internet Things J.*, 2020, doi: 10.1109/JIOT.2020.2967772.

[47] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," 2019, doi: 10.1109/VTS-APWCS.2019.8851649.

[48] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2968399.

[49] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory," *IEEE Internet Things J.*, 2019, doi: 10.1109/JIOT.2019.2940820.

[50] M. Abadi *et al.*, "Deep learning with differential privacy," 2016, doi: 10.1145/2976749.2978318.

[51] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," 2015, doi: 10.1145/2810103.2813687.

[52] Y. Liu, Z. Ma, X. Liu, S. Ma, R. H. Deng, and K. Ren, "Boosting Privately : Federated Extreme Gradient Boosting for Mobile Crowdsensing," pp. 1–11.

[53] A. Triastcyn and B. Faltings, "Federated Generative Privacy," *IEEE Intell. Syst.*, 2020, doi: 10.1109/MIS.2020.2993966.

[54] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv*. 2018.

[55] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," 2019.

[56] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "On-Device federated learning via blockchain and its latency analysis," *arXiv Prepr.*, 2018.

[57] C. Briggs, Z. Fan, and P. Andras, "A Review of Privacy-preserving Federated Learning for the Internet-of-Things," in *Studies in Computational Intelligence*, Springer, 2020.

[58] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," *arXiv*. 2019.

[59] X. Zhang, S. Ji, H. Wang, and T. Wang, "Private, Yet Practical, Multiparty Deep Learning," 2017, doi: 10.1109/ICDCS.2017.215.

[60] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, and B. He, "A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection," pp. 1–41, 2019, [Online]. Available: http://arxiv.org/abs/1907.09693.

[61] S. Hardy *et al.*, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv*. 2017.

[62] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," *IEEE Trans. Inf. Forensics Secur.*, 2018, doi: 10.1109/TIFS.2017.2787987.

[63] A. Shamir, "How to Share a Secret," *Commun. ACM*, 1979, doi: 10.1145/359168.359176.

[64] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," 2006, doi: 10.1007/11681878_14.

[65] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, 2013, doi: 10.1561/0400000042.

[66] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated Learning for Keyword Spotting," 2019, doi: 10.1109/ICASSP.2019.8683546.

[67] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," 2002, doi: 10.1145/775047.775142.

[68] J. Vaidya and C. Clifton, "Privacy preserving Naïve Bayes classifier for vertically partitioned data," 2004, doi: 10.1137/1.9781611972740.59.

[69] J. Vaidya, C. Clifton, M. Kantarcioglu, and A. S. Patterson, "Privacy-preserving decision trees over vertically partitioned data," *ACM Trans. Knowl. Discov. Data*, 2008, doi: 10.1145/1409620.1409624.

[70] J. Vaidya and C. Clifton, "Privacy-preserving K-means clustering over vertically partitioned data," 2003, doi: 10.1145/956750.956776.

[71] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *arXiv*. 2018.

[72] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*. 2010, doi: 10.1109/TKDE.2009.191.

[73] L. Zhao *et al.*, "InPrivate Digging: Enabling Tree-based Distributed Data Mining with Differential Privacy," 2018, doi: 10.1109/INFOCOM.2018.8486352.

[74] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated Electronic Health Records," *Int. J. Med. Inform.*, 2018, doi: 10.1016/j.ijmedinf.2018.01.007.

[75] A. C. Zhou, Y. Xiao, Y. Gong, B. He, J. Zhai, and R. Mao, "Privacy Regulation Aware Process Mapping in Geo-Distributed Cloud Data Centers," *IEEE Trans. Parallel Distrib. Syst.*, 2019, doi: 10.1109/TPDS.2019.2896894.

[76] T. Yang *et al.*, "APPLIED FEDERATED LEARNING: IMPROVING GOOGLE KEYBOARD QUERY SUGGESTIONS," *arXiv*. 2018.

[77] S. Wang *et al.*, "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning," 2018, doi: 10.1109/INFOCOM.2018.8486403.

[78] L. Li, Y. Fan, M. Tse, and K. Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, 2020, doi: 10.1016/j.cie.2020.106854.

[79] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, "Federated learning of out-of-vocabulary words," *arXiv*. 2019.

[80] A. Hard *et al.*, "Federated learning for mobile keyboard

prediction," *arXiv*. 2018.

[81] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv*. 2019.

[82] J. Feng, C. Rong, F. Sun, D. Guo, and Y. Li, "PMF: A privacy-preserving human mobility prediction framework via federated learning," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, 2020, doi: 10.1145/3381006.

[83] K. Sozinov, V. Vlassov, and S. Girdzijauskas, "Human activity recognition using federated learning," 2019, doi: 10.1109/BDCloud.2018.00164.

[84] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, 2019, doi: 10.1109/MNET.2019.1800286.

[85] T. Yu *et al.*, "Learning context-aware policies from multiple smart homes via federated multi-task learning," 2020, doi: 10.1109/IoTDI49375.2020.00017.

[86] B. Hu, Y. Gao, L. Liu, and H. Ma, "Federated Region-Learning: An Edge Computing Based Framework for Urban Environment Sensing," 2018, doi: 10.1109/GLOCOM.2018.8647649.

[87] X. Han, H. Yu, and H. Gu, "Visual inspection with federated learning," 2019, doi: 10.1007/978-3-030-27272-2_5.

[88] S. R. Kurupathi and W. Maass, "Survey on Federated Learning Towards Privacy Preserving AI," 2020, doi: 10.5121/csit.2020.101120.

[89] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, 2019, doi: 10.1145/3298981.

[90] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," 2008, doi: 10.1007/978-3-540-88313-5-13.

[91] R. L. Rivest, M. L. Dertouzos, and L. Adleman, "On data banks and privacy homomorphisms," *Found. Secur. Comput.*, 1978.

[92] Y. Lin, Y. Wang, S. Han, W. J. Dally, and H. Mao, "DEEP Gradient compression: Reducing the communication bandwidth for distributed training," *arXiv*. 2017.

[93] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated Learning for Healthcare Informatics," *J. Healthc. Informatics Res.*, 2020, doi: 10.1007/s41666-020-00082-4.

[94] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," 2019.

[95] H. Eichner, T. Koren, H. B. McMahan, N. Srebro, and K. Talwar, "Semi-cyclic stochastic gradient descent," 2019.

[96] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," 2020, doi: 10.1109/WorldS450073.2020.9210355.

[97] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv*. 2020.

[98] J. Schneider and M. Vlachos, "Mass personalization of deep learning," *arXiv*. 2019.

[99] V. Smith, C. K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," 2017.

[100] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv*. 2020.

[101] L. Lyu, H. Yu, J. Zhao, and Q. Yang, "Threats to Federated Learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020.

[102] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Access*. 2020, doi: 10.1109/ACCESS.2020.3013541.

[103] S. Feng and H. Yu, "Multi-participant multi-class vertical federated learning," *arXiv*. 2020.

[104] T. D. Cao, T. Truong-Huu, H. Tran, and K. Tran, "A federated learning framework for privacy-preserving and parallel training," *arXiv*. 2020.

[105] Q. Wu, K. He, and X. Chen, "Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge based Framework," *IEEE Comput. Graph. Appl.*, 2020, doi: 10.1109/OJCS.2020.2993259.

[106] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare," *IEEE Intell. Syst.*, 2020, doi: 10.1109/MIS.2020.2988604.

[107] D. Liu, T. Miller, R. Sayeed, and K. D. Mandl, "FADL:Federated-Autonomous Deep Learning for Distributed Electronic Health Record," *arXiv*. 2018.

[108] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT," *IEEE Trans. Ind. Informatics*, 2020, doi: 10.1109/TII.2019.2942190.

[109] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Futur. Gener. Comput. Syst.*, 2021, doi: 10.1016/j.future.2020.10.007.

[110] H. Wang, M. Yurochkin, Y. Sun, Y. Khazaeni, and D. Papailiopoulos, "Federated learning with matched averaging," *arXiv*. 2020.

[111] J. Ma, J. C. Ho, Q. Zhang, L. Xiong, J. Lou, and X. Jiang, "Privacy-preserving tensor factorization for collaborative health data analysis," 2019, doi: 10.1145/3357384.3357878.

[112] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," 2019, doi: 10.1145/3338501.3357371.

[113] K. Mandal and G. Gong, "PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks," 2019, doi: 10.1145/3338466.3358926.

[114] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and Verifiable Federated Learning," *IEEE Trans. Inf. Forensics Secur.*, 2020, doi: 10.1109/TIFS.2019.2929409.

[115] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, and Q. Yang, "Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography," *arXiv*. 2019.

[116] P. Schoppmann, B. Balle, J. Doerner, S. Zahur, and D. Evans, "Secure Linear Regression on Vertically Partitioned Datasets," *IACR Cryptol. ePrint Arch.*, 2016.

[117] I. Giacomelli, S. Jha, M. Joye, C. D. Page, and K. Yoon, "Privacy-preserving ridge regression with only linearly-homomorphic encryption," 2018, doi: 10.1007/978-3-319-93387-0_13.

[118] Y. Zhuang, G. Li, and J. Feng, "A survey on entity alignment of knowledge base," *Jisuanji Yanjiu yu Fazhan/Computer Res. Dev.*, 2016, doi: 10.7544/issn1000-1239.2016.20150661.

[119] P. Christen, *Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection*. 2012.

[120] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "SecureBoost: A lossless federated learning framework," *arXiv*. 2019.

[121] Q. Jing, W. Wang, J. Zhang, H. Tian, and K. Chen, "Quantifying the performance of federated transfer learning," *arXiv*. 2019.

[122] S. Caldas, V. Smith, and A. Talwalkar, "Federated Kernelized Multi-Task Learning," *Conf. Syst. Mach. Learn.*, 2018.

[123] S. Sharma, C. Xing, Y. Liu, and Y. Kang, "Secure and Efficient Federated Transfer Learning," 2019, doi: 10.1109/BigData47090.2019.9006280.

[124] Y. Chen, Y. Ning, and H. Rangwala, "Asynchronous online federated learning for edge devices," *arXiv*. 2019.

[125] Y. Chen, X. Sun, and Y. Jin, "Communication-Efficient Federated Deep Learning with Layerwise Asynchronous Model Update and Temporally Weighted Aggregation," *IEEE Trans. Neural Networks Learn. Syst.*, 2020, doi: 10.1109/TNNLS.2019.2953131.

[126] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, "Decentralized collaborative learning of personalized models over networks," 2017.

[127] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, "Collaborative deep learning in fixed topology networks," 2017.

[128] S. J. Reddi *et al.*, "Adaptive federated optimization," *arXiv*. 2020.

[129] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: Communication-Efficient Algorithms for Federated Learning,"

Icml, 2020.

[130] K. Wang, C. Kiddon, H. Eichner, and D. Ramage, "Federated Evaluation of On-device Personalization," 2018.

[131] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," 2017.

[132] M. Khodak, M. F. Balcan, and A. Talwalkar, "Adaptive gradient-based meta-learning methods," *arXiv*. 2019.

[133] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," *arXiv*. 2019.

[134] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv*. 2019.

[135] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arXiv*. 2020.

[136] P. Richtárik and M. Takáč, "Distributed coordinate descent method for learning with big data," *J. Mach. Learn. Res.*, 2016.

[137] S. Zhang, A. Choromanska, and Y. Lecun, "Deep learning with elastic averaging SGD," 2015.

[138] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," *arXiv*. 2019.

[139] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2016.

[140] B. S. Kašin, "Diameters of some finite-dimensional sets and classes of smooth functions," *Math. USSR - Izv.*, 1977, doi: 10.1070/IM1977v011n02ABEH001719.

[141] D. Rothchild *et al.*, "FetchSGD: Communication-Efficient Federated Learning with Sketching," *arXiv*. 2020.

[142] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local SGD," *arXiv*. 2018.

[143] C. Yang, S. Wang, Q. Wang, K. Bian, M. Xu, and X. Liu, "Heterogeneity-aware federated learning," *arXiv*. 2020.

[144] V. W. Anelli, Y. Deldjoo, T. Di Noia, and A. Ferrara, "Towards Effective Device-Aware Federated Learning," 2019, doi: 10.1007/978-3-030-35166-3_34.

[145] Z. Xu, Z. Yang, J. Xiong, J. Yang, and X. Chen, "ELFISH: Resource-aware federated learning on heterogeneous edge devices," *arXiv*. 2019.

[146] I. I. Eliazar and I. M. Sokolov, "Measuring statistical heterogeneity: The Pietra index," *Phys. A Stat. Mech. its Appl.*, 2010, doi: 10.1016/j.physa.2009.08.006.

[147] A. Shafahi *et al.*, "Poison frogs! Targeted clean-label poisoning attacks on neural networks," 2018.

[148] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv*. 2017.

[149] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," *arXiv*. 2018.

[150] L. Su and J. Xu, "Securing Distributed Gradient Descent in High Dimensional Statistical Learning," *Perform. Eval. Rev.*, 2019, doi: 10.1145/3309697.3331499.

[151] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," 2019, doi: 10.1109/SP.2019.00029.

[152] L. Zhu and S. Han, "Deep Leakage from Gradients," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020.

[153] A. Pyrgelis, C. Troncoso, and E. de Cristofaro, "Knock Knock, Who's There? Membership Inference on Aggregate Location Data∗ ," *arXiv*. 2017, doi: 10.14722/ndss.2018.23183.

[154] B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved Deep Leakage from Gradients," *arXiv*. 2020.

[155] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models under the GAN: Information leakage from collaborative deep learning," 2017, doi: 10.1145/3133956.3134012.

[156] K. Wei *et al.*, "Federated Learning with Differential Privacy: Algorithms and Performance Analysis," *IEEE Trans. Inf. Forensics Secur.*, 2020, doi: 10.1109/TIFS.2020.2988575.

[157] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," 2019, doi: 10.1145/3338501.3357370.

[158] H. Takabi, E. Hesamifard, and M. Ghasemi, "Privacy Preserving Multi-party Machine Learning with Homomorphic Encryption," *Proc. Work. Priv. Multi-Party Mach. Learn.*, 2016.

[159] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "DeepSecure: Scalable provably-secure deep learning," *arXiv*. 2017, doi: 10.1109/dac.2018.8465894.

[160] J. Stremmel and A. Singh, "Pretraining Federated Text Models for Next Word Prediction," *arXiv*. 2020.

[161] F. Hartmann, S. Suh, A. Komarzewski, T. D. Smith, and I. Segall, "Federated learning for ranking browser history suggestions," *arXiv*. 2019.

[162] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal client behavior detection in federated learning," *arXiv*. 2019.

[163] J. Lee, J. Sun, F. Wang, S. Wang, C. H. Jun, and X. Jiang, "Privacy-preserving patient similarity learning in a federated environment: Development and analysis," *J. Med. Internet Res.*, 2018, doi: 10.2196/medinform.7744.

[164] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, "Federated Face Anti-spoofing," *arXiv*. 2020.

[165] Y. Liu *et al.*, "FedVision: An online visual object detection platform powered by federated learning," *arXiv*. 2020, doi: 10.1609/aaai.v34i08.7021.

[166] A. M. Elbir and S. Coleri, "Federated Learning for Vehicular Networks," *arXiv*. 2020.

[167] W. Y. B. Lim *et al.*, "Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach," *arXiv*. 2020.

[168] P. Kairouz *et al.*, "Advances and open problems in federated learning," *arXiv*. 2019.

[169] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv*. 2019.

[170] N. Guha, A. Talwalkar, and V. Smith, "One-Shot Federated Learning," no. 2, pp. 1–5, 2019, [Online]. Available: http://arxiv.org/abs/1902.11175.

[171] N. Guha and V. Smith, "Model Aggregation via Good-Enough Model Spaces," pp. 1–21, 2018, [Online]. Available: http://arxiv.org/abs/1805.07782.

[172] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially Private Meta-Learning," no. 2018, pp. 1–18, 2019, [Online]. Available: http://arxiv.org/abs/1909.05830.