A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond

Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, *Senior Member, IEEE*, Chamseddine Talhi, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Driven by privacy concerns and the visions of deep learning, the last four years have witnessed a paradigm shift in the applicability mechanism of machine learning (ML). An emerging model, called federated learning (FL), is rising above both centralized systems and on-site analysis, to be a new fashioned design for ML implementation. It is a privacy-preserving decentralized approach, which keeps raw data on devices and involves local ML training while eliminating data communication overhead. A federation of the learned and shared models is then performed on a central server to aggregate and share the built knowledge among participants. This article starts by examining and comparing different ML-based deployment architectures, followed by in-depth and in-breadth investigation on FL. Compared to the existing reviews in the field, we provide in this survey a new classification of FL topics and research fields based on thorough analysis of the main technical challenges and current related work. In this context, we elaborate comprehensive taxonomies covering various challenging aspects, contributions, and trends in the literature, including core system models and designs, application areas, privacy and security, and resource management. Furthermore, we discuss important challenges and open research directions toward more robust FL systems.**

***Index Terms*—Artificial intelligence (AI), deep learning (DL), distributed intelligence, federated learning (FL) applications, FL, machine learning (ML), privacy, resource management, security.**

I. Introduction

NOWADAYS, people are generating an unprecedented amount of data through connected devices, such as smartphones, Internet-of-Things (IoT) devices, wearable medical devices, etc. With a wealth of data available and the fact that machine learning (ML) models are data hunger, artificial intelligence (AI) is now omnipresent and de rigueur across major stakeholders, and making our lives more efficient. In a nutshell, what is driving AI's explosion today is deep learning (DL). It has unleashed countless applications used everyday by people worldwide. On the other hand, with DL's rapid evolution, the existing approaches continue to support thecloud-centric architecture, where data are centrally stored and processed. Besides the unacceptable latency and high cost engaged by such practices, data privacy and security remain the major issues. Without serious privacy consideration, sensitive data are highly exposed to disclosure, attacks, and cyber risks. Among the worst breaches recorded in the 21st century [1], Equifax (with 147.9 million customers affected in 2017), Marriott (with 500 million customers affected in 2018), and eBay (with 145 million users affected in 2014) are in recent memory. In this context, a new regulation by the European Union, called general data protection regulation (GDPR), has been enforced. It secures and protects personal data by setting rules and limiting data sharing and storage, which makes the digital future built on trust.

In line with the aforementioned rules and regulations and to take a step further in data preservation, on-site ML [2] and federated learning (FL) [3] have been advanced as alternative solutions to centralized systems. While on-device (which we refer to as *on-site*) ML keeps raw data locally by pushing ML tasks from the cloud to the devices, each device builds its own model without benefiting from peer's data and experience. Therefore, FL was introduced to overcome such problems, while still preserving privacy and reducing the huge overhead of data collection. It is a decentralized approach in terms of training data and on-device processing of computations dedicated to train a model. In FL, raw data are kept on end user devices, which cooperate on training a joint model. On a central server, only locally computed updates and analysis results are received and aggregated for an enhanced global model benefiting from the distributed learning. The new model is then shared with the clients to share knowledge among them. The devices of the users/clients in current studies are varying between smartphones [4], [5], IoT devices [6]–[9], healthcare devices [10], [11], robots [12], vehicles [13], [14], and many more.

Since the emergence of FL in 2016, there has been a growing interest in this field with a wide-range of applications, challenges and problems relevant to this novel paradigm, which motivated us to write this survey. Subsequently, few

Manuscript received June 18, 2020; revised August 10, 2020 and September 21, 2020; accepted October 7, 2020. Date of publication October 12, 2020; date of current version March 24, 2021. This work was supported in part by MITACS, in part by Ericsson Canada, in part by ETS Montreal, and in part by Lebanese American University. *(Corresponding author: Mohsen Guizani.)*

Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, and Chamseddine Talhi are with the Department of Software Engineering and IT, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada (e-mail: sawsan.abdul-rahman.1@ens.etsmtl.ca; hanine.tout.1@ens.etsmtl.ca; cc-hakima.ould-slimane@etsmtl.ca; chamseddine.talhi@etsmtl.ca).

Azzam Mourad is with the Department of Computer Science and Mathematics, Lebanese American University, Beirut 961, Lebanon (e-mail: azzam.mourad@lau.edu.lb).

Mohsen Guizani is with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/JIOT.2020.3030072

TABLE I
OVERVIEW OF EXISTING SURVEY PAPERS RELEVANT TO THE PRESENT WORK

Refs	Focus Point
[15]	FL in Mobile Edge Networks
[16]	Three Architectures for Different FL Settings
[17]	Implementation Challenges in FL
[18]	Wireless Communications in FL
[19]	Recent Advances and Open Problems and Challenges in FL
[20]	Threat Models and Major Attacks in FL
[21]	Categorization for FL Systems

recent survey papers[1] and preprints have been published to cover the FL area with different focuses. Their themes presented in Table I are summarized as follows. The survey in [15] focuses on FL for mobile-edge networks while highlighting the challenges related to communication cost, resources, privacy and security. In addition, it shows some FL applications for the edge network. Based on the characteristics of the data distribution, Yang *et al.* [16] discussed the categorization and architectures of different FL settings, which involve horizontal FL, vertical FL, and federated transfer learning. Li *et al.* [17] focused on the implementation challenges and their current approaches in four fields: 1) communication; 2) systems heterogeneity; 3) statistical heterogeneity; and 4) privacy. Niknam *et al.* [18] emphasized on the wireless communications where possible FL applications could be applied. The survey in [19] studies the recent existing initiatives in FL, and highlights various open research questions and challenges in this field. The latter work is originated from two-day Google workshop in Seattle. Lyu *et al.* [20] take a different direction by surveying the threats that compromise FL systems. They focus mainly on poisoning and inference attacks, which modify the desired model behavior. In [21], a survey on FL systems has been conducted, in which existing studies are categorized based on: data partitioning, ML model, privacy technique, communication-based architecture, and scale and motivation of federation. Moreover, techniques for designing FL systems and some case studies have been presented.

In this context, given the high emergence, applicability and potential high impact of FL in different research areas, to the best of our knowledge, the literature still lacks a comprehensive survey spanning over its different core modeling, applications, technical and deployment aspects and directing researchers to contribute each in their field. This fact motivated us to perform a thorough analysis of the raised problems and contributions in the literature and build this survey embedding a new classification followed by different taxonomies and key challenges in a variety of FL topics and research fields, including the core system model and design, application areas, privacy and security, and resource management. We believe that the proposed survey shall offer an in-depth and in-breadth overview clearly distinguishing and classifying the raised problems and contributions and shall assist the research community in elaborating relevant approaches advancing different emerging technologies and timely topics. In summary, the major contributions of this work compared to existing surveys are stated as follows.

1) We elaborate on the evolution of the deployment architectures of ML-based analysis, provide a comprehensive examination of FL topics and research fields classifying the efforts and contributions where the FL paradigm is of the current trend in the research and industry, and offer an in-depth review and thorough analysis covering key technical aspects of the FL core system model and design. We further discuss the challenges and interesting open research directions that pave the way for upcoming generations of FL solutions. The proposed research directions are categorized based on the proposed FL fields and topics, i.e., system model and design, application areas, privacy and security, and resource management.
2) We build a taxonomy of FL application areas covering all the fields where FL approaches are introduced so far. The provided analysis shall assist researchers interested in ML solutions and wishing to start or continue working in the areas of the Gboard, healthcare, IoT, edge computing, networking, robotics, grid-world, models, recommender systems, cybersecurity, online retailers, wireless communications, and electrical vehicles (EVs).
3) We elaborate an additional in-depth study of the literature identifying and analyzing the key contributions that address privacy and security problems within the FL paradigm. These are fundamental aspects in FL as, in the presence of malicious parties, data can still be subject to disclosure and poisoning attacks. Accordingly, a thorough review is provided covering all the approaches, including the cryptographic protocols, different privacy techniques, data poisoning attacks, model update poisoning attacks, and defenses to poisoning attacks.
4) We provide a thorough analysis of resource management mechanisms proposed for FL settings and develop a taxonomy of the optimization approaches with respect to their objective functions and considered parameters. Such resources include the clients' reliability, network link quality, and central aggregation server. Our study touches different aspects of system characteristics, wireless resources, model quality, and offloading with hierarchical organization.

The remainder of this article is organized as follows. Section II examines and compares the different ML-based architectures. Section III provides preliminary background about the FL architecture and design. Section IV presents the new classification of FL topics and research fields. Taxonomies of the FL system model and design and application areas

[1] While preparing this survey, [15] and [17] have been released yet taking different directions.

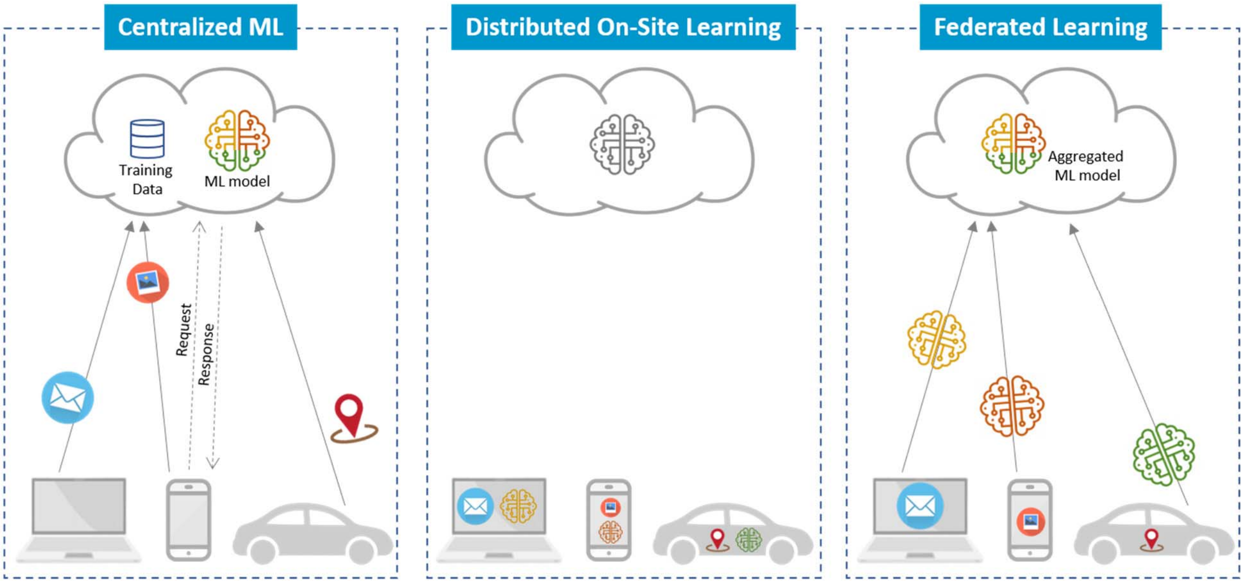

Fig. 1. Centralized versus Distributed on-site versus FL architectures: in centralized learning (left), data are sent to the cloud, where the ML model is built. The model is used by a user through an API by sending a request to access one of the available services. For distributed on-site learning (middle), each device builds its own model using its local data set. After the first interaction with the cloud to distribute a model to the devices, no more communication with the cloud is needed. In FL (right), each device trains a model and sends its parameters to the server for aggregation. Data are kept on-devices and knowledge is shared through an aggregated model with peers.

are presented in Sections V and VI. An in-depth analysis of the privacy, security, and resource management literature is provided in Sections VII and VIII. Section IX discusses future directions for FL research, followed by a conclusion in Section X.

II. Evolution of Machine Learning Architectures

This section elaborates on the evolution of ML architectures from centralized to distributed on-site and recently up to FL as illustrated in Fig. 1.

A. *Centralized Learning*

ML in general, and DL in particular, are finding their ways into our everyday life as we are becoming more fascinated by AI decision making. DL applications are ranging from as simple as Netflix is following in Google and Facebook footsteps to improve its services, to as sophisticated as self-driving cars [22], smart healthcare [23], fraud detection [24], earthquake prediction [25], and many more. What is behind DL success is the tremendous amount of data generated by mobile and IoT devices. In typical methods, the conventional wisdom is to continuously stream generated data into the cloud, where it is analyzed, more features are extracted, and models are better trained on high-performance servers. Such a method is illustrated in the centralized ML scenario in the left-hand side of Fig. 1. Amazon Web Services, Google Cloud, and Microsoft Azure are among the available ML as-a-service providers [26], where models can be deployed and used at scale. When there are lots of interactions with available services in the cloud, more training data are gathered and more intelligent ML-based applications are therefore produced. However, the privacy of available data used for training and for the astounding success of DL is becoming a rising concern for the users. Such data could be very private and of any type, such as personally identifiable information (e.g., driver's license, passport information, etc.), payment data (e.g., bank accounts, credit card numbers, etc.), protected health information (e.g., diagnosis and medical records, etc.), confidential data (e.g., financial documents, etc.), and others. When these data are shared with the cloud, it is most likely that users privacy becomes compromised with eavesdropping attacks. Other issues arise in the cloud/centralized-based approaches: 1) latency, as data could be transmitted hundreds, even thousands of miles away to reach the cloud and 2) data transfer cost, as moving data over the network into and out of the cloud computing is not free of charge. To overcome such problems, on-site ML has been advanced, where some of ML tasks are moved to the devices with powerful resources.

B. *Distributed On-Site Learning*

With the increasing risks of moving data to a centralized entity, there is a need for real-time intelligence motivating distributed on-site ML, where training, predictions, and inferences are based on live-streaming data. Rather than sending a request along with the private data from a user to the cloud, on-site ML engages the server to distribute a pretrained or a generic ML model to the devices, as illustrated in the middle section of Fig. 1. After deploying the model, each device can then personalize it by training using its local data, can perform some predictions for its data to predict its outcome, or can run the inferences computation to infer some testing samples and learn about the data generation process. In such systems, privacy advantage is definite, as data does not leave its hosts. On-device intelligence has been applied in many applications, such as skin cancer detection [27], medical applications [28], smart classrooms [29], neural network assisted services [30],

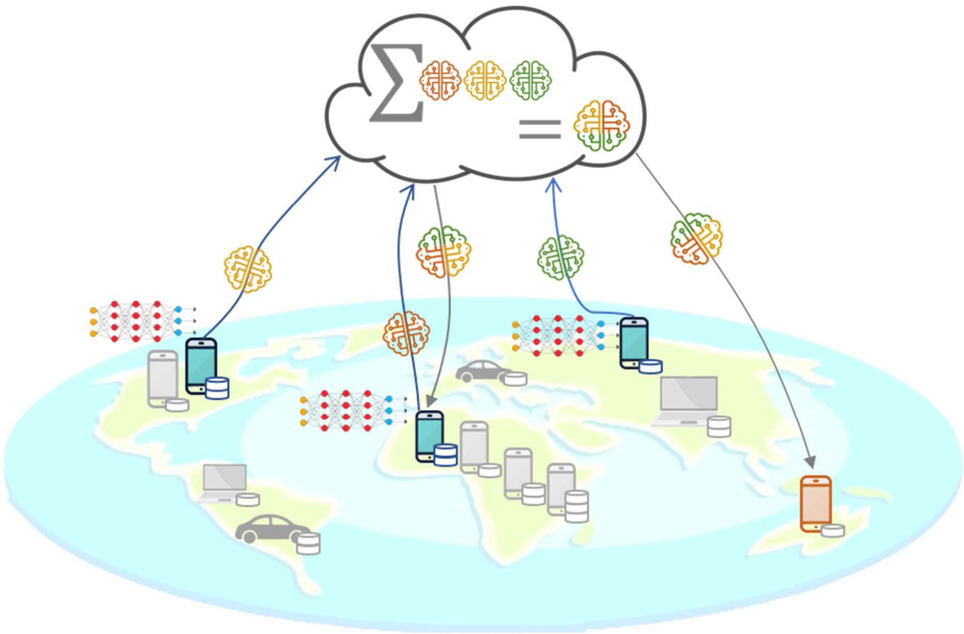

Fig. 2. Life cycle of FL: 1) training in a distributed fashion, where raw data are kept on-devices, and each selected client locally trains a model and sends its parameter to the server; 2) aggregation of the received models performed on the server; and 3) distribution of the new model to the clients.

etc. Nevertheless, the no round-trip fashion between a cloud and the devices limits the generated local models to each user experience without any benefit from peer's data. To this end, FL has been advanced, where users' computation is federated while still preserving privacy.

C. Federated Learning

Google researchers coined FL in 2016, and since then, it has been sweeping the world by experiencing vigorous growth in both academia and industry. Going beyond on-device ML, FL was developed to also move the training task to the device itself, while federating local models and learning. Its main objective is building a framework toward privacy-preserving ML. The right-hand side of Fig. 1 shows the FL process compared to the other existing approaches. Between sending local yet private data to the server and benefiting from ML applications, performing ML tasks on-devices without benefiting from peer's data, and precluding direct access to raw data and federating locally training ML models, the latter is more likely to be chosen by users. Therefore, FL preserves data privacy and reduce data communication overhead by keeping raw data on-devices and aggregating locally computed model updates.

III. Preliminary: FL Architecture and Design

We present in this section the process of FL, its production application, and its formal problem definition [3], [31], [32].

A. Production Application and Opensource Frameworks

FL was first tested on Gboard [33], Google keyboard for Android. It supports multilingual typing ranging from searching Google and sharing its results from the keyboard to autocorrections, voice typing, and glide typing. Based on the user behavior when Gboard shows some suggestions on the screen, local learning is performed and FL finds its way by enhancing future suggestions/interactions with the user. Thus, better features, such as next-word prediction, word completion, corrections, and many more are provided. To implement and experiment FL on decentralized data, the following opensource frameworks are in development/available: tensorflow federated (TFF) [34], federated AI technology enabler (FATE) [35], PySyft [36], PaddleFL [37], and Clara training framework [38]. In the research field, image classification and language modeling were the first widely adopted models for proposing an FL-based framework. To test their performances, Modified National Institute of Standards and Technology [39] (MNIST) for handwritten digits and the Canadian Institute For Advanced Research [40] (CIFAR) for images are the popular data sets used in the literature experiments.

B. FL Life Cycle and Protocol

Fig. 2 depicts the life cycle of FL. The process is divided into several continuous communication rounds, which are completed once the global model reaches the desired accuracy. The server first generates a generic model, then each round follows the steps as follows.

1) A subset of clients is selected by the server. While the typical conditions for the device selection lie in being in charge, idle, and on unmetered connection, only few works [41], [42] addressed this aspect.
2) Only selected clients download the current model parameters/weights from the server and initialize the local ML model with such weights.
3) Using its local training data, each selected client trains and optimizes the global model. As in typical and most used techniques, the client runs stochastic gradient descent (SGD) to compute the update. With the communication bandwidth constraint, computing one

Algorithm 1 `Federated Averaging` Algorithm [3]. The K Clients Are Indexed by k; B, E, and η Represent the Local Minibatch Size, Number of Local Epochs, and Learning Rate, Respectively

```
Server executes:
  initialize $w_0$
  for each round $t = 1, 2, \ldots,$ do
    $m \leftarrow max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of $m$ clients)
    for each client $k \in S_t$ in parallel do
        $w_{t+1}^{k} \leftarrow$ ClientUpdate($k, w_t$)
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k}$

ClientUpdate($k, w$):                                   ▷ Run on client $k$
  $\beta \leftarrow$ (split $P_k$ into batches of size $B$)
  for each local epoch $i$ from 1 to $E$ do
    for batch $b \in \beta$ do
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return $w$ to server
```

gradient and sending it back to the server are not sufficient enough. Instead, some number of minibatch gradient descent steps over multiple epochs are processed in one round in order to perform better model update and to reduce the communication cost.

4) Once the training is completed, the clients send the optimized parameters to the server. Some clients might dropout during the training or the parameter transmission phases due to poor connection, limited computation resources, a large amount of training data, etc. Therefore, a percentage of failed clients beyond what the server can handle is reported, and the process continues with the received number of updates. In case the number of clients reporting in time is not enough, the current active round is abandoned [31].
5) The server aggregates the clients updates after weighting them based on their data set size. Its pseudocode is provided in Algorithm 1. A new shared model is therefore produced, and to be better enhanced in the next iterations.

C. Problem Formulation

FL focuses on supervised ML, which maps input values x_i to output label y_i in order to predict unseen data. The input-output pair (x_i, y_i) is of size n and the goal is to find the model parameters w as vector while training. The model training process aims to minimize a loss function $f_i(w)$, which tells how good the model is when predicting ith data sample with the vector w. Based on the ML model, the problem can be convex or nonconvex. Since FL is built on the nonconvex neural network, its optimization algorithm of finite-sum function is depicted as follows [3]:

$$\min f(w), \qquad \text{where} \quad f(w) = \frac{1}{n}\sum_{i=1}^{n} f(x_i, y_i, w)$$
$$f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w). \qquad (1)$$

As in FL data from clients is never assembled, objective (1) should then be modified. Assume K clients participate in the learning rounds, each holding n_k data samples with $n_k = |P_k|$. P_k is the partition assigned to each client k from the whole data set P, with $P = \cup_{k=1}^{K} P_k$. Therefore, the new loss function, representing the global loss, is formulated as a weighted sum of the local loss functions $F_k(w)$ as follows [3]:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \qquad \text{where} \quad F_k(w) = \frac{1}{n_k}\sum_{i \in P_k} f_i(w). \qquad (2)$$

IV. FL Technical Challenges and Research Fields: New Classification

In this section, we discuss the FL technical challenges and provide a new classification of the main research fields addressing them. The distributed architectural aspect, quality of the collected data, type of devices hosting the learning models, communication and aggregation mechanisms, involvement of different parties, and applicability to different applications have raised diversity of challenging problems to be addressed by researchers in different fields. To start with, the answer to "why FL is not just as typical distributed learning settings" lies in its following challenges and properties.

1) *Nonindependent and Nonidentically Distributed (Non-IID) Data:* Each client generates his own data set based on his unique behavior and usage of the device. Such data remain local, decentralized, and not seen by other clients, which makes each device data not representative yet nonidentically distributed from the whole population. Moreover, the dependency of the data can be produced with the usage of the same device by different members of the family, such as mother–child or husband–wife. This is also the case when the same user dedicates the usage of one device while performing an activity x and another when performing an activity y, which result into some mutual dependency in data with different distribution.
2) *Unbalanced Data:* The different usage of devices, the clients local environment, and the noninteraction between clients result in vastly varying amounts of generated training data.
3) *Massively Distributed Data:* The participants in FL can form multiple millions of clients, ranging from mobile phones to IoT devices, organizations/institutions, vehicles, and many more. It has been reported in [3] that the number of the participants is expected to be larger than the average number of samples per participant.
4) *Unreliable Device Connection:* Network connectivity widely varies from one client to another. Most often, clients are under slow, limited, expensive, and unavailable connections, which significantly reduce the number of available ones at once. In addition, among the available clients, many might not be able to participate in each learning round due to their different computation capabilities.
5) *Limited Device Memory:* When mobile phones, in general, and IoT devices, in particular, are involved in the

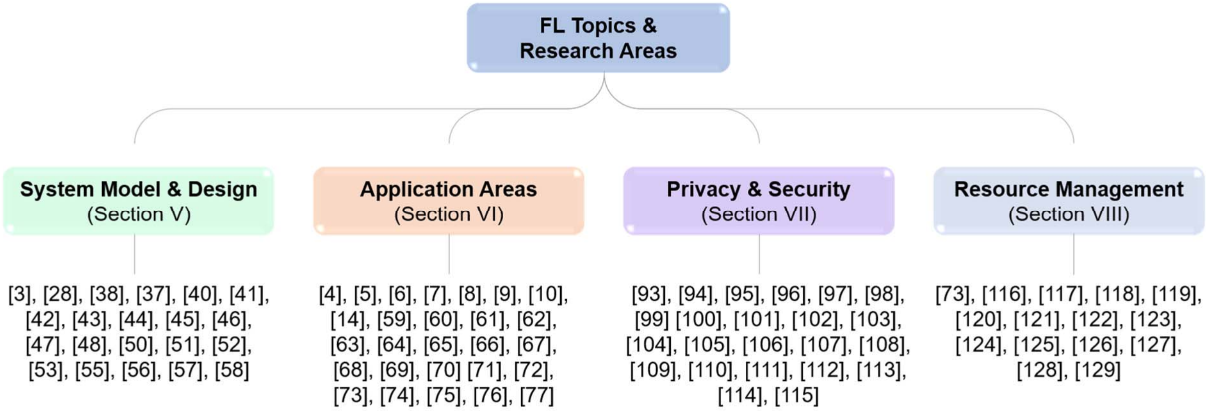

Fig. 3. Classifying FL topics and research areas into four main categories: 1) system model and design; 2) application areas; 3) privacy and security; and 4) resource management.

learning process, their available memory budgets are usually limited. Moreover, as the batch size increases, the memory footprint increases. This might either cause devices dropout in the training phase, or force simple models with small batch sizes to be executed on devices.

6) *Poisoning Attacks:* The anonymity of the clients might allow an attacker to behave as normal user and be selected to participate in the FL process. Hence, the attacker can take advantage during the training phase by feeding poisoned data, yet deviating the model toward miss-classification.

A deep analysis of the aforementioned technical challenges and recent contributions in the literature motivated us to propose in this survey the new classification illustrated in Fig. 3. We believe that the proposed classification would distinguish clearly the raised problems and assist the research community in elaborating relevant contributions advancing different emerging technologies and timely topics. Accordingly, we classify the existing FL research fields in terms of the core system model and design (Section V), application areas (Section VI), privacy and security (Section VII), and resource management (Section VIII).

V. FL System Model and Design

After analyzing the existing research studies, we provide in this section the efforts and contributions targeting the FL core system model and design. As illustrated in Fig. 4, these contributions and approaches are classified into five main areas: 1) communication cost; 2) client selection; 3) optimization and aggregation algorithms; 4) Non-IID; and 5) incentives.

A. Communication Cost

We observe that the FL process revolves around many communication rounds between a server and clients. The latter in a typical approach download generic ML model for a local computation of the updates and send them back to the server. Before moving to the next iteration, computation of the model aggregation is performed on the central node. To provide small communication footprints, the following research efforts have been advanced.

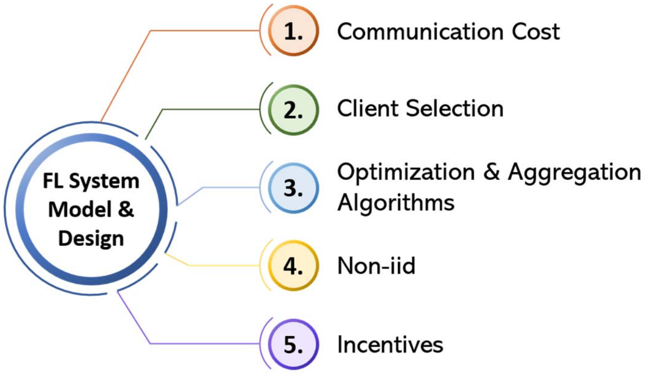

Fig. 4. Classification of the FL system model and design contributions.

Konečný *et al.* [32] proposed to investigate two methods that are either combined on one device or only one of them being adopted in order to minimize the communication cost. The first method, structured updates, imposes a predefined structure for the updates by proposing two types: 1) low and 2) random masks. The low rank divides the model parameters into two matrices, one of them is fixed and only the second is sent to the cloud. On the other hand, a random mask can generate matrices structures in a way that the nonzero values are only sent instead of the whole entries. Regarding the second proposed method, sketched updates, it requires updating the full model, then compressing it in a lossy manner before being sent to the server. Once received, the server extracts the model to be aggregated with the others. The conducted experiments test the models on 100 devices each training 500 examples in a task of image classification. The results show reduction in the upload communication. The communication cost minimization, presented in [43], aims to reduce the size of the models generated by both the server and the clients, while updating the FL process as follows. First, the server generates a smaller submodel with fewer parameters using the federated dropout technique. Then, lossy compression on the resulting model is performed on the server side and sent to the clients. The latter applies a decompression to start training. Once done, the updates are, in turn, lossily compressed and sent to the server, which decompresses and aggregates the final model. As the bottleneck of federated averaging method lies in the restricted communication bandwidth that delays the clients from uploading their updates, the work in [44] proposes faster aggregation model. This is achieved using *over-the-air computation* principles by both joint device selection and beam-forming design. Furthermore, the efficiency of the designed algorithms is supported by sparse and low-rank modeling. The contribution in [45] aims to fulfill the requirements of FL with the following: 1) enable both downstream and upstream compression; 2) make the model robust to unbalanced non-IID data with small batch sizes; and 3) handle the big number of participating clients. The proposed compression-based framework considers ternarization and sparsification, in addition to optimal Golomb encoding of the weights. The motivation of the work in [46] is to propose an enhanced FL framework, by not only reducing the communication cost but also improving the model accuracy. The first objective is achieved by building an asynchronous strategy, which allows splitting shallow from deep layers in a deep neural network, and makes the clients send, more frequently, shallow-related parameters as they perform better on the central model. On the other hand, the model accuracy is improved by considering in the aggregation the models trained in previous rounds rather than the ongoing round only. The contribution of reducing the communication cost in [47] is the following. Instead of training a single model on the client side, a two-stream model is adopted. As such, the clients use the shared global model as a reference during the whole training phase and, based on it, update their local model through backpropagation. Moreover, the maximum mean discrepancy constraint is also used to allow extracting more generalized features while training depending on the peer knowledge in the two-stream model. The work in [48] targets the practical FL usage for speech data when detecting wake word, such as “Hey Siri” and “OK Google.” The proposed solution replaces the standard FedAvg algorithm with an Adam-based adaptive averaging strategy, and the conducted experiments reduce the number of communication rounds when targeting a specific recall value. The bottleneck of FL’s communication overhead has been addressed in [49] by proposing a communication-mitigated FL (CMFL) solution. Instead of sending all the client updates to the cloud, the proposed algorithm reduces the FL communication rounds by sending only the updates being identified as relevant. At each learning phase, the clients receive the global tendency of the aggregated model to decide whether their local updates are worth being sent to the cloud and good enough to improve the global model. In the conducted experiments, CMFL are compared to three existing solutions and the results show how CMFL outperforms them in terms of communication efficiency.

Discussion: We perceive that naively following FL protocol results in communicating a full model in each round, which may reach a size of gigabytes in huge parameters-based models [50]. With the large number of participating devices and the slow network bandwidth in such distributed settings, communication overhead becomes a bottleneck in FL. As throughout the FL process the client updates are exchanged in each round, connectivity becomes a major concern, especially when the uplink has a lower network connection compared to downlink. The main goal of the approaches presented in this section is to reduce the communication cost.

B. Client Selection

The selection of clients in a typical FL considers nothing but the devices being charged, idle, and connected to unmetered network (i.e., WiFi). Among such devices, a random number is thus determined to initiate the communication with and register their participation. However, relying only on these criteria when dealing with heterogeneous clients in terms of communication and computation resources entails many drawbacks, such as long training time. To address such problems, few works have been proposed. Nishio and Yonetani [41] discussed different characteristics of the clients, which can affect the efficiency of the whole training process. The limited computational resources of some clients would impose longer time to update the model. Additionally, longer upload time will be needed under poor wireless channel conditions. The proposition consists of a new FL protocol FedCS. As opposed to the original algorithm, FedCS necessitates the participating clients to communicate with a mobile-edge computing (MEC) server information about their resources, mainly the upload and update time of the model parameters. Accordingly, the latter determines the subset of clients able to complete the FL steps of downloading, updating and uploading the model within a certain deadline. This work is then extended in [42] to cover the client selection aspect, in addition to resource scheduling algorithms. In the former, two set of clients are selected, one to update the models, and the other to upload their own data to the server by providing some incentives. As such, the server first updates the model using IID-based raw data, then updates it using the aggregated models. Besides, the model performance is measured using some validation data.

Discussion: When dealing with random selection of participants in FL, the training progress along with the final model deployment will be dependent of the performance of selected clients. FL becomes more at risk for bottleneck when heterogeneous clients with different limited resources train a shared model. Eventually, longer training processing time, unresponsive clients, longer transmission time, and many dropouts during the process are more likely to be faced. In this section, approaches related to the participants selection are presented, where more efforts are still in need.

C. Optimization and Aggregation Algorithms

The goal of FL is to train and generate high-quality global model through the learning rounds. With high-dimensional data distributed on-devices, the following approaches have been proposed to efficiently federate client-provided model updates. First, the importance of decentralizing the data on mobile devices by locally training and updating the models is strongly shown in [3]. The contributions of this article are: 1) selecting the practical algorithm, FederatedAveraging, that can best serve for the implementation of FL and 2) proving that the approach can be practically used by assessing extensive evaluation. The authors trained the model by considering two examples: 1) image classification and 2) language models for voice recognition. The following set of model architectures are applied: a multilayer perceptron (MLP), convolutional neural network, 2-layer LSTM, and world LSTM with a large number of parameters. Konecnỳ *et al.* [51] discussed the inefficiency of existing algorithms to be deployed within FL. These algorithms, whether designed to run on a single computer or in distributed settings, cannot meet with the following FL requirements: 1) massively distributed data points that are stored across large number of nodes; 2) Non-IID that represents the Non-IID data generated by each node; and 3) unbalanced data whereby each node can hold different amount of training data. To satisfy such needs, a federated stochastic variance-reduced gradient is proposed, which is capable of converging to an optimal classification accuracy in just few iterations. Nilsson *et al.* [52] presented the aforementioned optimization algorithms: FedAvg—federated averaging [3], FSVRG—federated stochastic variance reduced gradient [51], and CO-OP—cooperative ML [53]. They benchmarks these three algorithms, in addition to a centralized optimization, in order to compare their performance. The comparison is performed on the MNIST data set, where data are distributed in an IID and non-IID fashions. The results show that regardless of the data distribution, FedAvg outperforms the other federated algorithms. As for the centralized optimization, it outperforms FedAvg in the non-IID partitioning, but both have similar performance with IID fashion. Mohri *et al.* [54] showed that the original FL thoroughly depends on the uniform distribution of clients data while minimizing the loss function. However, this bias the models toward specific clients, making FL an inadequate system. Therefore, the authors propose an agnostic FL framework, which optimizes the aggregated model when any combination of the client distributions occur. A new fast stochastic optimization solution is also implemented to solve the mentioned problem. Liu *et al.* [55] highlighted the enormous number of communication rounds needed in FL to improve the model accuracy, which results in unendurable latency and network saturation. To reduce the required number of communication rounds, a hierarchical federated averaging algorithm is proposed by deploying mobile-edge servers to act as intermediary between the clients and the cloud. The proposed solution engages initially, at the edge server, several local aggregations for the clients' models to be sent at later stage to the cloud for global aggregation. The experiments have been first conducted to compare the client-edge to the edge-cloud divergence, which represents more noniidness in the data distribution for larger divergence. The results show that the model accuracy is more affected by the noniidness among the edges than the noniidness among the clients. Furthermore, the results prove that reducing the communication rounds is achieved by using less edge and more cloud aggregations. The massive number of communication rounds between the central server and the clients are substituted with only one round in [56] to overcome the critical bottleneck of communication in FL. The proposed approach is explored in two settings. First, the one-shot FL, representing the single communication round, entails that each device trains its model until completion. As numerous clients participate in FL, their number is controlled by selecting 1) the devices that has built models achieving a desired performance; 2) the ones according to their local amount of data; and 3) random devices from the network. The models' aggregation of the selected clients is then performed using ensemble learning technique instead of the naïve averaging. The second configuration follows a semi-supervised learning, where a set of unlabeled data are accessible by the server that allows the distillation of the resulting global model. Anelli *et al.* [57] focused on the aggregation algorithm in FL rather than following the standard FedAvg, which relies only on the clients data set size. Thus, a set of criteria about the clients is selected to base each client contribution on. Next, priority levels are assigned to these criteria and an online adjustment is used for the parameters aggregation.

Discussion: This section offers distributed optimization algorithms and aggregation strategies to be applied in the practical FL system. These algorithms and strategies have been advanced after it has been demonstrated in [51] that existing algorithms are not suitable to the settings of FL presented in Section IV. Since communication compared to computation is much more expensive in FL, implementing optimization and aggregation algorithms, which minimize the number of rounds with fast convergence of the model and without causing burdens on the backbone network, is of utmost importance.

D. Non-IID Data

Some propositions have been made to handle the non-IID data problem, which bias the model especially when training is performed using SGD. Zhao *et al.* [58] addressed the problem of decreased accuracy under skewed non-IID data. This engages that each client device trains only a solo class of data depending on its own behavior. The proposed solution aims to improve the accuracy level by sharing a small set of data encompassing a uniform distribution over the classes (labels) with all the participating clients. Besides the shared data, each client uses its local private data to build the ML model. Experiments have been performed on CIFAR-10 that is used for image prediction, and the results show increased accuracy by almost 30% with just 5% of globally shared data. The non-IID problem in FL has been addressed in [42] by proposing a Hybrid-FL. The latter provides some incentives to the clients encouraging them to upload their data to the server. While the selection of such clients does not exceed the 1% of the population, a signification IID-based data are assembled on the server. Subsequently, the gathered data are trained to form a model, which is aggregated with the other models received by the clients using their non-IID data. Data-uploading clients, selection of clients and model-uploading clients are all scheduled in this work based on heuristic algorithms. Smith *et al.* [59] showed first that FL faces two challenges referred as statistical and system when performed over a distributed number of nodes. The statistical challenges arise when the model should be learned from non-IID distributed data generated by different nodes. As for the system challenges, they are faced since contributed devices have unbalanced data and different capacities in terms of communication, storage, and computation, which cause some fault tolerance and stragglers. In this article, the authors prove that multitask learning, which learns from separate models instead of a single global model, can naturally address the statistical problems. In addition, they propose a novel optimization method, MOCHA, to handle the system challenges. This method divides the problem into subproblems and demands each controller on the nodes to specify a value for a defined parameter according to the node's network connection, power and storage capacity. Finally, some experiments are conducted based on real federated data sets: Google Glass (GLEAM), Human Activity Recognition, Land Mine, and Vehicle Sensor.

Discussion: This section presents the efforts made after a study showing that, when using FedAvg with highly skewed non-IID data, the accuracy of convolutional neural network can be drastically decreased by up to 11%, 51%, and 55% for MNIST, CIFAR-10, and keyword spotting data sets, respectively. Basically, deploying deep neural networks in FL relies on SGD. In the latter case, training data should represent the entire population distribution in order not to cause bias in the gradient estimates [60]. While such a property is based on IID data distribution, FL follows a non-IID fashion, as independent clients are generating data based on their own behavior and usage, and accordingly the original FedAvg algorithm has been implemented but without guaranteed performance.

E. Incentives

While existing approaches focus on optimizing different FL aspects, few have considered the unwillingness of the clients to participate in the training rounds or the selection of clients with low-quality model updates. It is considered in [61] that some clients, if selected by the server, tend not to waste their resources with the limited computation and communication capabilities. The authors address such a problem by designing, based on the contract theory, an incentive mechanism that motivates the users to contribute in FL. Models trained with high-quality data give better accuracy with less local model iterations. Therefore, the higher the quality data of a client x is, the more rewards are given to x. Kang *et al.* [62] proposed a reputation-based selection of reliable workers to defend against unreliable ones, which chooses the candidates with high-accuracy and efficient training data. Such reputation is evaluated using a subjective logic model relying on the client's past interactions and their behaviors with other FL services. Moreover, an incentive mechanism is designed with some rewards in order to motivate the clients in what resources they can contribute.

Discussion: It is assumed in typical FL that all the selected clients by the server are always available and tend to begin the learning process whenever chosen. However, such an optimistic assumption does not reflect real-world scenarios. Quite a number of devices are most likely to dropout throughout the process, or even reject to join due to resource costs and constraints. Moreover, to faster converge the global model, encouraging the clients with high-quality data is highly in need. As a result, the presented incentive-based approaches have been proposed to address these concerns.

VI. FL Application Areas

We provide in this section another taxonomy for FL application areas, which is summarized in Table II. It shows in which area each application-based paper falls and clearly explores the attention gained in each domain. We also highlight what is responsible of locally training the models, what each paper targets, and which ML algorithm is used, in addition to the implemented aggregation method. Since data are of utmost importance in ML and DL, we show as well which data set the authors chose in their proposed approaches. Researchers at Google work on enhancing language modeling from user-generated data on the Gboard application [4], [5], [63], [64]. Others find FL great fit in the healthcare domain [10], [11], [65]–[71], where patient privacy is balanced with ML by keeping patient data on-premise in the hospital. As smartphones gained traction in FL, so did IoT devices [8], [9], [72], [73]. Moreover, FL has as well found its way into many other areas, such as edge computing [74], networking [75], robotics [12], grid-world [76], FL enhancement [77], recommender system [78], cybersecurity [79], online retailers [80], wireless communication [18], and electric vehicles [81]. In the sequel, we present relevant research efforts in each of these domains.

A. Gboard Application

FL has been initially used on Google virtual keyboard, Gboard, to power its features. The work in [4] presents FL to improve the query suggestion of Gboard. There are different requirements that the clients should meet in order to validate their eligibility to participate in the FL process. These conditions are relevant to environmental requirements, device specifications and language restrictions. On the other hand, other constraints on the FL tasks are defined by the server, which includes the goal number of clients to participate in the process, the minimal number of clients needed to run a round, how frequently the training is done, a time threshold to wait for receiving clients updates and the fraction of clients that have to report back in order to commit a round. The performed evaluations show that the training examples count is higher in the evening while the loss is higher during the day. Observations of live deployment further show sometimes slight drop between the expected and actual query click-through rate. Gboard has also used FL in [5] in order to train a more complex neural network model demonstrating a better performance than a model trained on centralized data. Ramaswamy *et al.* [63] have proved the ability of the recurrent neural network to predict emoji from text on Gboard through FL. Chen *et al.* [64] adapted FL, more precisely federated character-based recurrent neural network, to learn out-of-vocabulary (OOV) words. The latter are undefined words that are encountered as input but are not found in the system's dictionary. While preserving data privacy in the FL settings, the proposed solution learns OOV words by aggregating the knowledge of many clients from local OOV words. The experiments demonstrate the feasibility of the approach using: 1) publicly available data set containing social comments and 2) data generated from Gboard on real device.

B. Healthcare

Traditionally, healthcare records from distributed sites are moved to a central database for analysis, which entails many

TABLE II
TAXONOMY OF FL APPLICATION AREAS

Refs	Area	Training Device Type	Goal	Trained Model	Aggregation Algorithm	Dataset
[4]	Gboard App	Mobile phones	Language Modeling: Keyboard search suggestion	Logistic Regression	FedAvg	Keyboard Live-Traffic
[5]			Language Modeling: Next-word prediction	RNN		
[63]			Language Modeling: Emoji Prediction	RNN - LSTM		
[64]			Language Modeling: Out-of-Vocabulary Learning	RNN - LSTM		Reddit [82]
[10]	Healthcare	Hospitals	Mortality Prediction	Neural Network	A proposed FADL	eICU [83]
[11]			Mortality & Hospital time stay Prediction	Deep Learning	FedAvg	
[65]		Scenario 1: Hospitals Scenario 2: Patients	Hospitalization Prediction	Sparse SVM	A proposed CPDS	EHR - Boston Medical Center
[66]		Phones connected to patients' devices	Anomaly Detection in Medical Systems	Neural Network	Average (not weighted) of parameters	MIMIC [84]
[67]		Organizations	Human Activity Recognition	Deep Neural Networks	n/a	UCI HAR [85]
[68]		Centers	Analysis of brain changes in neurological diseases	Feature Extraction	Alternating Direction Method of Multipliers	- ADNI [86] - PPMI [87] - MIRIAD - UK Biobank
[69]		Institutions	Brain Tumour Imaging Classification	CNN: U-Net	FedAvg	BraTS 2018 [88]
[70]				Deep Neural Networks		
[71]		Electroencephalography (EEG) Devices	EEG Signal Classification	CNN	FedAvg	MindBigData [89]
[72]	IoT Systems	Gateways monitoring IoT devices	Anomaly Detection	RNN - GRU	FedAvg	Self-Collected Data
[8]		IoT Objects or Coordinator (Cloud Server - Edge Device)	Lightweight Learning for resource-constrained devices	Deep Neural Networks	n/a	- MNIST [39] - Spambase [90]
[9]		IoT Devices	Computation Offloading	Double Deep Q Learning	n/a	n/a
[73]		Mobile Phones & Mobile Edge Computing Server	Improvement of IoT Manufacturers services	Partitioned Deep model training	n/a	MNIST [39]
[74]	Edge Computing	User Equipment Edge nodes	Computation Offloading Edge cashing	Reinforcement Learning	n/a	Self-Collected Data
[75]	Networking	Machine Type Devices (MTD)	Resource block allocation & Power transmission	Markov Chain	Aggregation of MTDs Traffic Models	n/a
[12]	Robotics	Robots	Robots Navigation Decision	Reinforcement Learning	A proposed Knowledge Fusion Algorithm	Gazebo simulator
[76]	Grid-world	Agents	Building Q-network Policies	Q-network	Multilayer Perceptron	- WHS [91] - WHG [92] - Cooking Tutorial
[77]	FL Enhancement	Edge nodes	Determination of the aggregation frequency	Gradient-descent-based ML models	FedAvg	- MNIST [39] - Energy [93] - User Knowledge Modeling [94] - CIFAR [40]
[78]	Recommender System	Any user device (e.g. laptops, phones)	Generation of personalized recommendations	Collaborative Filtering	Gradients Aggregation to update factor vectors	- Self-simulated Data - MovieLens [95] - In-house Production
[79]	Cybersecurity	Gateways monitoring Desktop nodes	Anomaly Detection	Autoencoder	FedAvg	CICIDS2017 [96]
[80]	Online Retailers	Customers	Click-Stream Prediction	RNN - GRU	FedAvg	Chinese Online Retailer
[18]	Wireless Communication	AR-enabled users	Edge cashing	Autoencoder	n/a	MovieLens
		Radios	Spectrum Management	*a spectrum utilization model*		n/a
		Entities in the core network	5G Core Network	n/a		*horizontally or vertically fragmented data*
[81]	Electric Vehicles	Vehicles	Failure Prediction of EVs	RNN - LSTM	Weighted Average based on loss function	Real-world EV dataset

complications, including the strict regulations and sensitivity of transferring such data, and other hurdles of slowing down information flow in healthcare where timely updates are critically important. FL is applied in [10] to address these issues. However, the authors interpret that with a large number of data sources with different amounts of data having different properties, it will be hard to achieve a tradeoff between what the model is globally learning in the light of local information from each data source. Therefore, the authors propose a new strategy called FADL, where the first layer of the neural network model is trained in a federated way using data from all sources, while the other layers of the neural network model are trained locally in each data source. Their proposition shows accuracy similar to a centralized analysis, which outperforms the application of the regular FL for distributed electronic health record (EHR). Huang *et al.* [11] proposed a community-based FL algorithm to predict mortality and hospital stay time. Electronic medical records are clustered into communities inside each hospital based on common medical aspects. Each cluster learns and shares particular ML model, which improves the efficiency and performance of the latter being customized for each community rather than general global one shared among all hospitals and hence patients. In [65], FL is also leveraged to predict hospitalizations during a target year for patients having heart disease using EHRs data spread among multiple data sources. Two scenarios are considered. The first is a semicentralized scenario where each agent/data source is holding multiple samples, while the second one is fully decentralized where each agent is holding one sample. In the first scenario, these agents are hospitals that process data of their patients and exchange messages with other hospitals to predict hospitalization. As for the second scenario, these agents are the patients who maintain their personal data and exchange messages among each other to collaboratively answer the hospitalization question. While information processing may happen at any of these levels, the proposed cluster primal dual splitting shows improved convergence rate compared to other alternatives. Intrusion detection systems based on FL is designed in the field of medical cyber–physical systems [66]. Private data from patients' devices (e.g., heart rate, blood oxygen saturation, etc.) are locally trained to enhance a global model, which can be used by the same or other patients to detect malicious activities. To provide high-performance model, homogeneous patients with similar characteristics are clustered, and each cluster creates its personalized local and global model. A federated transfer learning approach for wearable healthcare devices is proposed in [67]. While data might be distributed in different clouds and might not be exchanged due to imposed regulations, the proposition applies federate transfer learning in order to share knowledge. Additionally, this practice allows the needed customization of the models as different users have different characteristics and activity patterns. A framework for FL is proposed in [68] for the analysis of biomedical data. Using the feature selection and alternating direction method of Multipliers for the local task and aggregation method, respectively, this work investigates subcordinal brain changes in multiple diseases such as neurological disease. Sheller *et al.* [69] and Li *et al.* [70] focused on medical image prediction for brain tumor segmentation while considering FL. Their solutions allow the collaboration of multiple institutions by sharing their locally computing models. The latter is trained using U-net and DNN in [69] and [70], respectively. Moreover, differential-privacy techniques are implemented in [70] to prevent data leakage. FL is also used in [71] to classify electroencephalography (EEG) signals collected from various devices. From different area of the brain, signals are captured from many EEG devices, each responsible for training a CNN model to be sent for aggregation.

C. IoT Systems

To limit the vulnerabilities of large-scale IoT devices, FL is implemented in IoT systems. Due to the intensive computation loads engaged on-devices, edge computing is envisioned to supplement and offload tasks from IoT to edge nodes. Nguyen *et al.* [72] proposed an intrusion detection system based on anomaly detection for IoT. Different security gateways, each monitoring the traffic of one particular device type, locally train the gated recurrent unit model and send it to an IoT security service for aggregation. Such a system works without user intervention and is able to detect novel attacks. Jiang *et al.* [8] proposed a lightweight learning model for resource-constrained devices, especially in the IoT system. First, the proposed solution applies the Gaussian random projection at the devices level in order to obfuscate training data. Next, for the participating devices that do not have enough computational resources for training, a coordinator takes over. Ren *et al.* [9] take into account that proxy data on the edge level is less relevant to the data stored on IoT devices. Therefore, the latter is responsible for training the models, while edge nodes perform the updates aggregation. Computation task offloading is the use case considered in this solution to show the efficiency of integrating FL in IoT, with the help of edge computing when training deep reinforcement learning. Many aspects are considered in [73] to implement a fully secured FL approach for IoT. First, considering the limited computation resources of IoT devices, a mobile phone collects the device data, extracts features using the CNN network, and adds the Laplace noise to perturb the extracted features. Next, dense layers are trained in the MEC server. Afterward, the models, hashed and signed by participating devices using their private keys, are sent to a blockchain. To detect and prevent compromised clients from sending malicious updates, miners are responsible of verifying the identity of the senders by checking their signatures, and then downloading the models and aggregating their parameters. Subsequently, one selected minor encrypts the final model and uploads it to the blockchain. The selection of the miner among the clients is temporary and depends on some rewards given by a designed reputation-based crowdsourcing system. If correct and efficient model is uploaded, the client gets rewards and increases his reputation. Otherwise, he gets penalties with deduced reputation. Such incentive mechanism prevents the clients from misbehaving by providing them some services from the IoT manufacturer.

D. Other Application Areas

To start with, FL has been implemented in edge systems while integrating deep reinforcement learning in [74]. For the edge-to-cloud scenario, edge cashing is optimized by allowing edge nodes to train the shared model. As for the user equipments-to-edge scenario, computation offloading is optimized with User Equipments as clients training the model.

Habachi *et al.* [75] leveraged FL in order to dynamically allocate resource blocks and transmit power for machine-type devices that might be on regular or alarm mode. On the other hand, federated reinforcement learning in robotics is applied in [12]. The work allows robots to fuse and transfer their learning experience in order to quickly adapt to new environmental settings. Assuming that various agents can all benefit from joining a federation when building decision policies, Zhuo *et al.* [76] proposed an FL method based on reinforcement learning aiming to learn *Q*-network policy for agents by only sharing limited encrypted information among them. Wang *et al.* [77] introduced an adaptive approach to determine a tradeoff between local model updates and global aggregation parameters, which is able to minimize the learning loss under the resource constraints of the clients. As many clients can participate in FL, Din *et al.* [78] proposed a collaborative filtering method for FL settings. The work generates a recommender system by personalizing recommendations for a user on the basis of feedback of other clients. The federation method has been proved to be applicable without loss of accuracy. FL is developed for anomaly detection in [79]. Using blockchain technology, the proposed solution supports the auditing of autoencoder models learned from different nodes to detect anomalies. Yoo *et al.* [80] chose to apply FL to online retail business activities. From browsing sessions, data generated from each user's click-stream is analyzed and trained using the gated recurrent unit in FL settings. This enhances the prediction of the consumer's next browsing activities. Niknam *et al.* [18] preserved the privacy of the data in wireless communication. After introducing FL and its salient features, the authors discuss several possible applications in this field, while mainly focusing on content cashing and data computing in the edge, spectrum management, and 5G core network. Lu *et al.* [81] analyzed driver behavior metrics to predict the failure of EV in terms of battery and associated accessories. Among LSTM, the gradient-boosted decision tree and random forest, the former shows better prediction and has been then deployed as the ML model in the proposed FL-based framework.

E. Discussion

After discussing the current FL-based applications, we analyze the relevant lessons learned and opened challenges. Given the demand and urgency of guaranteeing the privacy of data, the growing number of applications at an unprecedented rate is adopting FL, which played a remarkable role and improved their quality.

1) In the IoT, the first challenge is that all of the IoT system-level characteristics, such as: a) the heterogeneity in the device's capability, in terms of hardware, connectivity, power and b) the size of the network and the constraints on each device affecting their ability to be active in the FL process, make impediments, including stragglers and fault tolerance more prevailing than in other environments such as data centers. Furthermore, communication methods should be efficient as they are much more expensive in such an environment.
2) While the reinforcement learning-based FL solution is able to fuse the learning experience and transfer it for navigation in a new environment, making FL-enabled robotics navigation deal with various input/output dimensions in order to offer a wider range of assistance in robotics systems, is still an open challenge.
3) While FL proves it ability to preserve privacy in recommendation systems, there are still many challenges to address in this area. First, coping with online learning to benchmark the system, in other words, analyzing real-life systems having continuous asynchronous updates coming from clients. Additionally, handing over methods for analyzing the communication capacity and efficiency is challenging in such systems. Furthermore, the challenge of providing techniques to secure the recommendation systems learning models from attacks and threats.
4) For cybersecurity, coming up with an aggregation algorithm that can deal with all of the hardware heterogeneity, unreliable connectivity and spasmodically connected nodes for mitigating the poisoning attacks before storing the weights updates on the blockchain is yet challenging.
5) One important challenge in the wireless communication is the robustness of the models where any of the communication bandwidth, noise, interference, and other aspects are factors that can intensify the channel bottleneck. In addition, the convergence time is another considerable challenge, where it depends not only on the local nodes and centralized aggregator but also the quality of the communication channel among them, which should be considered when optimizing the frequency of exchanging the updates and the one of aggregation. Finally, the wireless channel quality between the aggregator and any of the local learners affects the training process which is further challenging.

VII. Privacy and Security

Although the first-order concern in FL was revolved around fulfilling rigorous privacy protections by preventing data sharing, novel challenges related to privacy and security have jumped up. Recent efforts have clearly proved that the transmission of the model updates can still reveal sensitive information about clients [19], [97], and even worse can induce security issues [98]. In this section, we overview the pertinent approaches addressing these concerns.

A. Privacy

Existing privacy-preserved algorithms can still put users privacy at risk. As demonstrated in [99], attackers in FL may leak information from the clients training data. The authors show that malicious client is able to infer the existence of exact data points in the training set such as specific locations. Moreover, how properties from participating clients data can be inferred are as well investigated. In consequence, serious privacy guarantees are required to secure FL models. As participants can freely join and disconnect from a communication round throughout the process, FL settings give rise to various

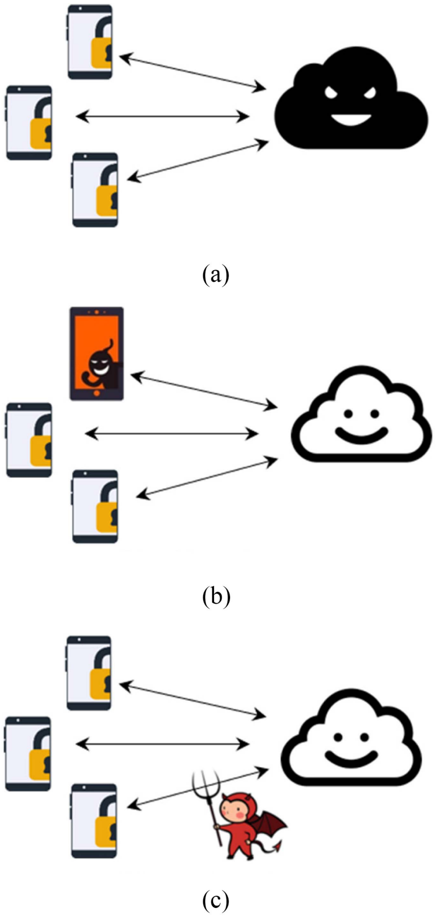

Fig. 5. Different malicious actors in vulnerable FL systems. (a) Malicious server. (b) Insider adversary. (c) Outsider adversary.

threat models and vulnerabilities from many actors. The latter are presented in Fig. 5 and classified as follows.

1) *Malicious Server:* An honest-but-curious server can inspect users updates without altering the model. In contrast, potential malicious server not only can inspect the updates but also can tamper the model. Wang *et al.* [100] are the first to consider attacks in FL from a malicious server rather than clients. The proposed framework incorporates multitask generative adversarial networks, where discrimination on the user identity is achieved by attacking client-level privacy.
2) *Insider Adversary:* Similar to the aforementioned actors, honest-but-curious and malicious clients participating in the learning rounds exist.
3) *Outsider Adversary:* When communicating the updates between trusted clients and the server, eavesdroppers on the channel can show up.

In the light of such threats, recent propositions have been advanced in order to prevent data leakage. One of these works was presented by Ma *et al.* [98]. They investigated the issues related to privacy and security in the FL system. First, several protection solutions have been discussed, when applying privacy at both client and server sides, in addition to when applying security for the whole FL framework. Next, privacy and security issues have been classified as convergence, data poisoning, scaling up, and model aggregation. For each category, some experiments and possible solutions have been proposed for a secured privacy-preserving FL system. On the other hand, other researchers have used different privacy techniques, which are illustrated in Fig. 6 and fall mainly under the umbrellas of cryptographic protocols and differential privacy (DP).

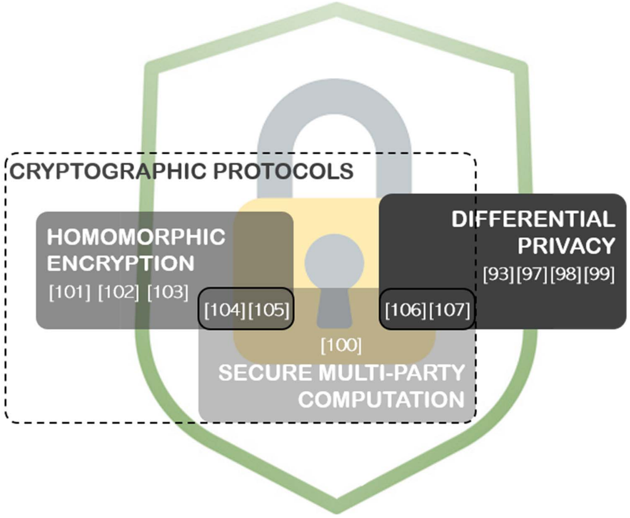

Fig. 6. Main privacy techniques used for privacy-preserved FL systems.

Differential Privacy: Works by injecting some noise in order to mask the influence of the client on the model parameters [97]. Geyer *et al.* [101] introduced an algorithm aiming to address data leakage using DP. In the proposed algorithm, two methods are used: 1) random subsampling, where in each communication round the server selects a random subset of clients to share the global model with and 2) a Gaussian technique is applied to distort the aggregated update, yet assuring that this does not exceed certain limit as it will add undesirable noise that affects the accuracy of the learning process. A new version of the federated averaging algorithm is proposed in [102], where moments accountant is used to satisfy user-level privacy. In this work, random rather than a fixed number of clients is selected per round. Moreover, flat and per-layer clipping strategies are imposed per-user updates. Besides, different estimators for the parameters aggregation, and the Gaussian noise to the final model, are as well used. Choudhury *et al.* [103] applied a DP mechanism for healthcare applications. Experiments on real-world health data sets have been conducted, and the results show that, without DP, FL has close performance to the centralized system. Moreover, significant loss for the studied healthcare applications is reported when applying DP, even though it increases the privacy level. This shall motivate the researchers to consider such applications for the future DP-based systems.

Secure Multiparty Computation (SMC): It is a subfield of cryptographic protocols with the goal of revealing nothing but the output when multiple parties jointly perform an arbitrarily function over their private input. A study in [104] has used SMC to build FL systems. The proposed protocols consider secret sharing, which adds new round at the beginning of the process for the keys sharing, double-masking round that protects from malicious server, key agreement that efficiently

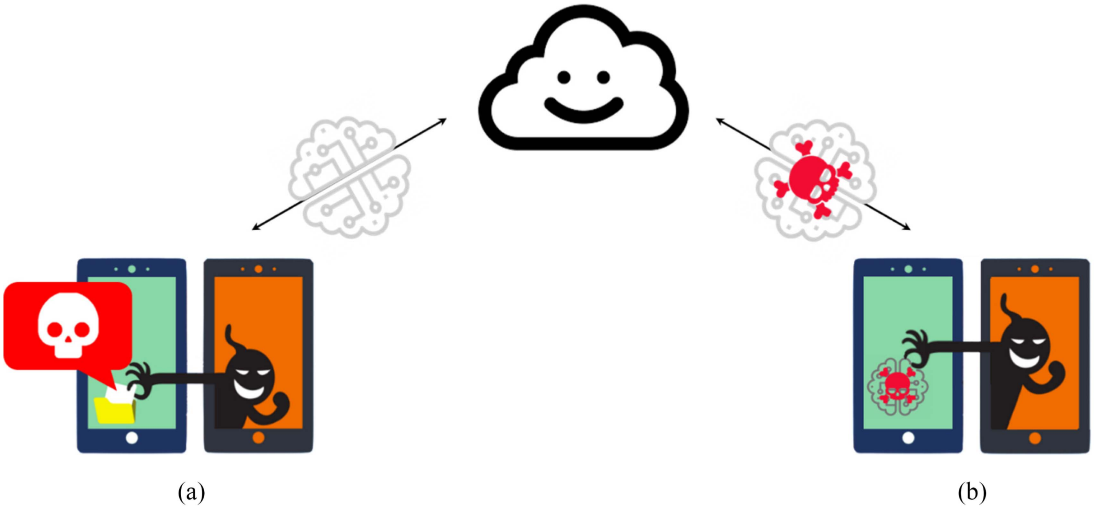

Fig. 7. Data and model update poisoning attacks in FL environment. (a) Data poisoning. (b) Model update poisoning.

exchange secrets, and server-mediated key agreement that minimizes trust.

Homomorphic Encryption (HE): It is a form of encryption that protects clients data by performing computation directly on ciphertexts [105]. SecureBoost, a lossless tree-boosting system for privacy preserving, is proposed in [106] using HE. The novelty of this article lies in collaborating models of multiple parties having data vertically rather than horizontally partitioned. In other words, the data set is split based on a feature dimension over different parties, each considered as either active or passive. The latter holds a data matrix for his assigned feature set, while only the former holds the labeled class in addition to his own data matrix, and acts as the dominating server in FL. Each party in SecureBoost trains a tree-bosting model to finally build a tree ensemble model. SecureBoost has been found to be robust in terms of accuracy compared to the nonfederated tree-boosting systems, while maintaining data privacy. In [107], vertically partitioned data are handled for a private FL using HE. Specifically, in cross-feature space, logistic regression is privately federated using Paillier encryption. Moreover, entity resolution errors that affects the learning process is analyzed.

Hybrid Protocols: Another line of work uses combined techniques to more protect raw data. Federated transfer learning is proposed in [108] to build a privacy-preserved FL framework. For minimal adjustment to the NN architecture, HE to multiparty computation is used in this approach. In [109], the privacy-preserved FL system is built using transfer learning across heterogeneous feature space. The proposed approach, which is provided under HE and secret sharing settings, involves the following steps: 1) secure domain adaptation; 2) secure feature mapping; 3) secure FL; 4) secure model integration; and 5) local model inference. The work in [110] emphasizes on the need of computing multiparty aggregation, in which none of the participants reveals its update, not only among each other but also to the aggregator. The proposition encompasses different cryptographic primitives, which includes secret sharing, key agreement, authenticated encryption, pseudorandom generator, signature scheme, and public-key infrastructure. Moreover, the combination of DP and secure aggregation has been discussed in this article. Truex *et al.* [111] proposed to implement both DP and SMC in an hybrid approach. It has been shown in the experiments that the proposed solution is able to train decision trees, SVM, and CNN models.

Other Techniques: Beyond DP, SMC, and HE techniques, Chang *et al.* [112] has built a system to protect against poisoning attacks. Rather than sending the model parameters to the server, the proposed approach allows to share the knowledge of built models in a black-box setting after being extracted and aggregated. Such a solution is based on knowledge transfer algorithms and supports heterogeneous-based models. The work in [113] aims to detect causative attacks, where adversaries feed the classifier with malicious activities that negatively impact the final model. The proposed approach ensures the integrity of DL training processes. The proposed solution in [114] allows clients to encode and compress the parameters of a trained neural network. The server then decodes them for aggregation, resulting in an end-to-end encrypted scheme, which guarantees that the updates are unexposed to the server, and are secured during the communication.

B. Security

Beyond malicious actors targeting user privacy, FL systems can be vulnerable to other type of attacks and potential points of failure. The latter are generally caused unintentionally by users, such as when training with messy data, noisy labels, etc. On the other hand, adversaries might attempt to harm the performance of the model depending on their intentions. Fig. 7 illustrates two types of attacks that adversarial attackers can target: data poisoning and model update poisoning.

Data Poisoning: Throughout FL learning process, one or more clients, who correctly behaved when participating in one or many previous rounds, may lately act maliciously and poison the joint model. Such adversary is able to manipulate the training phase through *clean-label* and *dirty-label* attacks. As the name applies, the latter allows to directly replace the labels with miss-classified ones, whereas the former looks innocuous as it injects poisoned data causing the model itself to miss-behave without any control from the attacker side over the labeling. *Label-flipping*, which is a special case of *dirty-label* attacks, has been proved in [115] to be one of the FL vulnerabilities. Based on the conducted experiments, it is clearly demonstrated that with only two malicious sybils, the final model is highly affected. This highlights the problem of equivalence influence of all received updates in the FL system. The authors also show that existing defenses in ML are not applicable to FL settings, especially with sybil-based attacks. Therefore, they have proposed a new solution to prevent such attacks based on contribution similarity of the clients. This type of attack can be mitigated using the aforementioned DP technique.

Model Update Poisoning: Instead of injecting malicious data into the training set, model update poisoning attacks directly corrupt the global model by attempting to fool the local model. Compared to data poisoning attacks, model poisoning ones look less natural but are much more effective as shown in [116] and [117]. The intruder can either perform independently or with other colluding participants. Moreover, Bagdasaryan *et al.* [116] introduced stealthy backdoor into the global model by proving that any client involved in the FL steps can present a hidden backdoor functionality in the shared global model. They show that a single-shot attack from one attacker is enough to achieve 100% accuracy on the backdoor task. In their conducted experiments on the word-prediction task, eight participants out of 80 000 are considered malicious, and are able to achieve 50% accuracy on backdoor compared to 400 intruders in the data poisoning attacks.

Defenses to Poisoning Attacks: Few approaches have been advanced in order to secure the system against poisoning attacks. In [118], the support of a central coordinator in vanilla FL is replaced by blockchain. In such practice, local models are shared and verified in the blockchain network while providing rewards to the clients. The overall latency of the learning process is formulated and minimized in this work. Yin *et al.* [119] proposed, in an IoT environment, a secure system for data collaboration. To ensure privacy and security of the data, efficient data access control is built using the blockchain paradigm under the settings of FL, which guarantees secure collaboration for large-scale distributed data computation. Ilias and Georgios [105] considered a scenario where one client has the problem to solve, some hold the appropriate data, and others have devices with enough computational resources. For such a scenario, an encryption scheme is proposed, in which the initial client creates public and private keys and encrypts the model parameters. Appropriate clients then collaborate to utilize the offered resources with the private encrypted data in order to successfully train the model. Blockchain technology and data integrity are as well used in the proposed approach for a more robust FL solution.

C. Discussion

In this section, privacy and security have been discussed when the clients update and send the model parameters. The main idea of FL was to bring ML models to the data source in order not to bring the data to the model, therefore guaranteeing data privacy. However, we have seen that malicious actors can not only reveal personal data from the client updates but also poison the training data and the learning model. Current works on FL security and privacy propose lossless methods and prove their efficiency while preserving the original accuracy. However, some of these techniques impose significant extra communication cost, while other methods incorporate a bunch of hyperparameters that not only affect the accuracy but also distress the communication. When strong security and privacy guarantees are indispensable, new techniques that limit the power of any potential adversarial party can enable stronger guarantees and lead to improved performance. Moreover, a fusion between compression techniques and DP would offer advanced benefits. Furthermore, security and privacy constraints might diverge from one device to another or even across the pieces of data on a single one, which is challenging. Therefore, new techniques that can address a variety of samples data-specific and device-specific boundaries look promising from such perspective.

VIII. Resource Management

FL is applied in dynamic environments, in which the clients have constrained resource devices and are communicating through bandwidth-constrained networks where some devices can share the same link. Therefore, many contributions have been focusing on resource management to take the best decision related to the selected clients, learning hyperparameters, number and duration of training rounds, and aggregation strategies. In this context, various optimization problems have been defined and solved assuming the availability/predictability of subsets of the following metrics (Fig. 8).

1) *Clients Reliability:* Resources (CPU, energy), location traceability (GPS coordinates), local training time, and quality of updated parameters (accuracy, loss). Some related work assume the availability of the "actual" values of these metrics at each learning round while others adopt various approaches to "predict" these values.
2) *Network Link Quality:* Uplink/downlink bandwidth either already available or possibly allocated.
3) *Central Aggregation Server:* Aggregation time, global model accuracy, and loss.

The proposed optimization approaches in the literature target various objectives, including global model (accuracy and loss) convergence time, clients consumed resources, and wireless links usage. In this regard, we provide in Table III a taxonomy of the FL resource management approaches with respect to their objective functions and considered parameters. In the following, we summarize these contributions by highlighting the main optimization problem, followed by the approaches specific to wireless networks and those covering clients models parameters. Finally, we present approaches relying on computation offloading and clients hierarchy.

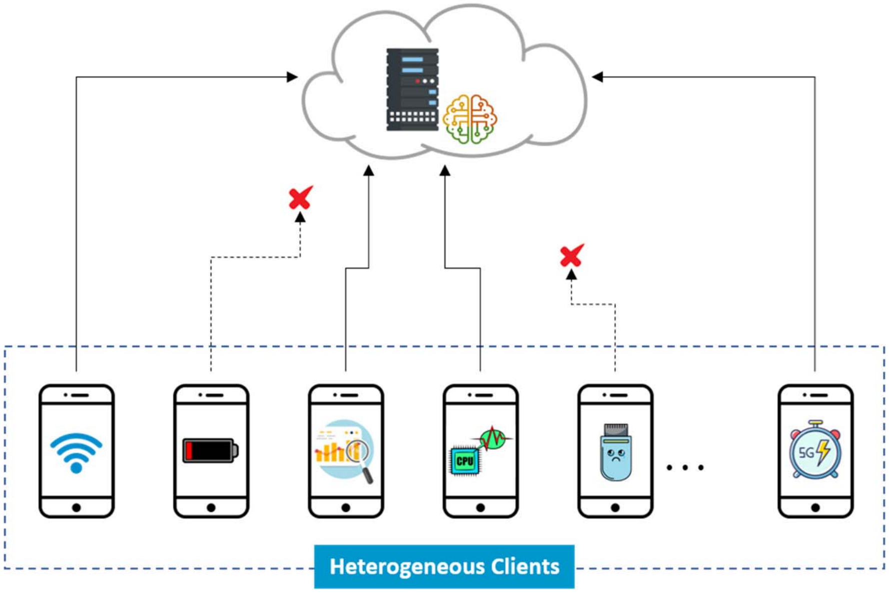

Fig. 8. Various metrics considered by FL resource management strategies.

Main Optimization Problem: Most of the researchers have investigated various strategies of client selection and resource allocations and analyzed their impact on model convergence. Ren *et al.* [120] addressed the problem of accelerating the DNN training tasks by jointly optimizing the local training batchsize and wireless resource allocation. They investigate both the CPU-based and GPU-based training tasks scenarios. The key idea of this contribution is the definition of a global loss decay function targeting the training batchsize, based on which a learning efficiency criterion is elaborated considering the ratio between global loss decay and end-to-end latency. Wang *et al.* [77] have performed analysis on the convergence bound for FL while considering non-IID data distributions. Moreover, they propose a control algorithm achieving the intended tradeoff between local update and global aggregation while minimizing the loss function with limited resource budget. Jin *et al.* [121] formulated clients selection problem as an online time varying nonlinear integer program, which minimizes the total cumulative usage of the computation and communication resources, subject to the server capacity and the long-term convergence requirements for both local and the aggregated models on each device and on the sever, respectively. They design an online learning algorithm to make fractional control decisions based on both previous system dynamics and training results.

Approaches Related to Wireless Networks: A special attention has been given to study FL resource allocation in the context of wireless networks. Tran *et al.* [122] formulated an FL problem over wireless networks that captures the following tradeoffs: 1) learning time versus clients energy consumption by adopting the Pareto efficiency model and 2) computation versus communication learning time by calculating the optimal learning accuracy. Shi *et al.* [123] proposed an approach for maximizing the convergence rate of the FL training with respect to time by formulating a joint bandwidth allocation and scheduling problem for minimizing the training time and attain the desired model accuracy. For the bandwidth allocation problem, they design an efficient binary search algorithm, while for maximum device scheduling, they adopt a greedy approach for achieving a tradeoff between the latency and learning efficiency in each round. Chen *et al.* [124] formulated the joint learning, wireless resource allocation and client selection problem as an optimization problem minimizing the FL loss function. A closed-form expression is proposed to quantify the impact of wireless factors on the FL convergence rate. They use the Hungarian algorithm for finding the optimal user selection and resource allocation in order to minimize the FL loss function. Khan *et al.* [125] presented an approach for self-organizing FL over wireless networks. They adopt a heuristic algorithm for minimizing the global FL time while considering the local energy consumption and resource blocks. Yang *et al.* [126] proposed a model for analyzing and characterizing the performance of FL. Tractable expressions are derived for the convergence rate of FL considering the effects of both scheduling schemes and intercell interference. Moreover, they studied the effectiveness (convergence rate) of random scheduling, round robin, and proportional fair scheduling policies. From the studied contributions, we can see that the main challenge of FL management in the context of mobile and wireless networks is optimally sharing the bandwidth between participating clients. As for managing energy consummation, it is based on: 1) reducing the transmission of model parameters updates and 2) optimizing the local model training. While, the former strategy is efficient, the latter strategy is constrained by the heterogeneity of wireless devices and their other computation tasks.

Approaches Covering Clients Model Parameters: Recently, some researchers have started investigating scheduling techniques directed by the model improvements opportunities during FL rounds. Amiri *et al.* [127] designed scheduling

TABLE III
TAXONOMY OF THE RESOURCE MANAGEMENT OPTIMIZATION APPROACHES WITH RESPECT TO THEIR OBJECTIVE FUNCTIONS AND CONSIDERED PARAMETERS

Refs	**Client Resources (CPU, Energy)**	**Local Model Train / Upload**	**Global Model Download**	**Client Parameters Quality**	**Client Location**	**Optimization Target**
[77]	- Available CPU - Set by Clients	- Train: energy, time	n/a	n/a	n/a	- Maximize accuracy - Minimize loss
[120]	- Available CPU, RAM - Set by Clients	- Train: time - Upload: time	- Time - Bandwidth	n/a	n/a	- Minimize loss - Minimize batch size
[121]	- Available CPU - Predicted	- Upload: bandwidth	- Device CPU - Bandwidth	- Accuracy per round	n/a	- Minimize device resources consumption - Minimize allocated bandwidth
[122]	- Available CPU - Set by Clients	- Train: energy, time - Upload: Bandwidth, energy, time	- Device CPU, energy - Bandwidth	n/a	n/a	- Minimize clients resource consumption - Minimize Convergence time
[123]	n/a	- Train: time - Upload: time	n/a	n/a	n/a	- Minimize loss - Minimize time
[124]	n/a	- Upload: energy, bandwidth, Packet Error Rate	- Bandwidth	n/a	n/a	- Minimize loss
[125]	- Available CPU - Set by Clients	- Train: time, energy - Upload: time	n/a	n/a	n/a	- Minimize loss - Minimize allocated bandwidth
[126]	n/a	- Train: time - Upload: time	n/a	n/a	n/a	- Maximize accuracy
[127]	n/a	- Upload: bandwidth, time	n/a	- Accuracy per round	n/a	- Maximize accuracy
[128]	- Available Energy - Set by Clients	- Upload: energy, bandwidth	n/a	n/a	n/a	- Minimize energy - Maximize accuracy - Minimize loss
[129]	- Available CPU - Predicted	- Train: time - Upload: time	- Time	- Accuracy per round	n/a	- Minimize convergence time
[130]	n/a	- Train: time - Upload: time	- Time	- Accuracy per round	n/a	- Maximize accuracy
[131]	- Available CPU - Actual/Predicted	- Upload: Success/Failure	- Success/Failure	n/a	- Actual/Predicted	- Maximize accuracy - Maximize valid participants ratio
[132]	- Available Energy - Set by Clients	- Train: time, energy - Upload: time, energy, bandwidth	n/a	n/a	n/a	- Minimize time - Minimize energy
[133]	n/a	- Upload: bandwidth	- Bandwidth	n/a	- Actual	- Maximize accuracy - Minimize loss

policies for deciding on the subset of devices to handle the transmission within each round based on both channel conditions and significance of local model updates. Experimental results show that the proposed approach offers better long-term performance than scheduling-based only on one of the two metrics. The contribution of [128] provides a long-term perspective for resource allocation in wireless networks where clients share a common wireless link. The approach is based on the experimental observation demonstrating that selecting fewer clients during the initial learning rounds and gradually increasing this number is the strategy having the best impact on learning performance. The authors formulate a stochastic optimization problem for selecting a client and allocating bandwidth while considering long-term client energy limitations. A key design element of this contribution is leveraging the Lyapunov technique and constructing a virtual energy deficit queue for each client. Chai *et al.* [129] proposed a tier-based FL (TiFL) system classifying the clients into tiers based on the performance of their training while applying adaptive tier-based clients selection. To deal with heterogeneity in resources and data, the scheduling algorithm adopts a "credits" budget for each tier. Ren *et al.* [130] proposed a scheduling policy to exploit both diversity in multiuser channels and diversity in the importance of the edge devices' learning updates (measured by gradient divergence). They propose a new probabilistic scheduling framework to yield unbiased update aggregation. Huang *et al.* [131] proposed a proactive algorithm that selects mobile clients based on the prediction of their future training and reporting qualities. The adopted approach consists of two main parts: 1) predicting users' mobility trajectory patterns and their smartphones' App-usage habits and 2) a deep reinforcement learning-based client-selection algorithm handling the unexpected dynamic events occurred in a metropolitan MEC environment. The monitored/predicted metrics are CPU, bandwidth, GPS coordinates, and success/failure of global parameter downloading and local parameter uploading. Although considering the quality of clients model updates is very promising to improve the efficiency of FL management techniques, these approaches are facing many challenges. First, there is no traceability between the communicated models parameters and the actual local training activities. Second, it is very difficult to identify situations of (non-IID) data. Finally, there is no guarantee that a client that provided good parameters in the previous rounds will provide parameters of the same quality in future rounds

Offloading-Based Approaches: An interesting direction is the offloading to edge nodes and hierarchical organization. Luo *et al.* [132] introduced a hierarchical federated edge learning framework, in which model aggregation is partially offloaded to edge servers from the cloud. A joint computation and communication resource allocation and edge association problem is formulated and solved. Bandwidth, time, and energy constraints are considered while optimizing convergence time and resource consumption. Abad *et al.* [133] proposed an approach targeting a heterogeneous cellular network. The FL is orchestrated among the mobile users within their cells by small base stations, which periodically communicate the model updates to the macro base station for global consensus. Their approach ensures efficient communication through joint sparsification and periodic averaging and a resource allocation strategy for minimizing the end-to-end latency. While appealing for FL computation offloading, these approaches have limited applicability in the context of mobile devices (mainly smartphones) where there is almost no possibility for organizing devices. However, these approaches are expected to have great impact on FL adoption in the context of wireless sensor networks where most sensor devices have severe resources limitations and are usually organized in a hierarchy around more powerful edge devices.

Discussion: The main target of the presented approaches is to reach the best global learning performance (minimizing loss and/or maximizing accuracy) while optimizing resource consumption. However, the efficiency of these approaches depends on the honesty of clients when communicating required metrics (CPU, time, etc.) or on the reliability of prediction algorithms. In addition, a central server has no verification opportunities over the size and quality of data used to train clients local models. This explains why only few of the existing approaches ([127] and [129]) take into consideration the quality of clients model parameters. Indeed, a central server has no control over the resources monitoring tools of involved clients. The wireless network bandwidth is the only resource dynamically assigned to clients and managed by existing approaches provided that a scheduling entity is located at a network node like a base-station.

IX. Research Directions

FL is an emerging yet innovative learning paradigm. Although many research efforts have addressed different architectural, technical, application, and deployment aspects, more efforts are still needed for FL to mature. Besides, many demanding open directions need to be explored, and new possibilities for FL applications and improvements need to open up. In this context, following the same classification of FL topics and research areas depicted in Fig. 3, we present in this section some of the challenges and future directions as a large potential for practitioners and researchers.

A. Core System Model and Design

This category spans over different technical aspects of FL, including the used ML algorithms, optimization, and aggregation mechanisms, techniques for communicating the models, deployment models and data distributions, and adopted frameworks, among others. In this regard, the baseline aggregation algorithm, *federated averaging*, has been developed to only consider the data set size to aggregate and weigh the updated models. However, the convergence of such algorithm is application-dependent and more sophisticated methods are worth investigating. New methods can help reach the desired accuracy with a less number of communication rounds, which reduce the communication cost. Moreover, algorithms other than neural networks are highly encouraged in FL implementation. Such algorithms with smaller model size can also help minimize the communication and computation cost. Even though several encouraging approaches have been proposed in this context, there is still a lot of room for future work. Furthermore, another fundamental aspect in FL is the selection of participating clients. Typically, from one round to another, different sets of clients are selected at complete [3] or quasi [41] randomness. When the selection comes to some clients with limited resources, such as IoT devices, not only longer processing time is engaged by the client, but also failure in completing the training task might occur, and accordingly affect the model accuracy. Therefore, the random selection of clients leads to less number of updates sent by the clients and hence some FL rounds will be discarded [31]. Thus, more efforts are needed to optimize the FL client selection while considering the network characteristics and the survivability of the devices chosen for training the models.

B. Application Areas

The wider set of efforts and contributions are targeting the application areas, in which the healthcare and IoT systems are the widest targeted fields. In another direction, in-edge FL proved good performance efficiency with minimal learning overhead, yet several challenges still need to be considered in this area. First, elaborating customized techniques for optimizing the learning computation tasks is still challenging. Additionally, scheduling methods for the collaborative AI tasks, whether on the edge nodes or the mobile devices, is needed.

Moreover, autonomous vehicles and unmanned aerial vehicles (UAVs) are promising fields which could have plenty of useful applications, such as taxis, food delivery, medical delivery, VR applications, inspections, public safety, accident reports, traffic monitoring, etc. The UAV applications are classified into three categories [134]: 1) delivery systems; 2) real-time multimedia streaming; and 3) intelligent transportation systems, each exposed to many wireless and security challenges. To address the latter, Challita *et al.* [134] have introduced an FL-based solution for the first and third category without providing a complete framework. Accordingly, investigating the appropriate FL approaches for autonomous vehicles and UAVs-based systems might be a promising direction to invest in.

Furthermore, Aïvodji *et al.* [135] have proposed a use case for smart homes in the context of FL. In their solution, different users sharing the same smart device can benefit from the trained model, and different devices in the smart home can benefit from other devices' data and models. In this context, when smart home devices are hit with attacks, IDS-based architecture could be implemented, where we can assume that: 1) all connected devices have enough resources to perform the training task; 2) none of the devices has the needed resources and a guardian can take care of the training; and 3) some of the devices are capable of training the models. For this described architecture, Aïvodji *et al.* [135] have presented a full simulated test-bed toward its implementation. The smart home environment might constitute an excellent match to investigate the deployment of FL.

On the other hand, most of the existing solutions consider labeled data for FL applications. However, in real scenarios, it is challenging to have labeled data set, or even high-quality labeled one. Therefore, emerging solutions to address such limitation are highly needed.

C. Privacy and Security

Although the privacy and security have been among the initial objectives for adopting FL as pertinent solution, the distributed aspect has raised additional problems to address, such as revealing sensitive information about users or poisoning local data and shared models. Although recent efforts adopted different privacy-based solutions, some challenges are still ahead. When DP is used, various levels of noise are injected, which result in several drawbacks. First, the noise can hurt the built model leading to loss in the accuracy. Acceptable accuracy can be only maintained with a small number of devices participating. Furthermore, such practice does not protect data privacy against malicious server. On the other hand, even though cryptographic methods are considered lossless, intensive communication overhead will be entailed hereby, and some methods are even not powerful to the extent that they can detect poisoning attacks. As a result, a call for designing robust privacy preserved and secure systems is urged, where formal guarantee of privacy and security is needed with tight accuracy loss.

D. Resource Management

Due to the heavy computation needed for ML training and learning in general, resource management plays a major role for achieving pertinent, sustainable and efficient FL-based solutions. In this regard, few works have started integrating edge computing into FL [8], [73], [74] for supporting end devices with additional computation resources. However, robust systems are still required in two main directions. First, with the critical bottleneck of FL, which lies in the communication bandwidth, some collaboration between edge nodes could decide on the best clients updates to be sent to the cloud, how frequent to send the updates, in addition to other criteria that help reduce the communication rounds. Second, since FL is not only embracing mobile phones but rather a wider range of devices, such as IoT, vehicles, etc., the training task may be moved or offloaded to the edge nodes to release intensive computation from resource-constrained devices [136], [137].

X. Conclusion

FL has emerged as an innovative learning paradigm, which copes with the growing computational capacities of devices, such as smartphones, wearable devices, and autonomous vehicles coupled with concerns about protecting private data. Motivated by the increasing demand of storing data locally and pushing ML computation to the end devices while reducing data communications overhead, many efforts have been undertaken by researchers to apply such FL training settings in numerous disciplines. In this context, this article presented in-depth and in-breadth investigation of the FL architecture, design, and deployment while comparing it to the centralized and distributed on-site ML-based systems. Moreover, a new classification of the FL topics and research fields was provided based on thorough literature review along with taxonomies for its crucial technical and emerging aspects, including the core system model and design, application areas, privacy and security, and resource management. Finally, few challenges and new research directions tailored for the future perspectives of FL have been discussed. We believe that the proposed approach in which we surveyed FL can offer fundamental insights into the future research progress and field advancement.

References

[1] D. Swinhoe. (Apr. 17, 2020). *The 15 Biggest Data Breaches of the 21st Century*. Accessed: Apr. 17, 2020. [Online]. Available: https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html

[2] B. K. Mathew, J. C. Ng, and J. L. Zerbe, "Using proxies to enable on-device machine learning," U.S. Patent App. 15 275 355, Jan. 25, 2018.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2016, pp. 1273–1282.

[4] T. Yang *et al.*, "Applied federated learning: Improving google keyboard query suggestions," 2018. [Online]. Available: arXiv:1812.02903.

[5] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," 2018. [Online]. Available: arXiv:1811.03604.

[6] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Netw.*, early access, Sep. 1, 2020, doi: 10.1109/MNET.011.2000286.

[7] S. A. Rahman, A. Mourad, M. El Barachi, and W. Al Orabi, "A novel on-demand vehicular sensing framework for traffic condition monitoring," *Veh. Commun.*, vol. 12, pp. 165–178, Mar. 2018.

[8] L. Jiang, R. Tan, X. Lou, and G. Lin, "On lightweight privacy-preserving collaborative learning for Internet-of-Things objects," in *Proc. Int. Conf. Internet Things Design Implement.*, 2019, pp. 70–81.

[9] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.

[10] D. Liu, T. Miller, R. Sayeed, and K. Mandl, "FADL: Federated-autonomous deep learning for distributed electronic health record," 2018. [Online]. Available: arXiv:1811.11400.

[11] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *J. Biomed. Informat.*, vol. 99, Mar. 2019, Art. no. 103291.

[12] B. Liu, L. Wang, and M. Liu, "Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4555–4562, Jan. 2019.

[13] A. Mourad, H. Tout, O. A. Wahab, H. Otrok, and T. Dbouk, "Ad-hoc vehicular fog enabling cooperative low-latency intrusion detection," *IEEE Internet Things J.*, early access, Jul. 10, 2020, doi: 10.1109/JIOT.2020.3008488.

[14] S. A. Rahman, A. Mourad, and M. El Barachi, "An infrastructure-assisted crowdsensing approach for on-demand traffic condition estimation," *IEEE Access*, vol. 7, pp. 163323–163340, 2019.

[15] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[16] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, p. 12, 2019.

[17] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, Aug. 2020.

[18] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities and challenges," 2019. [Online]. Available: arXiv:1908.06847.

[19] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019. [Online]. Available: arXiv:1912.04977.

[20] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020. [Online]. Available: arXiv:2003.02133.

[21] Q. Li, Z. Wen, and B. He, "Federated learning systems: Vision, hype and reality for data privacy and protection," 2019. [Online]. Available: arXiv:1907.09693.

[22] L. Li, K. Ota, and M. Dong, "Humanlike driving: Empirical decision-making system for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6814–6823, Aug. 2018.

[23] S. U. Amin, M. S. Hossain, G. Muhammad, M. Alhussein, and M. A. Rahman, "Cognitive smart healthcare for pathology detection and monitoring," *IEEE Access*, vol. 7, pp. 10745–10753, 2019.

[24] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, "Real-time credit card fraud detection using machine learning," in *Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. (Confluence)*, Jan. 2019, pp. 488–493.

[25] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake prediction based on spatio-temporal data mining: An LSTM network approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 1, pp. 148–158, Jan. 2020.

[26] P. Dube, T. Suk, and C. Wang, "Ai gauge: Runtime estimation for deep learning in the cloud," in *Proc. 31st Int. Symp. Comput. Archit. High Perform. Comput. (SBAC-PAD)*, Oct. 2019, pp. 160–167.

[27] X. Dai, I. Spasić, B. Meyer, S. Chapman, and F. Andres, "Machine learning on mobile: An on-device inference app for skin cancer detection," in *Proc. 4th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Jun. 2019, pp. 301–305.

[28] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, Jul. 2013.

[29] A. Pacheco, E. Flores, R. Sánchez, and S. Almanza-García, "Smart classrooms aided by deep neural networks inference on mobile devices," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, 2018, pp. 605–609.

[30] Y. Kim, J. Kim, D. Chae, D. Kim, and J. Kim, "μlayer: Low latency on-device inference using cooperative single-layer acceleration and processor-friendly quantization," in *Proc. 14th EuroSys Conf.*, 2019, pp. 1–15.

[31] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," 2019. [Online]. Available: arXiv:1902.01046.

[32] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016. [Online]. Available: arXiv:1610.05492.

[33] B. McMahan and D. Ramage. (2017). *Google Ai Blog*. Accessed: Feb. 2, 2020. [Online]. Available: http://www.mi.fu-berlin.de/inf/groups/ag-ti/theses/download/Hartmann_F18.pdf

[34] *Tensorflow Federated: Machine Learning on Decentralized Data*. Accessed: Apr. 14, 2020. [Online]. Available: https://www.tensorflow.org/federated/

[35] *Federated Ai Ecosystem—Collaborative Learning and Knowledge Transfer With Data Protection*. Accessed: May 10, 2020. [Online]. Available: https://www.fedai.org/

[36] *PYSYFT: A Library for Encrypted, Privacy Preserving Machine Learning*. Accessed: May 10, 2020. [Online]. Available: https://github.com/OpenMined/PySyft

[37] *PFL: Federated Deep Learning in Paddlepaddle*. Accessed: May 10, 2020. [Online]. Available: https://github.com/PaddlePaddle/PaddleFL

[38] *Nvidia Developer Blog: Federated Learning Powered by Nvidia Clara*. Accessed: May 10, 2020. [Online]. Available: https://devblogs.nvidia.com/federated-learning-clara/

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[40] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Rep., 2009.

[41] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[42] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL: Cooperative learning mechanism using non-IID data in wireless networks," 2019. [Online]. Available: arXiv:1905.07210.

[43] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," 2018. [Online]. Available: arXiv:1812.07210.

[44] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," 2018. [Online]. Available: arXiv:1812.11750.

[45] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 3400–3413, Sep. 2020.

[46] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.

[47] X. Yao, C. Huang, and L. Sun, "Two-stream federated learning: Reduce the communication costs," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, 2019, pp. 1–4.

[48] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 6341–6345.

[49] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 954–964.

[50] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–8.

[51] J. Konecnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016. [Online]. Available: arXiv:1610.02527.

[52] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastruct. Deep Learn. (DIDL@ Middleware)*, 2018, pp. 1–8.

[53] Y. Wang, "Co-OP: Cooperative machine learning from mobile devices," M.S. thesis, Univ. Alberta, Edmonton, AB, Canada, 2017.

[54] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," 2019. [Online]. Available: arXiv:1902.00146.

[55] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Edge-assisted hierarchical federated learning with non-IID data," 2019. [Online]. Available: arXiv:1905.06641.

[56] N. Guha, A. Talwlkar, and V. Smith, "One-shot federated learning," 2019. [Online]. Available: arXiv:1902.11175.

[57] V. W. Anelli, Y. Deldjoo, T. Di Noia, and A. Ferrara, "Towards effective device-aware federated learning," in *Proc. Int. Conf. Italian Assoc. Artif. Intell.*, 2019, pp. 477–491.

[58] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018. [Online]. Available: arXiv:1806.00582.

[59] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[60] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," 2011. [Online]. Available: arXiv:1109.5647.

[61] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. IEEE VTS Asia–Pac. Wireless Commun. Symp. (APWCS)*, Aug. 2019, pp. 1–5.

[62] 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.*, vol. 6, no. 6, pp. 10700–10714, Jan. 2019.

[63] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," 2019. [Online]. Available: arXiv:1906.04329.

[64] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, "Federated learning of out-of-vocabulary words," 2019. [Online]. Available: arXiv:1903.10635.

[65] 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. Informat.*, vol. 112, pp. 59–67, Apr. 2018.

[66] W. Schneble, "Federated learning for intrusion detection systems in medical cyber-physical systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Washington, Bothell, WA, USA, 2018.

[67] Y. Chen, J. Wang, C. Yu, W. Gao, and X. Qin, "FedHealth: A federated transfer learning framework for wearable healthcare," 2019. [Online]. Available: arXiv:1907.09173.

[68] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, 2019, pp. 270–274.

[69] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, 2018, pp. 92–104.

[70] W. Li *et al.*, "Privacy-preserving federated brain tumour segmentation," in *Proc. Int. Workshop Mach. Learn. Med. Imag.*, 2019, pp. 133–141.

[71] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, and Q. Yang, "HHHFL: Hierarchical heterogeneous horizontal federated learning for electroencephalography," 2019. [Online]. Available: arXiv:1909.05784.

[72] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. Sadeghi, "DIoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 756–767.

[73] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system," 2019. [Online]. Available: arXiv:1906.10893.

[74] 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.*, vol. 33, no. 5, pp. 156–165, Sep. 2019.

[75] O. Habachi, M.-A. Adjif, and J.-P. Cances, "Fast uplink grant for NOMA: A federated learning based approach," 2019. [Online]. Available: arXiv:1904.07975.

[76] H. H. Zhuo, W. Feng, Q. Xu, Q. Yang, and Y. Lin, "Federated reinforcement learning," 2019. [Online]. Available: arXiv:1901.08277.

[77] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[78] M. A.-U. Din *et al.*, "Federated collaborative filtering for privacy-preserving personalized recommendation system," 2019. [Online]. Available: arXiv:1901.09888.

[79] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Appl. Sci.*, vol. 8, no. 12, p. 2663, 2018.

[80] H. Yoo, S. Yao, L. Sun, and X. Du, "Using machine learning to address customer privacy concerns: An application with click-stream data," in *Proc. SSRN*, 2019, Art. no. 3314787.

[81] S. Lu, Y. Yao, and W. Shi, "Collaborative learning on the edges: A case study on connected vehicles," in *Proc. 2nd {USENIX} Workshop Hot Topics Edge Comput. (HotEdge)*, 2019, pp. 1–9.

[82] R. Al-Rfou, M. Pickett, J. Snaider, Y.-H. Sung, B. Strope, and R. Kurzweil, "Conversational contextual cues: The case of personalization and history for response ranking," 2016. [Online]. Available: arXiv:1606.00372.

[83] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The EICU collaborative research database, a freely available multi-center database for critical care research," *Sci. Data*, vol. 5, Sep. 2018, Art. no. 180178.

[84] A. E. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, May 2016, Art. no. 160035.

[85] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assisted Living*, 2012, pp. 216–223.

[86] (2017). *ADNI—Alzheimer's Disease Neuroimaging Initiative*. Accessed: Apr. 2, 2020. [Online]. Available: http://adni.loni.usc.edu/

[87] *PPMI—Parkinson's Progression Markers Initiative*. Accessed: Apr. 2, 2020. [Online]. Available: https://www.ppmi-info.org/data

[88] B. H. Menze *et al.*, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Trans. Med. Imag.*, vol. 34, no. 10, pp. 1993–2024, Dec. 2014.

[89] *Mindbigdata*. Accessed: Apr. 8, 2020. [Online]. Available: http://www.mindbigdata.com/opendb/index.html

[90] *Spambase Data Set*. Accessed: Apr. 7, 2020. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase

[91] S. R. Branavan, H. Chen, L. S. Zettlemoyer, and R. Barzilay, "Reinforcement learning for mapping instructions to actions," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Nat. Lang. Process. AFNLP*, vol. 1, 2009, pp. 82–90.

[92] *Wikihow—Home and Garden*. Accessed: Apr. 2, 2020. [Online]. Available: https://www.wikihow.com/Category:Home-and-Garden

[93] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy Build.*, vol. 140, pp. 81–97, Apr. 2017.

[94] H. T. Kahraman, S. Sagiroglu, and I. Colak, "The development of intuitive knowledge classifier and the modeling of domain dependent data," *Knowl. Based Syst.*, vol. 37, pp. 283–295, Jan. 2013.

[95] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.

[96] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *Proc. ICISSP*, 2018, pp. 108–116.

[97] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," 2018. [Online]. Available: arXiv:1812.00984.

[98] C. Ma *et al.*, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul./Aug. 2020.

[99] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE SP*, 2018, pp. 691–706.

[100] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2019, pp. 2512–2520.

[101] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017. [Online]. Available: arXiv:1712.07557.

[102] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2017. [Online]. Available: arXiv:1710.06963.

[103] O. Choudhury *et al.*, "Differential privacy-enabled federated learning for sensitive health data," 2019. [Online]. Available: arXiv:1910.02578.

[104] K. Bonawitz *et al.*, "Practical secure aggregation for federated learning on user-held data," 2016. [Online]. Available: arXiv:1611.04482.

[105] C. Ilias and S. Georgios, "Machine learning for all: A more robust federated learning framework," in *Proc. 5th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, vol. 1, 2019, pp. 544–551.

[106] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "SecureBoost: A lossless federated learning framework," 2019. [Online]. Available: arXiv:1901.08755.

[107] S. Hardy *et al.*, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017. [Online]. Available: arXiv:1711.10677.

[108] Y. Liu, T. Chen, and Q. Yang, "Secure federated transfer learning," 2018. [Online]. Available: arXiv:1812.03337.

[109] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2019, pp. 2552–2559.

[110] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 1175–1191.

[111] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Security*, 2019, pp. 1–11.

[112] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "CRONUS: Robust and heterogeneous collaborative learning with black-box knowledge transfer," 2019. [Online]. Available: arXiv:1912.11279.

[113] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Inf. Sci.*, vol. 522, pp. 69–79, Feb. 2020.

[114] H. Li and T. Han, "An end-to-end encrypted neural network for gradient updates transmission in federated learning," in *Proc. IEEE Data Compression Conf. (DCC)*, 2019, pp. 589–589.

[115] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2018. [Online]. Available: arXiv:1808.04866.

[116] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," 2018. [Online]. Available: arXiv:1807.00459.

[117] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," 2018. [Online]. Available: arXiv:1811.12470.

[118] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "On-device federated learning via blockchain and its latency analysis," 2018. [Online]. Available: arXiv:1808.03949.

[119] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6348–6359, Jul. 2020.

[120] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning system," 2019. [Online]. Available: arXiv:1905.09712.

[121] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, and X. Wang, "Resource-efficient and convergence-preserving online participant selection in federated learning," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, Singapore, Dec. 2020, pp. 1205–1221.

[122] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, 2019, pp. 1387–1395.

[123] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," 2019. [Online]. Available: arXiv:1911.00856.

[124] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "Performance optimization of federated learning over wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[125] L. U. Khan, M. Alsenwi, Z. Han, and C. S. Hong, "Self organizing federated learning over wireless networks: A socially aware clustering approach," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2020, pp. 453–458.

[126] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Aug. 2020.

[127] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Update aware device scheduling for federated learning at the wireless edge," 2020. [Online]. Available: arXiv:2001.10402.

[128] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," 2020. [Online]. Available: arXiv:2004.04314.

[129] Z. Chai *et al.*, "TiFL: A tier-based federated learning system," in *Proc. ACM Symp. High Perform. Parallel Distrib. Comput. (HPDC)*, Stockholm, Sweden, Jun. 2020, pp. 125–136.

[130] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling in cellular federated edge learning with importance and channel awareness," 2020. [Online]. Available: arXiv:2004.00490.

[131] H. Huang, K. Lin, S. Guo, P. Zhou, and Z. Zheng, "PROPHET: Proactive candidate-selection for federated learning by predicting the qualities of training and reporting phases," 2020. [Online]. Available: arXiv:2002.00577.

[132] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu. (2020). *HFEL: Joint Edge Association and Resource Allocation for Cost-Efficient Hierarchical Federated Edge Learning*. [Online]. Available: https://arxiv.org/abs/2002.11343.

[133] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2020, pp. 8866–8870.

[134] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, "Machine learning for wireless connectivity and security of cellular-connected UAVs," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 28–35, Feb. 2019.

[135] U. M. Aïvodji, S. Gambs, and A. Martin, "IOTFLA: A secured and privacy-preserving smart home architecture implementing federated learning," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2019, pp. 175–180.

[136] T. Dbouk, A. Mourad, H. Otrok, H. Tout, and C. Talhi, "A novel ad-hoc mobile edge cloud offering security services through intelligent resource-aware offloading," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1665–1680, Sep. 2019.

[137] H. Sami and A. Mourad, "Dynamic on-demand fog formation offering on-the-fly IoT service deployment," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 1026–1039, Jun. 2020.

Sawsan AbdulRahman received the M.S. degree in computer science from the Lebanese American University, Beirut, Lebanon, in 2017. She is currently pursuing the Ph.D. degree with the École de Technologie Supérieure, Montreal, QC, Canada.

She is a Researcher with Ericsson, Montreal. Her research interests include AI, machine learning, and security.

Mrs. AbdulRahman is a reviewer in several prestigious conferences and journals.

Hanine Tout (Member, IEEE) received the Ph.D. degree in software engineering from the École de Technologie Supérieure (ÉTS), Montreal, QC, Canada, in 2018.

She is currently a Postdoctoral Fellow between ÉTS and Ericsson, Mississauga, ON, Canada, where she is leading two industrial projects in the areas of AI, federated learning, machine learning, security, 5G, and cloud-native IMS.

Dr. Tout is a TPC member and reviewer of prestigious conferences and journals.

Hakima Ould-Slimane received the Ph.D. degree in computer science from Laval University, Quebec, QC, Canada, in 2011.

She is currently a Researcher and a Lecturer with the École de Technologie Supérieure, Montreal, QC, Canada. Her research interests include mainly information security, cryptography, preserving data privacy in smart environments, reliability of collaborative computing, and formal methods.

Azzam Mourad (Senior Member, IEEE) received the M.Sc. degree in CS from Laval University, Quebec, QC, Canada, in 2003, and the Ph.D. degree in ECE from Concordia University, Montreal, in 2008.

He is currently an Associate Professor of computer science with the Lebanese American University and an Affiliate Associate Professor with the Software Engineering and IT Department, Ècole de Technologie Superieure, Montreal. He published more than 100 papers in international journal and conferences on security, network and service optimization and management targeting IoT, cloud/fog/edge computing, vehicular and mobile networks, and federated learning.

Dr. Mourad has served/serves as an Associate Editor for IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE NETWORK, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, *IET Quantum Communication*, and IEEE COMMUNICATIONS LETTERS, the General Chair of IWCMC2020, the General Co-Chair of WiMob2016, and the Track Chair, a TPC member, and a reviewer for several prestigious journals and conferences.

Chamseddine Talhi received the Ph.D. degree in computer science from Laval University, Quebec City, QC, Canada, in 2007.

He is an Associate Professor with the Department of Software Engineering and IT, ÉTS, University of Quebec, Montreal, QC, Canada. He is leading a research group that investigates smartphone, embedded systems, and IoT security. His research interests include cloud security and secure sharing of embedded systems.

Mohsen Guizani (Fellow, IEEE) received the B.S. (with Distinction) and M.S. degrees in electrical engineering, and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively.

He is currently a Professor with the Computer Science and Engineering Department, Qatar University, Doha, Qatar. Previously, he served in different academic and administrative positions with the University of Idaho, Moscow, ID, USA; Western Michigan University, Kalamazoo, MI, USA; University of West Florida, Pensacola, FL, USA; University of Missouri–Kansas City, Kansas, MO, USA; University of Colorado–Boulder, Boulder, CO, USA; and Syracuse University, Syracuse, NY, USA. He has authored nine books and more than 600 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid.

Prof. Guizani received three teaching awards and four research awards. He is a recipient of the 2017 IEEE Communications Society Wireless Technical Committee Recognition Award, the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks, and the 2019 IEEE Communications and Information Security Technical Recognition Award for outstanding contributions to the technological advancement of security. He is currently the Editor-in-Chief of the *IEEE Network Magazine*, serves on the editorial boards of several international technical journals, and the Founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* (Wiley). He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, the chair, and the general chair of a number of international conferences. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Senior Member of ACM.