Received September 8, 2021, accepted September 22, 2021, date of publication November 10, 2021,
date of current version November 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3127448

The Internet of Federated Things (IoFT)

RAED KONTAR[1], NAICHEN SHI[1], XUBO YUE[1], SEOKHYUN CHUNG[1], EUNSHIN BYON[1], (Member, IEEE), MOSHARAF CHOWDHURY[2], (Member, IEEE), JIONGHUA JIN[1], (Member, IEEE), WISSAM KONTAR[3], NEDA MASOUD[4], MAHER NOUIEHED[5], CHINEDUM E. OKWUDIRE[6], GARVESH RASKUTTI[7], ROMESH SAIGAL[1], KARANDEEP SINGH[8], AND ZHI-SHENG YE[9], (Senior Member, IEEE)

[1]Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48105, USA
[2]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105, USA
[3]Department of Civil and Environmental Engineering, University of Wisconsin–Madison, Madison, WI 53715, USA
[4]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48105, USA
[5]Department of Industrial Engineering, American University of Beirut, Beirut 1107 2020, Lebanon
[6]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48105, USA
[7]Department of Statistics, University of Wisconsin–Madison, Madison, WI 53715, USA
[8]Department of Learning Health Sciences, University of Michigan, Ann Arbor, MI 48105, USA
[9]Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 119077

Corresponding author: Raed Kontar (alkontar@umich.edu)

This work was supported in part by the National Science Foundation under Grant CPS1931950.

ABSTRACT The Internet of Things (IoT) is on the verge of a major paradigm shift. In the IoT system of the future, IoFT, the ''cloud'' will be substituted by the ''crowd'' where model training is brought to the edge, allowing IoT devices to collaboratively extract knowledge and build smart analytics/models while keeping their personal data stored locally. This paradigm shift was set into motion by the tremendous increase in computational power on IoT devices and the recent advances in decentralized and privacy-preserving model training, coined as federated learning (FL). This article provides a vision for IoFT and a systematic overview of current efforts towards realizing this vision. Specifically, we first introduce the defining characteristics of IoFT and discuss FL data-driven approaches, opportunities, and challenges that allow decentralized inference within three dimensions: (i) a global model that maximizes utility across all IoT devices, (ii) a personalized model that borrows strengths across all devices yet retains its own model, (iii) a meta-learning model that quickly adapts to new devices or learning tasks. We end by describing the vision and challenges of IoFT in reshaping different industries through the lens of domain experts. Those industries include manufacturing, transportation, energy, healthcare, quality & reliability, business, and computing.

INDEX TERMS Internet of Things, federated learning, global model, personalized model, meta-learning, future applications.

I. INTRODUCTION

A. PREAMBLE

At the early stages of the COVID-19 pandemic, companies that mass-produce personal protective equipment (PPE) required long ramp-up times to fulfill the urgent demand [84], [87]. The ramp-up time took longer than expected as supply chains across the globe were critically disrupted, with entire countries in lockdown and essential workers succumbing to the virus [56]. Realizing this, many citizens and small businesses tried to bridge the supply gap using readily available, and low-cost 3D printers [57], [63]. This attempt at so-called massively distributed manufacturing [233] helped fill PPE production gaps to some extent [57], [63]. However, it also revealed critical impediments to realizing massively distributed manufacturing in terms of standardizing production requirements, guaranteeing quality and reliability, and attaining high production efficiencies that can rival those of mass production [233]. For example, a large percentage of parts printed by citizens did not meet the quality requirements [123], [293]. Even when following standard 3D printing guidelines, several prints failed [292] while others experienced recurrent defects due to the use of models or methods that did not account for the specific environment in which the 3D printer is operating [274]. On the other hand, citizens that succeeded struggled to effectively broadcast their

The associate editor coordinating the review of this manuscript and approving it for publication was Sathish Kumar.

improved models or methods to other users to help improve quality across the network of manufacturers [71].

Now imagine an alternative future based on a cyber-physical operating system for massively distributed manufacturing. All 3D printers are IoT-enabled through wifi and smart sensors. In addition, printers now have computation power through AI chips (many 3D printers nowadays have such capabilities, ex: Raspberry Pi's [18], [234]). The printers collaboratively learn a model for 3D printing PPE accurately with the help of a central orchestrator, guiding the production to the desired quality level. To preserve privacy and intellectual property and allow for massive parallelization, raw data from each 3D printer is never shared with the central server; instead, printers exploit their compute resources at the edge by running small local computations and only sharing the minimal information needed to learn the model. This model, despite having a global state, is personalized to form a local model that accounts for individual-level external factors affecting each 3D printer.

In this alternative reality, responders can 3D print PPE at the desired quality level with little or no defects. Responders act quickly due to the massively parallelized efforts from many 3D printers and the effective utilization of network bandwidth. In addition, with their personalized 3D printing models, the responders are able to push 3D printers at faster speeds to shorten printing time while maintaining quality [76], [215], [234]. Accordingly, the PPE supply gap is successfully filled until mass production ramps up.

In this future, not only manufacturing benefits. Take healthcare wearable devices as an example. Compute power on such devices has been immensely increasing over the years. Now, personal data need not be uploaded to a central cloud system to learn an anomaly detection model for health signals. Instead, the ''cloud'' is replaced by the ''crowd'', where wearable devices store necessary data, perform local computations and send only needed model updates to the central authority. This decouples the ability to learn the model from storing data in the cloud by bringing training to the device as well, where a model can be learned across thousands of millions of wearable devices in geographically dispersed locations.

Let us now switch paradigms and replace smart devices with ''smart'' institutes. Different medical institutions can join efforts and collaboratively learn diagnostic models without directly sharing their electronic health records, as imposed by the Health Insurance Portability and Accountability Act (HIPAA). Now, diagnostic models can leverage largely diverse datasets and promote fairness through a decentralized learning framework that mitigates privacy risks and costs associated with centralized modeling. Learning can be done across institutes and individuals at multiple scales and in areas that this has not been possible or allowed before.

The future described above is not a far cry away. It has already been set into action as the immediate yet bold next step for the Internet of Things (IoT). It is the cultivation of Industry 4.0. A cultivation of advances in interdisciplinary fields in the past two decades: ranging from data science, edge computing, machine learning, operations research, optimization, data acquisition technologies, physics-guided modeling, and privacy, amongst many others.

In this article, we term this future of IoT as the **Internet of Federated Things (IoFT)**. The term ''federated'' refers to some level of internal autonomy of IoT devices and is inspired by the explosive interest during the past two years in **Federated Learning (FL)**: an approach that allows decentralized and privacy-preserving training of models [208]. With the help of FL, the decentralized paradigm in IoFT exploits edge compute resources in order to enable devices to collaboratively extract knowledge and build smart analytics/models while keeping their personal data stored locally. This paradigm shift not only reduces privacy concerns but also sets forth many intrinsic advantages including cost efficiency, diversity, and reduced computation, amongst many others to be detailed in the following sections.

B. PURPOSE AND UNIQUENESS

This paper is a joint effort of researchers across a wide variety of expertise to address the three questions below:

1) What are the defining characteristics of IoFT?
2) What are key recent advances and potential data-driven methods in IoFT that allow learning in one of the three dimensions stated below? what modeling, optimization, and statistical challenges do they face? and what are potential promising solutions?
   - **A Global model**: that maximizes utility across all devices. The global model aims at capturing the commonalities and intrinsic relatedness across data from all devices to improve prediction and learning accuracy.
   - **A Personalized model**: that tries to personalize and adapt the global model to data and external conditions from each device. This embodies the principle of multi-task learning, [238] where each device retains its own model while borrowing strength across all IoFT devices.
   - **A Meta-learning model**: that learns a global model which can quickly adapt to a new task with only a small amount of training samples and learning steps. This embodies the principle of ''learning to learn fast,'' [289] where the goal of the global model is not to perform well on all tasks in expectation, instead to find a good initialization that can directly adapt to a specific task.
3) How will IoFT shape different industries and what are the domain specific challenges it faces for it to become the standard practice? Through the lens of domain experts, we shed light on the following sectors: **manufacturing**, **transportation**, **energy**, **healthcare**, **quality & reliability**, **business** and **computing**.

Besides defining the central characteristics of IoFT, our paper's focus is summarized in two folds. The first is **data-driven modeling** where we categorize FL approaches in

IoFT into learning a global, personalized, and meta-learning model and then provide an in-depth analysis on modeling techniques, recent advances, possible alternative, and statistical/optimization challenges. The second focus is a **vision of IoFT**'s potential use cases, application-specific models, and obstacles within different application domains. Our overarching goal is to encourage researchers across different industries to explore the transformation from IoT to IoFT so that critical societal impacts brought by this emerging technology can be fully realized.

We note here that some excellent surveys on FL have been recently released. Most notably, Lim *et al.* [179] address FL challenges in mobile edge networks with a focus on communication cost, privacy and security, Niknam *et al.* [227] discuss FL application in wireless communications, especially under 5G networks, Li *et al.* [173] provide a thorough overview of implementation challenges in FL, Yang *et al.* [329] then categorize different architectures for FL, Rahman *et al.* [256] discuss the evolution of the deployment architectures with an in-depth discussion on privacy and security, while Aledhari *et al.* [5] highlight necessary protocols and platforms needed for such architectures, Kairouz *et al.* [133] study open problems in FL and recent initiatives while providing a remarkable survey on privacy-preserving mechanisms. Along this line, Lyu *et al.* [193] highlight threats and major attacks in FL. While our focus is on **data-driven modeling** for IoFT and how various application fields will be affected by the shift from IoT to IoFT, the surveys above serve as excellent complementary work for a bird's eye view of FL and hence IoFT.

The remainder of this paper is organized as follows. Sec. II highlights the past and present features of IoT-enabled systems leading to IoFT. Secs. III - V provide data-driven modeling approaches for learning a global, personalized, and meta-learning model, along with their challenges and promising solutions. Finally, Sec. VI poses central statistical and optimization open problems in IoFT. These open problems are from both a theoretical and applied perspective. Finally, Sec. VII provides a vision for IoFT within manufacturing, transportation, energy, healthcare, quality & reliability, business, and computing.

Throughout this paper, we use IoFT to denote the future IoT system we envision, while FL denotes the underlying data analytics approach for data-driven model learning within IoFT. Also, edge device, local device, node, user, or client are used interchangeably to denote the end-user based on the problem context.

C. IoFT WEBSITE AND CENTRAL DIRECTORY

While exploring data-driven modeling approaches to FL in IoFT, it became clear that real-life datasets (in engineering, health sciences, etc..) are pressingly needed to fully explore the disruptive potential of IoFT. While few already exist, they are based on artificial examples, and the few non-artificial datasets are mostly focused on mobile applications. However, for IoFT to become a norm in different industries, real-life datasets with defining features of the underlying system are needed to unveil the potential challenges and opportunities faced within different domains. Only with a deep understanding of the underlying system and domain, one formulates the right analytics. Towards this end, this paper features a supplementary website (`https://ioft-data.engin.umich.edu/`) managed by the University of Michigan. The website will serve as a central directory for IoFT-based datasets and will feature brief descriptions of each dataset categorized by its respective field with a link to the repository (research lab website, GitHub account, papers, etc..) where the data is contained. Our hope is to provide a means for model validation within different domains, encourage researchers to develop real-life datasets for IoFT and help with the outreach and visibility of their datasets and corresponding papers.

Website: https://ioft-data.engin.umich.edu/

II. INTERNET OF THINGS: THE PAST, PRESENT AND FUTURE

IoT-enabled systems possess three defining characteristics: tangible physical components that comprise the system, connectivity among components that enable data acquisition and sharing, and data analytics and decision-making capabilities that transform a merely "connected" system into a "smart and connected" system. These defining features of IoT enabled systems [40], [207], [252] are shown in Fig. 1. IoT has brought broad disruptive societal impacts, particularly on economic competitiveness, quality of life, public health, and essential infrastructure [195]. Companies around the globe have invested heavily in IoT, including: Google's Cloud IoT [100], Samsung's Active wearable device [276], Amazon's Webservices solutions [11], Rockwell's Connected Enterprise [269], Welbilt's Smart Home Appliances, to name a few. The value at stake is more than 15 trillion dollars, a number expected to triple in the next decade [9].

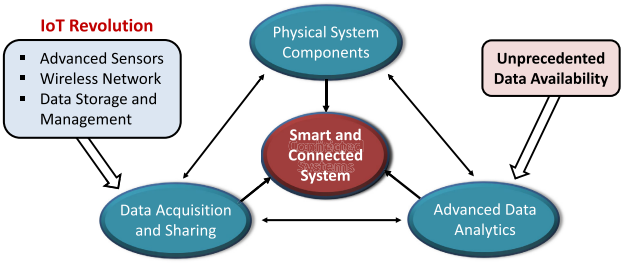

FIGURE 1. Key components of an IoT enabled system.

The essential feature of an IoT system is that data from multiple similar units and across multiple components within the system are collected during their operation, often in real-time. Since we have observations from potentially a large number of similar units, we can compare their operations,

share information, and extract common knowledge to enable accurate prediction and control. One can argue that such a notion of IoT dates back a long time before the Industrial Revolution, to the time when artisans producing crafts in geographically close locations used to gather to share knowledge and perfect/standardize the quality of their crafted product [291]. *A lot has changed since then*.

A. IoT: THE PRESENT

Starting with the industrial revolution came rapid advances in connectivity, automation, data science, cloud-based systems, among many others [164], [212]. An IoT sensor price dropped to $0.48 on average, and wide-area communication became readily available with around $36.13 billion connected IoT devices in 2018 [165]. Distributed computing allowed handling larger datasets than what was previously thought possible and cloud-based solutions for data storage and processing have become widely available for commercial use (ex: Amazon's AWS [10] or Microsoft's Azure [12]). This ushered in the present-day era of Industry 4.0 characterized by IoT-enabled systems [9]. In this present era, a typical IoT-enabled system structure is shown in Fig. 2. Take for example GM's OnStar® or Ford's SYNC® teleservice systems [1], [90], [235]. Vehicles enrolled for this service have their data in the form of condition monitoring (CM) signals uploaded to the cloud regularly. The cloud then acts as a back-office or data center that processes the data to keep drivers informed about the health of their vehicle. In the cloud, GM and Ford train models that can monitor and predict maintenance needs, amongst others. The data is also used to cross-validate the behavior of their learned models for continuous improvement. When the need arrives, service alerts are then sent to drivers.

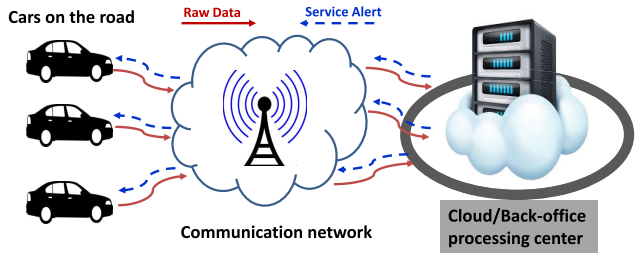

FIGURE 2. Present day IoT system.

Much like other IoT giants such as Google, Amazon and Facebook, GM and Ford have long adopted this centralized approach towards IoT: (i) gigantic amounts of data are uploaded and stored in the cloud (ii) models (such as predictive maintenance, diagnostics, text prediction) are trained in these data centers (iii) the models are then deployed to the edge devices. Needless to say, the need to upload large amounts of data to the cloud raises privacy concerns, incurs high costs, and benefits large enterprises capable of building their own private cloud infrastructures at the expense of smaller entities.

Here, distributed learning is often implemented in centralized systems to alleviate the huge computational burden via parallelization. In such systems, the clients are computing nodes within this centralized framework. Nodes can then access any part of the dataset, as data partitions can be continuously adjusted. In contrast, as described in the following sections, in IoFT, the data resides at the edge and is not centrally stored. As a result, data partitions are fixed and cannot be changed, shuffled, nor randomized.

B. IoT: THE FUTURE

With the tremendous increase in computational power on edge devices, IoT is on its way to move from the cloud/datacenter to the edge device, hence the aforementioned notion of substituting the "cloud" by the "crowd". In this IoT system of the future (IoFT), devices collaboratively extract knowledge from each other and achieve the "smart" component of IoT, often with the orchestration of a central server, while keeping their personal data stored locally. This paradigm shift is based on one simple yet powerful idea: with the availability of computing resources at the edge, clients can execute small computations locally instead of learning models on the cloud and then only share the minimum information needed to learn that model. As a result, IoFT decouples the ability to do analytics from storing data in the cloud by bringing training to the edge device as well. The underlying premise is that IoFT devices have computational (ex: AI chips) and communication (ex: wifi) capabilities.

Let us start with a simple example, assume the central orchestrator in Fig. 3 wants to learn the mean ($\bar{y}$) of a single feature (y) over all clients. Now assume that clients have some computational capabilities. To calculate $\bar{y}$, client i only needs to run a small calculation to compute their own mean ($\bar{y}_i$) and share it, rather than sharing their entire feature vector ($\mathbf{y}_i$). $\bar{y}_i$ is a sufficient statistic to learn $\bar{y}$.

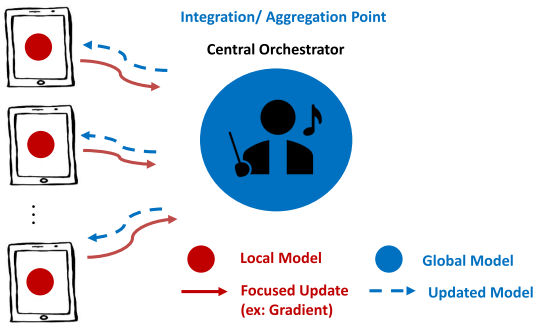

FIGURE 3. IoFT: IoT system of the Future.

In reality, models are often more complicated and require multiple communications between the central orchestrator and clients. For instance, and without loss of generality, assume that IoFT devices cooperate to learn a deep learning model through borrowing strength from each other, rather than using their own knowledge in isolation. In the decentralized realm of IoFT, model learning is often administered

by a central orchestrator and follows the cycle shown in Fig. 3. (i) The orchestrator (i.e., the central server) selects a set of IoFT devices meeting certain eligibility requirements and broadcasts an initial model to the selected clients. This model contains the neural network (NN) architecture, initial weights, and a training program. (ii) IoFT devices perform local computations by executing the program on their local data, and each device reports its focused update to the orchestrator. Here the program can be running stochastic gradient descent (SGD) on local data, and the focused update can be updated weights or a gradient. It is worth noting that the client might choose to encrypt their focused update or add noise to it for enhanced privacy at this stage. (iii) The central orchestrator collects the focused updates from clients and aggregates them to update the global model. (iv) This procedure is then iterated over several rounds until a stopping criterion, such as validation accuracy, is met. Through this process, the global model can account for knowledge from all IoFT clients, and each client can indirectly make use of the knowledge from other clients. Finally, the learned global model goes through a testing phase such as quality-A/B testing on held-out devices and a staged rollout on a gradually increasing number of devices.

This decentralized paradigm shift, made possible by compute resources at the edge, sets forth many intrinsic advantages that include:

- ***Privacy***: By bringing training to the edge device, users no longer have to share their valuable information, instead, it is kept local and never shared.
- ***Autonomy***: IoFT devices can be under independent control and opt-out of the collaborative training process at any time. Yet, with enhanced privacy in IoFT, clients will be more inclined to collaborate and build better models.
- ***Computation***: As the number of IoT devices skyrockets, computational and storage needs accumulated from these devices (say smartphones) is far beyond what any data center or cloud computing system can handle [350]. Instead, by exploiting compute and storage capacity at the edge, massive parallelization becomes a reality [121], [286].
- ***Cost***: Focused updates embody the principle of data minimization and contain the minimum information needed for a specific learning task. As a result, less information is transmitted to the orchestrator, which reduces communication costs and efficiently utilizes network bandwidth. Also, compute power at the edge device is now utilized. Hence storage and computational needs of the orchestrator are minimal. This is in contrast to distributed systems where massive utilization and synchronization of GPU and CPU power in the cloud is needed.
- ***Fast Alerts and Decisions***: In IoFT, upon deployment of the final model to clients, real-time decisions or service alerts are achieved locally at the edge. In contrast, cloud-based systems incur a lag in deployment, as decisions made in the cloud need to be transmitted to the clients (as shown in Fig. 2).
- ***Minimal Infrastructure***: With the increase in computing power of IoT devices and the gradual market penetration of AI chips [320], minimal hardware is required to achieve the transition to IoFT.
- **Fast encryption:** Encryption of focused updates can be done readily and with better guarantees compared to encrypting entire datasets.
- **Resilience**: Edge devices are resilient to failures at the orchestrator level due to the existence of a local model.
- ***Diversity and Fairness***: IoFT allows integrating information across uniquely diverse datasets, some of which have been restricted to be shared previously (recall medical institutes example). This diversity and ability to learn across geographically disperse locations promotes fairness by combining data across boundaries [29], [38].

Having recently realized its disruptive potential to traditional IoT, industries are eagerly trying to exploit IoFT in their operating systems and production. However, these efforts are in their infancy phase, awaiting broad implementations. Google pioneered some of the IoFT applications in their mobile keyboard ''Gboard'' [43], [106], [257], [331] and Android messaging [99] to improve next-word predictions and preserve privacy. Additionally, they introduced a decentralized framework to update android models on their Pixel phones [208]. In this framework, each android phone updates its model parameters locally and sends out the updated parameters to the Android cloud, which trains its central model from the aggregated parameters. BigTech giants have since started to catch up and utilize FL in their systems. Most notably Apple adopted FL in their QuickType keyboard, ''Siri'' and privacy protection protocols [7], [23]. As well as Microsoft in their device's telemetry data [68]. Further, FL has seen some application in optimizing mobile edge computing and communication [179], [307], computational offloading [307] and reliable network communication [275].

Most of the current IoFT applications are present within the technology industry and specifically tailored for mobile applications and few others. However, IoFT is expected to infiltrate all industries that benefit from knowledge sharing, data analytics, and decision-making. Indeed, the gradual use of FL in the technology industry has set in motion a timid yet insuppressible momentum for IoFT application in other sectors. For instance, in the healthcare field, FL is lately being used as a medium of collaboration between hospitals to share patients' electronic records and other medical data [29], [117], [142], [302]. In Sec. VII, we will present a deeper vision into how IoFT and FL will shape the future of various industries; those include manufacturing, transportation, energy, healthcare, quality & reliability, business, and computing.

1) CHALLENGES

IoFT as an emerging technology poses significant intellectual challenges. Interdisciplinary skills across diverse fields are

needed to bring the great promise of IoFT into reality. Below we highlight some of the challenges and shed light on their uniqueness compared to centralized IoT systems. This is by no means an exhaustive list as IoFT challenges vary widely across different application sectors as highlighted in Sec. VII.

- ***Statistical Heterogeneity***: IoFT devices often have local datasets that differ in both size and distribution. Recent papers have shown the unfortunate wide gap in the global model's performance across different devices due to their heterogeneity in distribution [306], [355] and size [77]. For instance, IoFT devices may have (i) unique outputs, labels, or features only observed within certain IoFT devices. (ii) Similar outputs but with dissimilar features (i.e., feature distribution skew) or vice versa. This statistical heterogeneity directly consequences IoFT's ability to reach out to many devices operating under different external factors and subject to geographic, cultural, and socio-economic differences. In contrast, traditional IoT systems offer a key, yet often subtle fundamental advantage: the ability to handle nonindependent or identically distributed (*i.i.d*) data by shuffling/randomizing the raw data collected in the cloud before learning; be it through distributed computing or learning on a single machine. This is not a luxury that IoFT possesses; rather, it is a price to pay for enhanced privacy.
- ***Personalization and Negative Transfer***: In the IoFT process described in Sec. II-B all clients collaborate to learn a global model; ''one model that fits all''. This integrative analysis of multiple clients implicitly assumes that these local datasets share some commonalities. However, with heterogeneity, negative transfer of knowledge may occur, which leads to decreased performance relative to learning tasks separately [151], [169]. One possible solution is through personalized modeling where global models are adapted for local clients (refer to Sec. IV for data-driven personalization approaches). Indeed, personalization may be the fundamental tool to overcome the heterogeneity barrier intrinsic to IoFT. Yet developing validation techniques to identify negative transfer and minimize it is a critical problem in FL.
- ***Communication Efficiency and Resource Management***: Communication can be a critical bottleneck for IoFT, especially with a large number of participants. Unlike cloud datacenters, edge devices in IoFT often have limited communication bandwidth with unstable and slow connection [150]. As a result, IoFT devices are often unreliable and can drop out due to battery loss or connectivity loss. Besides that, devices themselves are heterogeneous in their computational capabilities and memory budgets. Therefore, resource management in IoFT is of critical importance. Methods such as compressed communication [147], [297], client selection [326] and optimal trade-offs between convergence rates, accuracy, energy consumption, latency and communications [224], [267] are of high future relevance. Another possible approach is through incentive design to encourage reliable clients to participate in the training process and minimize dropout rates [134].
- ***Privacy***: Privacy remains one of the key challenges and motivators behind IoFT. IoFT systems are prone to poisoning attacks on both edge devices and the central server. Targeted data perturbations [14], [52], [187] to specific labels/instances or corrupting a large number of devices (i.e., fake devices) can immensely reduce accuracy. Further, a malicious server might be able to reconstruct raw data even through a focused update. As a result, secure computation, aggregation, and communication are needed in IoFT [20], [26]. So is adversarial data modeling to ensure robustness against corrupted data in case breaches are inevitable [198].
- ***Bias and Fairness***: IoFT systems can raise bias and fairness concerns. For example, sampling reliable phones with a larger bandwidth (i.e., more expensive phones) can lead to models mostly representative of people with certain socioeconomic statuses. Further, it is often important to build models that are competitive over different groups or attributes. This becomes a bigger challenge if such sensitive attributes are not shared. Therefore, fair FL is an important challenge to tackle within IoFT [172], [342]
- ***Other Statistical and Optimization Challenges***: We also refer readers to Sec. VI for both statistical and optimization challenges/opportunities and Sec. VII for domain-specific challenges in different sectors.

We here note that Secs. III, IV, V shed light on data-driven modeling approaches (global, personalized and meta-learning) aimed to tackle of the challenges above. However, we exclude (i) privacy and communication efficiency: since there are excellent surveys focused mainly on these challenges (refer to Sec. I-B) (ii) resource management: since literature in that area is still scarce.

2) IoFT STRUCTURES

The underlying structure and overall architecture of IoFT should be tailored to fit certain applications and overcome specific challenges. Current IoFT architectures are influenced by the data composition and the FL learning process. For instance, in the situation where multiple clients collaborate to learn a global model with the orchestration of a central server (as seen in Fig. 3), it is implicitly assumed that local datasets share a common feature space but have a different sample space - i.e. different clients. Such data composition is technically referred to as Horizontally partitioned data [329]. A typical FL system architecture for Horizontally portioned data (also known as Horizontal FL (HFL)), would exploit the availability of a common feature space. Notably, horizontally partitioned data are very common across different applications, making HFL the common practice in IoFT [43], [106], [257], [329], [331].

However, not all datasets share a common feature space which naturally poses the need for a different architecture.

Vertically partitioned data, which refers to datasets sharing a different feature space but similar sample space, is another familiar theme in various applications. Such datasets mostly appear in scenarios that involve joint collaboration between large enterprises. Consider as an example two different health institutes, each owning different health records yet sharing the same patients. Suppose you wish to build a predictive model for a patient's health using a complete portfolio of medical records from both healthcare institutes. Unlike HFL where each client trains a local model using their own data, training a local model requires data owned by other clients since each client holds a disjoint subset of the data. Accordingly, a typical FL system architecture for Vertically partitioned data (also known as Vertical FL (VFL) [329]) is designed to introduce secure communication channels between clients to share the needed training data, while preserving privacy and preventing data leakage from one provider to another. For this, VFL architecture may involve a trusted, neutral party to orchestrate the federation. The orchestrator aligns and aggregates data from participants to allow for collaborative model building using the joint data, see Fig. 4. Nonetheless, VFL remains less explored than HFL, and most of the currently developed structures can only handle two participants [108], [230], [330]. More challenging scenarios can occur when clients have datasets that share only partial overlap in the feature and sample spaces. FL in these cases can leverage transfer learning techniques to allow for collaborative model training [239], [329].

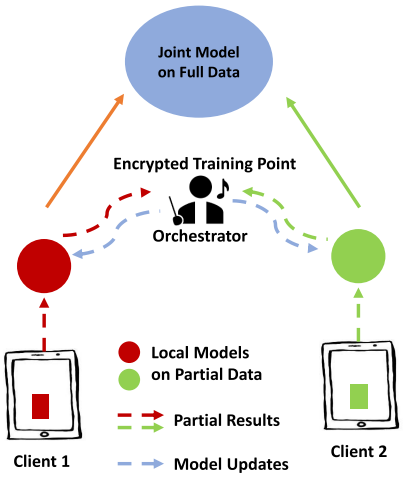

FIGURE 4. IoFT with vertically partitioned data.

The structures described above are designed to handle challenges arising from dataset partitioning. However, different challenges require new structures. One notable commonality of the above structures is the usage of a central orchestrator that coordinates the FL process in IoFT. The caveat, however, is that a central orchestrator is a single point of failure and can lead to a communication bottleneck with a large number of clients [177]. Accordingly, fully decentralized solutions can be explored to nullify the dependency on a central orchestrator. In fully decentralized architectures, communication with the central server is replaced by peer-to-peer communication, as seen in Fig. 5. In this setting, no central location receives model updates/data or maintains a global model over all clients, however, clients are set to communicate with each other to reach desired solutions. Notably, such peer-to-peer networks are better able to achieve scalability in situations with a large number of clients, thanks to their fully decentralized mechanism [140]; the current success of blockchains is a clear demonstration of this. Further, they offer additional security guarantees as it is difficult to observe the system's full state [21]. However, such architecture yields performance concerns. Some clients could be malicious in peer-to-peer networks and potentially corrupt the network (e.g., violate data privacy). Others could be unreliable and thus disrupt the communication channels. Consequently, a level of trust in a central authority in a peer-to-peer architecture can be of benefit in regulating the network's protocols.

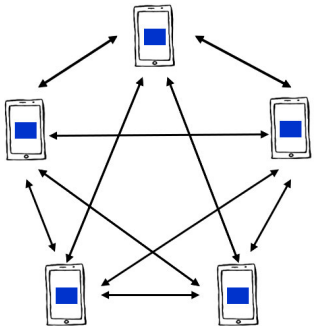

FIGURE 5. Peer-to-peer network.

The structures discussed here are by no means comprehensive, and several others exist in the literature (see [133], [173], [256], [329]). However, the common denominator here is that IoFT structures spawn from challenges of FL applicability to different scenarios. As IoFT is poised to infiltrate more and more fields, domain-specific challenges will dictate its architecture.

III. LEARNING A GLOBAL MODEL

Hereon, we discuss data-driven approaches for FL within IoFT. As aforementioned, we classify model building in FL into three categories: (i) a global model, (ii) a personalized model (iii) a meta-learning model. We then provide an in-depth overview of data-driven models, open challenges, and possible alternatives within these three categories.

As will become clear shortly, the current FL techniques mostly focus on predictive modeling using deep learning and first-order optimization techniques, specifically stochastic gradient descent (SGD). This is understandable as the immense data collected within IoFT often necessitates such an approach. Yet, as we discuss in the statistical/optimization perspective (Sec. VI) and applications (Sec. VII) sections, exploring FL beyond predictive models and deep learning is critical for its wide-scale implementation. Topics such as graphical models, correlated inference, zeroth and second order distributed optimization, validation & hypothesis

testing, uncertainty quantification, design of experiments, Bayesian optimization, optimization under conflicting objectives (see in Sec. VII-E), game theory and reinforcement learning, amongst others, are yet to be explored in the IoFT realm.

A. A GENERAL FRAMEWORK FOR FL

As highlighted in Fig. 3, IoFT allows multiple clients to collaborate and learn a shared model while keeping their personal data stored locally. This shared model is referred to as the global model as it aims to maximize utility across all devices. One can view the global model as: ''one model that fits all'', where the goal is to yield better performance in expectation across all clients relative to each client learning a separate model using its own data.

We start by constructing the objective function of a global model. Assume there are N clients (or local IoFT devices) and each client i has n_i number of observations. The general objective of training a global model is to minimize the average over the objective of all clients:

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) := \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{w}), \tag{1}$$

where $F_i(\boldsymbol{w})$ is usually a risk function on client i. This risk function can be expressed as

$$F_i(\boldsymbol{w}) = \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}_i}\left[\ell(f_{\boldsymbol{w}}(x_i), y_i)\right],$$

where $\mathcal{D}_i$ indicates the data distribution of the i-th client's data observations (x_i, y_i), $f_{\boldsymbol{w}}$ is the model to be learned parametrized by weights $\boldsymbol{w}$, and $\ell(\cdot,\cdot)$ is a loss function.

The risk function is usually approximated by the empirical risk given as $F_i(\boldsymbol{w}) = \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}_i}\left[\ell(f_{\boldsymbol{w}}(x_i), y_i)\right] \approx \frac{1}{n_i}\sum_{j=1}^{n_i}\left[\ell(f_{\boldsymbol{w}}(x_j), y_j)\right]$. Therefore, learning a global model in FL aims at minimizing the average of risks over all clients. However, unlike centralized training, in IoFT client i can only evaluate its own risk function $F_i(\boldsymbol{w})$ and the central server does not have access to the data from the clients. Client and central server training are thus decoupled.

Given this setting, Algorithm 1 is a general ''computation then aggregation'' [352] framework for FL. In each communication round, a central orchestrator selects a subset of clients ($\mathcal{S} \subseteq [N]$) and broadcasts the global model information to the subset. Each client then updates the global model using its own local data. Afterwards, clients send their updated models back to the central orchestrator/server. The orchestrator aggregates and revises the global model based on input from clients. The process repeats for several communication rounds until a stopping criterion, such as validation accuracy, is met. Note that we use $[N]$ to denote the set $\{1, 2, \ldots N\}$, D_i a client's dataset and the superscript ^{t} to represent the t-th communication round between the central server and selected clients, where $t \in \{1, .., T\}$.

One of the simplest FL algorithms is `FedSGD` [206], [359], a distributed version of SGD. `FedSGD` was initially used for distributed computing in a centralized regime. `FedSGD` partitions the data across multiple computing nodes. In every communication round, each node calculates the gradient from its local data using a single SGD step. The calculated weights are then averaged across all nodes. As a data-parallelization approach, `FedSGD` utilizes the computation power of several compute nodes instead of one. This approach accelerates vanilla SGD and has been widely used due to the growing size of datasets collected nowadays. Furthermore, since `FedSGD` only performs one step of SGD on a local node, averaging updated weights is equivalent to averaging gradients (η denotes steps size):

$$\mathbb{E}_i\left[\boldsymbol{w}^t - \eta\nabla F_i(\boldsymbol{w}^t)\right] = \boldsymbol{w}^t - \eta\mathbb{E}_i\left[\nabla F_i(\boldsymbol{w}^t)\right].$$

Despite being a viable option, traditional distributed optimization algorithms are often unsuitable in IoFT due to the large communication cost and the presence of heterogeneity. `FedSGD` transmits the gradient vector from one machine to the other after each single local optimization iterate. This issue is not critical in centralized distributed training when computation nodes are usually connected by large bandwidth infrastructure. However in IoFT, data lives on the edge device and not on a computing node. Communication with the central orchestrator at each gradient calculation is not feasible and may suffer immensely when the edge devices have limited communication bandwidth with unstable or slow connection.

To resolve this challenge, the seminal work of McMahan *et al.* [208] proposed a simple solution: `FedAvg`. The fundamental idea is that clients run multiple updates of model parameters before passing the updated weights to the central orchestrator. Specifically, in `FedAvg`, clients update local models by running multiple steps (e.g., E local steps) of SGD on their local objective $\min_{\boldsymbol{w}_i^t} F_i(\boldsymbol{w}_i^t)$. Upon receiving updated weights from clients, the *server_update* function simply calculates the average of the client models: $\boldsymbol{w}^{t+1} = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\boldsymbol{w}_i^t$. An illustration contrasting `FedAvg` and `FedSGD` is shown in Fig 6. Here one can also add flexibility by re-scaling the global update with a step size η_g, $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \eta_g\left(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}[\boldsymbol{w}_i^t - \boldsymbol{w}^t]\right)$.

Algorithm 1 Framework for Learning a Global Model

1: **Input:** Client datasets $\{D_i\}_{i=1}^N$, T, initialization for $\boldsymbol{w}$
2: **for** $t = 1, 2, \cdots T$ **do**
3: Orchestrator selects a subset of clients $\mathcal{S} \subseteq [N]$, broadcasts global model $\boldsymbol{w}^t$, or a part of it, to clients in $\mathcal{S}$.
4: **for** each $i \in \mathcal{S}$ **do**
5: Clients update model parameters $\boldsymbol{w}_i^{t+1} = client\_update\left(\boldsymbol{w}^t, D_i\right)$
6: Clients send updated parameters $\boldsymbol{w}_i^{t+1}$ to server.
7: **end for**
8: Orchestrator updates $\boldsymbol{w}^t$ by aggregating client updates $\boldsymbol{w}^{t+1} = server\_update\left(\left\{\boldsymbol{w}_i^t\right\}\right)$
9: **end for**

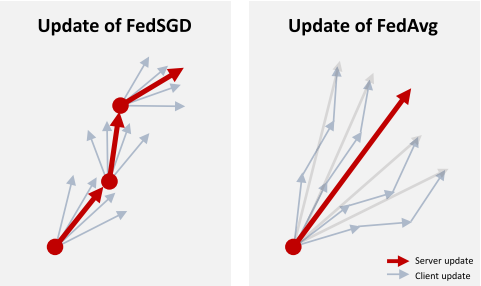

FIGURE 6. An illustration of `FedAvg` and `FedSGD`. Grey arrows represent gradients evaluated on the local client. Bold red arrows represent a global model update on the central server in one communication round. In `FedSGD`, each client performs one step of SGD, and sends the update to the server, while `FedAvg` allows each client to perform multiple SGD steps before averaging.

Indeed, despite its simplicity, `FedAvg` has seen wide empirical success within FL due to its communication efficiency and strong predictive performance on several datasets. To this day, `FedAvg` remains a standard benchmark that is often hard to beat. However, a major observed challenge was that the performance of `FedAvg` and `FedSGD` degrades significantly [208] when data across clients are heterogeneous, i.e. non-*i.i.d.* data. Here one should note that empirical results have shown that `FedAvg` requires fewer communication rounds than `FedSGD` even in the presence of heterogeneity [208].

B. TACKLING HETEROGENEITY

As previously discussed, an intrinsic property of IoFT is that the data distribution across clients is often imbalanced and heterogeneous. Unlike centralized systems, data cannot be randomized or shuffled prior to inference as it resides on the edge. For example, wearable devices collect data on users' health conditions such as heartbeats and blood pressure. Due to the many differences across users, the amount of data collected can significantly vary, and statistical patterns of these data are not alike, often with unique or conflicting trends. This heterogeneity degrades the performance of `FedAvg`. The reason is that minimizing the local empirical risk $F_i(\boldsymbol{w})$ is sometimes fundamentally inconsistent with minimizing the global empirical risk $F(\boldsymbol{w})$ when data are non-*i.i.d.* Mathematically, it also implies that $F(\boldsymbol{w}^*) \neq \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{w}_i^*)$, where superscript * indicates an optimal parameter. This phenomenon is known as client-drift [137]. Notice that if local datasets are *i.i.d.*, when the size of local datasets approaches infinity, $F_i(\boldsymbol{w})$ converges to the global empirical risk $F(\boldsymbol{w})$, hence optimal solutions coincide. In the following, we introduce some works trying to address the heterogeneity challenge.

One method to allay heterogeneity in FL is regularization. In the literature, regularization has been a popular method to reduce model complexity. As less complex models usually generalize better [16], [92], regularization attains better testing accuracy. In FL, regularization places penalties on a set of parameters in the objective function to encourage the model to converge to desired critical points. Researchers in FL have proposed several notable algorithms using regularization techniques to train global models with non-*i.i.d.* data. Perhaps the most basic one is `FedProx` [170] which adds a quadratic regularizer term (a proximal term) to the client objective:

$$\min_{\boldsymbol{w}_i^t} F_i(\boldsymbol{w}_i^t) + \frac{\mu}{2}\left\|\boldsymbol{w}_i^t - \boldsymbol{w}^t\right\|^2.$$

The proximal term $\frac{\mu}{2}\left\|\boldsymbol{w}_i^t - \boldsymbol{w}^t\right\|^2$ in `FedProx` limits the impact of client-drift by penalizing local updates that move too far from the global model in each communication round. Parameter μ controls the degree of penalization. It was also seen that `FedProx` allows each device to have a different number of local iterations E_i, which is especially useful when IoFT devices vary in reliability and communication/computation power. Experimental results show that `FedProx` can partially alleviate heterogeneity, while reducing communication cost due to the often faster convergence and ability of reliable clients to run more updates than others. Here it is important to note that despite reducing client-drift, `FedProx` is still based on in-exact minimization since it does not align local and global stationary solutions.

Besides `FedProx`, [3], [149], [283], [352] also develop a framework to tackle heterogeneity through regularization. Among this literature, `DANE` [283] was proposed for distributed optimization yet is readily amenable to FL settings. `DANE` uses a local objective:

$$\min_{\boldsymbol{w}_i^t} F_i(\boldsymbol{w}_i^t) - \left\langle \nabla F_i(\boldsymbol{w}^{t-1}) - \nabla F(\boldsymbol{w}^{t-1}), \boldsymbol{w}_i^t - \boldsymbol{w}^{t-1}\right\rangle + \frac{\mu}{2}\left\|\boldsymbol{w}_i^t - \boldsymbol{w}^{t-1}\right\|^2, \quad (2)$$

where μ is also a parameter for weighting the regularization and $\boldsymbol{w}^{t-1}$ is the global update at the previous communication round. Compared with the `Fedprox` objective, (2) adds one term that linearly depends on $\boldsymbol{w}_i^t$. This term aligns the gradient of the local risk to that of the global risk. To see it, one can calculate the gradient of (2) as $\nabla F_i(\boldsymbol{w}_i^t) - \left(\nabla F_i(\boldsymbol{w}^{t-1}) - \nabla F(\boldsymbol{w}^{t-1})\right) + \mu\left(\boldsymbol{w}_i^t - \boldsymbol{w}^{t-1}\right)$, where the term $\nabla F_i(\boldsymbol{w}^{t-1}) - \nabla F(\boldsymbol{w}^{t-1})$ approximates the difference between the local and global gradient by its value at the last communication round. It is shown that objective (2) can be interpreted as mirror descent. Interestingly, if the local loss function is quadratic, optimizing (2) can approximate performing Newton updates.

The exact minimization in (2) is sometimes infeasible, as edge devices usually have limited computation resources. To resolve the issue, Stochastic Controlled Averaging algorithm (`SCAFFOLD`) [137] replaces the exact minimization by several gradient descent steps on the local objective below,

$$\min_{\boldsymbol{w}_i^t} F_i(\boldsymbol{w}_i^t) - \left\langle \boldsymbol{c}_i^t - \boldsymbol{c}^t, \boldsymbol{w}_i^t\right\rangle. \quad (3)$$

where control variables $\boldsymbol{c}_i^t$ and $\boldsymbol{c}$ are defined as $\boldsymbol{c}_i^t = \nabla F_i(\boldsymbol{w}_i^{t-1})$, i.e. the local gradient at the end of the last communication round, and $\boldsymbol{c}^t = \frac{1}{N}\sum_i \boldsymbol{c}_i^t$. Objective (3) is akin

to (2), since $\langle \boldsymbol{c}_i^t - \boldsymbol{c}^t, \boldsymbol{w}_i^t \rangle$ also has the alignment effect, except that it does not have the proxy term $\frac{\mu}{2} \left\| \boldsymbol{w}_i^t - \boldsymbol{w}^{t-1} \right\|^2$. To show the update rule in communication round t, we use $\boldsymbol{w}_i^{t,e}$ to denote the weight at the e-th local iterate, and set $\boldsymbol{w}_i^{t,0} = \boldsymbol{w}^t$. In round t, the server samples a group of clients $\mathcal{S}$. For client i in $\mathcal{S}$, the local update of SCAFFOLD is:

$$\boldsymbol{w}_i^{t,e+1} = \boldsymbol{w}_i^{t,e} - \eta(\nabla F_i(\boldsymbol{w}_i^{t,e}) - \boldsymbol{c}_i^t + \boldsymbol{c}^t), \tag{4}$$

for $e \in \{0, \cdots, E-1\}$. After E iterations, clients send weights $\boldsymbol{w}_i^t = \boldsymbol{w}_i^{t,E}$ and gradients $\boldsymbol{c}_i^{t+1} = \nabla F_i(\boldsymbol{w}_i^t)$ to the server. The server takes the average of control variables $\boldsymbol{c}^{t+1} = \boldsymbol{c}^t + \frac{1}{N}\left(\sum_{i\in\mathcal{S}}\left[\boldsymbol{c}_i^{t+1} - \boldsymbol{c}^t\right]\right)$, and re-scales the updates for weights by η_g, $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \frac{\eta_g}{|\mathcal{S}|}\left(\sum_{i\in\mathcal{S}}[\boldsymbol{w}_i^t - \boldsymbol{w}^t]\right)$. Note here that c^t is taken over all N clients. For those that did not participate SCAFFOLD re-uses the previously computed gradients.

The idea behind SCAFFOLD is very intuitive. To solve (1), the ideal (centralized) update is that each client uses all client's data $\boldsymbol{w}_i^{t+1} = \boldsymbol{w}^t - \frac{\eta_g}{N}\sum_{i=1}^N \nabla F_i(\boldsymbol{w}^t)$. However such update rule is not possible in IoFT due to the need to communicate the gradients $\nabla F_i(\boldsymbol{w})$ with the orchestrator at every optimization iterate. To mimic the ideal update, SCAFFOLD uses $\boldsymbol{c}_i^t$ to approximates $\nabla F_i(\boldsymbol{w}_i^t)$ at using the last communication round, for all i. Then also $\boldsymbol{c}^t$ may approximate the gradient of the global risk, $\boldsymbol{c}^t = \frac{1}{N}\sum_{i=1}^N \boldsymbol{c}_i^t \approx \frac{1}{N}\sum_{i=1}^N \nabla F_i(\boldsymbol{w}_i^t)$. If this approximation holds, the update of SCAFFOLD becomes similar to ideal (centralized) update. One caveat in such update scheme is that $\boldsymbol{c}^t$ may not always equal (or approximate) the ideal value $\frac{1}{N}\sum_{i=1}^N \boldsymbol{c}_i^t$. Adding to that, $\boldsymbol{c}^t$ re-uses the previously computed gradients when clients do not participate. Therefore, when client participation rate is low, $\boldsymbol{c}^t$ can deviate far away from the ideal update leading to degraded optimization performance.

Empirically, SCAFFOLD requires fewer communication rounds to converge compared with FedAvg. A very similar algorithm is Federated SVRG [149], which applies stochastic variance reduced gradient descent to approximately solve (3). The update rule is $\boldsymbol{w}_i^{t,e+1} = \boldsymbol{w}_i^{t,e} - \eta\left(S_i\left(\nabla F_i(\boldsymbol{w}_i^{t,e}) - \boldsymbol{c}_i^t\right) + \boldsymbol{c}^t\right)$, where S_i is a diagonal matrix to rescale gradients. Federated SVRG reduces to SCAFFOLD when one sets S_i to the identity matrix.

As discussed, despite its efficiency on several FL tasks, SCAFFOLD does not work well in low client participation cases. To this end, FedDyn [3] uses a specially designed dynamic regularization to align gradients under partial participation. The objective on client i is defined as:

$$\min_{\boldsymbol{w}_i^t} F_i(\boldsymbol{w}_i^t) - \left\langle \nabla F_i(\boldsymbol{w}_i^{t-1}), \boldsymbol{w}_i^t \right\rangle + \frac{\mu}{2}\left\| \boldsymbol{w}_i^t - \boldsymbol{w}^{t-1} \right\|^2. \tag{5}$$

Objective (5) is also closely related to (2). In (2), when the weight $\boldsymbol{w}$ is near critical points of the global risk $F(\boldsymbol{w})$, $\nabla F(\boldsymbol{w})$ is close to 0, thus (2) reduces to (5). As a simple fixed points analysis, when all models start from $\boldsymbol{w}_i^{t-1} = \boldsymbol{w}^{t-1} = \boldsymbol{w}^*$, i.e. a critical point of the global loss, the optimal solution of (5) is still $\boldsymbol{w}^*$, thus local updates will stay at $\boldsymbol{w}^*$. FedDyn is proved to converge to critical points of the global objective with a constant stepsize. Also, to deal with partial client participation, FedDyn uses a SAG-style [281] averaging rule in *server_update*: instead of only averaging gradients from clients that participated in the training in one communication round, FedDyn estimates gradients on disconnected clients based on historic values and averages all gradients (or gradient estimates). In practice, FedDyn is shown to achieve similar test accuracy with much fewer communication rounds compared with FedAvg and FedProx, especially when client participation rate is low.

A closely related algorithm to FedDyn is Federated primal-dual (FedPD) [352]. FedPD and FedDyn have different formulations, but end up with the same update rule under some conditions. In FedPD, the optimization problem in (1) is reformulated to a constrained optimization problem

$$\min_{\boldsymbol{w}_0, \{\boldsymbol{w}_i\}} \frac{1}{N}\sum_{i=1}^N F_i(\boldsymbol{w}_i) \quad \text{subject to} \quad \boldsymbol{w}_1 = \ldots = \boldsymbol{w}_N = \boldsymbol{w}_0. \tag{6}$$

To solve the constrained optimization problem, FedPD introduces dual variables $\lambda_1, \cdots, \lambda_N$, then defines the augmented Lagrangian (AL) for client i to be $\mathcal{F}_i(\boldsymbol{w}_i, \lambda_i, \boldsymbol{w}_0) = F_i(\boldsymbol{w}_i) + \langle \lambda_i, \boldsymbol{w}_i - \boldsymbol{w}_0 \rangle + \frac{\mu}{2}\|\boldsymbol{w}_i - \boldsymbol{w}_0\|^2$. FedPD uses alternative descent on primal and dual variables to optimize $\mathcal{F}_i$. More specifically, FedPD first randomly initializes $\boldsymbol{w}_0$, λ_i, $\boldsymbol{w}_i$ for all clients. At round t, the algorithm updates $\boldsymbol{w}_i^{t+1}$ by optimizing $\mathcal{F}_i$ and fixing $\lambda_i = \lambda_i^t$ and $\boldsymbol{w}_0 = \boldsymbol{w}_0^t$. It then updates dual variables by $\lambda_i^{t+1} = \lambda_i^t + \mu(\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_{0,i}^t)$ and also $\boldsymbol{w}_{0,i}^{t+1} = \boldsymbol{w}_i^{t+1} + \frac{1}{\mu}\lambda_i^{t+1}$. After the local updates, FedPD makes a random choice with probability $1-p$, that all clients send updated $\boldsymbol{w}_{0,i}^t$ back to the orchestrator which updates $\boldsymbol{w}_0^{t+1} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{w}_{0,i}^{t+1}$ and broadcasts updated $\boldsymbol{w}_0^{t+1}$. With probability p, all clients set $\boldsymbol{w}^0 = \boldsymbol{w}_{0,i}^{t+1}$ and continue local training. Interestingly, by letting $\lambda_i = \nabla F_i(\boldsymbol{w}_i^{t-1})$, it was shown that FedPD is equivalent to FedDyn with full client participation on an algorithmic level [351]. However, different from FedDyn, FedPD does not directly apply to partial participation settings.

Another algorithm that uses a constrained optimization formulation is FedSplit [243]. FedSplit [243] applies Peaceman-Rachford splitting [58], [244]. More specifically, FedSplit concatenates $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_N$ into one long vector $\boldsymbol{\mathcal{W}} = (\boldsymbol{w}_1^{\mathrm{T}}, \ldots, \boldsymbol{w}_N^{\mathrm{T}})^{\mathrm{T}}$ and finds the optimal solution of $\frac{1}{N}\sum_{i=1}^N F_i(\boldsymbol{w}_i)$ on the subspace $\mathcal{E} = \{\boldsymbol{\mathcal{W}} | \boldsymbol{w}_1 = \cdots = \boldsymbol{w}_N\}$. The problem is also known as consensus optimization [272]. An important concept in consensus optimization is the normal cone defined as $\mathcal{N}_{\mathcal{E}}(\boldsymbol{\mathcal{W}}) = \mathcal{E}^{\perp}$ for $\boldsymbol{\mathcal{W}} \in \mathcal{E}$ and empty otherwise. At the optimal solution, the gradient should be in the normal cone of $\mathcal{E}$:

$$0 \in \nabla_{\boldsymbol{\mathcal{W}}} \sum_{i=1}^N F_i(\boldsymbol{w}_i) + \mathcal{N}_{\mathcal{E}}(\boldsymbol{\mathcal{W}}).$$

`FedSplit` treats gradient $\nabla_{\mathcal{W}}$ and normal cone $\mathcal{N}_{\mathcal{E}}$ as two operators, and uses Peaceman-Rachford splitting [58] to find a solution $\mathcal{W}$ that satisfies the optimality condition. After some derivations, the authors propose the following update rules. At communication round t, clients update their local weights $\boldsymbol{w}_i^t$, send them to server and store a local copy. In the following round, client i receives global update $\boldsymbol{w}^t$, and calculates:

$$\begin{cases} \boldsymbol{w}_i^{t+\frac{1}{2}} = \arg\min_{\boldsymbol{w}_i} F_i(\boldsymbol{w}_i) + \dfrac{\mu}{2} \left\| \boldsymbol{w}_i - \left(2\boldsymbol{w}^t - \boldsymbol{w}_i^t\right) \right\|^2 \\ \boldsymbol{w}_i^{t+1} = \boldsymbol{w}_i^t + 2\left(\boldsymbol{w}^t - \boldsymbol{w}_i^{t+\frac{1}{2}}\right). \end{cases}$$

The *server_update* simply averages $\boldsymbol{w}^{t+1} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{w}_i^{t+1}$. Intuitively, operator splitting adds a regularization term centered at $2\boldsymbol{w}^t - \boldsymbol{w}_i^t$ to the local objective. The carefully designed update rule has two advantages. Firstly it help alleviate client-drift: `FedSplit` convergences linearly to critical points of the global loss on convex problems. Also, it accelerates convergence: the theoretical convergence rate is faster than that of `FedAvg` on strongly convex problems.

In addition to algorithms applicable to general federated optimization problems, there are models designed specifically for neural networks to handle heterogeneity. For instance, researchers pointed out that re-permutation of neurons may cause declined performance in the aggregation step of FL. This re-permutation problem is due to the fact that different neural networks created by a weight permutation might represent the same function.

For example, consider a simple NN, $f_{\boldsymbol{w}}(x) = W_2\sigma\left(W_1 x\right)$ where σ is an activation function, $f_{\boldsymbol{w}}(x) \in \mathbb{R}^{d_y}$, $x \in \mathbb{R}^{d_x}$ and $W_1 \in \mathbb{R}^w \times \mathbb{R}^{d_x}$ and $W_2 \in \mathbb{R}^{d_y} \times \mathbb{R}^w$ are weight matrices. One can multiply a permutation matrix $\Pi \in \mathbb{R}^w \times \mathbb{R}^w$ to W_1 and W_2, $f_{\boldsymbol{w}}(x) = W_2\Pi^{\mathrm{T}}\sigma\left(\Pi W_1 x\right)$, and the function remains the same. However updates on different clients may be attracted to networks with a different permutation matrix Π. This can cause averaging over weights to fail. To cope with this, [343] propose a neuron matching algorithm called Probabilistic Federated Neural Matching (`PFNM`). `PFNM` assumes $\Pi_i W_1$ and $W_2\Pi_i^{\mathrm{T}}$ are generated by a hierarchical probabilistic model whose hyper-parameters are determined by global weights. Then `PFNM` uses Bayesian inference to estimate the hyper-parameters, and reconstructs the global model from the inference.

However, [303] argue that `PFNM` can only work on simple fully connected neural networks. To solve the problem, they extend `PFNM` to Federated Matched Averaging (`FedMA`) algorithm. `FedMA` updates weights of a neural network layer by layer. Firstly, clients train local NNs and send the trained first layer weights $W_i^{(1)}$ to the orchestartor, where $W_i^{(1)}$ denotes the weight vector of layer 1 from client i. The server uses matching algorithms such as the Hungarian algorithm in `PFNM` to estimate $\Pi_i^{(1)}$ that represents the permutation vector of first layer model weights for client i. Thus, $\Pi_i^{(1)} W_i^{(1)}$ become the matched weights after re-permutation. The server then averages the results $\overline{W}^{(1)} = \frac{1}{N}\sum_{i=1}^N \Pi_i^{(1)} W_i^{(1)}$, and broadcasts the averaged $\overline{W}^{(1)}$. After receiving $\overline{W}^{(1)}$, clients continue to train the remaining layers with the first layer fixed to $\overline{W}^{(1)}$. A similar match-then-average process repeats for remaining layers. For `FedMA`, the number of communication rounds equals the number of network layers. `FedMA` is reported to have strong performance on CIFAR-10 and a well known language dataset called Shakespeare. Additionally, the performance of FedMA improves with the increase of local epochs E, while that of `FedAvg` and `FedProx` drops after a threshold of E due to the discrepancy between local models (i.e. local weights wander away from each other). Thus `FedMA` enables clients to train more epochs between consecutive communications.

All approaches described above are of a frequentist nature. However, there has also been a recent push on improving global modeling through a Bayesian framework. The intuition is simple; rather than betting our results on one hypothesis ($\boldsymbol{w}$) obtained via optimizing the empirical risk, one may average over a set of possible $\boldsymbol{w}$ or integrate over all $\boldsymbol{w}$ weighted by their posterior probability $\mathbb{P}(\boldsymbol{w}|D = \{D_1, \cdots, D_N\})$. This is the underlying philosophy of marginalization compared to optimization, whereby in the frequentist approach predictions are obtained through substituting the posterior by $\mathbb{P}(\boldsymbol{w}|D) = \delta(\boldsymbol{w} = \hat{\boldsymbol{w}})$, where $\hat{\boldsymbol{w}}$ is the single optimized weight and δ is an indicator function. Indeed, this notion of Bayesian ensembling has seen a lot of empirical success in Bayesian deep learning [124], [196].

One such approach is `Fed-ensemble` [285]. `Fed-ensemble`, is a simple plug-in into any FL algorithm that aims to learn an ensemble of K-models without additional communication costs. To do so, `Fed-ensemble` follows a random permutation sampling scheme where at each communication round, every client trains one of the K models and then aggregation happens for each model separately (using `FedAvg` or other FL approaches). This approach corresponds to a variational inference scheme [27], [347] for estimating a Gaussian mixture variational distribution whose centers are randomly initialized at the beginning. Predictions on a new input x^* are then obtained by taking an average over the predictions of the K models

$$f(x^*) = \frac{1}{K}\sum_k f_{\boldsymbol{w}_k}(x^\star). \tag{7}$$

`Fed-ensemble` is also able to quantify predictive uncertainty. Using a neural tangent kernel argument, the authors show that all predictions from all K models converge to samples from the same limiting Gaussian process in sufficiently overparameterized regimes (see Fig. 7) where each mode can behave like a model trained by centralized training.

Another recent work taking insights from Bayesian inference is `FedBE` [42]. It performs statistical inference on the client-trained models and uses knowledge-distillation (KD) to update the global model. Intuitively, the goal of KD is to use high-quality base models from a global distribution $\mathbb{P}(\boldsymbol{w})$ to direct the global model update. More specifically, after receiving $\{\boldsymbol{w}_i\}_{i\in\mathcal{S}}$ from clients, the server fits them with

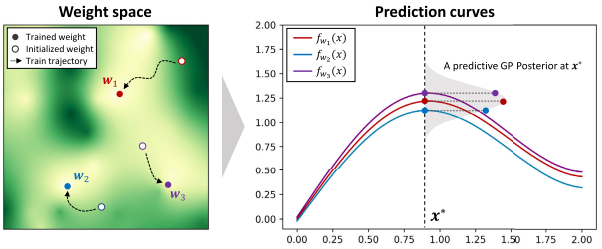

FIGURE 7. An illustration of an ensemble of $K = 3$ models. The three model weights on the left figure correspond to the three predictions on the right. Although the weights are well separated, the predictions admit the same limiting posterior distribution.

a Gaussian or Dirichlet distribution and then samples from the estimated distribution to form an ensemble of K models $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K\}$. Similar to (7), the ensemble prediction on a new point x^* is given by $y_{ensemble} = \frac{1}{K}\sum_{i=1}^{K} f_{\boldsymbol{w}_k}(x^*)$. In *server_update*, the global model $f_{\boldsymbol{w}}$ is trained to mimic the average prediction of models in the ensemble by minimizing the discrepancy between the two predictions evaluated on an additional unlabeled dataset D^{add} on the server:

$$\boldsymbol{w}^{t+1} = \arg\min_{\boldsymbol{w}} \mathbb{E}_{x\sim D^{\text{add}}}\left[\text{Div}(y_{ensemble}(x), f_{\boldsymbol{w}}(x))\right],$$

where Div denotes a divergence measure, here cross-entropy. The updated $\boldsymbol{w}^{t+1}$ is then sent to all clients. The authors empirically show that the ensemble and knowledge distillation turns out to be more robust to non-*i.i.d.* data than `FedAvg`. This approach however requires storing additional data on server which is not always feasible.

C. EFFICIENT & EFFECTIVE OPTIMIZATION

Several studies attempt to improve `FedAvg` by adopting adaptive optimization algorithms to the FL realm. They show theoretically or empirically that the improved algorithms can converge faster and accelerate global model training. In general, acceleration can be achieved by either improving the server aggregation step (*server_update*) or client updates (*client_update*). `FedAdam` and `FedYogi` [266] bring the well known Adam [143] and Yogi [265] algorithms to FL through augmenting the *server_update* function by adaptive stepsizes. More specifically, `FedAdam` and `FedYogi` use a second order moment estimate v_t to adaptively adjust the learning rate. v_0 is initialized at the beginning. Upon receiving $\boldsymbol{w}_i^t$ from clients, server calculates $\Delta \boldsymbol{w}_i^t = \boldsymbol{w}_i^t - \boldsymbol{w}^t$, and averages them $\Delta_t = \sum_{i\in\mathcal{S}} \frac{1}{|\mathcal{S}|}\Delta \boldsymbol{w}_i^t$. `FedAdam` updates v_t as:

$$v_t = \zeta v_{t-1} + (1-\zeta)\,\Delta_t^2,$$

and `FedYogi` as:

$$v_t = v_{t-1} - (1-\zeta)\,\Delta_t^2 \text{sign}(v_t - \Delta_t^2),$$

where ζ is a parameter for exponential weighting. The update rule for both `FedAdam` and `FedYogi` is:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \eta \frac{\Delta_t}{\sqrt{v_t} + \epsilon},$$

where ϵ is a small constant for numerical stability. Though the proved theoretical convergence rates of `FedAdam` and `FedYogi` are only comparable to those of `FedAvg`, the adaptive methods show strong performance on several FL tasks. Considering the success of adaptive stepsize methods in numerous important fields including language models [301], GANs [282], [332], amongst others, we believe their use in FL is promising. A related algorithm in this vein is federated averaging with server momentum (`FedAvgM`) [186], which uses server momentum in the *server_update* step.

Besides modifying *server_update*, multitudes of algorithms redesign the *client_update* function. For instance, there are some attempts to expedite local training by combining accelerating techniques in optimization.

`FedAc` [337] is a federated version of an accelerated SGD. Instead of updating a single variable $\boldsymbol{w}_i$ as `FedAvg` does, `FedAc` updates three sequences $\{\boldsymbol{w}_i, (\boldsymbol{w}_i)^{ag}, (\boldsymbol{w}_i)^{md}\}$ iteratively on the client side by the following rules for several steps:

$$\begin{cases} \left(\boldsymbol{w}_i^t\right)^{md} \leftarrow \zeta_1 \boldsymbol{w}_i^t + (1-\zeta_1)\left(\boldsymbol{w}_i^t\right)^{ag} \\ \quad \boldsymbol{g}_i \leftarrow \nabla F_i(\left(\boldsymbol{w}_i^t\right)^{md}) \\ \left(\boldsymbol{w}_i^t\right)^{ag} \leftarrow \left(\boldsymbol{w}_i^t\right)^{md} - \eta_1 \boldsymbol{g}_i \\ \quad \boldsymbol{w}_i^t \leftarrow (1-\zeta_2)\,\boldsymbol{w}_i^t + \zeta_2\left(\boldsymbol{w}_i^t\right)^{md} - \eta_2 \boldsymbol{g}_i. \end{cases}$$

$\zeta_1, \zeta_2, \eta_1, \eta_2$ are four hyper-parameters. Among them ζ_1, ζ_2 are exponential averaging parameters, and η_1, η_2 are stepsize parameters. The server averages $\boldsymbol{w}_i^t$ and $\left(\boldsymbol{w}_i^t\right)^{ag}$ from sampled clients, and broadcasts the averaged $\boldsymbol{w}^{t+1}$ and $\left(\boldsymbol{w}^{t+1}\right)^{ag}$, which clients will take as initialization of $\boldsymbol{w}_i^{t+1}$ and $\left(\boldsymbol{w}_i^{t+1}\right)^{ag}$ in the next communication rounds. The algorithm then proceeds till convergence. Reference [337] theoretically prove that `FedAc` can achieve a linear convergence rate faster than `FedAvg` when global risk $F(\boldsymbol{w})$ in (1) is strongly convex. Empirical results show that `FedAc` saves communication cost when there are many devices in the network.

`LoAdaBoost` [118] adaptively determines the training epochs of clients by monitoring the training loss on each client and adjusting the training schedule accordingly. More specifically, after one communication round, clients send training losses, in addition to updated weights to the server. The server estimates the median of the training loss F_i, $F_{median} := median(\{F_i\}_{i=1,\cdots,N})$. In the next round, all clients train for a certain amount of epochs $\frac{E'}{2}$, where E' is the average budget of epochs. If the training loss is lower than F_{median}, the local training is deemed to have reached its goal in this round, and the updated weight will be directly sent back to server. If the training loss is higher than F_{median} on client i, then the model underfits client i. As a result, `LoAdaBoost` will train the model on client i for extra epochs until the local training loss is lower than L_{median} or the total epochs exceed $\frac{3}{2}E$, whichever comes faster. Such dynamic training schedules allow `LoAdaBoost` to take resources of clients into consideration, thus can better utilize computation power on edge devices and enable faster training.

D. SAMPLING CLIENTS

Due to the often sheer size and unreliability of edge devices participating within IoFT, not all clients can participate in each communication round of the training process as shown in Algorithm 1. Therefore, choosing the appropriate subset $\mathcal{S}$ at each communication round between the orchestrator and client is of utmost importance in FL. Here we shed light on some existing schemes, other possible alternatives and their implications.

We first start by noting that an alternative approach to write the global objective in (1) is through giving different weights to client risk function. This is given as

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) := \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}) = \mathbb{E}_i[F_i(\boldsymbol{w})], \tag{8}$$

where p_i is a weight such that $p_i \geq 0$ and $\sum_{i=1}^{N} p_i = 1$. In IoFT, it is common to have datasets of different sizes. Thus, a natural choice is to set $p_i = \frac{n_i}{n}$ where n is the total data size across all clients $n = \sum_{i=1}^{N} n_i$. Clearly if all clients have the same dataset size n_i, objective (8) reduces to (1).

Indeed, although most algorithms for FL use (1), both `FedAvg` and `FedProx` (among the earliest methods) use (8) by adding weights p_i. `FedAvg` samples clients $\mathcal{S} \subseteq [N]$ uniformly with probability $\mathbb{P}_i = \frac{1}{N}$, and averages client models with weights proportional to their local dataset size n_i: $\boldsymbol{w}^{t+1} = \frac{1}{\sum_{i\in\mathcal{S}} n_i} \sum_{i\in\mathcal{S}} n_i \boldsymbol{w}_i^{t+1}$. On the other hand, `FedProx` samples clients with probability $\mathbb{P}_i = p_i = \frac{n_i}{n}$ and averages client models with equal weights: $\boldsymbol{w}^{t+1} = \frac{1}{|\mathcal{S}|} \sum_{i\in\mathcal{S}} \boldsymbol{w}_i^{t+1}$. These sampling probability and weights are chosen to make client updates unbiased estimates of global updates - i.e. unbiased estimates of $\sum_{i=1}^{N} p_i \nabla F_i(\boldsymbol{w})$.

However, both sampling schemes may have some drawbacks. For example, uniform sampling may be inefficient since the orchestrator can often sample unreliable clients or clients with very small datasets. Dataset size-based sampling addresses this issue, but it may raise fairness concerns as some clients are rarely sampled and trained. This also makes the training procedure more prone to adversarial clients with large datasets that can directly impact the training process.

To form better sampling schemes and accelerate training, adaptive sampling techniques have also been proposed. These FL algorithms update the sampling probability $\mathbb{P}_i^t$ after each communication round from historical statistics [51], [53], [162], [323]. Such methods usually sample clients on which the model fits worse, more often. Intuitively, when a model incurs high training loss or large gradient norms on client i, client i is not performing well under the current model and should be trained for more epochs.

There are a range of choices for measuring the performance of a model on the client. Among them, there is a set of literature that calculates the sampling probabilities adaptively using gradient norms of the clients [138], [185], [219]. Generally this can be given as

$$\mathbb{P}_i^t \propto \exp\left(\gamma \left\| \nabla F_i(\boldsymbol{w}_i^t) \right\|^2\right),$$

where γ is some constant. Other approaches sample clients based on their training loss [30], [53] where the gradient is substituted by the local loss $F_i(\boldsymbol{w}_i^t)$ at the end of each training round. In this set of literature, exploration and exploitation schemes are also used to continuously update the sampling probability. Client selection usually improves model performance and speeds up the training process. For example, [162] can achieve $1.2\times$ to $14.1\times$ speed up in terms of time-to-accuracy compared with vanilla `FedProx` or `FedYogi`.

Due to prevailing statistical and system heterogeneity among clients, we believe client sampling techniques will be of great significance when practitioners try to deploy FL frameworks in IoFT. An effective sampling scheme can efficiently exploit differences in client's resources while at the same time improving training speed and accuracy. Further, studying the connections between adaptive sampling and client re-weighting schemes (see Sec. III-E) used for fairness is an interesting topic worthy of investigation.

E. FAIRNESS ACROSS CLIENTS

In IoFT, it is crucial to ensure that all edge devices have good prediction performance. However, the key challenge is that devices with insufficient amounts of data, limited bandwidth, or unreliable internet connection are not favored by conventional FL algorithms. Such devices can potentially end with bad predictive ability. Besides this notion of individual fairness, group fairness also deserves attention in FL. As FL penetrates many practical applications, it is important to achieve fair performance across groups of clients characterized by their gender, ethnicity, etc. Before diving into the literature, we first start by formally defining the notion of fairness. Suppose there are d groups (e.g., ethnicity) and each client can be assigned to one of those groups. Group fairness can be defined as follows.

Definition 1: Denote by $\{a_{\boldsymbol{w}}^i\}_{1\leq i\leq d}$ the set of performance measures (e.g., testing accuracy) of a trained model $\boldsymbol{w}$. For trained models $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$, we say $\boldsymbol{\theta}$ is more fair than $\tilde{\boldsymbol{\theta}}$ if $Var(\{a_{\boldsymbol{\theta}}^i\}_{1\leq i\leq d}) < Var(\{a_{\tilde{\boldsymbol{\theta}}}^i\}_{1\leq i\leq d})$, where Var denotes variance.

When $d = N$, this definition is equivalent to the individual fairness. Definition 1 is widely adopted in most FL literature [119], [172], [214], [345], [348]. This notion of fairness might be different from traditional definitions such as demographic disparity [85], equal opportunity and equalized odds [107] in centralized systems. The reason is that those conventional definitions cannot be extended to FL as there is no clear notion of an outcome which is optimal for an edge device [133]. Instead, fairness in FL can be defined as equal access to effective models (e.g., the accuracy disparity [344] or the representation disparity [172]). Specifically, the goal is to train a global model that incurs a uniformly good performance across all devices or groups [133].

Despite the importance of fairness, unfortunately, very limited work exist along this line in FL. As will become clear shortly, the few works in this area mainly focus on a client

re-weighting scheme through exploiting the weighted global objective in (8) instead of (1).

GIFAIR-FL [342] is the first algorithm that can handle both group and individual fairness in FL. Specifically, it achieves fairness by penalizing the spread in the loss among client groups. This can be translated to the following optimization problem:

$$H(\boldsymbol{w}) = \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}) + \lambda \sum_{1\le j<k\le d} \left|L_j(\boldsymbol{w}) - L_k(\boldsymbol{w})\right|,$$

where λ is a regularization parameter and

$$L_j(\boldsymbol{w}) = \frac{1}{|\mathcal{A}_j|} \sum_{i\in\mathcal{A}_j} F_i(\boldsymbol{w})$$

is the average loss for group j and $\mathcal{A}_j$ is the set of indices of devices who belong to group i. The original formulation of $H(\boldsymbol{w})$ can be further simplified as

$$H(\boldsymbol{w}) = \sum_{i=1}^{N} p_i \left(1 + \frac{\lambda}{p_i|\mathcal{A}_{s_i}|} r_i(\boldsymbol{w})\right)$$
$$F_i(\boldsymbol{w}) := \sum_{i=1}^{N} p_i H_i(\boldsymbol{w})$$

where

$$r_i(\boldsymbol{w}) = \sum_{1\le j\ne s_i\le d} \text{sign}(L_{s_i}(\boldsymbol{w}) - L_j(\boldsymbol{w})),$$

and $s_i \in [d]$ is the group index of device i. $r_i(\boldsymbol{w})$ can be viewed as a scalar that related to the statistical ordering of L_{s_i} among client group losses. Therefore, to collaboratively minimize $H(\boldsymbol{w})$, each edge device i is minimizing $H_i(\boldsymbol{w})$, a scaled version of the original local loss function $F_i(\boldsymbol{w})$. The central server will aggregate local parameters and update $\{r_i(\boldsymbol{w})\}_{i=1}^N$ at every communication round. From the expression of $r_i(\boldsymbol{w})$, one can see that a higher value of $r_i(\boldsymbol{w})$ will be assigned to the client that has higher group loss. Therefore, GIFAIR-FL will impose higher weights for clients with bad performances. Furthermore, those weights will be dynamically updated at every communication round to avoid possible model overfitting. Reference [342] has shown that GIFAIR-FL will converge to an optimal solution or a stationary point even when heterogeneity exists.

Agnostic federated learning (AFL) [214] is another algorithm that re-weights clients at each communication round. Specifically, it solves a robust optimization problem in the form of

$$\min_{\boldsymbol{w}} \max_{\max p_1,\ldots,p_N} \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}).$$

AFL computes the worst-case combination of weights among edge devices. This approach is robust but may be conservative since it only focuses on the largest loss and thus causes pessimistic performance to other clients. Du *et al.* [75] further refine the notation of AFL by linearly parametrizing weight parameters by some kernel functions. Upon that Hu *et al.* [116] combine minimax optimization with gradient normalization to formulate a new fair algorithm FedMGDA+.

Inspired by fair resource allocation for wireless networks problems, [172] propose the q-FFL algorithm for fairness. They slightly modify the loss function and add a power q to each user

$$\min_{\boldsymbol{w}} \sum_{i=1}^{N} \frac{p_i}{q+1} F_i^{q+1}(\boldsymbol{w}). \tag{9}$$

The intuition is that q tunes the amount of fairness: the algorithm will incur a larger loss to the users with poor performance. Therefore, q-FFL can ensure uniform accuracy across all users.

To the best of our knowledge, fairness is an underinvestigated yet critical area in the FL setting. We hope this section can inspire the continued exploration of fair FL algorithms.

IV. LEARNING A PERSONALIZED MODEL

As highlighted in previous sections, heterogeneity is a fundamental challenge for IoFT. IoFT devices often exhibit highly heterogeneous trends and behaviors due to differences in operational, environmental, cultural, socio-economic and specification conditions [152], [153], [341]. For instance, in manufacturing, operational differences involve changes in the speed, load, or temperature a product experiences. As a result, data distribution across edge devices can be vastly heterogeneous, that one single global model cannot perform consistently well on all edge devices. This also has severe fairness implications as devices with limited data and unreliable connection will not be favored by many FL algorithms due to higher weights (recall FedAvg and its variants) given to devices with more data or those that can participate more often in the training process. Indeed, in the past few years, multiple papers have shown the wide gap in a global model's performance across different devices when heterogeneity exists [106], [131], [133], [288], [306].

One straightforward solution to address the challenges above is through personalization. As shown in Fig. 8, instead of using one global model for all edge devices, personalized FL fits tailor-made models for IoFT devices while levering information across all those devices. The rest of this section will discuss current personalization approaches, their drawbacks and potential alternatives. We divide the personalization techniques into fully personalized and semi-personalized. For fully personalized algorithms, each edge device retains its own individualized model, and for semi-personalized algorithms, models are tailor-made only to a group of clients. In Sec. V, we will further discuss personalization from a meta-learning perspective.

A. FULLY PERSONALIZED

From a statistical perspective, let $x_i \sim \mathbb{P}_x$, but let the conditional distributions of y_i given x_i vary across IoFT devices.

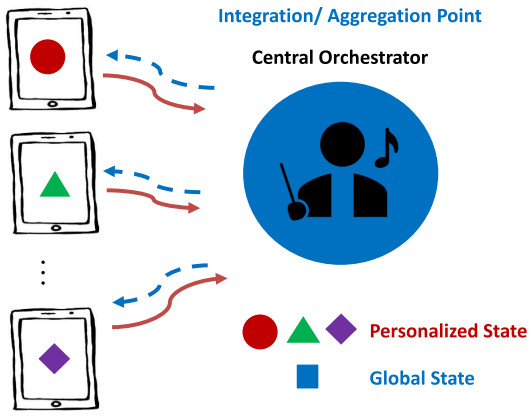

FIGURE 8. Personalized IoFT.

One can write this as $y_i \sim \mathbb{P}^i_{y|x}\left(f_{\boldsymbol{\beta}_i}(x_i)\right)$ where clients share the same f (a linear model, neural network) yet with different parameters $\boldsymbol{\beta}_i$. In this situation, the difference in the data distributions $\mathbb{P}^i_{x,y} = \mathbb{P}_x\mathbb{P}^i_{y|x}$ across clients can be explained by the difference in $\mathbb{P}^i_{y|x}$. This is often referred to as a concept shift and implies a change in the input-output relationship across clients [209], [294]. For example, in manufacturing the same design setting can have different effects on the manufactured product given external factors such as operational speed or load. Also, take the sequence prediction task on mobile phones as an example: for different users, the word following "I live in ..." should be different [155]. This example corresponds to a concept shift: x is assumed the given part of the sentence 'I live in', and y is the next word to predict. In this situation, $\mathbb{P}^i_{y|x}$ should be customized for different clients even if x is the same.

This section discusses current approaches to address a concept shift across clients, their drawbacks, and promising alternatives. Modeling for a shift in $\mathbb{P}_x$ is highlighted in our statistical perspective (Sec. VI-A).

To accommodate client specific concept shifts while leveraging global information one can extend the global FL model in (1) to the following general objective for personalized FL:

$$\min_{\boldsymbol{w},\boldsymbol{\beta}} F(\boldsymbol{w},\boldsymbol{\beta}) := \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{w},\boldsymbol{\beta}_i), \tag{10}$$

where $\boldsymbol{w}$ are shared global parameters while $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i\}_{i=1}^N$ is a set of unique parameters for each client.

The current literature aiming to address a concept shift can be broadly split into two categories: (i) weight sharing and (ii) regularization. **It will also become clear shortly that many current approaches follow a train-then-personalize philosophy which may be dangerous in some instances**.

1) WEIGHT SHARING

The first set of literature solves (10) by using different layers of a neural network to represent $\boldsymbol{w}$ and $\boldsymbol{\beta}_i$ [178], [306]. The underlying idea is that base layers process the input to learn a shared feature representation across clients, and top layers learn task-dependent weights based on the features.

`FedPer` [306] fits global base layers, and personalizes top layers. As an example, a fully connected multi-layer neural network can be expressed as $f_{\boldsymbol{w}}(x) = W_l\sigma\left(W_{l-1}\sigma\left(\ldots\sigma\left(W_1x\right)\right)\right)$, where l are the number of network layers. Recall from Sec. III-B, σ denotes an activation function and W_j's are weight matrices. In this example, `FedPer` takes W_1 to W_B as base layers that characterize $\boldsymbol{w}$, and W_{B+1} to W_n as personalized layers that characterize $\boldsymbol{\beta}_i$ in (10). In one communication round, client i uses SGD to update $\boldsymbol{w}$ and $\boldsymbol{\beta}_i$ simultaneously. However, different from `FedAvg`, only $\boldsymbol{w}$ is transmitted to the server where it is then aggregated. `FedPer` is found to perform better than `FedAvg` on image classification tasks such as CIFAR-10 and CIFAR-100. On these datasets, the authors show that having the last one or two basic residual blocks of Resnet-34 personalized can yield the best testing performance. Similarly, `LG-FedAvg` [178] takes top layers as a global weight $\boldsymbol{w}$ and base layers as personalized weights $\boldsymbol{\beta}_i$. The intuition is to learn customized representation layers for different clients, and to train a global model that operates on local representations. Additionally, by carefully designing the loss of representation learning, the generated local representation can confound protected attributes like gender, race, etc.

2) REGULARIZATION

In contrast to splitting of global and local layers, other recent work treat neural networks holistically and learn personalized $\boldsymbol{\beta}_i$'s by exploiting regularization [70], [288].

Perhaps the most straightforward method to personalize via regularization is to follow a **train-then-personalize (TTP)** approach. As the name suggests, this approach trains the global model on all clients then adapts it to individual devices. The simplest way for the adaptation is fine-tuning [8], [336], which is also widely employed in computer vision and natural language processing [255]. More specifically, in the TTP approach, we have a two step procedure. Step 1 - *Train*: clients collaborate to train a global model $\boldsymbol{w}^* \triangleq \boldsymbol{w}^T$ using `FedAvg` (or its variants) - recall T is the last communication round. Step 2 - *Personalize*: clients make small local adjustments based on their local data to personalize $\boldsymbol{w}^*$. Notice that for such methods, $\boldsymbol{\beta}_i$'s and $\boldsymbol{w}$ are in the same parameter space thus it's possible to perform addition or calculate the difference of these weight vectors. Weight regularizing methods thus usually allow all the weight vector to differ across clients, instead of forcing some coordinates of these weight vectors to be exactly the same.

A simple means for the personalization step is to start from $\boldsymbol{\beta}_i = \boldsymbol{w}^*$ and perform a few steps of SGD to minimize the local loss function $\min_{\boldsymbol{\beta}_i} F_i(\boldsymbol{\beta}_i)$. Indeed, this approach to fine-tuning is shown to generalize better than fully local training or global modeling on next word prediction [295] and image classification tasks (e.g. [83], [190]). In this same essence, one may exploit regularization to encourage the

weights of personalized models to stay in the vicinity of the global model parameters to balance each client's shared knowledge and unique characteristics. For instance, using ideas from FedProx, the personalization step can encourage $\boldsymbol{\beta}_i$ to remain within a vicinity of the global solution $\boldsymbol{w}^*$ as shown below.

$$\min_{\boldsymbol{\beta}_i} \left(F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \left\| \boldsymbol{\beta}_i - \boldsymbol{w}^* \right\|^2 \right). \tag{11}$$

Other forms of regularization can also be used. For instance, by employing the popular elastic weight consolidation model (EWC) [144] that is often used in continual learning, we can control $\boldsymbol{\beta}_i$ as

$$\min_{\boldsymbol{\beta}_i} \left(F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \sum_j \mathcal{FI}_j \left\| \beta_j - w_j^* \right\|^2 \right),$$

where $\mathcal{FI}_j$ are diagonal elements of the Fisher information matrix.

Some recent approaches [70], [174] have exploited the ideas above but in an iterative manner, where local and global parameters in the train-then-personalize step are obtained by alternating optimization methods. Among them, Ditto [174] simply proposed the following bi-level optimization problem for client i:

$$\begin{aligned} &\min_{\boldsymbol{\beta}_i} \quad F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \left\| \boldsymbol{\beta}_i - \boldsymbol{w}^* \right\|^2 \\ &\text{s.t.} \quad \boldsymbol{w}^* \in \arg\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{w}). \end{aligned} \tag{12}$$

To solve this formulation, Ditto uses the following update rule. In communication round t, clients firstly receive a copy of global weight $\boldsymbol{w}^t$ which is updated to $\boldsymbol{w}_i^{t+1}$ using multiple (S)GD steps on the local risk function $F_i(\boldsymbol{w})$; much like FedAvg. In the meantime, clients i also obtains β_i by multiple descent steps on the regularized loss (12):

$$\boldsymbol{\beta}_i \leftarrow \boldsymbol{\beta}_i - \eta \nabla F_i(\boldsymbol{\beta}_i) - \eta\mu \left(\boldsymbol{\beta}_i - \boldsymbol{w}^t \right).$$

At the end of the training round, client i sends only global weight $\boldsymbol{w}_i^{t+1}$ back to server. Server simply averages received weights $\boldsymbol{w}^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{w}_i^{t+1}$. Empirically, Ditto has shown strong personalization accuracy on multiple commonly used FL datasets.

In Ditto, the global weight update is independent of personalized weights and follows the FedAvg procedure. Hence global weights cannot learn from the performance of personalized weights. To integrate the update of $\boldsymbol{w}$ and $\boldsymbol{\beta}_i$, [70] proposes Moreau envelope FL (pFedMe) for personalization. pFedMe formulates the following bi-level optimization problem:

$$\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} \min_{\boldsymbol{\beta}_i} \left[F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \left\| \boldsymbol{w} - \boldsymbol{\beta}_i \right\|^2 \right].$$

pFedMe gets its name because $\min_{\boldsymbol{\beta}_i} \left[F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \left\| \boldsymbol{w} - \boldsymbol{\beta}_i \right\|^2 \right]$ is the Moreau envelope of $F_i(\boldsymbol{w})$. In the inner level optimization, personalized weights $\boldsymbol{\beta}_i$ minimize the local risk function in the vicinity of reference point $\boldsymbol{w}$, and in the outer level minimization, $\boldsymbol{w}$ is minimized to produce a better reference point. This objective is closely related to model-agnostic meta-learning (MAML). Sec V will cover more details about meta-learning algorithms. The optimal solution of pFedMe satisfies the relation $\boldsymbol{\beta}_i^*(\boldsymbol{w}) = \boldsymbol{w} - \frac{1}{\mu} \nabla F_i(\boldsymbol{\beta}_i^*)$. Through some calculus, the gradient with respect to $\boldsymbol{w}$ is: $\nabla_{\boldsymbol{w}} [F_i(\boldsymbol{\beta}_i^*(\boldsymbol{w})) + \frac{\mu}{2} \left\| \boldsymbol{w} - \boldsymbol{\beta}_i^*(\boldsymbol{w}) \right\|^2] = \mu(\boldsymbol{\beta}_i^*(\boldsymbol{w}) - \boldsymbol{w})$. In practice, at one communication round, selected clients receive global weight $\boldsymbol{w}$ and find an approximate optimal solution $\tilde{\boldsymbol{\beta}}_i^*$ to the inner optimization problem $\min_{\boldsymbol{\beta}_i} \left[F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \left\| \boldsymbol{w} - \boldsymbol{\beta}_i \right\|^2 \right]$ via SGD or its variants. Client i then multiplies the update by $\mu\eta_l$, $\Delta \boldsymbol{w}_i = \mu\eta_l(\tilde{\boldsymbol{\beta}}_i^* - \boldsymbol{w})$, and sends $\Delta \boldsymbol{w}_i$ to the server. μ is set to $15 \sim 30$ in the case studies. The server then averages received $\Delta \boldsymbol{w}_i$ to renew the global model as $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta_g \sum_i \Delta \boldsymbol{w}_i$. η_g is another hyperparameter whose value is $1 \sim 4$ in experiments. With proper hyper-parameter choices, the convergence rate of pFedMe is proved to be faster compared with FedAvg.

Local loopless GD (L2GD) [105] simply sets $\boldsymbol{w}$ to be the average of $\boldsymbol{\beta}_i$, $\boldsymbol{w} = \overline{\boldsymbol{\beta}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\beta}_i$. The objective is:

$$\min_{\boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} \left[F_i(\boldsymbol{\beta}_i) + \frac{\mu}{2} \left\| \boldsymbol{\beta}_i - \overline{\boldsymbol{\beta}} \right\|^2 \right].$$

To optimize the objective, L2GD chooses to update $\frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{\beta}_i)$ and $\frac{1}{N} \sum_{i=1}^{N} \frac{\mu}{2} \left\| \boldsymbol{\beta}_i - \overline{\boldsymbol{\beta}} \right\|^2$ by a random experiment. More specifically, at round t, with probability $1 - p$, each client will perform one step of GD to minimize local training loss $\boldsymbol{\beta}_i^t$ as $\boldsymbol{\beta}_i^t \leftarrow \boldsymbol{\beta}_i^t - \frac{\eta}{N(1-p)} \nabla F_i(\boldsymbol{\beta})$. With probability $1 - p$, clients will send updated models $\boldsymbol{\beta}_i^t$ back to the server, unless the model is not updated since the previous synchronisation. The *server_update* consists of two steps: first, the server takes the average of $\boldsymbol{\beta}_i^t$ to obtain $\overline{\boldsymbol{\beta}^t} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{w}_i^t$; second, the server calculates the initialization of client i's weight on the $t+1$-th communication round as $\boldsymbol{\beta}_i^{t+1} = \left(1 - \frac{\alpha\mu}{Np}\right) \boldsymbol{\beta}_i^t + \frac{\alpha\mu}{Np} \overline{\boldsymbol{\beta}^t}$ and sends it to the corresponding client. Here α is a tunable hyperparameter.

Finally we shed light on another formulation of the train then personalize approach provided by [65]. Reference [65] learn the global parameter $\boldsymbol{w}^*$ using traditional global modeling approaches, yet, in the personalization step the local objective is given as

$$\min_{\boldsymbol{\beta}_i} F_i(\zeta_i \boldsymbol{\beta}_i + (1 - \zeta_i) \boldsymbol{w}^*), \tag{13}$$

The tuning parameter ζ_i balances the importance of the local and global model. When users data are *i.i.d.*, the authors argue that the global model will have better generalization and suggest choosing a smaller ζ_i. On the other hand, when data are heterogeneous, they choose to use a larger ζ_i to encourage personalization.

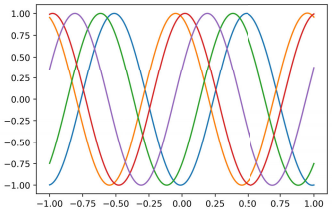

FIGURE 9. Underlying true sine functions from five different clients.

3) A COUNTER-EXAMPLE

The rationale behind regularization approaches is that the global model learns shared knowledge. Then, by encouraging local weights to stay close to the global model, every client can borrow strength from this shared knowledge. Unfortunately, this simple intuition does not apply to all problems. As a simple counterexample, assume client i's ground truth is a sine function:

$$f_{\theta_i}(x) = \sin(2\pi(x + \theta_i)),$$

where θ_i is client-dependent. We assume that among clients, θ_i admits a uniform distribution on [0, 1]. Sine functions are the basis of almost all periodic functions and a phase shift is common for vibration signals, which usually have strong similarity and large shifts.

Now, if we train a global model to minimize the population risk:

$$\min_{\boldsymbol{w}} \mathbb{E}_i[||f_{\boldsymbol{w}} - f_{\theta_i}||_2^2].$$

where $f_{\boldsymbol{w}}$ is a global model parametrized by weight $\boldsymbol{w}$, and $||\cdot||_2$ is a functional on [0, 1] defined as:

$$||f||_2^2 = \int_0^1 f(x)^2 dx.$$

Then $f_{\boldsymbol{w}}$ should minimize:

$$\begin{aligned}
&\arg\min_{f_w} \mathbb{E}_{\theta_i}\left[\int_0^1 \left(f_{\boldsymbol{w}}(x) - \sin(2\pi x + 2\pi\theta_i)\right)^2 dx\right] \\
&= \arg\min_{f_w} \mathbb{E}_{\theta_i}\left[\int_0^1 f_{\boldsymbol{w}}(x)^2 - 2f_{\boldsymbol{w}}(x)\sin(2\pi x + 2\pi\theta_i)dx\right] \\
&= \arg\min_{f_w} \mathbb{E}_{\theta_i}\left[\int_0^1 f_{\boldsymbol{w}}(x)^2 dx\right].
\end{aligned}$$

Therefore, the unique minimizer is $f_{\boldsymbol{w}}(x) = 0$ for every x in [0, 1]. Clearly, a global model does not learn anything from such a dataset. Now, assume there exists such a weight vector $\boldsymbol{w}_{zero}$ such that $f_{\boldsymbol{w}_{zero}} = 0$. Now examine the performance of personalized FL models. `Ditto`'s local objective in (12) becomes a local empirical risk plus a regularization:

$$\int_0^1 \left(f_{\boldsymbol{\beta}_i}(x) - \sin\left(2\pi x + 2\pi\theta_i\right)\right)^2 dx + \frac{\mu}{2}||\boldsymbol{\beta}_i - \boldsymbol{w}_{zero}||^2.$$

Not only no useful information about common patterns in the sine function is shared among clients, the regularization will further exacerbate the problem by forcing $\boldsymbol{\beta}_i$ close to $\boldsymbol{w}_{zero}$ which is clearly a bad point as its predicts a zero everywhere on the function. Similar phenomena can be witnessed on other regularization based train-then-personalize approaches such as `L2GD`.

In the example above, data across clients have strong commonalities as they all share the same functional form with only one parameter θ explaining their variations. Nevertheless, a train-then-personalize approach or its iterative counterparts fail due to a faulty global model. So what are potential alternative approaches ?

4) ALTERNATIVE SOLUTIONS

One approach to circumvent the need for a global model in FL is through multi-task learning (MTL) which aims to leverage commonalities across different but related outputs to improve prediction and learning accuracy [36]. In MTL, a shared representation across all tasks is built to allow the inductive transfer of knowledge [152], [339]. Here each task is a client/edge device.

An illustrative example of MTL vs. train-then-personalize is shown in Fig. 10. The main difference is that MTL directly estimates $\boldsymbol{\beta}_i$ without the need to learn a global model $\boldsymbol{w}$.

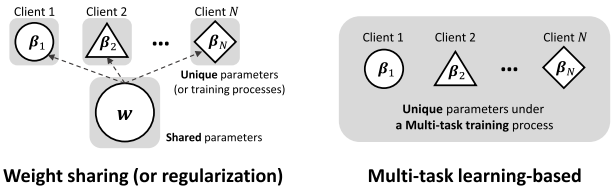

FIGURE 10. A comparison between train-then-personalize and multi-task learning approaches to personalization in IoFT.

Often, the shared representation in MTL is induced via regularization to facilitate information transfer across tasks. This can be written as

$$\min_{\boldsymbol{\beta},\Omega}\left\{\frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\beta}_i) + \mathcal{R}(\mathcal{B}, \Omega)\right\},$$

where $\boldsymbol{\beta}_i$ are the personalized weights, B is a matrix whose ith row is $\boldsymbol{\beta}_i$, $\mathcal{R}$ is a regularization term and Ω is an $N \times N$ matrix that models relationships amongst clients.

The formulation above has been studied extensively in centralized regimes [97], [136], [156], [251], [353]. Yet literature on MTL in FL is still very limited. One of such approaches is `MOCHA` [288] which defines the objective below

$$\min_{\boldsymbol{\beta},\Omega}\sum_{i=1}^{N}\ell(\boldsymbol{\beta}_i^{\mathrm{T}}x_i, y_i) + \mu_1\mathrm{Tr}\left(\mathcal{B}^{\mathrm{T}}\Omega\mathcal{B}\right) + \mu_2\left\|\mathcal{B}\right\|_F^2,$$

where, the first term defines a loss function for linear models, $\mathrm{Tr}\left(\mathcal{B}^{\mathrm{T}}\Omega\mathcal{B}\right)$ induces knowledge transfer such that negative off-diagonal entries of Ω encourages the alignment of two clients' local weights and $\|\cdot\|_F$ represents a Frobenius norm for shrinkage. Reference [288] extends the primal-dual formulation in [353] to distribute the update of $\mathcal{B}$ over clients

while keeping Ω fixed. MOCHA is shown to outperform baseline algorithms on datasets including GLEAM, Human Activity Recognition, and Vehicle Sensors.

Although the idea of MOCHA is intriguing the objective is only confined to the problems whose losses are convex, since the authors use dual algorithms to solve it and strong duality is not guaranteed for general non-convex functions. Here, future work that exploit MTL to learn a graphical model through Ω may open interesting new research directions in understanding the underlying commonality structure across clients.

It is also worth noting that other alternative routes can be taken instead of MTL. One such route is to first validate the performance of a the global model prior to personalization. Along this line, Fed-ensemble [285] proposes an alternative approach for personalization via the learned K global models described in Sec. III-B. After training a diverse group of K models, Fed-ensemble evaluates the loss of model k on a local validation dataset D_i^{val} by $\hat{F}_{k,i} = \frac{1}{n_i}\sum_{j=1}^{n_i} \ell(f_{\boldsymbol{w}_k}(x_{i,j}), y_{i,j})$. Then each model $\boldsymbol{w}_k$ is assigned a weight $\alpha_{k,i} = \exp(-\gamma \hat{F}_{k,i}) / \sum_{k=1}^{K} \exp(-\gamma \hat{F}_{k,i})$ where γ is some constant. The prediction on a new sample x^* is given by a weighted ensemble $\sum_{k=1}^{K} \alpha_{k,i} f_{\boldsymbol{w}_k}(x^*)$. The intuition is to check which models perform well on a local dataset, then assigns higher weights $\alpha_{k,i}$ to better fitted models in the ensemble.

B. SEMI-PERSONALIZED

A possible alternative to global or fully personalized modeling is semi-personalized modeling. Semi-personalized FL fits a stylized model for a group of clients. This approach balances between the need for N individualized models or one model that fits all. This is highlighted in Fig. 11. Usually these algorithms cluster clients into $G \ll N$ groups and assume the data distribution of clients inside one group is homogeneous.

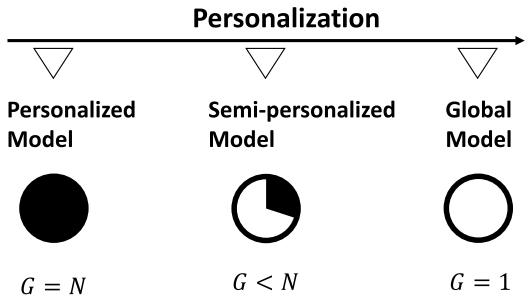

FIGURE 11. Degrees of personalization.

Reference [203] proposed an intuitive user clustering method. The number of clusters $G < N$ is predetermined. Each cluster is represented by a cluster parameter in $\{\boldsymbol{w}_1, \ldots \boldsymbol{w}_G\}$. The objective is:

$$\min_{\{\boldsymbol{w}_1,\ldots\boldsymbol{w}_G\}} \sum_{i=1}^{N} \min_{j\in 1,\cdots,G} F_i(\boldsymbol{w}_j)$$

The inner level of minimization taken over cluster index j assigns one client i to the cluster with the lowest training loss, and the outer level of minimization taken over $\boldsymbol{w}_j$'s optimizes cluster model weights. Authors propose a HypCluster algorithm to optimize the objective. In HypCluster, G is predetermined. In each communication round, all cluster weights are broadcasted to all clients, and each client chooses one with the lowest loss. Afterwards, clients train the corresponding model by SGD on their own local dataset. On EMNIST, HypCluster with $G = 2$ has higher test accuracy than FedAvg and AFL [214].

Clustered FL (CFL) [279] clusters clients dynamically. CFL measures the similarity between two clients in the following manner: suppose in communication round t, the update of clients i and j is $\Delta \boldsymbol{w}_i^t$ and $\Delta \boldsymbol{w}_j^t$, respectively. The cosine similarity is defined as $\alpha_{ij} = \frac{\left\langle \Delta \boldsymbol{w}_i^t, \Delta \boldsymbol{w}_j^t \right\rangle}{\|\Delta \boldsymbol{w}_i^t\| \left\|\Delta \boldsymbol{w}_j^t\right\|}$. For one communication round, server calculates the cosine similarity intra and across clusters. If clients in one cluster are homogeneous, the similarity of these clients should be large compared with similarity across clusters, then CFL simply performs FedAvg for this cluster. If otherwise, clients inside one cluster are heterogeneous, the similarity of these clients is low, and CFL will divide them into two subclusters. CFL then repeats the procedure on the two subclusters. As the algorithm proceeds, clients can automatically be divided into different subclusters. CFL ends when gradient norms on all clients are small and no further sub-dividing is needed.

Semi-personalized modeling and client clustering in IoFT are important questions that still require much further investigation. In the statistical perspective (Sec. VI) we pose some open questions that are critical along this research direction.

V. META-LEARNING

Meta-learning is the science of observing how different learning algorithms perform on a wide range of tasks and then learning new tasks more efficiently based on prior experience [50], [112], [217], [226], [263], [300]. Meta-learning tries to learn a global model that can quickly adapt to a new task with only a few training samples and optimization steps. This process is also known as "Learning to Learn Fast," where the goal of the model is not to perform well on all tasks in expectation, instead **to find a good initialization** that can directly adapt to a specific task. Therefore, meta-learning can be viewed as an approach to enable fast personalization and fine-tuning.

Meta-learning opens a unique opportunity to resolve many challenges in IoFT, such as scalability, fast adaptability, and improved generalization. With proper prior knowledge, a well-trained task can be rapidly generalized to new tasks with few samples. This few-shot property becomes especially crucial in IoFT where each device only has a small amount of data and does not have access to a centralized repository of all datasets.

A. FREQUENTIST PERSPECTIVE

From a frequentist perspective, the seminal work of Finn *et al.* [88] proposed one of the first model-agnostic meta-learning (MAML) algorithms. Given N different tasks $\{\mathcal{T}_i\}_{i=1}^N$, MAML optimizes a meta-loss function in the form of

$$\min_{\boldsymbol{w}} \sum_{i=1}^{N} F_{\mathcal{T}_i}(\boldsymbol{w} - \eta \nabla F_{\mathcal{T}_i}(\boldsymbol{w})). \tag{14}$$

In (14), $\boldsymbol{w}$ can be viewed as an initialization used to perform one gradient update and obtain $\boldsymbol{w}' = \boldsymbol{w} - \eta \nabla F_{\mathcal{T}_i}(\boldsymbol{w})$. MAML minimizes the objective $\sum F_{\mathcal{T}_i}(\boldsymbol{w}')$ given an initial parameter $\boldsymbol{w}$. Therefore, the goal of MAML is to find an optimal initial parameter $\boldsymbol{w}^*$ such that one gradient step on a new task can incur maximally effective behavior on that task. This idea is similar to fine-tuning in which several steps of gradients are performed given a tuned parameter $\boldsymbol{w}^*$.

The idea of meta-learning can be naturally extended to FL where each device is treated as a task and the goal is to learn an initialization for fast personalization on the edge devices. Indeed, recently researchers have tried to introduce the notion of meta-learning to FL to enable personalization and few-shot learning [173], [209]. Along this line, [39], [180], and [82] introduce MAML to FL. They reformulate the global objective of FL as (15). In the meta-learning literature, this global objective function is also known as meta-loss. In this section, we will use them interchangeably.

$$\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{w} - \eta \sum_{e=1}^{E} \nabla F_i^e(\boldsymbol{w})). \tag{15}$$

Here, $\nabla F_i^e(\boldsymbol{w})$ is the gradient after e steps of SGD. Recall that t refers to the communication round, while $e \in \{0, \cdots, E-1\}$ denotes the optimization iterates at the edge device. It is easy to see that (15) is different from the conventional objective function (1). The formulation (15) allows each user to exploit $\boldsymbol{w}$ as an initial point and update it with respect to its local data (e.g., running E steps of gradient descent). When $E = 1$, the above objective function can be simplified as

$$\frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{w} - \eta \nabla F_i^1(\boldsymbol{w})) = \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{w} - \eta \nabla F_i(\boldsymbol{w})) \tag{16}$$

and the gradient on device i can be computed as

$$\nabla F_i(\boldsymbol{w} - \eta \nabla F_i(\boldsymbol{w})) = \big(I - \eta \nabla^2 F_i(\boldsymbol{w})\big) \nabla F_i(\boldsymbol{w} - \eta \nabla F_i(\boldsymbol{w})). \tag{17}$$

Based on this formulation, [82] propose the `Per-FedAvg` algorithm to efficiently optimize (15). Given an initial parameter $\boldsymbol{w}^t$, local user i runs E steps of SGD. For simplicity, we assume $E = 1$. Therefore,

$$\boldsymbol{w}_i^t = \boldsymbol{w}^t - \eta_1 \nabla F_i(\boldsymbol{w}^t), \tag{18}$$

where η_1 is the learning rate for each local user. Afterwards, each local user calculates the gradient evaluated at $\boldsymbol{w}_i^t$ and the Hessian evaluated at $\boldsymbol{w}^t$. The user then sends the gradient $\nabla F_i(\boldsymbol{w}_i^t)$ and the Hessian $\nabla^2 F_i(\boldsymbol{w}^t)$ back to the central server. The central server aggregates them as:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta_2 \frac{1}{N} \sum_{i=1}^{N} \big(I - \eta_1 \nabla^2 F_i(\boldsymbol{w}^t)\big) \nabla F_i(\boldsymbol{w}_i^t),$$

where η_2 is a learning rate on the central server. This procedure is repeated several times till some exit conditions are met. One notable thing is that, in the central server, the gradient is evaluated at $\boldsymbol{w}_i^t$ rather than $\boldsymbol{w}^t$. This naturally arises from the meta-loss function (15). Reference [82] demonstrate the advantages of `Per-FedAvg` on image classification tasks. Interestingly, [131] interpret `FedAvg` as the linear combination of `FedSGD` and MAML when ignoring the second-order term $\eta \nabla^2 F_i(\boldsymbol{w})$ in (17). Specifically, assume each client runs E steps of local updates, then the gradient of `FedAvg` can be written as

$$\begin{aligned} & g_{FedAvg} \\ &:= \frac{1}{N} \sum_{i=1}^{N} \nabla F_i(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{E} \nabla F_i^e(\boldsymbol{w}) \\ &= \sum_{i=1}^{N} \frac{1}{N} \nabla F_i^1(\boldsymbol{w}) + \sum_{i=1}^{N} \frac{1}{N} \sum_{e=2}^{E} \nabla F_i^e(\boldsymbol{w}) \\ &= \sum_{i=1}^{N} \frac{1}{N} \nabla F_i^1(\boldsymbol{w}) + \sum_{i=1}^{N} \frac{1}{N} \sum_{e=2}^{E} \nabla F_i(\boldsymbol{w} - \eta \sum_{j=1}^{e-1} \nabla F_i^j(\boldsymbol{w})) \\ &= g_{\texttt{FedSGD}} + \sum_{e=2}^{E} g_{MAML}(e-1), \end{aligned} \tag{19}$$

where $g_{\texttt{FedSGD}}$ is the gradient of `FedSGD` and $g_{MAML}(e)$ is the gradient of MAML with e steps of local updates. Though the optimization of this linear combination in (19) is not strictly equivalent to `FedAvg` (due to the second-order term), this interpretation sheds light on the intrinsic connection between FL and meta-learning.

Based on this observation, [131] slightly modify the `Per-FedAvg` algorithm as follows: first run `FedAvg` (or another conventional FL algorithm) at the early stage of training and then switch to MAML (in (15)) or Reptile [225] to fine-tune the model. Through many empirical results, the authors argue that this combined strategy ensures fast and stable convergence compared to directly optimizing the federated MAML objective in (15). Besides, this paper also delivers an important message: no single FL is a panacea for all problems, instead, different dataset requires different inference strategies or a combination of them.

There are also few other variants of meta-learning-algorithms that can be readily applied to FL. For instance, `MetaSGD` [39], [176] specifies a coordinate-wise learning rate $\boldsymbol{\eta}$. Instead of constraining the learning rate to be a fixed positive scalar, they define $\boldsymbol{\eta}$ as a vector that is of the same size as $\boldsymbol{w}$ and each element in it can be positive, negative or zero. Different from the traditional definition of a learning rate,

$\boldsymbol{\eta}$ encodes both the update direction and rate. To understand the intuition behind $\boldsymbol{\eta}$, it is very helpful to first see the MetaSGD procedure in detail. In MetaSGD, both $\boldsymbol{w}$ and $\boldsymbol{\eta}$ are treated as model parameters to optimize. Specifically, at communication round t, MetaSGD first samples $\mathcal{S} \subseteq [N]$ clients and divides the data for each client into a training set D^{train} and validation set D^{val}. Here, MetaSGD then runs one step of SGD using $\boldsymbol{\eta}^t$ and $\boldsymbol{w}^t$ on each sampled client to obtain updated parameters $\{\boldsymbol{w}_i^t\}_{i=1}^N$:

$$\boldsymbol{w}_i^t \leftarrow \boldsymbol{w}^t - \boldsymbol{\eta}^t \circ \nabla F_i(\boldsymbol{w}^t)$$

where $\circ$ is an element-wise product operation and $F_i(\boldsymbol{w}^t)$ is the loss function evaluated on the training set at communication round t. Afterward, MetaSGD updates all model parameters as

$$(\boldsymbol{w}^{t+1}, \boldsymbol{\eta}^{t+1}) \leftarrow (\boldsymbol{w}^t, \boldsymbol{\eta}^t) - \eta_{\texttt{MetaSGD}} \frac{1}{N} \sum_{i=1}^{N} \nabla F_i^{\text{val}}(\boldsymbol{w}_i^t)$$

where $\eta_{\texttt{MetaSGD}}$ is a scalar learning rate and $\nabla F_i^{\text{val}}(\boldsymbol{w}_i^t)$ is the local gradient evaluated on the validation set D^{val}. Note that $\boldsymbol{w}_i^t$ is a function of both $\boldsymbol{w}^t$ and $\boldsymbol{\eta}^t$ since $\boldsymbol{w}_i^t \leftarrow \boldsymbol{w}^t - \boldsymbol{\eta}^t \circ \nabla F_i(\boldsymbol{w}^t)$. Therefore, the gradient of $F_i(\boldsymbol{w}_i^t)$ can be taken with respect to $\boldsymbol{w}^t$ and $\boldsymbol{\eta}^t$. From this algorithm, one can see that $\boldsymbol{\eta}$ is learned from all tasks to avoid possible model over-fitting on a specific task. The intuition is that the gradient $\nabla F_i(\boldsymbol{w}_i)$ on the training set can potentially lead to over-fitting, especially when the sample size is small. Therefore, $\boldsymbol{\eta}$ acts as a regularization role to control the sign and magnitude of $\nabla F_i(\boldsymbol{w}_i)$.

B. BAYESIAN PERSPECTIVE

Besides the frequentist perspective on meta-learning, there are recent yet few efforts to explore Bayesian meta-learning. Below we discuss some recent advances. We note that the works discussed below focus on meta-learning and not FL. However as shown earlier those concepts are naturally related. We will also provide examples on how to extend the models to FL.

Bayesian meta-learning simply defines $\boldsymbol{w}$ as a random variable and takes a Bayesian route to estimate its posterior. This posterior then serves as a ''prior'' for a new task. Such an approach would allow uncertainty quantification for predictions on both the central server and local clients [308]. Notably, [334] propose a Bayesian counterpart of MAML (BMAML) for fast adaption and uncertainty quantification.

Instead of finding a single parameter to minimize (15), BMAML aims at finding a posterior distribution of the parameter such that one can quantify uncertainties. To achieve so, the gradient method used in MAML is replaced by one of its Bayesian counterparts - Stein variational gradient descent (SVGD) [184]. The SVGD method combines the strengths of SGD and Markov chain Monte Carlo (MCMC) such that one can sample from a posterior distribution to quantify uncertainties. As described in [184], SVGD maintains M instances of model parameters, called particles. Those particles can be viewed as samples from the posterior distribution of the model parameter.

Here we detail the BMAML algorithm. At the global optimization iterate t, which is equivalent to the communication round in the FL, BMAML starts with M initial particles $\{\boldsymbol{w}_m^t\}_{m=1}^M$ that are sent to the clients. Each client then applies E steps of SVGD to obtain updated parameters as shown below:

$$\{\boldsymbol{w}_{i,m}^{t,E}\}_{m=1}^M = SVGD(\{\boldsymbol{w}_m^t\}_{m=1}^M; D_i, \eta),$$

where η and $SVGD(\{\boldsymbol{w}_{i,m}^t\}_{m=1}^M; D, \eta)$ is the SVGD algorithm that aims to collect samples from the posterior $\mathbb{P}(\boldsymbol{w}|D_i)$ during local training.

Afterwards, the updated task-dependent parameters $\{\boldsymbol{w}_{i,m}^{t,E}\}_{i\in[N],m\in[M]}$ are used to calculate the gradient of the meta-loss $\sum_i \nabla F_i(\{\boldsymbol{w}_{i,m}^{t,E}\}_{m=1}^M)$ and perform a one step update

$$\{\boldsymbol{w}_m^{t+1}\}_{m=1}^M \leftarrow \{\boldsymbol{w}_m^t\}_{m=1}^M - \eta \frac{1}{N} \sum_i \nabla F_i(\{\boldsymbol{w}_{i,m}^{t,E}\}_{m=1}^M).$$

Recall that the gradient of the meta-loss is evaluated at the updated particles $\{\boldsymbol{w}_{i,m}^{t,E}\}_{m=1}^M$ and the global optimization is performed over $\{\boldsymbol{w}_m^t\}_{m=1}^M$. The overall idea of BMAML is very similar to MAML (Eq. (15)): the goal is to train a model that can quickly adapt to a new task.

Besides BMAML and its variants, there are also many other studies that formulate meta-learning from a Bayesian perspective [17], [78], [89], [101], [104], [158], [159], [222], [242], [262], [270], [298], [334], [361]. Most notably, [298] and [241] consider applied deep kernel methods to learn complex task distributions in a few-shot learning setting. References [159] and [324] introduce deep parameter generators that capture a wide range of parameter distributions. However, this method is not amenable to first-order stochastic optimization methods and therefore may scale poorly.

The works discussed above focus on meta-learning and not FL. However as shown earlier those concepts are naturally related. To give an example, in the Bayesian MAML, each local user i can be treated as a task i. During training, each local user performs SVGD to obtain M updated weight parameters $\{\boldsymbol{w}_{i,m}^{t,E}\}_{m=1}^M$. The local user i then sends the gradient of local loss evaluated at $\{\boldsymbol{w}_{i,m}^{t,E}\}_{m=1}^M$ to the central server. The central server collects information from all users and updates the meta-loss function to obtain $\{\boldsymbol{w}_m^{t+1}\}_{m=1}^M$. At the end of each communication round, the central server broadcasts $\{\boldsymbol{w}_m^{t+1}\}_{m=1}^M$ to all devices. This cycle is repeated until some exit conditions are met.

Interestingly, there is also a trend that formulates meta-learning from a stochastic process perspective [94], [141], [189], [194]. Most notably, [95] introduce neural processes (NPs) that combine advantages of neural networks and Gaussian processes (GPs). Fig. 12 illustrates the idea behind NPs. Given a point (x_i, y_i), the algorithm first defines a representation function $\Phi_i = h((x_i, y_i))$ that maps inputs into a feature space. Here Φ is a NN. Then it defines a latent distribution over the feature representation. Specifically,

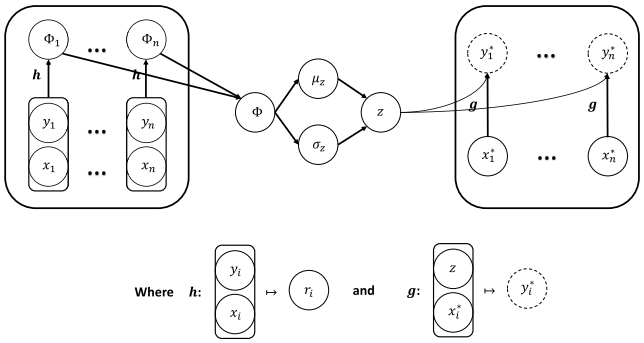

FIGURE 12. The diagram of neural processes.

$z \sim \mathcal{N}(\mu_z(\Phi), I\sigma_z(\Phi))$ where $\Phi = a(\{\Phi_i\}_{i=1}^N) = \frac{1}{n}\sum_i \Phi_i$ and a is called the aggregator. In turn this latent distribution mimics a GP as it defines a prior over Φ [317]. Now, given this latent distribution a conditional decoder g, learned through variational inference, takes inputs sampled from z and testing data x^* to make predictions y^*. One key feature of NP is that it encodes input data into a single order-invariant global representation. This representation captures the global uncertainty which allows sampling at a global level. The similarity to meta-learning in NPs is the fact that we are learning a prior over the latent representation. This prior then acts as an starting point for predictions at a new point.

To extend NP to a federated framework, one can develop some aggregation strategies that allow N edge devices to collaboratively learn the global feature z. For example, all edges devices can use `FedAvg` to train a global encoder h. This shared encoder is then used to calculate the global latent distribution z.

As shown above, various meta-learning approaches may be readily extended to FL. We hope this review will help inspire continued exploration along this direction as we envision that meta-learning will have a great impact in IoFT. Further, exploring and contrasting of meta-learning models in FL may help guide practitioners chose amongst existing methods given the data properties and features.

VI. STATISTICAL AND OPTIMIZATION PERSPECTIVE

A. STATISTICAL PERSPECTIVE

Much of the current work on FL within IoFT has focused on algorithm development, and thus far, there has been little in the context of developing statistical models. In this subsection, we discuss challenges and interesting open directions for FL from a statistical perspective. In particular, several statistical challenges such as heterogeneity, dependence, and sample bias under privacy and communication constraints need to be addressed. Part of the challenge is developing a suitable modeling framework that allows the assessment and validation of methods handling the challenges above.

Below, we discuss areas where statistics can make a significant contribution. This is by no means an exhaustive list as IoFT is still in its infancy phase, and new challenges will arise as IoFT infiltrates new applications.

1) DEPENDENCE (BEYOND EMPIRICAL RISK MINIMIZATION)

Statistical dependence in IoFT is a common challenge since clients may be dependent (due to geographic or spatial dependence, common features, etc.) or data within each client may be correlated. Current FL algorithms operate under an empirical risk minimization (ERM) framework and assume independence both within and across clients. In correlated settings, even classic SGD will lead to a biased estimator of the gradients [50] as the loss function cannot be simply summed over all data points. Adding to that, correlation needs to be learned with additional challenges posed by communication and privacy constraints in IoFT.

Yet here lies a significant opportunity: if learned effectively, a dependence structure across clients can be exploited to improve prediction, update sampling schemes and better allocate resources. More specifically, statistical approaches for dealing with dependence typically involve learning a suitable dependence structure (e.g. graphical model) amongst the clients of interest and then exploiting the structure (e.g., [146], [245], [261]) for inference and prediction. Following this line of thinking, the challenges are to (i) develop an FL approach to learn a suitable dependence structure amongst the clients; (ii) exploit the learned dependence structure for improved inference and prediction within FL; (iii) develop an FL approach to deal with correlated data points within each client. While there are numerous techniques to address (i)-(iii) in the standard centralized setting (e.g., [181], [240], [264]), their applications to IoFT are yet to be explored.

a: NETWORK LEARNING

Learning networks (especially through graphical modeling) is a problem that has received significant attention and research in the statistics and machine learning literature [181], [264], [338]. In IoFT, when there is dependence amongst the clients, learning a graphical model/network structure potentially improves overall performance. However, there remains the open challenge of adapting and implementing these graphical modeling algorithms that learn pairwise sufficient statistics to respect communication and differential privacy constraints. At the heart of the challenge, if our goal is to learn a network structure amongst nodes, second-order statistics are required in the computation. From a privacy and communication perspective, this requires communication between all pairs of nodes in order to compute these second-order sufficient statistics. One possible solution is to carry over ideas from differential privacy to network learning problems. Also, statistical ideas such as sketching or randomization [73], [74], [200], [258]) may improve privacy whilst still learning sufficient statistics.

b: CORRELATION

Handling correlated data within each client is also an essential challenge as it goes beyond ERM. Here stochastic processes, such as Lévy processes and auto-regressive models, can play

an important role. One example is multivariate Gaussian process (GPs) models, where a covariance matrix encodes dependence both within and across clients. Yet again, privacy and communication considerations in IoFT yield decentralized estimation of a covariance matrix, a challenging task. Here, one may assume that the covariance matrix/kernel is parameterized by a small set of parameters (e.g. Mattern, RBF). Very recent work has shown that despite biased gradients, SGD can still converge in correlated settings, specifically GPs [41]. A natural question to think about is whether a natural extension exists for FL.

c: VALIDATING DEPENDENCE MODELS

The final challenge associated with network learning for IoFT is to address whether the learned network improves predictive power, and if so, which approaches are best. For example, would performing network learning improve predictive performance compared to clustering nodes and doing personalization? Since the ultimate goal is predictive power (although the network may contain helpful information), this presents a natural validation metric for network learning methods. Due to the modeling framework, predictive power can be validated theoretically using refined bounds that incorporate dependence, simulation studies and real data examples.

2) UNCERTAINTY QUANTIFICATION & BAYESIAN METHODS

As seen in Secs. III-V, very few approaches are able to quantify uncertainty. Besides, `Fed-ensemble` and `Fed-BE`, FL methods have mainly focused on point predictions. Yet a model should acknowledge the confidence in its prediction. Therefore, further exploration into Bayesian methods is important for the application of IoFT within different domains.

One possible route is to place a prior over $\boldsymbol{w}$ (and possibly personalized weights β_i) $\mathbb{P}(\boldsymbol{w}, \boldsymbol{\beta}_i)$ and try to estimate the posterior $\mathbb{P}(\boldsymbol{w}, \boldsymbol{\beta}_i | D = \{D_1, \cdots, D_N\})$. Clearly, if $f_{\boldsymbol{w}}(x)$ is a complex function such as a neural network, $\mathbb{P}(\boldsymbol{w}, \boldsymbol{\beta}_i | D)$ is usually intractable, yet one may hope to extract some samples from it. Here recent advances in approximate posterior sampling such as Stochastic Gradient Langevin Dynamics (SGLD) [315] or Stein Variational Gradient Sescent (SVGD) [184] may be possible solution techniques. Besides the approximate sampling schemes above, hierarchical Bayes may be worth investigating, as IoFT has an inherent hierarchy between the orchestrator and clients. The challenge to be addressed here is how to estimate a hierarchical Bayes model in a decentralized fashion that preserves edge privacy and minimizes communication.

As an alternative to a prior on model parameters, one can take a functional route by directly specifying a prior on the functional space $f_{\boldsymbol{w}}(x)$; commonly done using Gaussian processes (GP). Indeed, GPs have a long history of success in engineering applications [103], [132], [250], [313] and their success in IoFT may pave the way for many new applications. However, the challenge is that GPs are based on correlations and do not conform to the ERM paradigm FL is currently based on. *Interestingly, learning a prior on the functional space may also help in personalization and meta-learning. A learned GP innately acts as a prior through which predictions are obtained by conditioning on new data.*

3) STATISTICAL HETEROGENEITY & PERSONALIZATION

In IoFT, statistical heterogeneity is a central challenge as individual devices may have different data patterns and potentially collect different amounts and types of data. As highlighted in Secs. IV and V, personalization (and meta-learning) are one way to overcome the heterogeneity challenge by allowing clients to retain their individualized models while still borrowing strength from each other. However, personalization poses many exciting challenges and open questions. Below we list a few.

i. It is essential to investigate when a personalized model is needed and understand the trade-off between a global model and device-specific features. Intuitively, when there is significant heterogeneity among clients, fitting personalized models would be better than using a global model. Also, as pointed in [151], when data is highly heterogeneous, negative transfer of knowledge may occur where each client can generate better models using their data in isolation compared to sharing knowledge with other clients. Reference [178] conducted some analysis on additive models that matched this intuition. Yet, literature on approaches that characterize the heterogeneity of data and decide on whether personalization is needed is largely missing.

ii. Following the idea above, one can also decide on how many personalized models to build. As described in Sec. IV-B, this entails inferring the number of clients clusters where data within each cluster is homogeneous. A key question remains: how to cluster clients in IoFT? Clients usually send back a set of weights or gradients, however, are these summary statistics sufficient to recover true client clusters? If not, what sufficient statistics can achieve such a goal, and do they guarantee privacy? Here one may pose a question: what client statistics are needed such that clustering (say using K-means) using all data (D) in a centralized regime and clustering using the client statistics in IoFT will yield similar outcomes.

iii. Personalization may come at the price of privacy. Reference [357] shows that input images can be reconstructed from unperturbed gradient signals, which opens the possibility of gradient attacks. Precautions like adding noise or quantization can somewhat reduce the risk. However, some experiments [312] have shown that there exists trade-offs between privacy and performance. Therefore, it is essential to understand this trade-off better and propose methods that can improve both simultaneously. Differential privacy may be a valuable tool along this line [312].

iv. A probabilistic approach to personalization is still needed. Ideas such as random effects have built the

statistical foundation for incorporating unit-to-unit heterogeneity. They may be of great value if extended to the decentralized IoFT settings.

4) VALIDATION AND HYPOTHESIS TESTING

To this point, there has been little work in FL on suitable statistical validation and hypothesis testing procedures. While there may be a general reluctance to impose statistical models since they are never truly correct and there are advantages to being model-agnostic; imposing statistical models provides several potential benefits. First and foremost, modeling provides a way to develop and assess a hypothesis testing approach.

Most, if not all, prior work on FL has focused on deep learning algorithms due to their predictive power. If we are interested in questions associated with statistical estimation and inference, it makes sense to extend FL to incorporate models that are interpretable in addition to having good predictive capabilities. Algorithms that come to mind include kernel methods (see e.g. [13], [148], [259], [260]), Gaussian processes (see e.g. [102], [151], [261], [340]) and other approaches. Like deep neural networks, all of these approaches exhibit non-linearities and function complexity while being more amenable to statistical inference.

One example of a statistical model was given in Sec. IV where one can model the conditional distribution as:

$$y_i \sim \mathbb{P}^i_{y|x}\big(f_{\boldsymbol{w}_i}(x_i)\big),$$

Here, clients share the same f (a linear model, kernel, Gaussian process, neural network) yet with different parameters $\boldsymbol{w}_i$ which allows for personalization. This represents N semi-parametric models [24] and a key question is what structure to impose on the $\boldsymbol{w}_i$'s which incorporate the degree of statistical heterogeneity and dependence? Assumptions such as graphical models, low-rank models, sparse models and many others may be incorporated (e.g., [32], [110]). Graphical models naturally lend themselves to network learning while low-rank models naturally lend themselves to clustering of nodes.

5) OTHER OPEN QUESTIONS

a: DOMAIN ADAPTATION

The joint distribution of data pairs for client i is given as $\mathbb{P}^i_{x,y} = \mathbb{P}_x\mathbb{P}^i_{y|x}$. Current models assume that $x \sim \mathbb{P}_x$ across clients yet there is a change in the conditional distribution $\mathbb{P}^i_{y|x}$, i.e. input-output relationship across clients. What happens if there is a covariate shift, $\mathbb{P}^i_x$, across clients. Indeed, this is not uncommon in IoFT as different clients may observe a wide range of unseen input on other clients (e.g. unique defect types of failure modes). Here domain adaptation may play a key role where the input is first mapped to a shared feature space, and then inference is made on the feature embeddings. A simple example is:

$$f_{\boldsymbol{\beta}_i,\boldsymbol{w}}(x) = g_{\boldsymbol{w}} \circ \Phi_{\boldsymbol{\beta}_i}(x),$$

where Φ is a personalized encoder that projects to the feature space and g is a global decoder with shared parameters $\boldsymbol{w}$ across all clients.

b: DECENTRALIZED & COLLABORATIVE DESIGN OF EXPERIMENTS (DOE)

DOE is critical within many domains [37], [49], [135], [322], [325]. In the realm of IoFT, a key question is how can DOE be achieved? For instance, for expensive experiments or computer models, DOE often uses a sequential strategy to find the next-to-sample design points that best help in estimating an unknown response surface [120] or providing statistical inference. In IoFT, such an expensive computer experiment may reside on the central server or perhaps each client has its own computer model. To this end, how can decentralized DOE be achieved? How can sequential designs learn a global computer model, given that clients may be of different fidelities [122]? How can computer models borrow strength from each other for better calibration [249] while preserving privacy? How can we distribute the experimental design process across clients given resource limitations of each client? Such questions will be of key importance in IoFT and open a new area of exploration for DOE. These same questions extend to Bayesian optimization which also aims at optimal sequential sampling, yet with the goal of finding optima of an unknown response surface.

c: VERTICAL IoFT

Current literature is mainly focused on horizontal IoFT where the edge denotes different clients. Yet more exploration should be oriented towards vertical IoFT, where we have the same client (ex: patient), yet information is stored across different system components (ex: hospitals). Here, feature extraction shall play a critical role, where features from different components are extracted and then jointly trained for a holistic model. Techniques such as sketching or random embeddings may be of use to preserve data privacy.

B. OPTIMIZATION PERSPECTIVE

Several optimization algorithms have been proposed in recent years to learn a global or personalized model collaboratively within IoFT; see Secs. III-V for details. However, significant theoretical and computational hurdles associated with solving such problems remain unresolved. In this section, we discuss FL from an optimization perspective. More specifically, we categorize the main existing streams of work and provide insights on potential open directions.

Several algorithms have been recently proposed to mitigate the issue of heterogeneity. A class of these algorithms add a local regularization term to the client's objectives. Popular examples of such algorithms include `FedProx` [170], `FedDyn` [3], `DANE` [283] and its federated counterpart `FedDANE` [171], `SCAFFOLD` [137], `FedPD` [352]. While existing methods tackle heterogeneity by adding a regularization term, it is worth exploring algorithms that add adaptive ball constraints when minimizing local objectives.

Such methods can control the variability of local parameters and can align the stationary solutions of local and global objectives when the radius of the ball constraints converges to zero in the limit.

The convergence guarantees and complexity rates of many of these algorithms were established under a variety of assumptions; see [133], [137], [352]. When the clients are identical, FedAvg coincides with FedSGD [206], [359] (also known as local SGD) and both converge asymptotically. In the (strongly)-convex case with *i.i.d* data, FedAvg converges to a global solution with optimal complexity order. Despite its success in many applications, the former algorithm suffers when applied to FL settings with non-*i.i.d* data. More specifically, in the presence of heterogeneity, the discrepancy between the average local client optimum and the global optimum results in a drift in local updates, often called client drift. This drift can significantly affect the convergence guarantees and complexity rate of the algorithm. In particular, [352] provides a problem instance for which FedAvg with constant step-size diverges to infinity. Reference [137] show that the client-drift effect in unavoidable even if we use full batch gradients and all clients participate in each communication round.

To mitigate the issue of heterogeneity, FedProx uses a proximal term that suppresses the variance among local client solutions. The method is shown to converge without any boundedness assumptions on the local gradients. Despite more stable convergence, the method is still based on inexact minimization since it does not align local and global stationary solutions. When all clients have the same optima, and full batch gradients are used, FedAvg and FedProx have the same complexity rate. Other algorithms that directly tackle client-drift in FL appear in the work of [137], [352], and [3]. These algorithms are shown to converge without any bounded client gradient assumptions. Among them, FedDyn requires fewer communication rounds between the central orchestrator and clients compared to SCAFFLOD.

Another FL method class uses an adaptive choice of step size for the client and server optimizer steps. This approach is motivated by the practical benefits that repeatedly appeared in adaptive optimization when training machine learning models. Examples of FL-versions of such algorithms are FedAdam and FedYogi [266]. We refer the readers to Sec. III-C for a more breadth and depth discussion on the algorithms presented above.

1) OPEN DIRECTIONS

Despite the focus on algorithm development and their corresponding optimization mechanisms, there are potential optimization questions still to be explored for FL.

a: CHOICE OF THE NUMBER OF LOCAL STEPS E

One popular example of algorithms proposed for FL is FedSGD, a distributed version of SGD. While utilizing parallel computations yields efficient training for large datasets, such methods incur high communication cost since they require passing the gradient vector of clients to the server at every iteration. To remedy the high communication cost, FedAvg was proposed. As detailed in previous sections, this method applies multiple local SGD steps in each communication round before updating the global parameter at the server. Despite being widely used in practice, several recent results have shown degrading performance in the presence of heterogeneity [355]. One particular issue is that client heterogeneity can introduce a wide discrepancy between local and global objectives, resulting in a drift in local updates. While a higher number of local updates E reduces the communication cost, it can magnify this client drift. Similarly, a low E directly implies a high communication cost. Hence, the choice of E presents a trade-off between convergence stability and communication cost. Some interesting open questions worth investigating are i) How should we choose the number of local steps E? ii) In the presence of heterogeneity, can we choose a different E_i across clients? iii) Can we utilize an underlying client heterogeneity structure to decide on E_i?

b: HIGHER ORDER ALGORITHMS

Almost all existing FL algorithms belong to the class of first-order methods. Designing second-order methods tailored for FL remains an exciting area to explore. This class of algorithms can potentially perform better than first-order algorithms when the objective is highly non-linear or ill-conditioned. Such methods aim at effectively using the curvature information for faster convergence. To overcome the computational drawback of computing the Hessian, estimating the curvature using first-order information is well-studied in quasi-newton methods, as well as their stochastic variants [34]. Motivated by their potential superior performance in several applications and under special structural properties of the objective, a natural potential research direction is investigating FL variants of second-order methods.

c: ZEROTH-ORDER ALGORITHMS

These methods utilize a heuristic derivative-free approach for updating the sequence of optimization iterates. Such methods can be helpful in problems with access to only noisy evaluations of the objective function [59], [268]. Several recently arising machine learning applications [48], [273] have brought significant attention to zeroth-order algorithms. Studying FL variants of zeroth-order algorithms is a potential research direction that can pave the path for interesting FL applications.

d: MIN-MAX OPTIMIZATION

Developing new methods for solving min-max optimization problems in FL settings is still premature and is worth investigating. Min-max optimization problems have recently appeared in a wide range of applications, including adversarial training of Neural Network, fair inference [15], [172], [214], training GANS [98], and many others. A wide-variety of algorithms have been proposed to solve these problems in non-FL settings. Most commonly, stochastic gradient

descent-ascent (SGDA) that applies an ascent step followed by a descent step at every iteration is used in practice. Applying SGDA and its variations is undesirable in FL since it requires communication at each iteration. A natural research direction is to investigate FL variants of such methods. One potential direction that can be explored when the maximization problem is (strongly)-concave is using duality theory to minimize the model parameters and the dual variables jointly.

e: RESOURCE ALLOCATION & CONSTRAINED OPTIMIZATION

As mentioned in Sec. II, we do not cover approaches for resource allocation in IoFT since literature in this area is scarce. However, in IoFT, clients themselves are heterogeneous in their capabilities in computation, memory, processing power, connectivity, amongst others. Modeling approaches should account for edge resources while at the same time ensuring uniform or comparable model performance across clients regardless of their capabilities. Resource allocation for optimal trade-offs between convergence rates, accuracy, energy consumption, latency, and communications are of high future relevance.

Along this line, client resources may be posed as constraints in the local and global objectives. However, constrained optimization is still to be investigated in IoFT. While deep learning models usually do not place constraints on the parameters, we envision that constrained optimization will arise within domain-specific applications. Similar and unique constraints across clients pose interesting questions on redefining local and global updates and understanding theoretical guarantees in such settings.

f: FULL DECENTRALIZATION

At the current stage, most developed methods for IoFT rely on a central orchestrator. Full decentralization is a step forward, where clients collaborate directly with each other without the guidance of an orchestrator (see Fig. 5). This may add an additional layer of privacy as it is difficult to observe the system's full state. With the increased penetration of blockchain applications, IoFT may become fully decentralized. Yet, many of the FL techniques described in this paper will need to be re-thought to make this possible. Also, besides statistical and optimization challenges, there lies fundamental challenges of trust and privacy as malicious clients may corrupt the network and violate privacy without a central authority taking corrective action. Consequently, a level of trust in a central authority in a peer-to-peer network can benefit in regulating the network's protocols.

VII. APPLICATIONS

In the previous sections, we have discussed defining features of IoFT and data-driven modeling approaches for decentralized inference. Yet, IoFT both shapes and is shaped by the application it encompasses. This boils down to a crucial question: how will IoFT shape different industries, and what domain-specific challenges it faces to become the standard practice? Through the lens of domain experts, we shed the light on the following sectors: **manufacturing**, **transportation**, **energy**, **quality & reliability**, **computing**, **healthcare**, and **business**. To keep notation simple, this section slightly exploits earlier notation.

A. MANUFACTURING

The fourth industrial revolution (Industry 4.0), which is undergirded by smart technologies like IoT, has brought disruptive impacts on the manufacturing industry [19], [166]. In the United States alone, 86 percent of manufacturers believe that smart factories built on Industry 4.0 will be the primary driver of competition by 2025. Furthermore, 83 percent believe that smart factories will transform the way products are made [314]. However, only five percent of US manufacturers surveyed in a recent study reported the full conversion of at least one factory to ''smart'' status, with another 30 percent reporting they are currently implementing initiatives related to smart factories [157]. This means that nearly two out of three (65 percent) manufacturers surveyed report no progress on initiatives that they overwhelmingly point to as their primary driver of near-term competitiveness in five years [157].

Distrust is listed as one of the dominant factors inhibiting the spread of Industry 4.0 [218]. The current paradigm of IoT where data is agglomerated in a central server does not foster trust. Instead, it breeds concerns about privacy and security [31]. Also, several time-sensitive applications could be advanced by Industry 4.0 but are inhibited by the current IoT paradigm. For example, through cloud computing, manufacturing machines could benefit from advanced control algorithms that significantly improve their performance [234]. However, with an IoT infrastructure reliant on the exchange of data with a centralized server, internet latency becomes a significant challenge [232]. Moving large amounts of data to and from a central server also demands high internet bandwidth. Another major challenge of the current IoT landscape is that it is poised to benefit large enterprises at the expense of small and medium-sized enterprises. Given the concerns around privacy and security, companies are inclined to use private rather than public cloud infrastructures [111]. Therefore, smaller companies are unlikely to have the capital to set up and maintain their own private cloud infrastructure. Even if they can set one up, they are unlikely to generate sufficient data volumes for meaningful big data analytics.

IoFT could help overcome the aforementioned challenges and create lots of new opportunities in present-day manufacturing. For example, it could enable vertical integration of IoT across a manufacturing ecosystem, which is key to capturing value from Industry 4.0 [218]. The ability for entities to keep their data private while collaborating on a shared model could allow original equipment manufacturers (OEMs), for instance, to integrate their data analytics with those of their suppliers to help improve quality across their entire supply chain. This benefits the OEMs as well as their suppliers. Similarly, developing a shared model without compromising

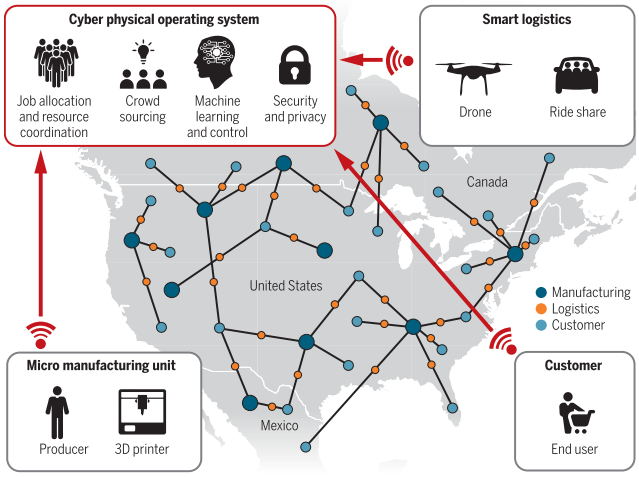

FIGURE 13. Cyber-physical operating system connecting and coordinating customers, micro-manufacturing units and smart logistics to enable massively distributed manufacturing. Reprinted from C Okwudire and H Madhyastha, Science 372, 341 (2021); Graphic: C. Bickel.

privacy could help level the playing field between large and small enterprises. Small companies who cannot afford a private cloud infrastructure can benefit from public cloud infrastructures without sharing their data. Moreover, even if they do not have large enough datasets for analytics, they can benefit from the data of other entities through a shared model. Furthermore, data analytics for time-sensitive applications can be run at the edge [216], closer to the device or machine, to reduce latency while also benefiting from a shared cloud-based model across several machines [197]. The same benefit extends to bandwidth-intensive applications.

IoFT will also be a key enabler of futuristic paradigms like massively distributed manufacturing (MDM), briefly described in Sec. 1. MDM involves the manufacture of products by a large, diverse, and geographically dispersed but coordinated network of individuals and organizations with agility and flexibility, but at near-mass-production quality, productivity, and cost-effectiveness [233]. A cyber-physical operating system (CPOS), which intelligently, efficiently, and securely coordinates large networks of cloud-connected, autonomous, and geographically-dispersed manufacturing resources will be needed to support MDM. The importance of operating systems to support present-day distributed manufacturing has been highlighted in recent works, together with ideas on how to realize them [62], [93]. However, in the context of MDM, some distinguishing features of CPOS are that: it will optimally allocate manufacturing jobs to the resources connected to it and leverage distributed and democratized delivery systems, like Uber/Lyft and drones, for logistics. It will apply machine learning to the data gathered from sensors to help assure and improve quality and optimize operations. Furthermore, it will leverage the ingenuity of humans via crowdsourcing of ideas to improve manufacturing operations across networks of manufacturers. It will leverage cybersecurity measures to protect the intellectual property and privacy of participants. CPOS will thus allow the collaboration of large, autonomous, heterogeneous, and geographically dispersed networks of manufacturers to rapidly respond to production demands and disruptions with agility and flexibility while ensuring high quality, productivity, and cost-effectiveness [233].

IoFT will allow CPOS to maintain the autonomy, privacy, and security of all the participants in MDM while enabling them to develop shared models that improve quality (and other performance metrics) across the entire system. MDM, enabled by CPOS and IoFT, promises to improve the responsiveness and resilience of manufacturing to urgent production demands (e.g., in emergencies like pandemics); promote mass customization and cost-effective low-volume production; gainfully employ lots of ordinary citizens in manufacturing (e.g., through the gig economy); and reduce the environmental footprint of manufacturing, by producing items closer to their points of use.

In a nutshell, Industry 4.0 is poised to transform the manufacturing industry, but it would need a transition from traditional IoT to IoFT for its promise to fully materialize. IoFT will help alleviate issues around privacy, security, cost, data scarcity, communication latency, and bandwidths that are slowing down the adoption of IoT solutions in the manufacturing industry. It will also help catalyze new paradigms of manufacturing, for example, massively distributed manufacturing. To facilitate the transition of the manufacturing industry from traditional IoT to IoFT, the challenges discussed in Sec.II have to be addressed in the context of manufacturing.

B. TRANSPORTATION

The prevalence of smart personal devices and the emergence of connected vehicle technology provide a plethora of opportunities for vehicles, travelers, and the transportation infrastructure to be in constant communication. This connectivity promises a safer and more sustainable transportation system with enhanced levels of mobility and accessibility. Connectivity allows subsystems that were previously modeled and optimized separately to be modeled as a single system, thereby capturing their interactions. This comprehensive modeling approach allows for increasing system throughput, which results in many benefits for travelers (e.g., lower prices, less congestion, more reliable travel times, lower levels of greenhouse gas emissions) and the transportation system (less pressure on the infrastructure). Take the example of traffic signal control systems. Traffic signal controllers are traditionally optimized locally, either per intersection or set intersections within an arterial. This optimization is based on local information: as vehicles approach an intersection, they activate loop detectors deployed in the pavement, sending a signal to road-side controllers. The controller then optimizes the traffic signal to minimize total delay. In an arterial setting, the optimization of the controllers at downstream intersections could be further informed by the state of the upstream intersections. Connected vehicle (CV) technology provides two unique opportunities for traffic signal control:

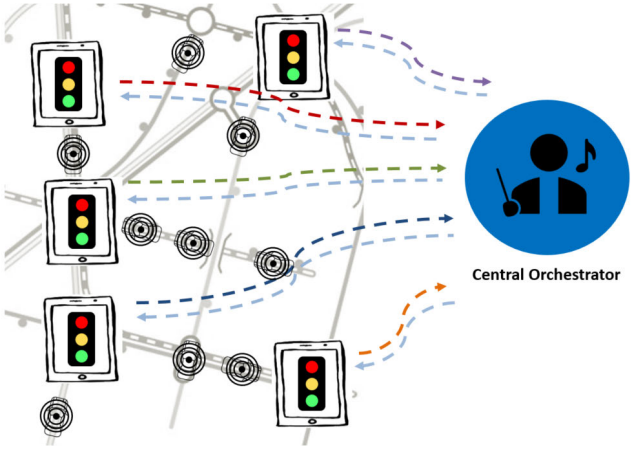

FIGURE 14. Network-level intersection control.

(1) controllers can be optimized proactively before vehicles arrive at intersections using the messages received from connected vehicles; and (2) arterial-level optimization can be advanced to network-level optimization by customizing basic safety messages (BSMs) transmitted by vehicles to include route-choice information or estimating this information based on standard BSMs ([4], [213], [311]).

Despite the benefits that connectivity can offer, the existing methodologies are generally not scalable to allow for fast decision-making in connected systems. This lack of scalability creates a critical bottleneck in leveraging connectivity in transportation applications, especially since the state of the environment in transportation systems changes dynamically. Yet here lies a critical opportunity for transportation systems: with more compute power on edge devices (ex: AI chips in autonomous cars), we may can exploit these resources to decentralize model training.

Take the example of a shared mobility system, such as ride-sourcing, ridesharing, or micro-mobility service. Although the principle of sharing resources has been used in transportation systems for several decades (e.g., transit or carpooling), the advent of smart and connected personal devices led to the unprecedented growth in shared mobility systems. Consider the well-known ride-sourcing company, Uber. From an operational point of view, Uber can be considered a fleet operator. However, traditional optimization-based fleet dispatching schemes are not scalable for Uber as it scales up its operation to entire cities, states, and countries. Uber can fulfill ride requests in densely-populated urban regions using myopic solution methodologies (e.g., dispatching the closest vehicle to a request's pick-up location) with short wait times, providing a high level of service. However, as the demand level diminishes in suburban areas, using myopic matching schemes leads to degradation of level of service, prompting Uber not to offer its services when demand follows below a threshold. The need for using a decentralized solution methodology for solving centralized matching problems in shared mobility systems has been acknowledged in the literature. The proposed solutions typically falls under one of the two categories of decomposition methods [61], [145], [205], [354] or partitioning and clustering approaches [246], [296]. Both families of solutions attempt to solve a large-scale optimization problem by means of solving smaller sub-problems, typically by adopting an iterative procedure that allows for asymptotically approaching the optimal solution.

Despite successfully striking a balance between solution quality and time, there are still practical challenges that limit the applicability of these methods. These challenges include the lack of a guarantee in finding a feasible solution within a specified period and the high set-up cost of the problem in a dynamic environment that is fast evolving. These challenges can be addressed through IoFT by exploiting edge resources to achieve massive model parrallelization where the computational burden is divided between local devices. Besides that, IoFT reduces communication and storage needs and can continuously update the model in real-time.

Besides that, due to the high computational complexity of optimization-based approaches, there has been a recent surge in interest to leverage deep learning in transportation applications (e.g., [168], [223]). The benefit of deep learning models is that once trained, their evaluation typically takes a fraction of a second, rendering them effective for real-time applications. However, training high-performing models requires immense amounts of data. In turn, FL can provide an elegant solution through efficiently training a global model by incorporating focused updates from several local (possibly heterogeneous) datasets, thereby enabling training generalizable deep learning models. Also, recent advances in FL can account for data heterogeneity through personalization, where each client retains a fine-tuned model based on its local data. These advances would facilitate using high-performing deep learning models to make operational decisions in dynamic systems. For example, adopting FL could allow Uber to learn a global matching policy customized for regional operations with minimal additional training to capture local idiosyncrasies. Such regional models are likely to outperform myopic algorithms that use only spatiotemporally local data.

The application of IoFT in transportation systems is not limited to connected intersections and shared mobility systems. Other existing applications that rely on spatiotemporal gathering of peers, such as vehicle platooning [163] and P2P wireless power transfer [2], can be enabled by IoFT. To effectively operate such systems, fast decision-making is necessary. FL can bridge the inherent trade-off between solution accuracy and computational complexity of finding a solution in such applications. Additionally, it is anticipated that the CAV technology will give rise to new applications that leverage connectivity and, therefore, require fast decision-making.

In addition to improving system throughput, IoFT can be used in transportation systems for privacy preservation purposes. In the age of autonomous vehicles, training models that can predict the motion of different traffic agents, e.g., vehicles, pedestrians, cyclists, etc., is of utmost importance.

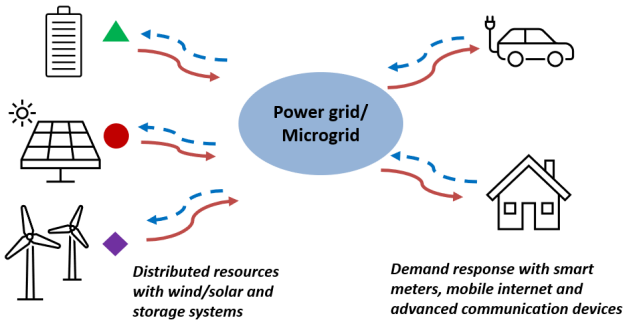

FIGURE 15. Recent developments in energy infrastructure and technology.

Typically, roadside sensors, such as cameras, can be used to obtain historical trajectories based on which trajectory prediction models can be trained. However, transmitting camera recordings or other identifiable data to a central server may create privacy concerns. IoFT can address these concerns as it allows for processing the data in the edge device, and only sending focused updates (such as gradients) needed for updating the global model to the server. Similarly, using FL, other models that rely on sensitive traveler data, such as mode choice, destination choice, and route choice models, can be effectively trained.

C. ENERGY

Modern society increasingly depends on complicated electric power systems. The US end-use of electricity reached 3.99 trillion kilowatt-hours (kWh) in 2019, and the total demand is expected to increase in the next decades [79], [81], [201]. Rapid developments in energy infrastructure and technology provide numerous opportunities for implementing new energy applications and services to meet demand, as depicted in Fig. 15. In particular, the market share of variable renewable energy sources, such as wind and solar, which provide local and distributed energy, grew to 19% in 2019 [79], [201]. It is expected that the electricity generation from renewables will double over the next 30 years [80]. Advanced communication capabilities, smart meter installations, mobile internet, and other smart technologies are enabling grid-responsive demand response and management services, such as shedding, shifting, and modulating load in peak and off-peak periods while minimizing occupant discomforts [91], [127], [167], [221], [237]. Additionally, increased use of battery storage technology and the growing penetration of electric vehicles will also change electricity supply and usage patterns [188], [304].

Facing the massive transformation, IoT can provide system-wide, integrated approaches to managing modern power systems. The IoT infrastructure's ability to capture and analyze data-intensive systems like the energy system can play a key role in managing renewables, demand response programs, electric vehicles, and other elements. Data collection and the use of intelligent algorithms can monitor and control the energy supply chain, including production, delivery, and consumption, so that suitable, cost-effective decisions can be made to balance supply and demand with minimal disruption to system operations. From the perspective of energy supply, since energy generation is an asset-intensive industry, data analytics can improve the efficiency of power production [113], [356]. On the demand side, buildings equipped with smart monitoring and communication devices can analyze end users' energy consumption, identify their needs, and transform consumers into prosumers, adjusting their demand in response to system conditions [113], [220], [356].

While IoT can empower data-intensive analytics by providing an integrated platform to collectively gather and process data, applying data science methods to analyze complex energy systems in the centralized IoT platform is not practical due to many untraceable complexities, making the IoFT paradigm more suitable.

First, massive data generated from sensors, actuators, and other devices in energy systems require real-time data analysis in high-dimensional regimes. For instance, condition monitoring sensors, such as the vibration sensors in wind turbine gearboxes, produce frequent high-dimensional observations, and smart meters in residential, commercial, and industrial buildings produce massive amounts of end-user data in high frequency. In the standard cloud-based IoT framework, where centralized cloud/data centers collect and process data, the energy consumption for big data processing is substantial, thus possibly negating the benefits of IoT for the energy industry.

Second, transmitting all the energy data to the central cloud can cause communication latency. Considering that the electric power end-use demand needs to be satisfied in real-time, such latency poses a severe risk to energy system operations. In power grid operations, the ancillary service allows the grid operator to maintain a balance between supply and demand at all times [360]. The ancillary service ranges in duration, ramping requirements, and magnitude [271], [349]. These ancillary services become more important as renewable energy sources rapidly replace fossil fuel generation. Wind and solar, the fastest-growing renewables, are characterized by significant variability, often with limited predictability [6], [33]. Smart and grid-interactive buildings can provide such grid flexibility by adjusting their end-use patterns [25], [210]. Storages are also an attractive option for providing ancillary services. Communication latency causes ineffective coordination of these ancillary service resources and negatively affects grid reliability. As such, fast local updates for enhanced electricity supply and demand predictions are essential for successful ancillary service implementation.

Lastly, the modern power system is characterized by distributed energy due to the growing penetration of renewables, storage, and demand response. In these distributed systems, individual stakeholders such as utilities and consumers can perform their own decision-making [35], [114], [280]. For example, utilities that manage their own renewable facilities may not want to share information with others to maximize

their profit. Those who participate in the demand response programs may wish to adjust their end-use demand upon grid request without revealing their energy use patterns to others due to privacy issues. The decentralized IoFT framework provides the right platform for such decentralized and distributed decision-making.

While IoFT can remedy the limitations of the centralized IoT by building individual models locally at each end node, there are several challenging issues. First, end nodes often have limited computing power to train data science/machine learning models. As discussed above, sensors, actuators, and smart meters produce massive data. Inefficient computation at end nodes can delay predictive decision-making, fault diagnosis/condition monitoring, and change point detection, among others [33], [54], [327]. New data science methods are needed to optimally guide the model learning process for achieving computational efficiency with theoretical and practical implications in the IoFT paradigm.

Next, unlike traditional power supply with fuel-based generators and end-users who passively consume energy, modern power systems consist of highly heterogeneous units with distinct supply/demand characteristics. On the demand side, technologies such as smart devices, demand management programs, and electric vehicles affect end-use patterns 24/7. On the supply side, energy units become more diverse and heterogeneous. Unlike traditional fuel-based sources, each renewable facility has distinctive power generation characteristics (e.g., facility layout, turbine type, each wind farm [335]). While the personalized learning discussed in Sec. IV can address the heterogeneity to some extent, managing a large number of heterogeneous supply/demand units with distinct energy characteristics is challenging. Hence, personalized prediction needs to be translated into effective collaborative management.

Finally, energy consumption is significantly affected by ambient environmental and other localized conditions [228]. Peak demand predictions vary 1.5–2% for every 0.5°C difference in predicted temperature [22], [229]. Electricity needs vary due to many spatially localized characteristics, such as densely populated areas experiencing urban heat island effects in summer that increase electricity demand compared to suburban and rural areas [125], [202], and EV charging stations exhibiting different charging patterns depending on localized characteristics. Renewable generations are also directly influenced by local weather and geographical conditions [126]. Expanding the use of IoFT for environmental modeling will require modeling the spatially and temporarily correlated environmental conditions while incorporating local heterogeneous characteristics.

In summary, modern power system faces technical challenges including computational scalability, efficiency, heterogeneity, localized characteristics, and distributed management. IoFT has the potential to address these challenges, and the successful development and implementation of IoFT will make the energy system (and its end users) ''smarter'' in terms of efficiency, flexibility, and economic competitiveness.

D. HEALTHCARE

Healthcare stands to benefit significantly from IoFT because several unique contextual factors suggest that the status quo has failed when it comes to deploying machine learning (ML): (i) many existing models fail to generalize; (ii) legal and ethical implications limit the appetite to share data; (iii) the vendors who administer the electronic health records (EHRs) that contain patients' data have an outsized influence on model deployment; (iv) national network-based research efforts started to adopt decentralized methods. This section will describe how these factors impact healthcare differently from other sectors and illustrate areas where IoFT is likely to thrive in this domain.

ML models are commonly used in early warning systems, diagnostic systems for radiology and pathology, and interpreting medical device output, such as in electrocardiography. While the medical literature contains numerous examples of apparently high-performing models, many of these studies suffer from poor generalization, which can either be demonstrated through an independent examination of its methods (by applying the PROBAST tool [319]) or through external validation of the findings. This was particularly evident early in the COVID-19 pandemic, where nearly all studied models in an extensive systematic review were considered poorly generalizable. Although the pandemic has affected millions of people in the U.S. alone, most health systems did not have a sufficient number of patients, or adequate diversity, to ensure model generalizability. The lack of generalizability is not merely theoretical; it has also been systematically demonstrated. In a recent study examining over a thousand cardiovascular clinical prediction models, 81% of validation studies found worse performance than was reported originally [316].

Generalizability improves when models are developed using pooled data from multiple health systems. Under the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule, healthcare data may be shared for the purposes of research if identifiers have been removed, or under certain circumstances, if patients have authorized the use of their data for research after approval by an institutional review board [231]. In most instances, data sharing between health centers also require legal agreements known as ''data use agreements'' or ''business associates agreements''. Even with such agreements in place, sharing of data between health systems may go against the expectations of the general public [248]. Thus, pooling data from multiple health systems while enabling better ML models may potentially damage public trust. Indeed, when a 2019 partnership between Ascension and Google was reported on by the Wall Street Journal, it resulted in public outcry [60].

The difficulty faced by health centers in combining data with other health systems has led to a vacuum that EHR

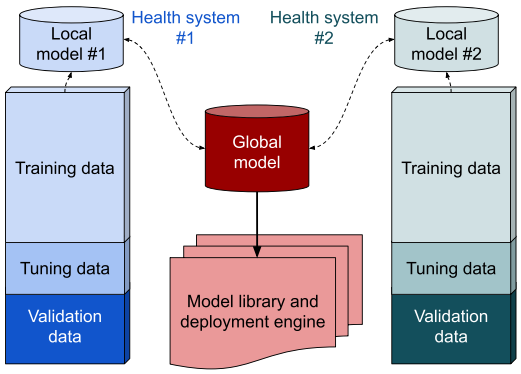

FIGURE 16. IoFT in healthcare.

vendors have largely filled. Indeed, two of the most widely used healthcare ML models in the U.S. include the Epic Deterioration Index (owned by Epic Systems in Verona, Wisconsin) and the APACHE-IV scores (owned by Cerner Corporation, Kansas City, Missouri) [287], [358]. Both models were developed using data from the EHR systems of multiple hospitals. Because EHR vendors have direct access to patient data (on behalf of their clients), they are well-positioned not only to combine data for analysis (with permission) but also to deploy the resulting models within their EHRs.

This siloing of data, while in the best interests of patients, has led to significant challenges in the development of high-quality, non-proprietary, freely available models. The healthcare informatics community has responded to this challenge, but there is still a long road ahead. In 2009, the development of the Shared Health Research Information Network (SHRINE) enabled federated querying of clinical data repositories [310]. Conceptually, federated querying of multi-hospital data allows researchers to identify optimal sites for ML model development based on rapid multi-system sample size determinations [309]. A federated querying system, known as the 4CE Consortium, was rapidly deployed to support COVID-19 research [28]. Along this line, IoFT has been applied to the development of multi-hospital models in healthcare [139], [278], [284]. In these setting FL was used to allow health systems to share access to a model library and deployment engine without directly sharing data, as depicted in Fig. 16.

IoFT will be important in healthcare because it enables the creation of high-quality ML models without privacy risks. However, IoFT will have to contend with Food and Drug Administration (FDA) regulations that treat ML algorithms as a type of software-as-medical-device (SaMD). Initial applications of IoFT in health will likely be limited to class I and II medical devices, including ML models used primarily to inform care decisions. Class III-IV medical devices (e.g., cardiac pacemakers) require extensive review and premarket approval by the FDA. Class I-II devices only require premarket notification demonstrating equivalence with a legally marketed device. As a result, IoFT will likely be applied in supporting early warning systems in hospitals, automating order entry, and smart scheduling of patient visits. In each of these scenarios, global patterns exist but differ locally to the extent that the combination of global and local models will likely achieve superior results without sacrificing patient privacy.

E. BUSINESS

Capturing and maintaining relationships between businesses raise many challenges for IoFT, and new methods to meet them are needed. To get an insight into these challenges, consider a business (the principal) that has the following decision to make: shall it build a facility to supply a particular item, or shall it 'outsource' it to another business (the agent)? In the former option, all the risks are carried by the principal, while in the latter these are shared with the agent. In the economy of today dominated by fast technological developments, outsourcing is often preferred. Despite the advantage of risk-sharing, outsourcing comes at a cost. The agent has more information about its operations and can use this information in its favor and at the expense of the principal, which is generally termed as 'moral hazard' in the economics literature [160], [290].

The example above highlights several key aspects of this relationship:

(i) First: Businesses are often intrinsically federated and are reluctant to share their proprietary business secrets. This federation forces decentralization of decisions. Thus, all the advantages of IoFT like localized computing, data privatization, security, and information privacy can be realized, along with the benefits of risk-sharing.

(ii) Second: A key challenge is that in business applications, agents may have different and often competing objectives. This becomes a new challenge for FL. Indeed, the models in Secs. III - V are focused on maximizing a common objective. But if such a formulation does exist and somehow includes agents' conflicting objectives, its successful implementation is contingent on true reporting by the agents. Otherwise, the principal has to monitor the agents' operations continuously. Depending on the situation, monitoring may be impossible or too expensive and thus may nullify the advantage of decentralization of operations.

(iii) Third: The principal can make arrangements with many independent agents, and each one of these may have an arrangement with other independent businesses supplying its services. This creates an organizational structure of relationships, forming a hierarchical structure of agents (i.e., a rooted tree with agents at various distances (levels) from the root). Here agents at intermediate levels (excluding the root agent and the agents at the end) play two roles: a principal to its subordinate nodes and an agent to the lower level node. This adds an additional level of complication for the use of FL in such business setups.

Though businesses are usually organized in a federated nature and are ideally suited for FL, the full potential of IoFT can only be realized if the above-stated problems are

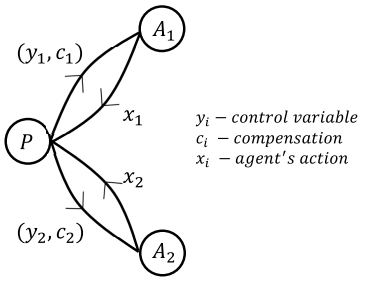

FIGURE 17. Example of decentralized decisions.

effectively solved. There are some recent encouraging developments towards this end. Below we highlight one possible solution.

In the single-agent case, an answer to this problem has been proposed by [277] by the design of a mechanism that mitigates moral hazard and effectively decentralizes decision making. In a continuous dynamic setup, at the epoch t the principal observes a noisy signal, $x(t)$ about the 'effort' of the agent and compensates $c(t)$ to the agent based on this signal. It is shown that in case the signal noise is generated by a Brownian motion (a Gaussian process), there exists a 'control' variable $y(t)$ that can affect the noise of the signal to a level at which the principal can make a good decision about the compensation for the agent. In a more realistic setting, the principal may create such relationships with several agents, each with private information, data, and objectives. There may also be interactions between agents' decisions, i.e., decisions by an agent may affect the outcomes of other agents' actions, and they may have conflicting objectives with each other and with the principal's, making the moral hazard problem harder to mitigate. This case has been studied by [192] who integrated the notion of Nash equilibrium into this model. Thus, if all the agents' decisions form a Nash equilibrium, no agent can gain by falsifying information when all the other agents do not. This mitigates the moral hazard, and the decisions arrived at can be implemented. Fig. 17 shows a representation of decentralized decision-making with two agents.

A brief overview of the solution methodology for a dynamic optimization problem in continuous time over a finite or an infinite horizon is as follows: the optimization problem faced by the principal is to find a policy which maximizes its expected discounted profit over the horizon, i.e., for the infinite horizon case:

$$\arg \min_{\{y(s),\, c(s):\, s\geq 0\}} \mathbb{E} \int_0^{\infty} e^{-r_P \times s} f_P\Big(x(s), y(s), c(s)|\alpha(s)\Big) ds$$

where r_P, f_P and α_P are respectively the discount factor, profit function and data of the principal; under the individual rational constraint that each agent's expected discounted profit over the horizon exceeds some predetermined minimum amount. When Brownian motions drive all randomness in the formulation, it can be seen that when the optimal policy is followed, the expected discounted profits of the principal and the agents are martingales. Using the principle of Bellman, this formulation is decomposed into federated optimization problems that are independently solved by each agent, while the principal solves a constrained Hamilton-Jacobi-Bellman (HJB) optimal control problem. The HJB solves for the continuation value ('value' function of dynamic programming) of the principal as a function of the state variables, which include the continuation values of each agent. It is obtained by using Ito's formula on the function and the fact that the expected profit of the principal is a Martingale when an optimal policy is adapted. In the case of a hierarchical system, agents at the intermediate levels independently solve both an optimization and a constrained HJB problem, thus achieving a key goal of IoFT.

In conclusion, the above-described decomposition mechanism between the principal and possibly several agents facilitates the use of federated learning. Each participant can effectively use the data collected from its operations and determine the profits, given the compensation it receives from the principal. The method described above is but one possible solution. The use of data-driven reinforcement learning is another viable option. Through this example, we hope to encourage researchers to explore decentralized decision-making within IoFT further.

F. QUALITY ENGINEERING

IoT as an enabling technology for real-time data sharing has stimulated a new paradigm in quality engineering, which expands quality control from the design and manufacturing stages to the whole product life cycle. For example, GE Prognostic Health Management Plus (PHM+) system uses its onboard sensors to collect engines' operating data during flight. These data are communicated through its secure network and analyzed by the central server to provide proactive maintenance services. Similarly, the automotive industry uses vehicles' onboard sensors to monitor vehicles' real-time driving performance, allowing for early warnings of potential problems. Additionally, they can deploy integrated vehicle-based safety systems (IVBSS) [86] to improve customers' driving experience and safety. Throughout the product life cycle, quality assurance is highly demanded for customer satisfaction, which is especially imperative for expensive or safety-sensitive products. Further, these in-field operational data can provide quick feedback for continuous quality improvement.

Online monitoring and fault diagnosis, which uses onboard sensors on products or in-situ process sensing signals during manufacturing, is one of the most critical research issues in quality engineering. Currently, most collected process sensing signals or quality inspection are multistream waveform signals; an image or video signals with very high frequencies [254], [305], [321]. Those require massive bandwidth and time to transmit the original data between local devices and the central orchestrator. More importantly, quality control requires fast decisions and real-time detection of anomalies. Therefore, decision-making ought to be on the edge and not on a central system. The shift towards IoFT,

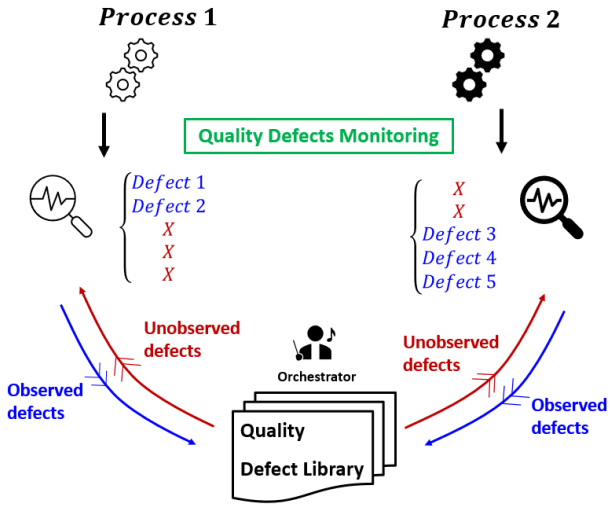

FIGURE 18. Quality control example in IoFT with missing anomalies.

will allow tackling both of these challenges. In IoFT, compute resources at the edge are used so that only summary statistics and low dimensional information is transmitted to the central server (or perhaps a peer in a peer-to-peer network). Also, models reside on the edge and can be deployed immediately. Therefore, the IoFT platform shows distinctive advantages in quality control for reducing the communication load and making real-time decisions.

That being said, there are some unique research issues to advance quality control methodologies under the IoFT platform. Below we iterate a few:

1) INSUFFICIENT DATA

Quality control models, such as anomaly detection or fault diagnosis models [182], [183], are poised to greatly benefit from IoFT. In statistical process control (SPC), many edge devices or clients may lack sufficient data to build a normal operating baseline for abnormal change detection. For instance, (i) clients may not have observed the full set of possible anomalies as shown in Fig. 18 (ii) new products/processes possess few data, so does small-scale clients (such as the 3D printing citizens in the Sec. I-A) and low volume manufacturing of rare and expensive products (ex: planes). IoFT as an emerging technology offers a medium to borrow strength across different clients for better SPC models while preserving copyrights and privacy. For instance, through meta-learning within IoFT, clients may directly adapt to new products/processes. Also, through domain adaptation, clients can learn across defect types observed only on some clients.

2) CONTINUAL LEARNING

IoFT allows knowledge to be readily shared. As a result, quality control models (e.g., anomaly detection) may be continuously updated to register new defects or improve detection and diagnosis accuracy for old ones [64]. Continuous process improvement requires updates of quality control models over the entire life cycle of a product.

3) HUMAN FEEDBACK AND EXPERT KNOWLEDGE

Upon the detection of an anomaly, most operators will do a post-inspection about the diagnosis results (i.e., false positive or false negative). Improving models upon such expert feedback will be of importance to IoFT. Indeed, much like data in IoFT, human knowledge is often decentralized, with different entities having expert knowledge on different elements of a system. Therefore, modeling approaches that combine expert knowledge, human feedback and data-driven models are needed. Such models have evolved recently under the notion of expert or physics-guided data-driven modeling. However, they are yet to be explored under the IoFT paradigm.

4) QUALITY CONTROL

As described in Sec. VII-A, quality control will immensely benefit from a shared library of knowledge, be it a library of in-control behaviors, common anomalies, root causes, etc. Many companies are reluctant to collaborate in building such a library due to privacy constraints. IoFT may bring this end goal to fruition.

In conclusion, quality control is set to greatly benefit from IoFT. Yet many challenges still need to be tackled to realize its potential.

G. COMPUTING

Intending to gain insights without exposing raw data, large technology companies such as Google, Apple, and Firefox started to deploy FL for computer vision and natural language processing tasks across user devices [44], [67], [109], [331]; others, including NVIDIA, apply FL to create medical imaging AI [175]; smart cities perform in-situ image training and testing on AI cameras to avoid expensive data migration [115], [130], [191]; and video streaming solutions use FL to interpret and react to network conditions [328]. However, we believe that these applications of FL are only scratching the surface, given that the applications of ML in computing are even broader, many of which can be deployed more widely and improved by leveraging FL. In the following, we present an incomplete overview of FL's many existing and potential applications in computing. The common theme across many of them is enabling information sharing between multiple administrative domains without sharing raw private data.

1) DATABASES

Indexes play a critical role in speeding up query processing in database management systems (DBMS). In recent years, learned indexes are gaining popularity, whereby an ML model replaces traditional index structures including B-Tree, Hash-Table, Bitmap, and so on. These learned indexes can be classified into two broad categories: static, read-only indexes [154] focus more on read-heavy workloads, while updatable indexes [69] can handle lookups as

well as inserts and deletes common in write-heavy workloads. Nonetheless, all of these works focus on applying ML to a single administrative domain, which restricts the use of learned indexes to scenarios that have already been observed within the domain and leaves them with potentially weaker performance on previously unseen workloads. Applying FL in this context will help collaborative training among multiple competing domains without sharing raw data. Indexes are only a part of the many research challenges faced in the database literature, and ML and FL can have possible applications in, among others, transaction processing, lock management, query planning/optimization, and cardinality estimation.

2) NETWORKING

Networks are inherently distributed, and networking protocols are no exception. As a result, FL is a natural fit for many networking problems where ML can be applied and has already been applied in limited scope (e.g., not being able to copy all data to a centralized location). Over the past decade, many networking problems have relied on ML techniques; for example, to infer datacenter topology [55], to determine hyperparameters for congestion control algorithms [318], for Internet-scale congestion control using deep reinforcement learning [128], for leveraging single and multiple paths in an adaptive manner [72], [96], and for routing [299]. They primarily relied on a single trust domain (e.g., a data center, an Internet AS, etc.) where everything is controlled by and cooperates with a single entity within which data can be shared. FL can expand the scope of many of these algorithms to be applied at a broader scale via privacy-preserving learning that may incentivize multiple domains to collaborate to learn a global model and then personalize to their own needs.

3) CLOUD COMPUTING

To cope with the increasing number of Internet users as well as IoT and edge devices, large organizations leverage tens to hundreds of data centers and edge sites. Collecting data related to end-user sessions, monitoring logs, and performance counters, and thereafter analyzing and personalizing this data can significantly improve the overall user experience. Traditional approaches to ML require collecting all these data to a centralized cloud data center, which is often impossible due to bandwidth constraints and data privacy regulations. IoFT is the natural choice in this context to address both of these concerns [161], [253].

4) VIDEO ANALYTICS

Cameras deployed for traffic control and surveillance continuously record and analyze large volumes of recorded videos using video analytics [115], [129], [346], which has been made possible by recent advances in computer vision. A key challenge in this context is training large models, typically in datacenters, before they are deployed in the wild. Traditional centralized training is expensive and narrow; the latter follows from the fact that the models are trained on relatively small training datasets. With the advent of smart cameras in the edge, i.e., cameras with onboard or nearby computing capabilities, we can leverage FL to train models with much bigger training datasets, which can significantly improve the accuracy of the models and keep them continuously updated.

5) VIDEO STREAMING

Videos constitute the bulk of the Internet traffic today, and live video streaming is a major contributor to this category. Client-side video players typically employ a variety of adaptive bitrate (ABR) algorithms to optimize users' quality of experience. Recent advances in ABR algorithms include using reinforcement learning to generate context-specific ABR algorithms [204] to more recently demonstrating that generating many such algorithms does not necessarily perform better than using FL to generate one model that works in conjunction with classic video streaming techniques [328]. A key research direction here would be adopting federated reinforcement learning to leverage the best of both approaches, and find a balance between the global and personalized ABR algorithms.

H. RELIABILITY

Reliability engineering is concerned with the failure behavior of a system under stated conditions. A failure can be catastrophic, meaning a complete, sudden, and often unexpected system breakdown, leading to significant or even total loss of system performance. It can also be a degradation-induced soft failure (e.g., the capacity drop of a lithium-ion battery). There are several ways to evaluate the reliability of a product, though generally, evaluation based on reliability data is most common. Reliability data are usually in the format of lifetime data or degradation data. However, in these datasets, failure data is scarce, given that most products are highly reliable and do not fail often. Nevertheless, IoFT, with its privacy-protecting protocols, provides a unique opportunity to overcome the challenges of scarce data in reliability engineering.

Throughout the ages, reliability data have been classified as sensitive information by companies. With millions of products released in the marketplace by manufacturers, it is no secret that reliability data is both extensive and comprehensive. Yet, its sensitivity hinders its usability. IoFT provides a unique opportunity to enable knowledge sharing from available datasets without compromising its privacy in such a scenario. For instance, edge compute resources can be exploited to replace existing reliability databases (e.g., product lifetime) with summary statistics (or prior) distribution of modeling parameters for each product/component. Further, scenarios exist where products have only a few lead manufacturers (e.g., the smartphones industry). In here, the designs from different manufacturers are distinct, implying a certain degree of heterogeneity [333]. For a smartphone manufacturer, the reliability information from previous generations of smartphone products can be more beneficial than information from other manufacturers. Such a setting poses another unique challenge for IoFT. In the following, we will

discuss the potential applications of IoFT in three different settings: (i) among a group of manufacturers producing a similar product, (ii) within a manufacturer, and (iii) within a reliability organization.

We first start with IoFT among a group of manufacturers. Consider reliability testing for evaluating a product's reliability, where reliability data is in the format of lifetime data subject to right censoring. Generally, the Weibull distribution with reliability function (scale α and shape β) and the log-normal distribution (location μ and scale σ) are two of the most commonly used distributions for describing a product's lifetime data [211]. Moving forward, we will focus on the Weibull distribution, though similar logic applies to different distributions. The Weibull shape parameter β is commonly believed to depend on the product type (i.e., failure mode due to the material used: e.g., corrosion of semiconductor material) or the failure mode due to customer usage (e.g., the user breaks their cellphone). Such parameter can be regarded as insensitive information to the product's reliability. On the other hand, the Weibull scale parameter α (also known as the characteristic life) is usually dependent on the effort of reliability investment from the manufacturers [211]. Suppose a manufacturer uses local data to evaluate product reliability. In that case, both parameters in the lifetime distribution have to be estimated, and the uncertainties in both parameters will affect the precision of the final reliability evaluation. Then, it is reasonable to advocate sharing information on β to decrease uncertainty in β, which eventually helps all manufacturers achieve a more accurate evaluation of product reliability. Additionally, since the information on α is unshared, a manufacturer cannot infer the product reliability of other manufacturers.

Operationally, we can use a Bayesian approach. Let us consider the Weibull distribution for demonstration and provide a rough sketch of the parameter updating process in an IoFT system. First, in large sample sizes, the posterior distribution of $\log\beta$ can be well approximated by a normal distribution ($\log\beta$ ensures the positiveness of β). Afterward, when a manufacturer has recently conducted a life test and requests an update, or when the central server randomly chooses a manufacturer and mandates an update, the manufacturer will first get a broadcast of the current posterior distribution of $\log\beta$. The manufacturer can then use this posterior distribution of β and the manufacturer's local posterior distribution on α, which might be obtained from previous product testing, as a prior distribution for the newly collected reliability data. A routine Bayesian update gives the new posterior of α and $\log\beta$. Then, the manufacturer can compute the mean and variance of the new posterior of $\log\beta$, and return the updated posterior distribution to the central server. Finally, the central server can then check the discrepancy between the broadcasted and the updated distributions to safeguard against data corruption during transmission or malicious attacks. If acceleration is used in the life test, then parameters in the acceleration model (as the activation energy in the Arrhenius model) contain no sensitive reliability information.

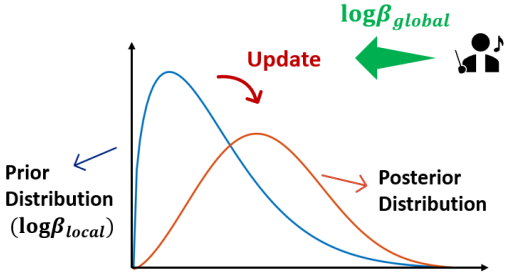

FIGURE 19. A schematic of FL in reliability testing by using Weibull as the global model.

Thus, such parameters can also be federated together with the Weibull shape. The same idea can be extended to accelerated degradation testing, where FL can be applied to the shape parameter of the mean degradation paths and the acceleration parameters. Fig. 19 provides a schematic view of the discussed protocol.

Next, we explore the application of IoFT within a manufacturer. The underlying idea is that when a certain product is sold to customers, the collection of user data for early prediction of product failure must comply with some privacy terms, thus being restricted. Given the computational and communication capabilities of the product, IoFT provides a unique advantage in the presence of privacy constraints. Consider as an example, lithium-ion batteries which are widely used in electric vehicles. It is well-known that the usage pattern has a significant impact on the state-of-charge (short-term) and the remaining useful life (long-term) of lithium batteries [247]. However, it is almost impossible to associate the usage pattern to these two important performance characteristics because of difficulties replicating the heterogeneity in users' behavior. FL provides an opportunity to train an accurate model for each characteristic. To do so, we need a global statistical model that associates the customer usage pattern, the charge-discharge pattern, and the ambient environments to the performance characteristics. The global statistical model can be a random-effects model that allows for heterogeneity among customers. Then the approaches introduced in Secs. III and IV can be used to learn a global model or a personalized model. Further, due to the different ambient environments of the users, there will be covariate shifts among users. Methods reviewed in Sec. IV can be perfectly adopted to solve these problems.

Third, we discuss IoFT implementation on reliability organizations. The major of reliability organizations collect field failure data on a large variety of components from various sources. The ultimate goal is to estimate the reliability of any new system based on the component reliability estimated from collected databases. Some large databases can be found in OREDA [236], Mahar *et al.* [199] and Denson *et al.* [66]. Since there are millions of components, the data reported in these databases are aggregated in such a way that only a few summary statistics are provided for each component. This aggregation is based on the assumptions of exponential

distributions, and it makes the fitting of a Weibull distribution extremely difficult [45]–[47]. However, FL provides a better solution to build such a database. Instead of recording these summary statistics, we can first agree upon a distribution for a component and then maintain a posterior distribution for the parameters. For example, the inverse Gaussian and the Birnbaum-Saunders distributions are commonly used for mechanical components, and Weibull is the most popular distribution in reliability. A conjugate distribution for the parameters, or a normal distribution for the transformed parameters (to ensure positivity), can be adopted for ease of use. Every time a partner of the database has new data to update, the database (which serves as the central server) can broadcast the current posterior of parameters for the component. The partner can then use this as prior and update the posterior with local data. The above rough idea can be materialized with the framework discussed in Sec. III.

In a nutshell, reliability of a manufactured product is usually shrouded with privacy concerns. Thus, implementing IoFT within a manufacturer is promising in solving the issues of data transmission and user privacy. On the other hand, IoFT across manufacturers is much more difficult. Nevertheless, with proper design of the information-sharing mechanisms, IoFT can tremendously help manufacturers increase the accuracy of reliability estimation and prediction without sacrificing confidentiality.

REFERENCES

[1] *Ford Sync*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.ford.com/technology/sync/

[2] M. Abdolmaleki, N. Masoud, and Y. Yin, "Vehicle-to-vehicle wireless power transfer: Paving the way toward an electrified transportation system," *Transp. Res. C, Emerg. Technol.*, vol. 103, pp. 261–280, Jun. 2019.

[3] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[4] S. M. A. B. A. Islam, A. Hajbabaie, and H. M. A. Aziz, "A real-time network-level traffic signal control methodology with partial connected vehicle information," *Transp. Res. C, Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102830.

[5] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020.

[6] J. Wang, A. AlShelahi, M. You, E. Byon, and R. Saigal, "Integrative density forecast and uncertainty quantification of wind power generation," *IEEE Trans. Sustain. Energy*, vol. 12, no. 4, pp. 1864–1875, Oct. 2021.

[7] Apple. (2019). *Designing for Privacy*. Accessed: Apr. 21, 2021. [Online]. Available: https://developer.apple.com/videos/play/wwdc2019/708

[8] M. Ghuhan Arivazhagan, V. Aggarwal, A. Kumar Singh, and S. Choudhary, "Federated learning with personalization layers," 2019, *arXiv:1912.00818*.

[9] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[10] AWS. (2019). *What is AWS?* Accessed: Jul. 18, 2020. [Online]. Available: https://www.youtube.com/watch?v=a9__D53WsUs

[11] AWS. (2021). *Amazon Web Services*. AWS. Accessed: Jul. 18, 2020. [Online]. Available: https://aws.amazon.com/

[12] Azure. (2018). *How Does Microsoft Azure Work?* Accessed: Jul. 18, 2020. [Online]. Available: https://www.youtube.com/watch?v=KXkBZCe699A

[13] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, Jun. 2008.

[14] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.

[15] S. Baharlouei, M. Nouiehed, A. Beirami, and M. Razaviyayn, "Rényi fair inference," 2020, *arXiv:1906.12005*.

[16] L. Peter Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, Nov. 2002.

[17] M. Bauer, M. Rojas-Carulla, J. B. Świkatkowski, B. Schölkopf, and R. E. Turner, "Discriminative k-shot learning using probabilistic models," 2017, *arXiv:1706.00326*.

[18] F. Baumann and D. Roller, "Additive manufacturing, cloud-based 3D printing and associated services—Overview," *J. Manuf. Mater. Process.*, vol. 1, no. 2, p. 15, Oct. 2017.

[19] A. Behrendt, A. Kadocsa, R. Kelly, and L. Schirmers, "How to achieve and sustain the impact of digital manufacturing at scale," McKinsey Company Quartley, New York, NY, USA, Tech. Rep., 2017.

[20] A. Beimel, A. Korolova, K. Nissim, O. Sheffet, and U. Stemmer, "The power of synergy in differential privacy: Combining a small curator with local randomizers," 2019, *arXiv:1912.08951*.

[21] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Personalized and private peer-to-peer machine learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 473–481.

[22] M. Bhandari, S. Shrestha, and J. New, "Evaluation of weather datasets for building energy simulation," *Energy Buildings*, vol. 49, pp. 109–118, Jun. 2012.

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

[24] P. J. Bickel and Y. Ritov, "Non- and semiparametric statistics: Compared and contrasted," *J. Stat. Planning Inference*, vol. 91, no. 2, pp. 209–228, Dec. 2000.

[25] B. Biegel, M. Westenholz, L. H. Hansen, J. Stoustrup, P. Andersen, and S. Harbo, "Integration of flexible consumers in the ancillary service markets," *Energy*, vol. 67, pp. 479–489, Apr. 2014.

[26] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld, "PROCHLO: Strong privacy for analytics in the crowd," in *Proc. 26th Symp. Operating Syst. Princ.*, Oct. 2017, pp. 441–459.

[27] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[28] G. A. Brat, G. M. Weber, N. Gehlenborg, P. Avillach, N. P. Palmer, L. Chiovato, J. Cimino, L. R. Waitman, G. S. Omenn, A. Malovini, and J. H. Moore, "International electronic health record-derived COVID-19 clinical course profiles: The 4CE consortium," *NPJ Digital Med.*, vol. 3, no. 1, pp. 1–9, 2020.

[29] 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.

[30] S. Bubeck, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[31] B. Buckholtz, I. Ragai, and L. Wang, "Cloud manufacturing: Current trends and future implementations," *J. Manuf. Sci. Eng.*, vol. 137, no. 4, Aug. 2015, Art. no. 040902.

[32] P. Buhlmann and S. van de Geer, *Statistical for High-Dimensional Data* (Springer Series in Statistics). New York, NY, USA: Springer, 2011.

[33] E. Byon, Y. Choe, and N. Yampikulsakul, "Adaptive learning in time-variant processes with application to wind power systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 997–1007, Apr. 2016.

[34] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A stochastic quasi-Newton method for large-scale optimization," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1008–1031, Jan. 2014.

[35] R. Carli and M. Dotoli, "Decentralized control for residential energy management of a smart users microgrid with renewable energy exchange," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 641–656, May 2019.

[36] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[37] A. R. Castillo, V. R. Joseph, and S. R. Kalidindi, "Bayesian sequential design of experiments for extraction of single-crystal material properties from spherical indentation measurements on polycrystalline samples," *JOM*, vol. 71, no. 8, pp. 2671–2679, Aug. 2019.

[38] K. Chang, N. Balachandar, C. Lam, D. Yi, J. Brown, A. Beers, B. Rosen, D. L. Rubin, and J. Kalpathy-Cramer, "Distributed deep learning networks among institutions for medical imaging," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 8, pp. 945–954, Aug. 2018.

[39] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," 2018, *arXiv:1802.07876*.

[40] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the Internet of Things: Literature review and challenges," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, Aug. 2015, Art. no. 431047.

[41] H. Chen, L. Zheng, R. A. Kontar, and G. Raskutti, "Stochastic gradient descent in correlated settings: A study on Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 2722–2733.

[42] H.-Y. Chen and W.-L. Chao, "FedBE: Making Bayesian model ensemble applicable to federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2021.

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

[44] M. Chen, A. T. Suresh, R. Mathews, A. Wong, C. Allauzen, F. Beaufays, and M. Riley, "Federated learning of N-gram language models," in *Proc. 23rd Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2019, pp. 121–130.

[45] P. Chen and Z.-S. Ye, "Random effects models for aggregate lifetime data," *IEEE Trans. Rel.*, vol. 66, no. 1, pp. 76–83, Mar. 2016.

[46] P. Chen and Z.-S. Ye, "Estimation of field reliability based on aggregate lifetime data," *Technometrics*, vol. 59, no. 1, pp. 115–125, 2017, doi: 10.1080/00401706.2015.1096827.

[47] P. Chen, Z.-S. Ye, and Q. Zhai, "Parametric analysis of time-censored aggregate lifetime data," *IISE Trans.*, vol. 52, no. 5, pp. 516–527, May 2020.

[48] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 15–26.

[49] R.-B. Chen, W. Wang, and C. F. J. Wu, "Sequential designs based on Bayesian uncertainty quantification in sparse representation surrogate modeling," *Technometrics*, vol. 59, no. 2, pp. 139–152, Apr. 2017.

[50] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Frank Wang, and J.-B. Huang, "A closer look at few-shot classification," 2019, *arXiv:1904.04232*.

[51] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," 2020, *arXiv:2010.13723*.

[52] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017, *arXiv:1712.05526*.

[53] Y. Jee Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," 2020, *arXiv:2010.01243*.

[54] Y. Choe, W. Guo, E. Byon, J. J. Jin, and J. Li, "Change-point detection on solar panel performance using thresholded LASSO," *Qual. Rel. Eng. Int.*, vol. 32, no. 8, pp. 2653–2665, Dec. 2016.

[55] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, "Managing data transfers in computer clusters with orchestra," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 98–109.

[56] CNBC. (2020). *Ford Temporarily Closes Two Plants After Three Workers Test Positive for Coronavirus*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.cnbc.com/2020/05/20/ford-closes-chicago-plant-after-two-workers-test-positive-for-covid-19.html

[57] CNN. (2020). *12-Year-Old Boy 3D Prints Masks for Frontline Workers*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.cnn.com/videos/us/2020/04/25/coronavirus-3d-print-ppe-12-year-old-pkg-whitfield-vpx.cnn

[58] P. L. Combettes, "Monotone operator theory in convex optimization," *Math. Program.*, vol. 170, no. 1, pp. 177–206, Jul. 2018.

[59] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. Philadelphia, PA, USA: SIAM, 2009.

[60] Rob Copeland. (2019). *Google's, Project Nightingale Gathers Personal Health Data on Millions of Americans*. Accessed: Apr. 25, 2021. [Online]. Available: https://www.wsj.com/articles/google-s-secret-project-nightingale-gathers-personal-health-data-on-millions-of-americans-11573496790

[61] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: Models and algorithms," *Ann. Oper. Res.*, vol. 153, pp. 29–46, Sep. 2007.

[62] J. E. Correa, R. Toro, and P. M. Ferreira, "A new paradigm for organizing networks of computer numerical control manufacturing resources in cloud manufacturing," *Proc. Manuf.*, vol. 26, pp. 1318–1329, 2018.

[63] D. Culver and B. Westcott. (2020). *3D Printing Enthusiasts Are Working From Home to Help Hospitals Fight Coronavirus*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.cnn.com/2020/04/18/tech/us-coronavirus-ventilator-3d-printer-intl-hnk/index.html

[64] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 5, 2021, doi: 10.1109/TPAMI.2021.3057446.

[65] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.

[66] W. Denson, W. Crowell, P. Jaworski, and D. Mahar, *Electronic Parts Reliability Data 2014*. Rome, NY, USA: Reliability Information Analysis Center, 2014.

[67] Apple Differential Privacy Team, "Learning with privacy at scale," *Apple Mach. Learn. J.*, 2017.

[68] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3571–3580.

[69] J. Ding, U. F. Minhas, J. Yu, C. Wang, J. Do, Y. Li, H. Zhang, B. Chandramouli, J. Gehrke, D. Kossmann, D. Lomet, and T. Kraska, "ALEX: An updatable adaptive learned index," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2020, pp. 969–984.

[70] T. C. Dinh, H. N. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. 34th Conf. Neural Inf. Process. Syst.*, 2020, pp. 21394–21405.

[71] J. Dolezal. (2020). *3D Printed Face Shields for Medics and Professionals—Join Us!* Accessed: Jul. 18, 2020. [Online]. Available: https://forum.prusaprinters.org/forum/coronavirus-covid-19/3d-printed-face-shields-for-medics-and-professionals-join-us/

[72] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," in *Proc. USENIX NSDI*, 2015, pp. 395–408.

[73] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *J. Mach. Learn. Res.*, vol. 6, pp. 2153–2175, Dec. 2005.

[74] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, "Faster least squares approximation," *Numer. Math.*, vol. 117, no. 2, pp. 219–249, Feb. 2011.

[75] W. Du, D. Xu, X. Wu, and H. Tong, "Fairness-aware agnostic federated learning," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2021, pp. 181–189.

[76] M. Duan, D. Yoon, and C. E. Okwudire, "A limited-preview filtered B-spline approach to tracking control–with application to vibration-induced error compensation of a 3D printer," *Mechatronics*, vol. 56, pp. 287–296, Dec. 2018.

[77] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang, "Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications," in *Proc. IEEE 37th Int. Conf. Comput. Design (ICCD)*, Nov. 2019, pp. 246–254.

[78] H. Edwards and A. Storkey, "Towards a neural statistician," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[79] EIA. (2020). *U.S. Energy Information Administration, U.S Energy Facts Explained*. Accessed: Apr. 16, 2021. [Online]. Available: https://www.eia.gov/energyexplained/us-energy-facts/

[80] EIA. (2020). *U.S. Energy Information Administration, Today in Energy*. Accessed: Mar. 29, 2021. [Online]. Available: https://www.eia.gov/todayinenergy/detail.php?id=42635

[81] EIA. (2021). *U.S. Energy Information Administration, Consumption and Efficiency*. Accessed: Apr. 16, 2021. [Online]. Available: https://www.eia.gov/consumption/

[82] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[83] M. Farooq and A. Hafeez, "COVID-ResNet: A deep learning framework for screening of COVID19 from radiographs," 2020, *arXiv:2003.14395*.

[84] FDA. (2020). *3D Printing in FDA'Ts Rapid Response to COVID-19*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.fda.gov/emergency-preparedness-and-response/coronavirus-disease-2019-covid-19/3d-printing-fdas-rapid-response-covid-19

[85] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 259–268.

[86] F. Feng, S. Bao, J. Jin, W. Sun, S. Saigusa, A. Tahmasbi-Sarvestani, and J. Dsa, "Estimation of lead vehicle kinematics using camera-based data for driver distraction detection," *Int. J. Automot. Eng.*, vol. 9, no. 3, pp. 158–164, 2018.

[87] K. Field. (2000). *COVID-19: How Quickly Can Manufacturing Respond to the Surge in Demand?* Accessed: Jul. 18, 2020. [Online]. Available: https://www.fierceelectronics.com/electronics/how-quickly-can-manufacturing-respond-to-surge-demand

[88] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[89] A. Pawar, "Modified model-agnostic meta-learning," in *Proc. IEEE Int. Conf. Mach. Learn. Appl. Netw. Technol. (ICMLANT)*, Dec. 2020, pp. 1–4.

[90] M. Fitzgerald and S. Schwinke. (2016). *General Motors Relies on IoT to Anticipate Customers Needs*. Accessed: Jul. 18, 2020. [Online]. Available: https://sloanreview.mit.edu/article/general-motors-relies-on-iot-to-keep-its-customers-safe-and-secure/

[91] B. Foster, D. Burns, J. Grove, D. Kathan, M. Lee, S. Peirovi, and C. Schilling, "Assessment of demand response and advanced metering," Federal Energy Regulatory Commission, Washington, DC, USA, Tech. Rep., 2017.

[92] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1. New York, NY, USA: Springer, 2001.

[93] D. J. Garcia, M. Mozaffar, H. Ren, J. E. Correa, K. Ehmann, J. Cao, and F. You, "Sustainable manufacturing with cyber-physical discrete manufacturing networks: Overview and modeling framework," *J. Manuf. Sci. Eng.*, vol. 141, no. 2, Feb. 2019, Art. no. 021013.

[94] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, "Conditional neural processes," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1704–1713.

[95] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, and Y. W. Teh, "Neural processes," 2018, *arXiv:1807.01622*.

[96] T. Gilad, N. Rozen-Schiff, P. B. Godfrey, C. Raiciu, and M. Schapira, "MPCC: Online learning multipath transport," in *Proc. 16th Int. Conf. Emerg. Netw. EXperiments Technol.*, Nov. 2020, pp. 121–135.

[97] A. R. Gonçalves, F. J. Von Zuben, and A. Banerjee, "Multi-task sparse structure learning with Gaussian copula models," *J. Mach. Learn. Res.*, vol. 17, no. 33, pp. 1–30, 2016.

[98] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[99] Google. (2019). *Your Chats Stay Private While Messages Improves Suggestions*. Accessed: Apr. 21, 2021. [Online]. Available: https://support.google.com/messages/answer/9327902?hl=en#zippy=

[100] GoogleCloud. (2021). *Googlecloud*. Accessed: Jul. 18, 2020. [Online]. Available: https://cloud.google.com/solutions/iot

[101] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner, "Meta-learning probabilistic inference for prediction," 2018, *arXiv:1805.09921*.

[102] Y. Gordon, "Some inequalities for Gaussian processes and applications," *Isr. J. Math.*, vol. 50, no. 4, pp. 265–289, 1985.

[103] R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Boca Raton, FL, USA: CRC Press, 2020.

[104] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical Bayes," 2018, *arXiv:1801.08930*.

[105] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," 2020, *arXiv:2002.05516*.

[106] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[107] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3315–3323.

[108] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017, *arXiv:1711.10677*.

[109] F. Hartmann, S. Suh, A. Komarzewski, T. D. Smith, and I. Segall, "Federated learning for ranking browser history suggestions," 2019, *arXiv:1911.11807*.

[110] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning With Sparsity: The Lasso and Generalizations* (Monographs on Statistics and Applied Probability), vol. 143. Boca Raton, FL, USA: CRC Press, 2015.

[111] C. Horn and J. Krüger, "Feasibility of connecting machinery and robots to industrial control services in the cloud," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–4.

[112] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," 2020, *arXiv:2004.05439*.

[113] N. Hossein Motlagh, M. Mohammadrezaei, J. Hunt, and B. Zakeri, "Internet of Things (IoT) and the energy sector," *Energies*, vol. 13, no. 2, p. 494, Jan. 2020.

[114] S. M. Hosseini, R. Carli, and M. Dotoli, "Robust optimal energy management of a residential microgrid under uncertainties on demand and renewable power generation," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 618–637, Apr. 2021.

[115] K. Hsieh, G. Ananthanarayanan, P. Bodik, S. Venkataraman, P. Bahl, M. Philipose, P. B. Gibbons, and O. Mutlu, "Focus: Querying large video datasets with low latency and low cost," in *Proc. USENIX OSDI*, 2018, pp. 269–286.

[116] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, "FedMGDA+: Federated learning meets multi-objective optimization," 2020, *arXiv:2006.11489*.

[117] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, "LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data," *PLoS ONE*, vol. 15, no. 4, Apr. 2020, Art. no. e0230706.

[118] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, "LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data," 2018, *arXiv:1811.12629*.

[119] W. Huang, T. Li, D. Wang, S. Du, and J. Zhang, "Fairness and accuracy in federated learning," 2020, *arXiv:2012.10069*.

[120] Y. Hung, V. R. Joseph, and S. N. Melkote, "Analysis of computer experiments with functional response," *Technometrics*, vol. 57, no. 1, pp. 35–44, Jan. 2015.

[121] Z. Huo, B. Gu, and H. Huang, "Training neural networks using features replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6659–6668.

[122] M. Imani, S. F. Ghoreishi, D. Allaire, and U. M. Braga-Neto, "MFBO-SSM: Multi-fidelity Bayesian optimization for fast inference in state-space models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jan./Feb. 2019, pp. 7858–7865.

[123] Rishi Iyengar. (2020). *Can 3D Printing Plug the Coronavirus Equipment Gap?* Accessed: Jul. 18, 2020. [Online]. Available: https://www.cnn.com/2020/04/16/tech/coronavirus-medical-equipment-3d-printing/index.html

[124] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," 2018, *arXiv:1803.05407*.

[125] D. E. Jahn, W. A. Gallus, P. T. T. Nguyen, Q. Pan, K. Cetin, E. Byon, L. Manuel, Y. Zhou, and E. Jahani, "Projecting the most likely annual urban heat extremes in the central united states," *Atmosphere*, vol. 10, no. 12, p. 727, Nov. 2019.

[126] Y. Jang and E. Byon, "Probabilistic characterization of wind diurnal variability for wind resource assessment," *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2535–2544, Oct. 2020.

[127] Y. Jang, E. Byon, E. Jahani, and K. Cetin, "On the long-term density prediction of peak electricity load with demand side management in buildings," *Energy Buildings*, vol. 228, Dec. 2020, Art. no. 110450.

[128] M. Bachl, T. Zseby, and J. Fabini, "Rax: Deep reinforcement learning for congestion control," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[129] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: Scalable adaptation of video analytics," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2018, pp. 253–266.

[130] J. Jiang, Y. Zhou, G. Ananthanarayanan, Y. Shu, and A. A. Chien, "Networked cameras are the new big data clusters," in *Proc. Workshop Hot Topics Video Anal. Intell. Edges (HotEdgeVideo)*, 2019, pp. 1–7.

[131] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019, *arXiv:1909.12488*.

[132] V. R. Joseph, L. Gu, S. Ba, and W. R. Myers, "Space-filling designs for robustness experiments," *Technometrics*, vol. 61, no. 1, pp. 24–37, Jan. 2019.

[133] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, and R. G. D'Oliveira, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[134] 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, Dec. 2019.

[135] L. Kang, V. R. Joseph, and W. A. Brenneman, "Design and modeling strategies for Mixture-of-Mixtures experiments," *Technometrics*, vol. 53, no. 2, pp. 125–136, May 2011.

[136] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proc. ICML*, 2011, pp. 521–528.

[137] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[138] A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," in *Proc. 35 th Int. Conf. Mach. Learn.*, no. 80, 2018, pp. 2525–2534.

[139] P. A. Keane and E. J. Topol, "AI-facilitated health care requires education of clinicians," *Lancet*, vol. 397, no. 10281, p. 1254, Apr. 2021.

[140] A.-M. Kermarrec and F. Taïani, "Want to scale in centralized systems? Think P2P," *J. Internet Services Appl.*, vol. 6, no. 1, pp. 1–12, Aug. 2015.

[141] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[142] Powell Kimberly. (2019). *NVIDIA CLARA Federated Learning to Deliver AI to Hospitals While Protecting Patient Data*. Accessed: Apr. 21, 2021. [Online]. Available: https://blogs.nvidia.com/blog/2019/12/01/clara-federated-learning/

[143] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[144] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, and D. Hassabis, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[145] C. Kloimüllner and G. R. Raidl, "Full-load route planning for balancing bike sharing systems by logic-based benders decomposition," *Networks*, vol. 69, no. 3, pp. 270–289, May 2017.

[146] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[147] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," 2019, *arXiv:1907.09356*.

[148] V. Koltchinskii and M. Yuan, "Sparsity in multiple kernel learning," *Ann. Statist.*, vol. 38, no. 8, pp. 3660–3695, 2010.

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

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

[151] R. Kontar, G. Raskutti, and S. Zhou, "Minimizing negative transfer of knowledge in multivariate Gaussian processes: A scalable and regularized approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3508–3522, Oct. 2021.

[152] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric-condition-based remaining useful life prediction incorporating external factors," *IEEE Trans. Rel.*, vol. 67, no. 1, pp. 41–52, Mar. 2018.

[153] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric modeling and prognosis of condition monitoring signals using multivariate Gaussian convolution processes," *Technometrics*, vol. 60, no. 4, pp. 484–496, Oct. 2018.

[154] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis, "The case for learned index structures," in *Proc. Int. Conf. Manage. Data*, May 2018, pp. 489–504.

[155] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS)*, Jul. 2020, pp. 794–797, doi: 10.1109/WorldS450073.2020.9210355.

[156] A. Kumar and H. Daume, "Learning task grouping and overlap in multi-task learning," 2012, *arXiv:1206.6417*.

[157] S. Laaper, B. Dollar, M. Cotteleer, and B. Sniderman, "Implementing the smart factory: New perspectives for driving value," Deloitte Insights, Deloitte, London, U.K., Tech. Rep., 2020.

[158] A. Lacoste, T. Boquet, N. Rostamzadeh, B. Oreshkin, W. Chung, and D. Krueger, "Deep prior," 2017, *arXiv:1712.05016*.

[159] A. Lacoste, B. Oreshkin, W. Chung, T. Boquet, N. Rostamzadeh, and D. Krueger, "Uncertainty in multitask transfer learning," 2018, *arXiv:1806.07528*.

[160] J.-J. Laffont and D. Martimort, *The Theory of Incentives: The Principal-Agent Model*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[161] F. Lai, J. You, X. Zhu, V. Harsha Madhyastha, and M. Chowdhury, "Sol: A federated execution engine for fast distributed computation over slow networks," in *Proc. USENIX NSDI*, 2020, pp. 273–288.

[162] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," 2020, *arXiv:2010.06081*.

[163] J. Larson, K.-Y. Liang, and K. H. Johansson, "A distributed framework for coordinated heavy-duty vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 419–429, Feb. 2014.

[164] Matt Leonard. (2019). *With Predictive Maintenance, Operators Seek Improved Uptime*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.supplychaindive.com/news/with-predictive-maintenance-operators-seek-improved-uptime/561684/

[165] Matt Leonard. (2019). *Declining Price of IoT Sensors Means Greater Use in Manufacturing*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.supplychaindive.com/news/declining-price-iot-sensors-manufacturing/564980/

[166] H. Leurent and E. D. Boer, *The Next Economic Growth Engine: Scaling Fourth Industrial Revolution Technologies in Production*. Geneva, Switzerland: World Economic Forum, 2018.

[167] D. Li, C. C. Menassa, V. R. Kamat, and E. Byon, "HEAT—Human embodied autonomous thermostat," *Building Environ.*, vol. 178, Jul. 2020, Art. no. 106879.

[168] M. Li, Z. Qin, Y. Jiao, Y. Yang, J. Wang, C. Wang, G. Wu, and J. Ye, "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proc. World Wide Web Conf.*, May 2019, pp. 983–994.

[169] M. Li and R. Kontar, "On negative transfer and structure of latent functions in multi-output Gaussian processes," 2020, *arXiv:2004.02382*.

[170] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, and V. S. A. Talwalkar, "Federated optimization in heterogeneous networks," in *Proc. 3rd MLSys Conf.*, 2018, pp. 429–450.

[171] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smithy, "FedDANE: A federated Newton-type method," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 1227–1231.

[172] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019, *arXiv:1905.10497*.

[173] 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, May 2020.

[174] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," 2020, *arXiv:2012.04221*.

[175] W. Li, F. Milletari, and D. Xu, "Privacy-preserving federated brain tumour segmentation," in *Machine Learning in Medical Imaging*. Cham, Switzerland: Springer, 2019.

[176] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv:1707.09835*.

[177] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," 2017, *arXiv:1705.09056*.

[178] P. Pu Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," 2020, *arXiv:2001.01523*.

[179] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[180] S. Lin, G. Yang, and J. Zhang, "Real-time edge intelligence in the making: A collaborative learning framework via federated meta-learning," 2020, *arXiv:2001.03229*.

[181] H. Liu, J. Lafferty, and L. Wasserman, "The nonparanormal: Semiparametric estimation of high-dimensional undirected graphs," *J. Mach. Learn. Res.*, vol. 10, pp. 1–37, Oct. 2009.

[182] J. Liu and J. Jin, "Diagnosing multistage manufacturing processes with engineering-driven factor analysis considering sampling uncertainty," *J. Manuf. Sci. Eng.*, vol. 135, no. 4, Aug. 2013, Art. no. 041020.

[183] K. Liu, X. Zhang, and J. Shi, "Adaptive sensor allocation strategy for process monitoring and diagnosis in a Bayesian network," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 452–462, Apr. 2014.

[184] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose Bayesian inference algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 5393–5404.

[185] R. Liu, T. Wu, and B. Mozafari, "Adam with bandit sampling for deep learning," in *Proc. NeurIPS*, 2020, pp. 5393–5404.

[186] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020.

[187] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," Purdue Univ., West Lafayette, IN, USA, Tech. Rep. 17-002, 2017.

[188] Z. Liu, Z. Zhang, R. Zhuo, and X. Wang, "Optimal operation of independent regional power grid with multiple wind-solar-hydro-battery power," *Appl. Energy*, vol. 235, pp. 1541–1550, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261918317781, doi: 10.1016/j.apenergy.2018.11.072.

[189] C. Louizos, X. Shi, K. Schutte, and M. Welling, "The functional neural process," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019.

[190] Z. Lu, X. Jiang, and A. Kot, "Deep coupled ResNet for low-resolution face recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 526–530, Apr. 2018, doi: 10.1109/LSP.2018.2810121.

[191] J. Luo, X. Wu, Y. Luo, A. Huang, Y. Huang, Y. Liu, and Q. Yang, "Real-world image datasets for federated learning," 2019, *arXiv:1910.11089*.

[192] Q. Luo and R. Saigal, "Dynamic multiagent incentive contracts: Existence, uniqueness, and implementation," *Mathematics*, vol. 9, no. 1, p. 19, 2021, doi: 10.3390/math9010019.

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

[194] C. Ma, Y. Li, and J. M. Hernández-Lobato, "Variational implicit processes," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4222–4233.

[195] S. Madakam, V. Lake, V. Lake, and V. Lake, "Internet of Things (IoT): A literature review," *J. Comput. Commun.*, vol. 3, no. 5, p. 164, 2015.

[196] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 13153–13164.

[197] H. V. Madhyastha and C. Okwudire, "Remotely controlled manufacturing: A new frontier for systems research," in *Proc. 21st Int. Workshop Mobile Comput. Syst. Appl.*, Mar. 2020, pp. 62–67.

[198] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[199] D. Mahar, W. Fields, J. Reade, P. Zarubin, and S. McCombie, *Non-electronic Parts Reliability Data 2011*. Rome, NY, USA: Reliability Information Analysis Center, 2011.

[200] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 3, pp. 697–702, 2009.

[201] T. Mai, D. Sandor, R. Wiser, and T. Schneider, "Renewable electricity futures study. executive summary," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep., 2012.

[202] C. Malings, M. Pozzi, K. Klima, M. Bergés, E. Bou-Zeid, and P. Ramamurthy, "Surface heat assessment for developed environments: Probabilistic urban temperature modeling," *Comput., Environ. Urban Syst.*, vol. 66, pp. 53–64, Nov. 2017.

[203] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," 2020, *arXiv:2002.10619*.

[204] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 197–210.

[205] N. Masoud and R. Jayakrishnan, "A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem," *Transp. Res. B, Methodol.*, vol. 99, pp. 1–29, May 2017.

[206] R. Mcdonald, M. Mohri, N. Silberman, D. Walker, and G. Mann, "Efficient large-scale distributed training of conditional maximum entropy models," in *Advances in Neural Information Processing Systems*, vol. 22, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2009, [Online]. Available: https://proceedings.neurips.cc/paper/2009/file/d81f9c1be2e08964bf9f24b15f0e4900-Paper.pdf

[207] McKinsey, "The age of analytics: Competing in a data-driven world," McKinsey & Company, Tech. Rep., 2016.

[208] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. 2017, pp. 1273–1282.

[209] E. B. P. Kairouz and H. B. Mcmahan, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1, 2021.

[210] M. McPherson and B. Stoll, "Demand response for variable renewable energy integration: A proposed approach and its impacts," *Energy*, vol. 197, Apr. 2020, Art. no. 117205.

[211] W. Q. Meeker and L. A. Escobar, *Statistical Methods for Reliability Data*. Hoboken, NJ, USA: Wiley, 1998.

[212] Microsoft. (2019). *2019 Manufacturing Trends Report*. Accessed: Jul. 18, 2020. [Online]. Available: https://info.microsoft.com/rs/157-GQE-382/images/EN-U.S.-CNTNT-Report-2019-Manufacturing-Trends.pdf

[213] R. Mohebifard and A. Hajbabaie, "Optimal network-level traffic signal control: A benders decomposition-based solution algorithm," *Transp. Res. B, Methodol.*, vol. 121, pp. 252–274, Mar. 2019.

[214] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. 36th Int. Conf. Mach. Learn.*, K. Chaudhuri and R. Salakhutdinov, Eds. vol. 97, Jun. 2019, pp. 4615–4625.

[215] Nicole Casal Moore. (Nov. 2017). *3-D Printing Gets a Turbo Boost From U-M Technology*. Accessed: Feb. 19, 2020. [Online]. Available: https://news.umich.edu/3-d-printing-gets-a-turbo-boost-from-u-m-technology/

[216] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam, "Delay mitigation in offloaded cloud controllers in industrial IoT," *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[217] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2554–2563.

[218] J. Nagy, J. Oláh, E. Erdei, D. Máté, and J. Popp, "The role and impact of Industry 4.0 and the Internet of Things on the business strategy of the value chain—The case of Hungary," *Sustainability*, vol. 10, no. 10, p. 3491, Sep. 2018.

[219] H. Namkoong, A. Sinha, S. Yadlowsky, and C. J. Duchi, "Radaptive sampling probabilities for non-smooth optimizatio," in *Proc. 34th Int. Conf. Mach. Learn.*, no. 70, 2017, pp. 2574–2583.

[220] S. Nan, M. Zhou, and G. Li, "Optimal residential community demand response scheduling in smart grid," *Appl. Energy*, vol. 210, pp. 1280–1289, Jan. 2018.

[221] M. Neukomm, V. Nubbe, and R. Fares, "Grid-interactive efficient buildings," U.S. Dept. Energy USDOE, Washington, DC, USA, Tech. Rep., 2019.

[222] C. Nguyen, T.-T. Do, and G. Carneiro, "PAC-Bayesian meta-learning with implicit prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 210, 2018, pp. 1280–1289.

[223] H. Nguyen, L.-M. Kieu, T. Wen, and C. Cai, "Deep learning methods in transportation domain: A review," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 998–1004, 2018.

[224] H. T. Nguyen, N. C. Luong, J. Zhao, C. Yuen, and D. Niyato, "Resource allocation in mobility-aware federated learning networks: A deep reinforcement learning approach," in *Proc. IEEE 6th World Forum Internet Things (WF-IoT)*, Jun. 2020, pp. 1–6.

[225] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.

[226] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.

[227] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[228] S. Ning, E. Byon, T. Wu, and J. Li, "A sparse partitioned-regression model for nonlinear system–environment interactions," *IISE Trans.*, vol. 49, no. 8, pp. 814–826, 2017.

[229] NOAA. (2021). *Climate Change: Global Temperature*. Accessed: Apr. 17, 2021. [Online]. Available: https://www.climate.gov/news-features/understanding-climate/climate-change-global-temperature

[230] R. Nock, S. Hardy, W. Henecka, H. Ivey-Law, G. Patrini, G. Smith, and B. Thorne, "Entity resolution and federated learning get a federated resolution," 2018, *arXiv:1803.04035*.
[231] OCR. (2009). *Office for Civil Rights, Research*. Accessed: Apr. 25, 2021. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/guidance/research/index.html
[232] C. E. Okwudire, X. Lu, G. Kumaravelu, and H. Madhyastha, "A three-tier redundant architecture for safe and reliable cloud-based CNC over public internet networks," *Robot. Comput.-Integr. Manuf.*, vol. 62, Apr. 2020, Art. no. 101880.
[233] C. E. Okwudire and H. V. Madhyastha, "Distributed manufacturing for and by the masses," *Science*, vol. 372, no. 6540, pp. 341–342, Apr. 2021.
[234] C. E. Okwudire, S. Huggi, S. Supe, C. Huang, and B. Zeng, "Low-level control of 3D printers from the cloud: A step toward 3D printer control as a service," *Inventions*, vol. 3, no. 3, p. 56, Aug. 2018.
[235] OnStar. (2021). *Welcome to Onstar*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.onstar.com/
[236] OREDA, *OREDA Offshore Reliability Data Handbook*. Bærum, Norway: Det Norske Veritas, 2009.
[237] F. Pallonetto, M. De Rosa, F. D'Ettorre, and D. P. Finn, "On the assessment and control optimisation of demand response programs in residential buildings," *Renew. Sustain. Energy Rev.*, vol. 127, Jul. 2020, Art. no. 109861.
[238] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.
[239] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[240] G. Park and G. Raskutti, "Learning large-scale Poisson dag models based on overdispersion scoring," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 631–639.
[241] M. Patacchiola, J. Turner, E. J. Crowley, M. O'Boyle, and A. Storkey, "Bayesian meta-learning for the few-shot setting via deep kernels," 2019, *arXiv:1910.05199*.
[242] M. Patacchiola, J. Turner, E. J. Crowley, M. O'Boyle, and A. Storkey, "Bayesian meta-learning for the few-shot setting via deep kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16108–16118.
[243] R. Pathak and J. Martin Wainwright, "FedSplit: An algorithmic framework for fast federated optimization," in *Proc. 34th Conf. Neural Inf. Process. Syst.*, 2020, pp. 7057–7066. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/4ebd440d99504722d80de606ea8507da-Paper.pdf
[244] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *J. SIAM*, vol. 3, no. 1, pp. 28–41, 1955.
[245] J. Pearl, *Causality: Models, Reasoning and Inference*, vol. 29. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[246] D. Pelzer, J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt, "A partition-based match making algorithm for dynamic ridesharing," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2587–2598, Oct. 2015.
[247] W. Peng, Z.-S. Ye, and N. Chen, "Bayesian deep-learning-based health prognostics toward prognostics uncertainty," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2283–2293, Mar. 2020.
[248] J. E. Platt, P. D. Jacobson, and S. L. R. Kardia, "Public trust in health information sharing: A measure of system trust," *Health Services Res.*, vol. 53, no. 2, pp. 824–845, Apr. 2018.
[249] M. Plumlee, "Bayesian calibration of inexact computer models," *J. Amer. Stat. Assoc.*, vol. 112, no. 519, pp. 1274–1285, Jul. 2017.
[250] M. Plumlee, "Computer model calibration with confidence and consistency," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 81, no. 3, pp. 519–545, Jul. 2019.
[251] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, Dec. 2008.
[252] M. E. Porter and J. E. Heppelmann, "How smart, connected products are transforming competition," *Harvard Bus. Rev.*, vol. 92, no. 11, pp. 64–88, 2014.
[253] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica, "Low latency geo-distributed data analytics," in *Proc. ACM Conf. Special Interest Group Data Commun.*, Aug. 2015.
[254] P. Qiu, "Big data? Statistical process control can help!" *Amer. Statistician*, vol. 74, no. 4, pp. 329–344, Oct. 2020.
[255] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," in *Proc. OpenAI*, 2019.
[256] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.
[257] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," 2019, *arXiv:1906.04329*.
[258] G. Raskutti and M. Mahoney, "A statistical perspective on randomized sketching for ordinary least-squares," *J. Mach. Learn. Res.*, vol. 17, no. 213, pp. 1–31, 2016.
[259] G. Raskutti, M. J. Wainwright, and B. Yu, "Minimax-optimal rates for sparse additive models over kernel classes via convex programming," *J. Mach. Learn. Res.*, vol. 13, pp. 398–427, Feb. 2012.
[260] G. Raskutti, M. J. Wainwright, and B. Yu, "Early stopping and non-parametric regression: An optimal data-dependent stopping rule," *J. Mach. Learn. Res.*, vol. 15, pp. 335–366, Jan. 2014.
[261] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[262] S. Ravi and A. Beatson, "Amortized Bayesian meta-learning," in *Proc. ICLR (Poster)*, 2019.
[263] A. Srinivasan, A. Bharadwaj, M. Sathyan, and S. Natarajan, "Optimization of image embeddings for few shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2021.
[264] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence," *Electron. J. Statist.*, vol. 5, pp. 935–980, Jan. 2011.
[265] S. Reddi, M. Zaheer, D. Sachan, S. Kale, and S. Kumar, "Adaptive methods for nonconvex optimization," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NIPS)*, 2018.
[266] J. S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: https://openreview.net/forum?id=LkFG3lB13U5
[267] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8388–8399.
[268] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *J. Global Optim.*, vol. 56, no. 3, pp. 1247–1293, 2013.
[269] Rockwell. (2021). *The Connected Enterprise*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.rockwellautomation.com/en-us/capabilities/connected-enterprise.html
[270] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[271] J. Ryan, E. Ela, D. Flynn, and M. O'Malley, "Variable generation, reserves, flexibility and policy interactions," in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, Jan. 2014, pp. 2426–2434.
[272] K. E. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math.*, vol. 16, pp. 3–43, Jan. 2016. [Online]. Available: https://stanford.edu/~boyd/papers/pdf/monotone_primer.pdf
[273] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017, *arXiv:1703.03864*.
[274] M. Salmi, J. S. Akmal, E. Pei, J. Wolff, A. Jaribion, and S. H. Khajavi, "3D printing in COVID-19: Productivity estimation of the most promising open source solutions in emergency situations," *Appl. Sci.*, vol. 10, no. 11, p. 4004, Jun. 2020.
[275] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Nov. 2019.
[276] Samsung. (2021). *Samsung Galaxy Watch Active 2*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.samsung.com/us/mobile/wearables/galaxy-watch-active-2/
[277] Y. Sannikov, "A continuous-time version of the principal–agent problem," *Rev. Econ. Stud.*, vol. 75, no. 3, pp. 957–984, Jul. 2008.
[278] K. V. Sarma, S. Harmon, T. Sanford, H. R. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. G. Raman, R. Kulkarni, B. J. Wood, P. L. Choyke, A. M. Priester, L. S. Marks, S. S. Raman, D. Enzmann, B. Turkbey, W. Speier, and C. W. Arnold, "Federated learning improves site performance in multicenter deep learning without data sharing," *J. Amer. Med. Inform. Assoc.*, vol. 28, no. 6, pp. 1259–1264, Jun. 2021.

[279] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints," 2019, *arXiv:1910.01991*.

[280] P. Scarabaggio, S. Grammatico, R. Carli, and M. Dotoli, "Distributed demand side management with stochastic wind power forecasting," *IEEE Trans. Control Syst. Technol.*, early access, Feb. 15, 2021, doi: 10.1109/TCST.2021.3056751.

[281] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," 2013, *arXiv:1309.2388*.

[282] C. Seward, T. Unterthiner, U. Bergmann, N. Jetchev, and S. Hochreiter, "First order generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4567–4576.

[283] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proc. 31st Int. Conf. Mach. Learn.*, E. P. Xing and T. Jebara, Eds., vol. 32, 2014, pp. 1000–1008.

[284] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, and S. Bakas, "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020.

[285] N. Shi, F. Lai, R. Al Kontar, and M. Chowdhury, "Fed-ensemble: Improving generalization through model ensembling in federated learning," 2021, *arXiv:2107.10663*.

[286] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," 2019, *arXiv:1909.09145*.

[287] K. Singh, T. S. Valley, S. Tang, B. Y. Li, F. Kamran, M. W. Sjoding, J. Wiens, E. Otles, J. P. Donnelly, and M. Y. Wei, "Evaluating a widely implemented proprietary deterioration index model among hospitalized COVID-19 patients," *Ann. Amer. Thoracic Soc.*, vol. 18, no. 7, pp. 1129–1137, 2021.

[288] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–19.

[289] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017, *arXiv:1703.05175*.

[290] S. E. Spear and S. Srivastava, "On repeated moral hazard with discounting," *Rev. Econ. Stud.*, vol. 54, no. 4, pp. 599–617, 1987.

[291] J. S. Srai, M. Kumar, G. Graham, W. Phillips, J. Tooze, S. Ford, P. Beecher, B. Raj, M. Gregory, M. K. Tiwari, B. Ravi, A. Neely, R. Shankar, F. Charnley, and A. Tiwari, "Distributed manufacturing: Scope, challenges and opportunities," *Int. J. Prod. Res.*, vol. 54, no. 23, pp. 6917–6935, Dec. 2016.

[292] Robest Stevens. (2020). *Why DIY 3D-Printed Face Masks and Shields Are So Risky*. Accessed: Jul. 18, 2020. [Online]. Available: https://slate.com/technology/2020/04/diy-3d-printed-face-masks-shields-coronavirus.html

[293] Alice Su. (2020). *Faulty Masks. Flawed Tests. China's Quality Control Problem in Leading Global COVID-19 Fight*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.latimes.com/world-nation/story/2020-04-10/china-beijing-supply-world-coronavirus-fight-quality-control

[294] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments*. Cambridge, MA, USA: MIT Press, 2012.

[295] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *Proc. Chin. Comput. Linguistics*, 2019, pp. 194–206.

[296] A. Tafreshian and N. Masoud, "Trip-based graph partitioning in dynamic ridesharing," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 532–553, May 2020.

[297] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu, "DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6155–6165.

[298] P. Tossou, B. Dura, F. Laviolette, M. Marchand, and A. Lacoste, "Adaptive deep kernel learning," 2019, *arXiv:1905.12131*.

[299] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Learning to route," in *Proc. ACM HotNets*, 2017, pp. 185–191.

[300] J. Vanschoren, "Meta-learning: A survey," 2018, *arXiv:1810.03548*.

[301] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[302] De Brouwer Walter. (2019). *The Federated Future is Ready for Shipping*. Accessed: Apr. 21, 2021. [Online]. Available: https://medium.com/_doc_ai/the-federated-future-is-ready-for-shipping-d17ff40f43e3

[303] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[304] J. Wang, S. Chung, A. AlShelahi, R. Kontar, E. Byon, and R. Saigal, "Look-ahead decision making for renewable energy: A dynamic 'predict and store' approach," *Appl. Energy*, vol. 296, Aug. 2021, Art. no. 117068.

[305] K. Wang, J. Li, and F. Tsung, "Distribution inference from early-stage stationary data streams by transfer learning," *IISE Trans.*, pp. 1–25, Mar. 2021.

[306] K. Wang, R. Mathews, C. Kiddon, H. Eichner, F. Beaufays, and D. Ramage, "Federated evaluation of on-device personalization," 2019, *arXiv:1910.10252*.

[307] 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.

[308] Z. Wang, Y. Zhao, P. Yu, R. Zhang, and C. Chen, "Bayesian meta sampling for fast uncertainty adaptation," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[309] G. M. Weber, "Federated queries of clinical data repositories: Scaling to a national network," *J. Biomed. Informat.*, vol. 55, pp. 231–236, Jun. 2015.

[310] G. M. Weber, S. N. Murphy, A. J. Mcmurry, D. MacFadden, D. J. Nigrin, S. Churchill, and I. S. Kohane, "The shared health research information network (SHRINE): A prototype federated query tool for clinical data repositories," *J. Amer. Med. Inform. Assoc.*, vol. 16, no. 5, pp. 624–630, Sep. 2009.

[311] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1913–1922.

[312] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farhad, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," 2019, *arXiv:1911.00222*.

[313] P. Wei, F. Liu, and C. Tang, "Reliability and reliability-based importance analysis of structural systems using multiple response Gaussian process model," *Rel. Eng. Syst. Saf.*, vol. 175, pp. 183–195, Jul. 2018.

[314] P. Wellener, "Deloitte and mapi smart factory study: Capturing value through the digital journey," Deloitte Insights MAPI, Deloitte, London, U.K., Tech. Rep., 2019.

[315] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 681–688.

[316] B. S. Wessler, J. Nelson, J. G. Park, H. McGinnes, G. Gulati, R. Brazil, B. Van Calster, E. Venema, E. Steyerberg, and J. K. Paulus, "External validations of cardiovascular clinical prediction models: A large-scale review of the literature," *medRxiv*, vol. 14, no. 8, 2021, Art. no. 007858.

[317] C. K Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006.

[318] K. Winstein and H. Balakrishnan, "TCP ex machina: Computer-generated congestion control," in *Proc. ACM SIGCOMM Conf.*, Aug. 2013, pp. 123–134.

[319] R. F. Wolff, K. G. Moons, R. D. Riley, P. F. Whiting, M. Westwood, G. S. Collins, J. B. Reitsma, J. Kleijnen, and S. Mallett, "PROBAST: A tool to assess the risk of bias and applicability of prediction model studies," *Ann. Internal Med.*, vol. 170, no. 1, pp. 51–58, 2019.

[320] Christina Wong. (2019). *AI Chips for Self Driving Cars Will a Be 10 Billion Market by 2024*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.nextbigfuture.com/2019/03/ai-chips-for-self-driving-cars-will-a-be-10-billion-market-by-2024.html

[321] W. H. Woodall and E. del Castillo, "An overview of George Box's contributions to process monitoring and feedback adjustment," *Appl. Stochastic Models Bus. Ind.*, vol. 30, no. 1, pp. 53–61, Jan. 2014.

[322] C. J. Wu and M. S. Hamada, *Experiments: Planning, Analysis, and Optimization*, vol. 552. Hoboken, NJ, USA: Wiley, 2011.

[323] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," 2020, *arXiv:2012.00661*.

[324] T. Wu, J. Peurifoy, I. L. Chuang, and M. Tegmark, "Meta-learning autoencoders for few-shot prediction," 2018, *arXiv:1807.09912*.

[325] F. Xiong, Y. Xiong, W. Chen, and S. Yang, "Optimizing Latin hypercube design for sequential sampling of computer experiments," *Eng. Optim.*, vol. 41, no. 8, pp. 793–810, Aug. 2009.

[326] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021.
[327] N. Yampikulsakul, E. Byon, S. Huang, S. Sheng, and M. You, "Condition monitoring of wind power system with nonparametric regression analysis," *IEEE Trans. Energy Convers.*, vol. 29, no. 2, pp. 288–299, Jun. 2014.
[328] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning *in situ*: A randomized experiment in video streaming," in *Proc. USENIX NSDI*, 2020, pp. 495–511.
[329] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Feb. 2019.
[330] S. Yang, B. Ren, X. Zhou, and L. Liu, "Parallel distributed logistic regression for vertical federated learning without third-party coordinator," 2019, *arXiv:1911.09824*.
[331] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving Google keyboard query suggestions," 2018, *arXiv:1812.02903*.
[332] Y. Yaz, C.-S. Foo, S. Winkler, K.-H. Yap, G. Piliouras, and V. Chandrasekhar, "The unusual effectiveness of averaging in GAN training," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[333] Z.-S. Ye, Y. Hong, and Y. Xie, "How do heterogeneities in operating environments affect field failure predictions and test planning?" *Ann. Appl. Statist.*, vol. 7, no. 4, pp. 2249–2271, Dec. 2013.
[334] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7343–7353.
[335] M. You, E. Byon, J. Jin, and G. Lee, "When wind travels through turbines: A new statistical approach for characterizing heterogeneous wake effects in multi-turbine wind farms," *IISE Trans.*, vol. 49, no. 1, pp. 84–95, Jan. 2017.
[336] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," 2020, *arXiv:2002.04758*.
[337] H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," in *Proc. 34th Conf. Neural Inf. Process. Syst.*, 2020, pp. 5332–5344. [Online]. Available: https://papers.nips.cc/paper/2020/file/39d0a8908fbe6c18039ea8227f827023-Supplemental.pdf
[338] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.
[339] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4349–4360, Oct. 2012.
[340] X. Yue and R. Kontar, "The Renyi Gaussian process: Towards improved generalization," 2019, *arXiv:1910.06990*.
[341] X. Yue and R. Kontar, "Variational inference of joint models using multivariate Gaussian convolution processes," 2019, *arXiv:1903.03867*.
[342] X. Yue, M. Nouiehed, and R. Al Kontar, "GIFAIR-FL: An approach for group and individual fairness in federated learning," 2021, *arXiv:2108.02741*.
[343] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7252–7261.
[344] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *Artificial Intelligence and Statistics*, 2017, pp. 962–970.
[345] Y. Zeng, H. Chen, and K. Lee, "Improving fairness via federated learning," 2021, *arXiv:2110.15545*.
[346] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. A. Lee, "AWStream: Adaptive wide-area streaming analytics," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2018.
[347] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008–2026, Aug. 2019.
[348] D. Y. Zhang, Z. Kou, and D. Wang, "FairFL: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 1051–1060.
[349] D. Zhang, C. C. Chan, and G. Y. Zhou, "Enabling industrial Internet of Things (IIoT) towards an emerging smart energy system," *Global Energy Interconnection*, vol. 1, no. 1, pp. 39–47, Jan. 2018.
[350] Mi Zhang. (2019). *Federated Learning: The Future of Distributed Machine Learning*. Accessed: Jul. 18, 2020. [Online]. Available: https://medium.com/syncedreview/federated-learning-the-future-of-distributed-machine-learning-eec95242d897
[351] X. Zhang and M. Hong. (2021). *On the Connection Between FEDDYN and FEDPD*. [Online]. Available: http://people.ece.umn.edu/~mhong/FedDyn_FedPD.pdf
[352] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A federated learning framework with optimal rates and adaptivity to non-IID data," 2020, *arXiv:2005.11418*.
[353] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," 2012, *arXiv:1203.3536*.
[354] M. Zhao, J. Yin, S. An, J. Wang, and D. Feng, "Ridesharing problem with flexible pickup and delivery locations for app-based transportation service: Mathematical modeling and decomposition methods," *J. Adv. Transp.*, vol. 2018, pp. 1–21, Jul. 2018.
[355] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.
[356] K. Zhou, S. Yang, and Z. Shao, "Energy internet: The business perspective," *Appl. Energy*, vol. 178, pp. 212–222, Sep. 2016.
[357] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019.
[358] J. E. Zimmerman, A. A. Kramer, D. S. McNair, and F. M. Malila, "Acute physiology and chronic health evaluation (APACHE) IV: Hospital mortality assessment for today's critically ill patients," *Crit. Care Med.*, vol. 34, no. 5, pp. 1297–1310, May 2006.
[359] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems*, vol. 23, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2010, pp. 733–742. [Online]. Available: https://proceedings.neurips.cc/paper/2010/file/abea47ba24142ed16b7d8fbf2c740e0d-Paper.pdf
[360] P. Zou, Q. Chen, Q. Xia, G. He, and C. Kang, "Evaluating the contribution of energy storages to support large-scale renewable generation in joint energy and ancillary service markets," *IEEE Trans. Sustain. Energy*, vol. 7, no. 2, pp. 808–818, Apr. 2016.
[361] Y. Zou and X. Lu, "Gradient-EM Bayesian meta-learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20865–20875.

RAED KONTAR received the B.E. degree in civil engineering in 2014, the M.S. degree in statistics in 2017, and the Ph.D. degree in industrial and systems engineering in 2018. He is currently an Assistant Professor with the Department of Industrial and Operations Engineering, University of Michigan. His research focuses on data science using probabilistic models. He is funded by NSF and NIH and has received seven best paper awards.

NAICHEN SHI received the Bachelor of Science degree in physics from Peking University. He is currently pursuing the Ph.D. degree with the Department of Industrial and Operations Engineering, University of Michigan. His research interests include federated learning and high dimensional Bayesian inference.

XUBO YUE received the Bachelor of Science degree in biomedical sciences and applied mathematics from the University of Macau and the Master of Science degree in biostatistics from the University of Michigan, Ann Arbor, where he is currently pursuing the Ph.D. degree with the Department of Industrial and Operations Engineering. His research interests include Gaussian processes, federated learning, and Bayesian optimization.

SEOKHYUN CHUNG received the Bachelor of Science and Master of Science degrees in industrial and management engineering from Korea University. He is currently pursuing the Ph.D. degree with the Department of Industrial and Operations Engineering, University of Michigan. His research interests include Bayesian predictive analytics on federated systems and generalization of deep neural networks.

EUNSHIN BYON (Member, IEEE) received the Ph.D. degree from the Industrial and Systems Engineering Department, Texas A&M University, College Station, TX, USA, in 2010. She is currently an Associate Professor with the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA. Her research interests include data analytics, quality and reliability engineering, condition monitoring, operations and maintenance optimization, and uncertainty quantification. She is a member of IIE and INFORMS.

MOSHARAF CHOWDHURY (Member, IEEE) is currently a Morris Wellman Assistant Professor of computer science and engineering with the University of Michigan, Ann Arbor, where he leads the SymbioticLab on application-infrastructure co-design in diverse networking conditions. He invented coflows and is a co-creator of Apache Spark. Artifacts from his research are widely used in modern cloud datacenters. He has received numerous individual awards, fellowships, and best paper awards, thanks to his amazing students and collaborators.

JIONGHUA (JUDY) JIN (Member, IEEE) received the Ph.D. degree from the University of Michigan. She is currently a Professor with the Department of Industrial and Operations Engineering, University of Michigan. Her major research interests include data analytics and quality engineering for improving complex system operations and quality. Her data fusion research emphasizes a synergistic fusion of engineering models and simulations with advanced data analytics methods across statistics, machine learning, system control, and optimization methods. She is an Elected Fellow of ASME, IISE, and INFORMS. Her research has received numerous awards, including the Presidential Award (NSF-PECASE), the NSF-CAREER Award, and 14 best paper awards.

WISSAM KONTAR received the Bachelor of Engineering degree in civil and environmental engineering from the American University of Beirut, Lebanon. He is currently pursuing the Ph.D. degree with the Department of Civil and Environmental Engineering, University of Wisconsin–Madison. His research interests include developing decision-making methodologies specifically tailored for transportation networks in the era of connected and automated systems.

NEDA MASOUD received the Bachelor of Science degree in industrial engineering, the Master of Science degree in physics, and the Ph.D. degree in civil and environmental engineering. She is currently an Assistant Professor with the Department of Civil and Environmental Engineering, University of Michigan. Her research interests include devising operational and planning tools to facilitate the transition into the next generation of transportation systems, which are envisioned to be connected, automated, electrified, and shared.

MAHER NOUIEHED received the M.S. degree in operations research engineering under the supervision of Prof. Sheldon Ross and the Ph.D. degree in industrial engineering from the University of Southern California, under the supervision of Prof. Meisam Razaviyayn and Prof. Jong Shi-Pang. He is currently an Assistant Professor with the Department of Industrial Engineering and Management, American University of Beirut (AUB). He is interested in studying the computational and theoretical aspects of mathematical optimization problems that arise from machine learning and various fields of science and engineering.

CHINEDUM E. OKWUDIRE received the Ph.D. degree in mechanical engineering from The University of British Columbia, in 2009. He joined the Mechanical Engineering Faculty, University of Michigan, in 2011. Prior to joining Michigan, he was the Mechatronic Systems Optimization Team Leader with DMG Mori, USA, based in Davis, CA, USA. His research interests include exploiting knowledge at the intersection of machine design, control and, more-recently, computer science, to boost the performance of manufacturing automation systems at low cost. He has received a number of awards, including the CAREER Award from the National Science Foundation, the Young Investigator Award from the International Symposium on Flexible Automation, the Outstanding Young Manufacturing Engineer Award from the Society of Manufacturing Engineers, the Ralph Teetor Educational Award from SAE International, and the Russell Severance Springer Visiting Professorship from UC Berkeley. He has coauthored a number of best paper award winning articles in the areas of control and mechatronics.

GARVESH RASKUTTI received the Ph.D. degree in statistics from UC Berkeley. He was a Postdoctoral Fellow with the Statistical and Applied Mathematical Sciences Institute (SAMSI). He is currently an Associate Professor in statistics with the University of Wisconsin–Madison. His research interests include high-dimensional statistics, statistical learning theory and algorithms, graphical models, and time series models.

ROMESH SAIGAL received the B.Tech. and M.Tech. degrees from IIT Kharagpur, Kharagpur, India, and the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA. He is currently a Professor of industrial and operations engineering with the University of Michigan, Ann Arbor, MI, USA. He has been on the faculty with the Haas School of Business, University of California Berkeley, and the Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA. He also teaches courses in optimization, stochastic processes, and financial engineering. His research interests include applications of financial engineering tools in managing risks in transportation and energy sectors.

KARANDEEP SINGH received the M.D. and M.M.Sc. degrees, and the master's degree in medical sciences in biomedical informatics from the Harvard Medical School. He completed his medical education at the University of Michigan Medical School. He is board certified in internal medicine, nephrology, and clinical informatics. He is currently an Assistant Professor of learning health sciences, internal medicine, urology, and information with the University of Michigan. He also directs the Machine Learning for Learning Health Systems Lab, which focuses on using machine learning and biomedical informatics methods to understand and improve health at scale. He also chairs the Michigan Medicine Clinical Intelligence Committee, which oversees the implementation of machine learning models across the health system. He completed his internal medicine residency at the UCLA Medical Center, where he served as a Chief Resident, and the nephrology fellowship at the combined Brigham and Women's Hospital/Massachusetts General Hospital Program in Boston, MA, USA.

ZHI-SHENG YE (Senior Member, IEEE) received the joint B.E. degree in material science, engineering, and economics from Tsinghua University, Beijing, China, in 2008, and the Ph.D. degree in industrial and systems engineering from the National University of Singapore, Singapore, in 2012. He is currently an Assistant Professor with the Department of Industrial Systems Engineering and Management, National University of Singapore. His research interests include reliability engineering, complex systems modeling, and industrial statistics.