This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3407584

Decentralized Federated Learning: A Survey and Perspective

Liangqi Yuan, *Student Member, IEEE*, Ziran Wang, *Member, IEEE*, Lichao Sun, *Member, IEEE*, Philip S. Yu, *Life Fellow, IEEE*, and Christopher G. Brinton, *Senior Member, IEEE*

***Abstract*—Federated learning (FL) has been gaining attention for its ability to share knowledge while maintaining user data, protecting privacy, increasing learning efficiency, and reducing communication overhead. Decentralized FL (DFL) is a decentralized network architecture that eliminates the need for a central server in contrast to centralized FL (CFL). DFL enables direct communication between clients, resulting in significant savings in communication resources. In this paper, a comprehensive survey and profound perspective are provided for DFL. First, a review of the methodology, challenges, and variants of CFL is conducted, laying the background of DFL. Then, a systematic and detailed perspective on DFL is introduced, including iteration order, communication protocols, network topologies, paradigm proposals, and temporal variability. Next, based on the definition of DFL, several extended variants and categorizations are proposed with state-of-the-art (SOTA) technologies. Lastly, in addition to summarizing the current challenges in the DFL, some possible solutions and future research directions are also discussed.**

***Index Terms*—Federated learning, decentralized learning, network, privacy preservation, internet of things (IoT).**

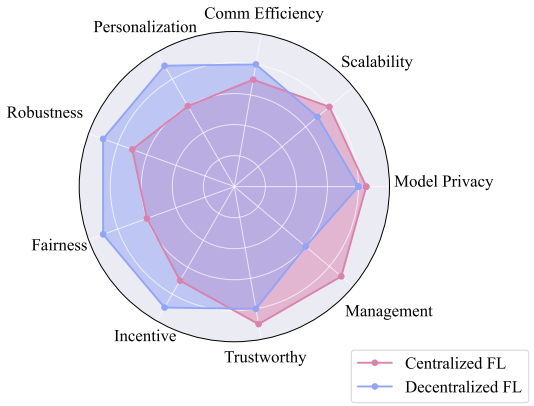


Fig. 1. Comparative analysis between centralized FL and decentralized FL across various performance metrics. Each axis represents a metric with the plotted values indicating the relative strength of the respective FL approach in that domain.

I. Introduction

FEDERATED learning (FL) is a decentralized learning paradigm with natural privacy-preserving capabilities, which shares only model weights instead of user data [1]. Federated learning was first proposed by Google researchers in 2016 [2] and was applied to build a language model collaboration framework on Google Keyboard to learn whether people clicked on recommended suggestions and contextual information [3]. In 2020, Google researchers expanded the concept of FL to federated analytics [4]–[10], extending from learning tasks to collaborative computing, data analysis, and inference, further deploying it within Google Keyboard. FL has demonstrated its excellent capabilities in various areas, including intelligent transportation, internet of things (IoT), healthcare, manufacturing, agriculture, energy, remote sensing, and more [11]–[28]. FL also breaks geographical limitations allowing efficient collaboration worldwide [29]. Researchers employed FL to aggregate data from 20 institutes worldwide to train a universal model to predict clinical outcomes of COVID-19 patients [30]. FL improves the generalization capability of the model to include knowledge of diverse data. Other researchers have also used FL to aggregate data from 71 sites for rare cancer boundary detection, which greatly enriches the dataset to support research on rare diseases [31].

Traditional FL focuses on the decentralized learning and centralized aggregation paradigm established by data parallelism. Data parallelism refers to the situation where the raw data of the clients is generated in parallel locally, and this raw data is neither sent out nor visible to others. Each client trains a model based on its local data and then communicates the model parameters with the server to ensure the effective integration of learning results from each client and obtain a global model. An FL taxonomy refers to the number and nature of clients participating in the learning network, including cross-silo and cross-device FL frameworks [1]. The clients in cross-silo FL usually are different organizations, research institutions, data centers, etc., which may have more reliable communication, computational resources, and a large amount of data. The clients in cross-device FL are huge mobile or IoT devices, which can encounter potential bottlenecks in communication and computation. Another FL taxonomy is considered for differences in data distribution among clients, including horizontal, vertical, and transfer [32]. In horizontal FL, clients have more similar sample features and fewer identical users. Clients in vertical FL have more similar users and fewer similar sample features. Federated transfer learning

Manuscript received May 03, 2024; accepted 21 May 2024. This work is supported by the ONR under Grant N000142112472, the NSF under Grant CNS-2146171, and the NSF under Grant CPS-2313109.

L. Yuan, Z. Wang, and C. G. Brinton are with the College of Engineering, Purdue University, West Lafayette, IN 47907, USA (e-mail: liangqiy@purdue.edu; ryanwang11@hotmail.com; cgb@purdue.edu).

L. Sun is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015, USA (e-mail: lis221@lehigh.edu).

P. S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA (e-mail: psyu@uic.edu).

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3407584

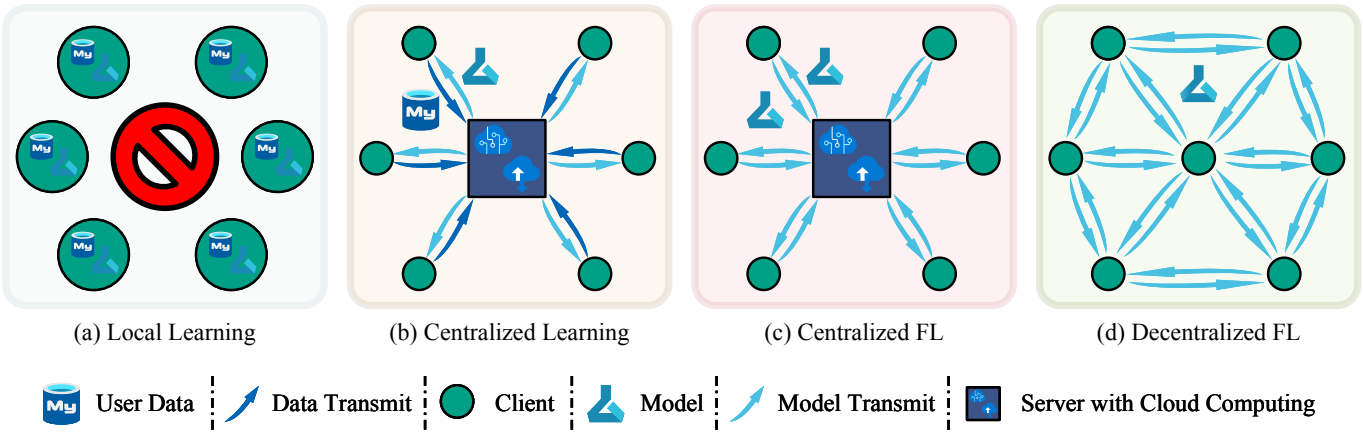


Fig. 2. Illustration of local learning, centralized learning, CFL, and DFL. (a) Clients are trained with local user data only. The clients neither share raw data nor communicate with each other. (b) After clients send the user data packets to the server, the server trains a general model using all the data. The generalized model is then shared with all clients. (c) Clients send the locally trained model parameters to the server. The server aggregates all the local models and then transmits the aggregated global model parameters to all the clients. (d) Clients share their locally trained model with other clients. Subsequent clients then continue to learn, personalize, and adapt the model locally, while also exchanging and propagating the model parameters that possess local knowledge.

clients have neither many similar sample features nor similar users.

In this paper, we present a thorough investigation into decentralized FL (DFL) and offer novel perspectives on its taxonomies. Distinguishing from the conventional centralized FL (CFL) that relies on a central server for aggregation, we specifically focus on the less-explored DFL framework, which operates independently of a central server. Fig. 1 illustrates a comparison between CFL and DFL across nine key evaluation metrics, which are the focal points of current state-of-the-art (SOTA) research and worthy of further investigation [33]–[48]. Given their inherent characteristics, CFL and DFL each exhibit unique advantages in different applications.

Fig. 2 shows the illustration of local learning, centralized learning, CFL, and DFL. In the local learning strategy, the user data and trained model of each client are only used locally, and they do not communicate with any other clients or servers, as shown in Fig. 2(a), but this may lead to overfitting. Alternatively shown in Fig. 2(b), the centralized learning strategy involves the transmission of raw data in the communication between clients and the server, which consolidates and centralizes the learning process but does not guarantee the privacy of the users. Both of these strategies are often used by researchers as baselines to compare with FL.

CFL is a centralized structure where a server will communicate, coordinate, and manage all clients. Fig. 2(c) shows the communication between clients and the server. Clients learn on local data and then upload the trained model parameters to the server. The server aggregates the local models and then shares the global model with the clients. The idea is that all clients contribute to one global model, and the one global model is applied to all clients. For CFL, clients only share the trained local model parameters with the server but not the users' raw data. FL not only protects users' privacy and improves learning efficiency, but also saves communication resources when the model size is much smaller than the data size.

DFL is a decentralized structure in which clients communicate and share model parameters with each other without any server. There are relevant designations in the recent literature, such as peer-to-peer FL [49], server-free FL [50], serverless FL [51], device-to-device FL [52], swarm learning [53], etc. Fig. 2(d) shows clients communicating directly with other clients without server coordination. Since there is no unified coordination and configuration of servers, the communication network between clients is more diverse. For the DFL discarding the server is considered to be more customizable, which can further save communication and computational resources with higher confidence in diverse variants. The pointing and peer connections in the communication network are adaptively configured and changed according to the scenario, which is one of the advantages of DFL. In addition to the typical line, ring, and fully connected peer connection types, it is conceivable to connect based on geographical neighbors, the similarity of clients, communication protocols, etc.

The concept of DFL was first proposed in the year 2018 [54]. As of June 1, 2023, a search on Google Scholar yields 1,350 results related to DFL, with a substantial number of 652 contributions coming from the year 2022 alone. The research associated with DFL exhibits a persistent exponential growth trajectory. DFL has received extensive attention as an emerging framework [55]–[60]. The most significant advantage of DFL is that it eliminates the communication resources needed for the server as an intermediary step and the high bandwidth requirements associated with it. Xu *et al.* [61] listed DFL, model compression, selective client communication, and low communication frequency as four ways to reduce communication costs. Lian *et al.* [62] demonstrated the advantages of decentralized learning over centralized learning, especially since the number of clients in decentralized learning is proportional to the speedup.

Although FL has shown unprecedented advantages, most of the current research has been limited to CFL. DFL, as an essential branch in FL, is proliferating and offering benefits over CFL. Recent surveys have focused more on CFL, with

TABLE I
COMPARISON OF RELATED SURVEYS OF DECENTRALIZED FEDERATED LEARNING

Survey	Time	Focused DFL Topic	Application Scenarios	Order	Protocol	Topology	Paradigm	Variability
[1]	2021	Introduction	✗	✗	✗	✗	✗	✗
[63]	2021	Introduction	✗	✗	✗	✗	✗	✗
[64]	2021	UAV		✗	✗	✗	✗	✗
[65]	2021	IoT	City	✗	✗	✗	✗	✗
[66]	2022	Blockchain	✗	✗	✗	✗	✗	✗
[67]	2022	Survey	✗	✗	✗	✗	✗	✗
[68]	2022	Survey & Reflection	IoT Finance	✗	✗	✗	✗	✗
[69]	2023	Tutorial	✗	✗	✗	✗	✗	✗
[70]	2023	Survey	✗	✗	✗	✗	✗	✗
[71]	2023	Topology	✗	✗	✗	✓	✗	✓
[72]	2023	CAV		✗	✗	✓	✗	✗
[73]	2023	Survey & Tutorial		✗	✓	✓	✗	✗
[74]	2024	Security and Privacy	✗	✗	✗	✗	✗	✗
Ours	2024	Survey & Perspective		✓	✓	✓	✓	✓

✓ Included, ✗ Not mentioned.
: Connected and Automated Vehicles (CAVs), : Healthcare, : Industrial IoT (IIoT), : Mobile Services,
: Unmanned Aerial Vehicle (UAVs) and Satellites, : Social Networks, : Artificial General Intelligence (AGI)

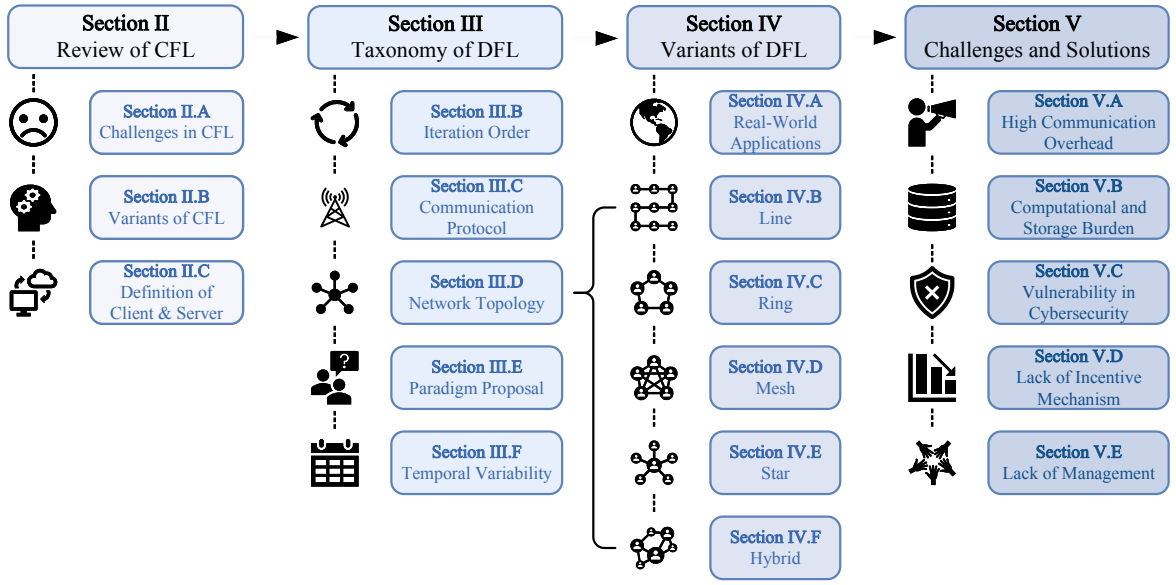


Fig. 3. Roadmap for this perspective paper.

less attention given to DFL [61], [75]–[77]. Furthermore, there is a lack of a comprehensive, in-depth, and insightful survey that establishes the logic of building a DFL system, including iteration, communication protocol, network topology, paradigm, and more. This paper begins with a review of CFL, summarizing its challenges and various extended variants as potential solutions that can be compared and analogized with DFL. As an emerging field, this survey aims to fill gaps in the DFL survey literature by covering perspective papers that are currently not included. It systematically integrates and categorizes the SOTA research in DFL. A detailed and comprehensive comparison of our survey with other related DFL surveys can be found in Table I.

The contributions of this paper are:

- We provide a description of CFL, summarize the challenges, and offer a detailed introduction to the various variants, their roles, addressed issues, and advantages.
- We systematically define and describe five taxonomies of DFL, including iteration order, communication protocol, network topology, paradigm proposal, and temporal variability. To the best of our knowledge, this is the first comprehensive and insightful perspective paper for DFL.

- Based on the network topology, we propose and envision five variants of DFL to categorize the recent literature, anticipate potential application scenarios, and highlight the advantages.
- We summarize five current challenges, possible solutions, and future research directions for DFL.

The presentation of this paper is summarized as shown in Fig. 3. Section II reviews the history of CFL, the existing challenges, and some variants as potential solutions. Section III provides the definitions and descriptions of DFL communication protocol, network topology, and paradigm proposal. Section IV demonstrates several variants in DFL, followed by Section V analyzing the challenges of DFL. Finally, Section VI provides a summary of this paper.

II. Review of Centralized Federated Learning

McMahan *et al.* [2] proposed the first mature and most popular FL algorithm, federated averaging (FedAvg). At each communication round, clients upload their trained local models to the server, and the server weighted averages all local models according to the number of client samples. Based on FedAvg, various derivation and optimization schemes exist to address the challenges in the FL algorithm [78], [79]. Li *et al.* [80] developed an advanced algorithm FedProx to penalize the bias of the local model to the global model by a proximal term. The advantage is to limit the significant variance and unstable convergence of local models due to overfitting on clients with system heterogeneity. Wei *et al.* [81] took into account the privacy leakage concern of model parameters uploaded by clients in FL and proposed to improve the differential privacy by adding noise before the client sends it to the server for aggregation. Also, the game trade-off between FL convergence and privacy preservation and the optimal communication rounds were highlighted.

Although the diverse derivations that exist complement the performance of FL, there are undeniable drawbacks, such as a single point of failure (SPoF) on the server. In this section, after presenting some of the challenges and limitations of the server, we show some variants of the solution and SOTA technologies.

A. Challenges in Centralized Federated Learning

For CFL, the server takes on many responsibilities and challenges, with large service providers, such as large organizations and research institutions, playing the role of server. While these large providers have unparalleled resources compared to small workshops, there are some concerns here as the number of clients grows endlessly [82].

1) ***Client Heterogeneity*** predominantly stems from three latent factors: individual, group, and systemic heterogeneity [80], [83]–[85]. Individual heterogeneity arises from differences inherent to each sample or individual, such as variances among patients in a hospital setting. Group heterogeneity is rooted in shared characteristics among subsets of samples, such as patients of different age groups, regions, or medical backgrounds. Systemic heterogeneity originates from variations introduced during data collection by the system, which can include discrepancies from different equipment, clinical practices, or data collection personnel.

2) ***Communication Resource*** is limited on both the server and client sides [86], [87]. Although FL has dramatically reduced the consumption of communication resources by sharing only model parameters instead of user raw data, communication resources are a serious problem considering the large number of parallel clients (up to one billion). In particular, when delays in communication cause the server to wait for clients with communication problems, it can also cause the whole FL framework to become highly inefficient. Some FL communication proposals have been proposed to improve communication efficiency [88]–[91].

3) ***Computational and Storage Resource*** on the server side are also challenged [92]–[94]. The server needs to store and aggregate the models of these billions of clients. Even though lightweight models are emerging recently [95], [96], the need to compute and store model data can easily reach petabytes in size [97]. Besides the current version of the massive local model, subconditionals and versioned storage of the global model may also be required. Additionally, as clients demand real-time processing of a large volume of inference tasks, this places high demands on the computational resources required during inference [98].

4) ***Fairness, Security, and Trustworthy*** have always been crucial concerns in CFL, with these factors significantly impacting the system's overall reliability, user confidence, and data integrity. A series of questions related to security and trust form the chain of suspicion: whether the server aggregation model is reasonable, whether the global model will have high performance across all clients, whether the global model is validated, how to use the global model securely, and whether the server is secure from attacks [99]. For security issues, there are different directions of research, including malicious attacks [100], data poisoning [101], anomaly detection [102], and privacy protection [81]. For trustworthy [103], fairness [104], incentive [105], and interpretability [106] in FL are also worthy research directions.

5) ***Unreliable Connection*** in FL can stem from factors such as unreliable communication conditions, malicious attacks, or server malfunctions, leading to delays, packet loss, or noise in model transmission [107]–[109]. As all clients typically communicate with a central server, an SPoF can halt the entire system's update process. Although employing multiple edge servers can distribute the risk of SPoF, it may still cause the system segments connected to an affected edge server to become unresponsive. [110], [111] have explored using blockchain technology to mitigate SPoF by replacing the central server role, but this approach diverges from the conventional centralized FL model.

B. Variants of Centralized Federated Learning

The network variants and extensions of CFL are designed to address the above challenges and adapt to different real-world application scenarios.

1) ***Hierarchical FL*** features a classic and popular client-edge-cloud architecture, where model parameters are infrequently transmitted between the edge and the cloud, effectively

reducing communication overhead [112]–[114]. It usually perform additional aggregations by setting up additional edge servers [115], which aim to spread the communication [116] and computing pressure and reduce the impact of SPoF. These additional edge servers are geographically closer to clients, resulting in less communication resource consumption and lower latency [117]–[119]. After one or more edge server aggregations, the edge servers then upload the edge global model to the cloud for aggregation into a global model. In addition to communication optimizations, the geographic proximity of edge servers to clients may also lead to better adaptation of edge servers to the connected clients. Edge servers and connected clients can be considered geographically personalized clusters. For example, by assigning edge servers to states in the United States, the state edge servers can be more personalized to the state's user scenarios and user habits, such as weather, number of users, time zone, ethnicity, age distribution, etc.

2) ***Personalized FL*** can be classified into two categories, i.e., global model personalization and personalized model architecture [120]–[122]. Global model personalization usually starts with a global model, and then the client personalizes this global model to fit the local user. Personalization is the behavior of the client independent of the server, such as federated transfer learning to transfer global model knowledge locally [123], [124]. The personalized model architecture changes the traditional FL architecture to develop a personalized model with user knowledge, which is the behavior of the server. A famous architecture is clustered FL that has been of interest to researchers [83], [125]. The client model in the personalized FL framework is closer to the user, so it is known for its high accuracy and confidence. In particular, it is a highly effective solution for non-independent and identically distributed (non-IID) data. When the aggregated global model deviates from the user, personalization can transfer the model and adapt it to different heterogeneities.

3) ***Split FL*** splits the model for learning, where the server is responsible for some model layers [126]–[128]. The only data sent by the client to the server are the hidden representations and/or gradients in the cut layer of the model. The client not only shifts part of the learning task to the server but also does not share the user data. Compared to traditional FL, the split FL framework has similar accuracy and communication efficiency with a lower learning burden on the client side and more robust privacy protection. However, split FL is still in its early stages and has significant limitations, such as the need to consume more communication resources. Especially the presence of SPoF on the server can have even greater consequences.

4) ***Graph FL*** is particularly effective for applications involving graph-structured data, such as social networks, transportation networks, molecular structures, etc [129], [130]. This effectiveness stems from the fact that clients possess local or independent graph data, which includes rich relational information about nodes and edges. More specifically, Graph FL encompasses three levels, depending on the level of detail clients have about the graph data. These levels include multiple graphs, different parts of multiple graphs, and parts of a single graph [131]. In addition to employing graph neural networks for graph data in FL, some researchers have also considered topological graphs between clients [132]. These topological graphs can be established based on various factors, exploring network connectivity conditions between clients and the server, data and model availability of clients, as well as similarities and data generality among clients [133]–[135].

5) ***Asynchronous FL*** is designed to overcome the limitations of FL frameworks that require clients to synchronize model updates. This approach aligns well with the real-world scenarios in which the client's data updates and model training vary significantly. In these systems, heterogeneous clients have different amounts of data and computational resources and can train and update their local models at any time without the server having to wait for all clients to synchronize [136], [137]. Asynchronous FL is particularly suitable for environments with dynamic changes, a large number of clients, and a wide distribution, such as mobile devices, which may conduct model training during idle times rather than during routine user interactions [138], [139].

Variants of CFL currently exist with exotic frameworks that may include single, multiple, sub, and master servers to optimize and target different problems. For example, Zhang *et al.* [140] proposed the $(\text{Com})^2$Net, a large-scale distributed computing framework capable of spanning space-air-ground, end-edge-cloud, and multi-data center environments, facilitating ubiquitous connectivity and collaborative computing in FL. In addition to various variants, a popular approach is to assemble various variants of the FL framework to target multiple issues [141].

C. Definition of Client and Server

This paper proposes a new taxonomy that focuses on the roles played by communication endpoints in FL. We argue that the roles of edge devices, compute clusters, institutions, and organizations are relative rather than absolute. For example, a university may act as a central server when managing edge devices within its campus, but it should be considered as a client when communicating with other universities. The classification of a role as a client can be determined by whether it generates raw data and stores the data locally. For instance, a healthcare institution both generates clinical data for patients and retains it in its own database without sharing it with other institutions. Additionally, clients and servers are not mutually exclusive roles. A healthcare institution, for example, can act as both a client and a server [31]. It generates raw patient data and performs local training while also serving as a server by receiving model parameters from other healthcare institutions, aggregating them, and sharing the updated model. In practice, it can be considered as a star topology network in DFL.

The emphasis on a star topology network instead of CFL is driven by the more pressing issues of fairness and trust in FL systems when an institution simultaneously possesses raw data for local training and assumes the role of a server for aggregation. Ensuring that this institution does not favor its local data during aggregation poses an open question. As the saying goes, One cannot be a judge, jury, and executioner.

TABLE II
DEFINITIONS AND DESCRIPTIONS OF DFL TAXONOMY

Taxonomy	Category	Description
Iteration Order	Sequential	Clients are synchronized to communicate one by one in a certain order.
	Random	Clients are synchronized to communicate one by one in a random order.
	Cycle	Clients are synchronized to communicate one by one in cycle.
	Parallel	All clients communicate asynchronously.
Communication Protocol	Pointing	Clients communicate in a specific form of one-peer-to-one-peer.
	Gossip	Clients communicate in a random form of one-peer-to-one-peer, which may be determined by the neighborhood principle, client model version, complete randomness, etc.
	Broadcast	Clients communicate in a form of one-peer-to-all-peers.
	Broadcast-gossip	Clients communicate in a form of one-peer-to-multiple-peers, which is also a combination form of Gossip and Broadcast.
Network Topology	Line	Clients communicate in a sequential pointing form.
	Bus	Clients send the model to all clients behind them in order.
	Ring	Clients communicate in a cycle pointing form.
	Mesh	Clients communicate with all other clients.
	Star	Clients communicate only with the central client.
	Tree	Clients communicate only with their central clients.
	Hybrid	Combination of multiple communication topologies.
Paradigm Proposal	Continual	Client learns directly from the model of the previous client.
	Aggregate	Client first aggregates the models of past clients and then learns on the aggregated models.
Temporal Variability	Static	Communication architecture will not change.
	Dynamic	Communication architecture may change with external factors, resource-saving purpose, fairness purpose, concept drift, etc.

Similarly, an institution should not act as both a client and a server. However, this situation is more common in the real world. On one hand, institutions with more raw data have greater influence and often initiate tasks. On the other hand, institutions with more data also tend to have more abundant computing resources, making them better suited for the server role. In today's world, where large institutions possess more resources and hold greater influence, addressing the interests, privacy, and fairness of non-representative clients is a challenge.

III. TAXONOMY OF DECENTRALIZED FEDERATED LEARNING

In this section, we begin by analyzing and comparing DFL and other related designations. Subsequently, we provide a well-organized, clear, and precise description of the various iterations, protocols, network topologies, paradigms, and variations in DFL, as presented in Table II. It is worth noting that the table comprises five distinct taxonomies, which may exhibit overlapping meanings as well as conflicting aspects, and can also be applied in a complementary manner. These taxonomies, representing the viewpoints of the authors, include summarizations of existing literature, extensions of understanding, and even inferences regarding potential definitions. This comprehensive approach aims to strengthen the comprehension and categorization of concepts in the field of DFL.

A. Iteration Order

In general, FL requires multiple iterations to converge, and iteration order represents the order of each client in each iteration or the way client queues are formed in DFL. In CFL, clients iterate in a parallel manner, and the order in which the server receives the client models does not affect the convergence of the system. However, in DFL, the iteration order of clients will significantly affect the performance of client models, and we continue to discuss this issue in depth in Section III-D. Depending on the specific usage scenario and task requirements, the client iteration order in DFL can be determined to be sequential, cyclic, random, parallel, dynamic, or other strategies. The choice of iteration order can impact the convergence and performance of the system, and it is important to consider the specific characteristics and constraints of the application when determining the appropriate order.

B. Communication Protocol

DFL is a network framework for sharing model weights based on the pointing, gossip, or broadcast protocol, with the goal of obtaining optimal models across all clients. Pointing is one of the simplest and most straightforward forms of establishing a communication relationship between two peers in a unidirectional, one-to-one, and specified form. The algorithms of gossip and broadcast have been well established for use in networks [142]. Gossip protocol is essentially a random one-peer-to-one-peer way for clients to share, disseminate, and learn knowledge in a stochastic communication method [143], [144]. It is a standard communication protocol in DFL and is already in its infancy [145], [146]. The broadcast protocol is a one-peer-to-all-peers approach that allows the client to broadcast its model to all clients [147].

Hybrid protocols are now more popular, with different gossip, broadcast, and their combined communication structures designed for different scenarios and constraints. Aysal *et al.* [148] proposed a method that combines gossip and broadcast protocols and can be considered as a one-peer-to-neighbor-peers approach, where the client first broadcasts to its neighbors before gossiping. Bellet *et al.* [149] introduced an

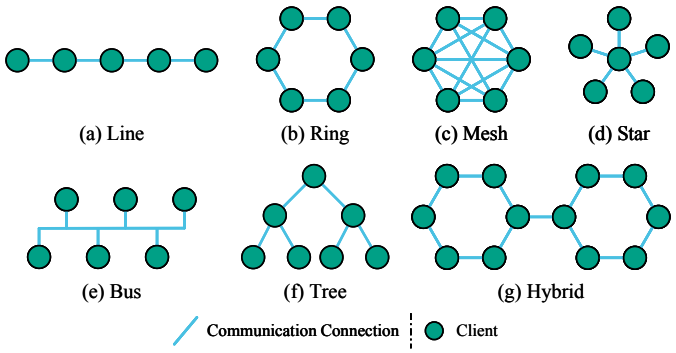


Fig. 4. Illustration of communication network topology.

algorithm that operates the agent asynchronously and performs broadcast communication between similar clients with a focus on obtaining personalized local models.

C. Network Topology

DFL networks are inspired by various network topologies, as detailed in sources such as [150]–[154]. Nedić *et al.* [142] highlighted and provided convergence proofs for several network structures, including grid, star, and fully connected topologies. Due to the absence of server-based adaptation, management, and propagation constraints, DFL networks exhibit a diverse range of configurations, as illustrated in Fig. 4. Drawing from Graph FL concepts, DFL networks can form a graph to represent structured relationships, with clients as nodes and their connections as edges. This graph-based approach in DFL enables the quantification of dependencies between clients based on characteristics such as heterogeneity and communication patterns, thus enhancing learning efficiency and model robustness. Note that the depicted line segment indicates a connection between clients, which can be either unidirectional or bidirectional. The data transmitted between clients do not only comprise their models, but may also include models from previous interactions. In addition, the computational scope of each client can encompass both local learning and aggregation.

For CFL, since all client models are hosted on the server, aggregation (i.e., averaging of all model parameters) is the mainstream and popular method for integrating knowledge from all clients. However, for DFL, the situation is much more complex. First, the network topology is diverse. There are diverse network topologies in DFL. At each communication round, clients may obey different protocols to transmit models to one or more other clients. Second, there are different versions of the model. Except for the synchronous DFL, there must be different versions of the model for other DFLs. The subsequent clients in the learning process will have models that incorporate more knowledge compared to the previous clients. Third, acquiring all client knowledge becomes more challenging. Without a centralized server for collaborative management, future clients face difficulties in accessing the knowledge of all previous clients, except for the immediate preceding clients, such as Fig. 4(a). Therefore, there is an urgent need for an alternative paradigm to complement and expand the FL landscape that is not well-compatible with aggregation.

D. Paradigm Proposal

We introduce an innovative taxonomy of DFL into two paradigms: `Continual` and `Aggregate`. The main differences between these two paradigms lie in the number of model updates exchanged between clients and whether aggregation takes place. The distinction between the paradigms also entails variations in other settings, such as learning rates. The `Aggregate` paradigm represents the archetypal FL algorithm, where each client receives the model from other clients, aggregates these models, and subsequently conducts local learning. Conversely, within the `Continual` paradigm, each client receives the model from merely one peer client and proceeds to learn directly based upon this particular model. Continual learning [155], [156], or be called incremental learning, provides a solid and grounded theory for `Continual`. A number of concepts and algorithms for federated continual learning are mentioned in the recent literature [157]–[160], which consider the process of dynamic data collection in the real world while addressing the issues of non-IID data, concept drift, and catastrophic forgetting. In DFL, the role of continual learning is more extensive:

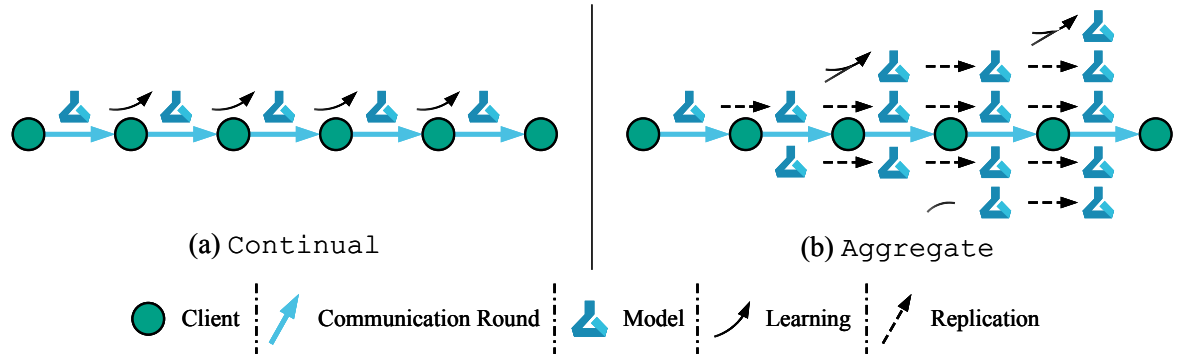
- The subsequent client will directly learn on the model of the previous client. Compared to local learning and `Aggregate`, the client is able to obtain a more personalized model while saving communication, computational, and storage resources.
- In storage-constrained frameworks, clients do not need to retain any additional model parameter data.
- In computation-constrained frameworks, clients also do not need to consume additional resources for aggregation calculations.
- The continuous generation of new data by clients is accommodated, and they do not need to wait for all data to be collected before starting the local learning process.
- For tasks that may have concept drift, clients are always provided with the latest version of the model.

Table III analyzes and summarizes the intrinsic, algorithm, advantage, challenge, and network topology of these two paradigms. The difference between these two paradigms is illustrated by the example of sequential pointing line DFL. In the `Continual` paradigm, the only content delivered to the subsequent client is the trained model. The subsequent client just continue learning on this model, as shown in Table III(a). In the `Aggregate` paradigm, the previous client transmits not only the trained local model but also all the previous models. The learning process performed by the subsequent client is divided into two parts, first aggregation and then learning, as shown in Table III(b).

In order to compare and illustrate the difference between the `Continual` and `Aggregate` paradigms, pointing, gossip, and broadcast, and different network topologies. Algorithm 1 shows two paradigms in the sequential pointing line DFL topology and Algorithm 2 shows pointing ring `Continual`

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3407584

TABLE III
PARADIGMS OF DFL

Paradigm	**`Continual`**	**`Aggregate`**
Intrinsic	Client receives a single model per iteration. Learning is performed on the received model without aggregation.	Client receives multiple models per iteration. Learning is performed locally after aggregating the previous models.
Algorithm	**For** each client **until convergence do:** 1) Receive the model from the previous client. 2) Perform local learning on the model. 3) Transmit the trained model to the next client.	**For** each client **until convergence do:** 1) Receive all other client models from the previous client (pointing and gossip) or other clients (broadcast and broadcast-gossip). 2) Aggregate all received models and perform local learning on the aggregated model. 3) Transmit the trained model and other client models to the next client (pointing and gossip) or transmit the trained model to all clients (broadcast and broadcast-gossip).
Advantage	• Each client involved in learning has a highly accurate, personalized, high-confidence local model. • Compared to CFL, they do not have the same set of issues on the server side, such as aggregation fairness.	
	• Fewer communication, computation, and storage resources are required. • More simple and straightforward, suitable for all scenarios.	• More powerful generalization ability on the obtained model. • Stronger ability to update new knowledge generated by the client.
Challenge	• Model performance strongly depends on the client iteration order. • Appropriate loss function, learning rate, and training epoch, which allows the model to learn the current client's knowledge while ensuring that the previous knowledge is not forgotten.	
	• Catastrophic forgetting of past client knowledge.	• Repetition and overemphasis on learning from past clients.
Network Topology (Take sequential pointing line DFL architecture as an example)		

and broadcast mesh `Aggregate` DFL. The difference between the `Continual` and `Aggregate` paradigms can be clearly seen in the pre-processing of the client before learning and the sharing of the model after learning. The additional requirements of the `Aggregate` paradigm for communication, computation, and storage have been highlighted. Under the `Aggregate` paradigm, the pointing and gossip protocols require the client to send more model data at once, while the broadcast and broadcast-gossip protocols require the client to send at a higher frequency. The ring topology can be seen as a cyclic variant of the line topology, and both network topologies are widely used by researchers due to their simple and straightforward structure. The line topology is a sufficient knowledge learning system for systems that do not generate new knowledge. However, in a system that is constantly generating new knowledge, the ring topology may be a more reasonable topology. It is not only able to re-update the knowledge in the system but also a feasible solution to catastrophic forgetting.

To further illustrate the learning and communication process among clients in these two paradigms, Fig. 5 demonstrates the learning process from the first client to the final client in the parameter space. It is important to note that we actually have several assumptions here. Firstly, the optimal solutions of the local models of all clients follow a multivariate normal distribution in the parameter space. Secondly, considering the systemic and statistical heterogeneity, some clients exhibit significant biases. Thirdly, although reasonable loss functions and learning rates are chosen, the models are not always trained to achieve optimal solutions. The communication and learning processes of the two paradigms, `Continual` and `Aggregate`, are as follows.

Step 1) Both paradigms initiate learning with the same initial model parameters and obtain the same Model 1 in Client 1.

Step 2) Both paradigms learn from Model 1 and reach the same Model 2 in Client 2. It's worth noting that the `Aggregate` paradigm is meaningful when there are two or more aggregated models available.

Step 3) In the `Continual` paradigm, Client 3 learns directly from Model 2 to obtain Model 3, while in the `Aggregate` paradigm, Model 1 and Model 2

Algorithm 1 Sequential pointing line `Continual` and pointing line `Aggregate` decentralized federated learning.

Input: Client set (C), training epoch (E), initial model (ω_0), loss function ($\mathcal{L}$), learning rate (η)
Output: Local models ($\{\omega_c | c \in C\}$)
for $c \in C$ **in sequence do**
  Copy the model from previous client $\omega_c \leftarrow \omega_{c-1}$
  Aggregate received models $\omega_c \leftarrow \text{Aggregation}\{\omega_1, \omega_2, ..., \omega_{c-1}\}$
  for $e = 1$ **to** $E - 1$ **do**
    Backpropagate and update the local model $\omega_c^{e+1} \leftarrow \omega_c^e - \eta \nabla \mathcal{L}$.
  end for
  Update the local model $\omega_c \leftarrow \omega_c^E$.
  Client c sends $\{\omega_1, \omega_2, ..., \omega_{c-1}\}$ and ω_c to the next client.
end for

Algorithm 2 Cycle pointing ring `Continual` and broadcast mesh `Aggregate` Decentralized Federated Learning.

Input: Client set (C), training epoch (E), initial model (ω_0), loss function ($\mathcal{L}$), learning rate (η)
Output: Local models ($\{\omega_c | c \in C\}$)
while $c \in C$ **in cyclic do** ▷ line vs. ring
  Copy the model from previous client $\omega_c \leftarrow \omega_{c-1}$
  Aggregate received models $\omega_c \leftarrow \text{Aggregation}\{\omega_1, \omega_2, ..., \omega_{c-1}\}$
  for $e = 1$ **to** $E - 1$ **do**
    Backpropagate and update the local model $\omega_c^{e+1} \leftarrow \omega_c^e - \eta \nabla \mathcal{L}$.
  end for
  Update the local model $\omega_c \leftarrow \omega_c^E$.
  Client c sends ω_c to the next client and all other clients. ▷ pointing vs. broadcast
end while

are first aggregated, and then Client 3 learns from the aggregated model to obtain Model 3. Note that the `Continual` paradigm is less complex than the `Aggregate` paradigm, as indicated by the length of the black arrow.

Step 4) In the `Aggregate` paradigm, the model aggregated by Client 4 is closer to the center of the Normal distribution than Client 3, so it is expected that the learning process for subsequent clients will be easier.

Step n) When the client is positioned towards the end of the queue, the learning difficulty in the `Continual` paradigm becomes random, depending on the deviation between the previous client and the current client. However, in the `Aggregate` paradigm, the learning difficulty is only influenced by the current client since the aggregated model is expected to be extremely close to the center of the normal distribution.

Based on the aforementioned assumptions and iterative process, we can make certain expectations regarding the accuracy, loss, convergence, and communication complexity of the clients in both paradigms during training. We come up with the following speculations:

1) The learning loss will exhibit periodic oscillations across client iterations and eventually converge in both paradigms.
2) The convergence of learning loss in the `Aggregate` paradigm is expected to be more stable. In the `Continual` paradigm, the learning difficulty depends on the discrepancy between the previous client and the current client's local data, in other words, it depends on the iteration order of clients. Under the assumption of a normal distribution, the learning difficulty in the `Aggregate` paradigm is determined by the heterogeneity of the current client's data, and most clients may have similar data distributions.
3) The convergence of learning loss in the `Aggregate` paradigm is also expected to be faster due to the decreasing learning difficulty as the client iterations progress. However, this acceleration in convergence is accompanied by an increase in communication overhead.
4) The `Continual` paradigm requires more communication rounds to achieve convergence, whereas the `Aggregate` paradigm incurs greater communication overhead per round.
5) The `Continual` paradigm exhibits stronger personalization, while the `Aggregate` paradigm demonstrates greater generalization. Depending on scenario requirements, both paradigms can achieve similar performance after convergence by adjusting weights.

E. Temporal Variability

The network topology of DFLs has recently undergone a shift from static to dynamic trends, adapting to the time-varying external environment [161]. The inspiration for the separation and clustering of network topologies comes from group behaviors observed in nature, such as fish schools and bee swarms [162]. When a school of fish encounters a predator, the entire school separates to avoid it. Similarly, in a bee swarm, a small number of scouts can lead the entire swarm, demonstrating the herd effect. Interestingly, migratory birds form V-shaped formations during long-distance flights to conserve energy, and the birds at the front rotate over time to distribute flight fatigue evenly. In the context of DFL networks, dynamic topologies may exhibit more robust, fair, and efficient performance compared to static topologies. The determination of dynamic topologies in DFL networks can be influenced by various factors, including:

- **External Interference.** Strong and unbreakable communication barriers, SPoF, malicious attacks, and other external factors can lead to changes in the network topology. In order to avoid the failure of the entire network, topology adjustments and discards are made.
- **Communication Resource Saving.** Clients have the ability to dynamically select their neighbors for each communication. By selectively choosing nearby clients, communication resources can be optimized and saved. Additionally, clients can dynamically elect the most central client as a leader during each communication, enhancing the efficiency and effectiveness of communication within the network.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3407584

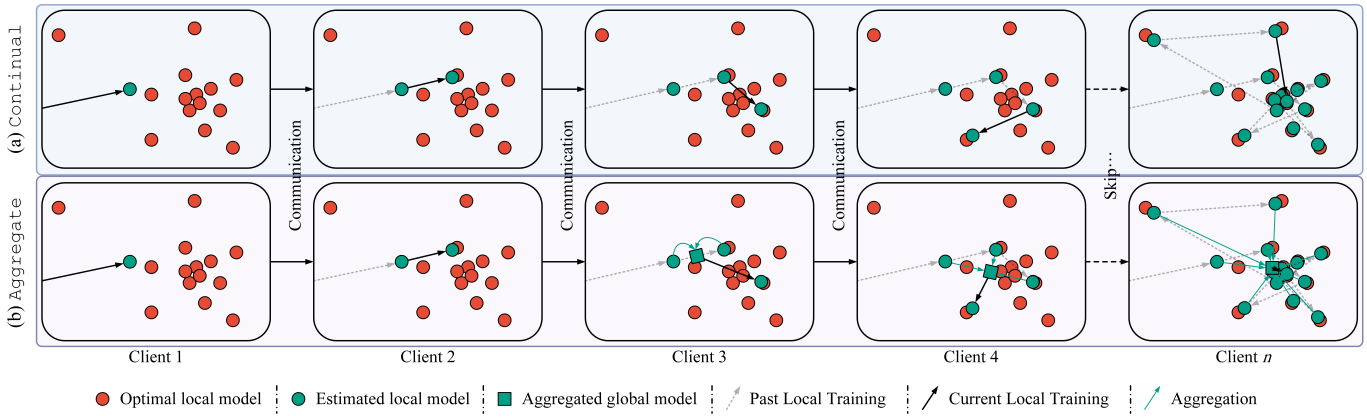


Fig. 5. Illustration of the two paradigms, `Continual` and `Aggregate`, for sequential pointing line DFL in the parameter space, showcasing their respective learning and communication processes. The length of the arrow represents both the learning difficulty and the magnitude of the model parameters that undergo changes during learning, which can be measured using the ℓ_2 norm. Shorter arrows are desired as they indicate more accessible, stable, and accurate model learning and convergence. Excessively long arrows suggest that the given loss function and learning rate may not produce the desired model outcome.

- **Fairness.** In order to ensure fairness among clients, a random selection process is employed for determining the communication target. This helps to prevent any bias or preference towards specific clients, ensuring equal opportunities for all participants.

The development of dynamic topological structures based on these factors shapes DFL networks to facilitate robust, efficient, and fair communication among clients.

IV. Variants of Decentralized Federated Learning

In Section III, we provide a comprehensive definition, introduction, and propose two paradigm for DFLs. In this section, we review the real-world applications in DFL, with a specific focus on its diverse applications across various domains and its real-world deployment. Taking inspiration from the CFL variants and considering the underlying network topologies depicted in Fig. 4, we propose several viable topology variants for DFL. These topology variants serve as alternative options for researchers to consider when deploying DFL. We discuss the advantages and limitations associated with each variant, enabling researchers to make informed decisions regarding the most suitable topology for specific usage scenarios.

A. Real-World Applications

The development of a DFL framework relies on several key factors, such as relevant application scenarios, sources of information acquisition, information processing units, and perceptual prediction modules, among others. With the establishment of the theoretical framework for networks, DFL has been adopted in various application domains, including vehicles, healthcare, industrial IoT, social networks, etc.

1) ***Connected and Automated Vehicles (CAVs)*** serve as a robust hardware infrastructure for DFL, leveraging onboard batteries, diverse sensors, computing units, storage devices, and more. Existing vehicle networking frameworks, such as vehicle-to-vehicle (V2V), have also laid the foundation for communication and networking experiences in DFL for CAV [72], [174]. Referred to as V2V FL, this approach enables the exchange and sharing of up-to-date knowledge among vehicles and has been explored in recent studies [175]–[182]. Lu *et al.* [183] proposed a vehicular DFL approach with a focus on privacy protection and mitigating data leakage risks in vehicular cyber-physical systems (VCPS). In their framework, roadside units (RSUs) are responsible for forwarding vehicle identities, vehicle data retrieval information, data profiles, data sharing requests, and related tasks. Once the V2V connection is established through the RSU intermediary, the model data is directly transmitted to the requesting vehicle.

2) ***Healthcare Institutions*** are inclined towards DFL frameworks over CFL due to their abundant patient privacy data, computational resources, and storage capabilities. As key stakeholders in healthcare institutions, clinicians play a crucial role in data collection, model training, data analysis, characterization, and providing experimental results and solutions. Unlike traditional server-centric approaches, clinicians have the flexibility to observe, analyze, fine-tune, and match models manually, offering more control and adaptability. Healthcare institutions widely employ DFL frameworks in various studies [61], [163]–[166], [184]–[186]. Warnat-Herresthal *et al.* [53] introduced a DFL framework called Swarm Learning, which addresses four use cases of heterogeneous diseases, including COVID-19, tuberculosis, leukemia, and lung pathologies. This framework incorporates a blockchain smart contract for enhanced security and dynamically selects a leader for aggregating model parameters in each iteration.

3) ***Industrial IoT (IIoT)***, as a cornerstone of Industry 4.0, greatly benefits from DFL, which offers robust and scalable solutions tailored to the complexities of modern manufacturing environments [187]–[189]. IIoT requires enhanced robustness, and DFL improves system resistance to SPoF, which is particularly crucial for devices on production lines [190]. The high autonomy of DFL makes it well suited for industrial settings with diverse physical conditions and operational environments. Moreover, IIoT can rely on the scalability of DFL to easily adapt to geographically dispersed production sites and dynamically accommodate new industrial devices into the

TABLE IV
SOME INSPIRING DFLS WITH DIFFERENT PROTOCOLS, TOPOLOGIES, PARADIGMS, AND VARIANTS.

Literature	Year	Paradigm	Type	Highlight
Chang *et al.* [163]	2018	`Continual`	• Sequential pointing line • Cycle pointing ring	• Introduced system heterogeneity artificially.
Sheller *et al.* [164]	2019	`Continual`	• Sequential pointing line • Cycle pointing ring	• Obtained a conclusion that catastrophic forgetting worsens as the number of clients increases.
Sheller *et al.* [165]	2020	`Continual`	• Sequential pointing line • Cycle pointing ring	• Considered the DFL framework to output a final model approach.
Huang *et al.* [166]	2022	`Continual`	• Sequential pointing line • Cycle pointing ring	• Introduced synaptic intelligence in `Continual` DFL to effectively improve model stability, especially for sequential pointing line topology.
Yuan *et al.* [167]	2023	`Continual`	• Random gossip ring	• Considered the highly dynamic and random nature of vehicle connectivity in vehicular networks and employed gossip-based communication to simulate this characteristic when deploying `Continual` DFL. • Provided a comprehensive comparison between CFL and DFL, such as knowledge dissemination mechanism, communication complexity, generalizability, compatibility, overhead, hidden concerns, etc.
Assran *et al.* [168]	2019	`Aggregate`	• Cycle broadcast-gossip mesh • Parallel broadcast mesh	• Performed a comparison of broadcast-gossip and broadcast protocols.
Roy *et al.* [169]	2019	`Aggregate`	• Random broadcast-gossip mesh	• Pre-requested model versions from other clients. • Considered the scenario where the DFL framework outputs a model to a new client.
Pappas *et al.* [170]	2021	`Aggregate`	• Parallel broadcast star	• Proposed a framework that combines star DFL with split learning.
Warnat *et al.* [53]	2021	`Aggregate`	• Dynamic pointing star	• Elected a leader dynamically via a blockchain smart contract that is used to aggregate model parameters.
Shi *et al.* [171]	2021	`Aggregate`	• Cycle broadcast-gossip hybrid	• Analyzed the convergence of broadcast-gossip in a hybrid network.
Chen *et al.* [172]	2022	`Aggregate`	• Cycle broadcast mesh	• Introduced the superposition property of the analog scheme to improve the parallelism of communication, which enables a significant reduction communication rounds.
Wang *et al.* [173]	2022	`Aggregate`	• Dynamic parallel broadcast-gossip hybrid	• Promoted more frequent communication in the central client to achieve fast convergence. • Promoted less frequent communication in the other clients to achieve low communication latency. • The proposed algorithm is applicable and generalized to all hybrid networks.

DFL system. Du *et al.* [191] designed a DFL framework for IIoT, where each client exchanges model parameters only with neighbors using a broadcast-gossip communication protocol to achieve model consensus. To enhance the efficiency of the gossip protocol and reduce communication overhead, they also consider the topology of the entire client network to facilitate asynchronous model exchanges between clients.

4) ***Mobile Services*** based on IoT devices provide a significant application scenario for DFL, leveraging the capabilities of smartphones, laptops, tablets, etc. These mobile IoT devices are equipped with various sensors, such as global positioning system (GPS), inertial measurement unit (IMU), cameras, sound sensors, and magnetic sensors, enabling them to acquire diverse sources of information. Unlike the relatively fixed connectivity of CAVs, mobile IoT devices offer more flexible systems and platforms to support a wide range of applications. In recent studies, DFL frameworks have been developed specifically for mobile IoT devices, aiming to leverage their computational power and sensor capabilities [192], [193]. While the traditional example of CFL, such as Google mobile keyboard prediction, is well-known [3], the transfer of such applications to DFLs is of great interest. For instance, building DFLs among individuals with similar professions, such as doctors, lawyers, or engineers, can enable personalized word recommendations tailored to their specific needs. Belal *et al.* [194] developed a smartphone-based DFL personalized recommendation system for New York City attractions and movies. By sharing model parameters with neighbors who have similar interests, the system achieves higher hit rates and faster convergence, enhancing the recommendation accuracy and user experience.

5) ***UAVs and Satellites***, operating in dynamic computing environments, possess vast amounts of sensitive data for remote sensing, target recognition, and military-related tasks, under the constraints of limited resources, making them well suited for the advantages of DFL [195]–[197]. For mobile

UAVs and satellites, bandwidth is a valuable resource. One potential solution is that using a DFL framework based on broadcast gossip tailored to dynamic geographic locations can significantly reduce bandwidth requirements and the consumption of communication resources [64]. Moreover, due to their dynamic nature and highly variable data, DFL can also enhance real-time responsiveness, adaptability, and efficiency of task execution. Han *et al.* [198] proposed a DFL framework that orchestrates satellite constellations, aggregating models within a satellite cluster and relaying models to other satellites via inter-satellite links, particularly considering the dynamic scenarios of low earth orbit satellites with varying orbits.

6) ***Social Networks***, as large-scale knowledge graphs, contain various users as nodes, content, and connections, making DFL highly effective for handling a widely dispersed and personalized user base, where each user is connected only to their neighbors [199]. Use cases in social networks include sentiment analysis, recommendation systems, publication systems, and influence analysis, among others [130], [200], [201]. Users in social networks come from diverse backgrounds, such as teachers and doctors, but may share common interests in topics like comics, leading to frequent interactions. The connections within and between user groups vary in proximity, thus emphasizing the need for a personalized and scalable DFL approach. Chen *et al.* [202] developed a DFL framework for social networks that establishes a user data structure with affine distributions. This structure helps capture the heterogeneity among users and reduces the loss associated with independently and identically distributed data.

7) ***Artificial General Intelligence (AGI)***, as one of the popular research areas today, represented by large language models (LLMs) like ChatGPT, has the potential to fundamentally change our lives [203]. The training of these models relies on vast amounts of computational resources and data, which can benefit from the collaborative learning paradigm of DFL to facilitate cooperation among different data centers [204], [205]. Unlike other application areas discussed earlier, such as healthcare, AGI models feature billions or even tens of billions of model parameters and are intended for general use cases, not just confined to a specific application. This poses unique challenges for the application of DFL in AGI. Qin *et al.* [206] proposed a method using FL for full-parameter fine-tuning of billion-scale LLMs, employing zeroth-order optimization with a random seed subset, reducing communication requirements to just a few random seeds and scalar gradients, totaling only a few thousand bytes. Meanwhile, Gao *et al.* [207] only proposed a purely theoretical design for decentralized LLMs. Therefore, the use of DFL for AGI remains a nascent field requiring further research [208].

B. Variant: Line

The base variant of DFL can be considered as a sequential pointing line, depicted in Fig. 4(a) and Fig. 6(a). This topology serves as the simplest and most straightforward illustration and comparison in this paper, as demonstrated in Table III(a), (b), and Algorithm 1. The line variant is frequently used as a baseline for comparison due to its ease of implementation, intuitiveness, and efficiency [163]–[166]. However, it has notable limitations, such as the inability to accommodate continuous learning of new knowledge within the system, the risk of catastrophic forgetting in the `Continual` paradigm, or redundant and excessive learning in the `Aggregate` paradigm, as well as limited generalization ability for starting clients and the vulnerability to a SPoF. Furthermore, the line variant lacks cyclic connections, limiting each client to a single iteration and preventing the system from fully converging. In particular, in the line variant, the clients at the front of the queue will have worse model performance. Given its prominent disadvantages and advantages, it can serve as a baseline or initial implementation for further research and development.

C. Variant: Ring

The ring variant corresponds to the cycle pointing line DFL, as depicted in Fig. 4(b) and Fig. 6(a). The cyclic form is commonly used in DFL as the model needs to be trained between clients to acquire new knowledge collected from other clients, thereby enhancing generalization. Based on the framework, not all past models need to be transferred for aggregation in each communication since many of them may already be outdated. The ring variant not only inherits the simplicity of the line variant but also becomes a popular approach in various research papers due to its ability to iterate indefinitely until convergence.

The ring topology is already considered mature in decentralized learning [145], [209] and is beginning to gain traction in DFL [210]. For example, Chang *et al.* [163] proposed two heuristics for DFL, including sequential pointing communication on each client for one iteration and multiple iterations to obtain the final model. Similarly, Sheller *et al.* [165] also considered sequential pointing or cycle continual learning in the client to generate the final model. Nguyen *et al.* [181] applied cycle pointing DFL to autonomous driving applications. Yuan *et al.* [167] proposed a random ring topology DFL framework, named FedPC, based on the gossip communication protocol for naturalistic driving action recognition. FedPC emphasizes the highly dynamic, random, and data-heterogeneous nature of vehicle connections in this context.

D. Variant: Mesh

A multidirectional ring, also known as a fully connected topology, or be called mesh, is a variant of the ring basic variant, depicted in Fig. 4(c) and Fig. 6(b). In the ring variant, each client needs to transmit multiple model parameters in each communication round, which can pose a burden on the network bandwidth. In contrast, the mesh variant requires each client to transmit its local model parameters to all other clients in each communication round. This approach entails higher communication frequency for the clients while also reducing the size of model packets transmitted per communication. The higher communication frequency and larger per-communication data packet overhead have their respective advantages and disadvantages, which can be traded off depending on the specific application context. However, when compared

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3407584

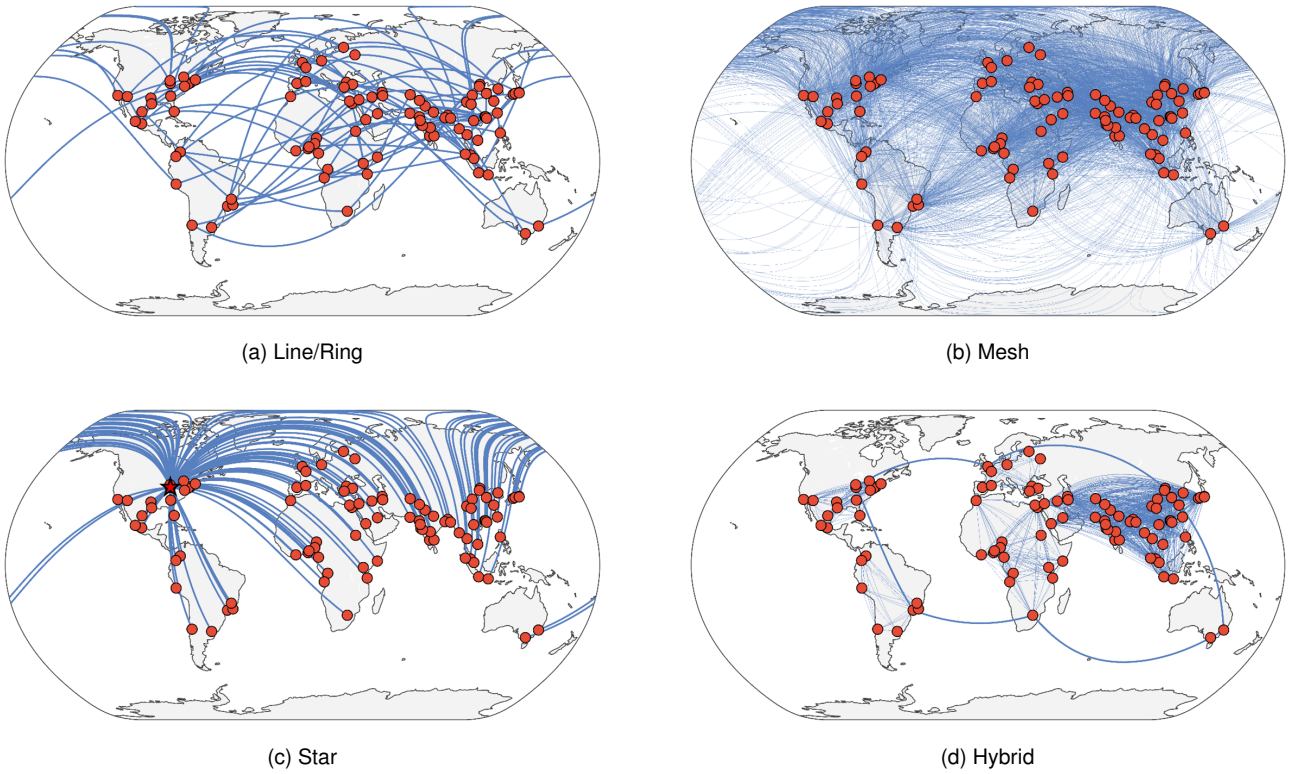


Fig. 6. Illustrations of imagined DFL network topologies in the real world: (a) line/ring, (b) mesh, (c) star, and (d) hybrid. The red dots represent clients, which can be universities, institutions, or organizations in some of the major cities in the world (determined by population). The blue lines depict the communication network among these clients. Depending on the chosen topology, the communication networks exhibit different communication distances, number of communication links, complexity, and other characteristics.

to the ring variant, the mesh variant significantly mitigates the impact of SPoF, which is a notable advantage of this variant.

Recent research has witnessed the emergence of mesh-based DFL approaches [61], [172], [180], [211]. Assran *et al.* [168] proposed Stochastic Gradient Push (SGP), a parallel broadcast-gossip mesh DFL approach. In the broadcast-gossip iteration, clients in SGP send their trained local models to a sparse selection of other clients in a parallel manner, and they also receive models from other selected clients. Each client then performs a weighted aggregation of its local model with the received models. Roy *et al.* [169] introduced the BrainTorrent framework, in which a requesting client communicates with all clients to obtain information about available model versions, and clients with new versions send their models to the requesting client for aggregation.

E. Variant: Star

The star variant resembles the CFL model, where one client assumes the role of the server to coordinate and interact with other clients, as depicted in Fig. 4(d) and Fig. 6(c). The star variant operates in two different modes. In the first mode, similar to CFL, the central client is responsible for receiving, aggregating, and distributing the local models. However, unlike CFL, the central client also generates original data and utilizes the models for perception and decision-making. This mode emphasizes a family-like relationship, where one member has more computational and communication power to assist the other clients. In the second mode of operation, the focus is on geographic interoperability among clients. As some clients in the community are geographically dispersed, there is a client that serves as the geographical center for these clients. To conserve communication resources, the surrounding clients transmit their models to the central client, which then forwards the models to the other clients.

Pappas *et al.* [170] introduced a split learning framework within a star DFL architecture, where clients train different layers of a model and update the model parameters with the central client. This approach allows for distributed model training and collaboration among clients. Another example of a star variant is the Swarm Learning framework proposed by Warnat-Herresthal *et al.* [53], which involves the dynamic election of a leader to aggregate model parameters. In Swarm Learning, the leader plays a central role in coordinating the aggregation process and facilitating collaboration among the clients.

F. Variant: Hybrid

The Hybrid variant of DFL encompasses a wide range of configurations, combining elements from various other variants. It is considered the most promising option for practical applications due to its adaptability to different scenarios [212]–[214]. However, the complexity of configuring a hybrid variant can pose challenges. One example of a hybrid variant, as depicted in Fig. 4(g), involves connecting two ring variants

through a central client. In this configuration, the hybrid variant provides global connectivity, allowing for the sharing of client models and knowledge within the framework. The two ring variants can also be treated as a single entity, with only one communication channel connected to the two central clients. Another illustration of a hybrid variant, shown in Fig. 6(d), involves dividing clients into organizations based on geographical locations (e.g., continents). Within each organization, a mesh topology network is established, and a leader is elected. These leaders then form a ring topology network among themselves. The hybrid variants do not have a fixed structure and can be customized to meet the specific requirements of real-world scenarios. The hybrid variant offers several advantages.

Firstly, the hybrid variant helps in saving communication resources. This is achieved through the knowledge dissemination between the leaders of two organizations, where only the aggregated global model is shared. By transmitting only the essential information, the hybrid variant reduces the communication overhead. Additionally, organized knowledge dissemination further enhances resource efficiency by minimizing the sharing of irrelevant or invalid information. This approach is particularly advantageous when establishing communication between two geographically distant organizations, as the single-line connection reduces the resource requirements for long-distance communication. Considering that the communication between organizations represents the dissemination of knowledge across states, countries, and continents [215], the clients representing the institutions establish a stable and well-structured communication connection to facilitate the exchange of knowledge between their respective organizations.

Secondly, the hybrid variant offers enhanced security. With two central clients in control, they have the ability to unilaterally disconnect the communication between organizations, ensuring the protection of their respective knowledge from potential leaks or unauthorized access. This adds an extra layer of security to the DFL system.

Thirdly, the hybrid variant provides a more personalized approach. Each organization's aggregated model is organization-specific, tailored to the unique characteristics of its local data. This personalized model may offer better applicability to the specific needs and requirements of the organization. While model knowledge is shared between the two organizations, the decision of whether to utilize the other organization's model is subject to further investigation and discussion. By thoroughly assessing the performance of the other organization's model, clients can ensure that their own model remains uncontaminated and unaffected by potentially inferior or incompatible models.

Xing *et al.* [216] proposed a hybrid DFL network that establishes connections only with neighboring clients, and model parameters are broadcast-gossiped only among these neighboring clients. Their approach takes into account various factors such as link blockages, channel fading, and mutual interference, to ensure efficient and reliable communication. Building upon this work, Shi *et al.* [171] further improved the convergence performance by incorporating coding strategies, gradient tracking, and variance reduction algorithms. In a similar vein, Wang *et al.* [173] developed a dynamic hybrid DFL framework called Matcha. Matcha introduces the concept of creating different network topologies at each iteration to enhance convergence speed. The algorithm consists of two main parts. Firstly, an initial network topology pre-processing step where Matcha performs matching decomposition on a base communication topology to obtain disjoint sub-graphs, including sub-graphs with only two-peer connections. Next, matching activation probabilities are computed to maximize the connectivity of the graph, and a new random topology graph is generated for each iteration. The key idea behind Matcha is to achieve faster convergence by enabling more frequent communication on connectivity-critical links (e.g., central clients) and reducing communication latency by decreasing the frequency of communication on other connections. Matcha is particularly advantageous for hybrid networks with unknown or dynamic central clients. However, it may not exhibit the same advantages in scenarios involving research institutions where central clients are known and pre-determined.

V. Challenge and Potential Solutions in DFL

Based on the current SOTA technology, this section aims to discuss and analyze potential challenges and future research directions for DFL. Additionally, the variants mentioned in Section IV can be regarded as potential solutions to address these challenges.

A. High Communication Overhead

DFL is widely recognized as an extremely communication resource-efficient approach compared to CFL. However, researchers are still striving for further savings in communication resources and reduced communication complexity [217]. We discuss the existing CFL frameworks in Section II-B, and we consider introducing viable methods to achieve efficient communication in DFL. Wang *et al.* [218] introduced a method called optimization of topology construction and model compression (CoCo) that aims to improve communication efficiency and convergence speed in DFL. CoCo achieves this by employing adaptive techniques for constructing the DFL network topology and assigning an appropriate model compression ratio to each participating client. It achieves this by adaptively constructing the DFL network topology and assigning an appropriate model compression ratio to each client.

In addition to model compression, it is also important to investigate how to leverage efficient communication lines and reduce the overall communication length. Variants such as star and hybrid variants, which select geocentric clients and resource-rich clients as leaders, have been proven to be effective solutions in this regard. Some researchers have also focused on addressing the bandwidth differences among different communication lines [219], [220]. It is worth noting that the dynamic hybrid variant proposed by Wang *et al.* [173] emphasizes the importance of communication efficiency and suggests frequent communication with key clients to achieve faster convergence. A considerable body of research emphasizes the importance of efficient communication in DFL and

proposes various strategies and methods to reduce complexity and optimize communication resources. Further exploration in this direction is expected to facilitate the potential deployment of DFL frameworks in real-world applications.

B. Computational and Storage Burden

Compared to the CFL and `Continual` paradigm, the `Aggregate` paradigm imposes significantly higher demands on client-side computational and storage resources. As there is no dedicated server in the `Aggregate` paradigm, clients are responsible for storing previous model parameters and performing aggregation computations alongside local model training. Consequently, the computational and storage burdens pose challenges for client hardware.

One potential solution is to adopt the transfer learning concept and fix the weights of the lower layers in all models. In this approach, the lower layers serve as feature extractors for a specific task and are expected to be similar across models, while the higher-level representations remain task-specific. By fixing these parameters, there is no need for gradient descent, aggregation computations, or communication-related to these layers. Moreover, this approach reduces storage requirements, thereby substantially mitigating the resource consumption of the client. Currently, with the widespread availability of high-performance GPU computing resources, the challenges related to computational and storage burdens are gradually diminishing. This is especially true in DFL scenarios where institutions and organizations serve as clients. However, in contexts such as mobile services dominated by smartphones and on-board units in vehicular edge devices, there is still value in researching ways to reduce computational complexity and optimize storage efficiency.

C. Vulnerability in Cybersecurity

Network security has always been a major challenge in FL, and this challenge is particularly prominent in DFL [221]–[224]. In the traditional CFL setting, clients communicate with a central server, typically operated by a research institution or a large commercial organization. While there is still potential for attacks and data poisoning between clients and the server, communication is generally more regulated and protected compared to DFL. In DFL, the knowledge exchange occurs directly among users within a local area network, with free and unrestricted sharing agreements, which poses an increased risk of privacy exposure. Malicious attacks from clients, poisoned data, free-riding attacks, and other malicious behaviors are all possible in this decentralized setting [225], [226].

Kuo *et al.* [227] proposed the integration of blockchain into a decentralized learning framework to enhance privacy protection, which can also be applied to DFL. Chen *et al.* [228] integrated a differential privacy mechanism based on blockchain technology. Bellet *et al.* [149] introduced an asynchronous and differential privacy algorithm in DFL to safeguard user privacy. He *et al.* [50] addressed trust issues between clients by employing an online push-sum algorithm to actively push local models to trusted clients. Shayan *et al.* [229] proposed the Biscotti DFL system, which incorporates multiple privacy and security protection techniques, including the Multi-Krum defense to prevent poisoning attacks, differential privacy noise to protect privacy, and secure aggregation. The future research direction in cybersecurity will involve the roles of attackers and defenders, focusing on developing targeted attack and defense mechanisms for different DFL variants.

D. Lack of Incentive Mechanism

In the absence of server management, the issue of fairness in aggregation has been effectively addressed in DFL. However, the lack of incentives and mutual distrust among clients can significantly impact their willingness to contribute knowledge. A key issue is the lack of incentives, which may lead to free-riding attacks where clients choose to benefit from the models without contributing their own knowledge [68], [230].

In the context of DFL, the feasibility of incentive mechanisms based on game theory, such as Stackelberg games [231], raises questions due to the requirements on game leaders, participants, and rewards. One potential solution could be the integration of reputation-based incentive mechanisms using blockchain and smart contracts. Kang *et al.* [105] proposed assigning reputation scores to clients to represent and quantify their reliability. Clients with higher contributions and reputations can receive greater rewards. However, designing effective and practical incentive mechanisms for DFL remains an open problem.

In cases where task providers do not exist or there are no explicit rewards, punitive incentives may also be a potential solution. Clients who fail to contribute or engage in malicious behavior could face penalties or reduced access to the benefits of the DFL framework. Further research is needed to explore and develop robust incentive mechanisms tailored specifically for DFL systems. Designing effective incentive mechanisms to encourage active participation, foster trust, and stimulate enthusiastic knowledge sharing will greatly facilitate the dissemination of knowledge in DFL.

E. Lack of Management

In DFL, the absence of a central server for managing all clients poses a significant challenge in receiving and sharing knowledge in an organized manner. The lack of central management can lead to confusion, particularly among clients with varying sample sizes, computational resources, and communication capabilities. In the ring variant, a client only needs to wait for the model parameters from the previous client, while in the mesh variant, a client needs to wait for model parameters from all other clients. Such dependencies on other clients for model transmission can result in deadlocks, causing the entire system to halt. Moreover, the communication among clients may not be robust, considering the possibility of SPoF. The absence of management is particularly problematic in the hybrid variant depicted in Fig. 6(d), where clients communicate globally. The lack of management can lead to reduced operational efficiency, confusion regarding model versions, and performance degradation.

To address the challenge of lack of management in DFL, researchers have proposed several approaches. One approach

is to pre-request the status of other clients, such as their model versions, before initiating knowledge transfer [169]. By obtaining accurate information, a client can then request the transfer of the entire model data. Some star variants enforce the knowledge dissemination flow among clients by designating a leader [178]. This leader is responsible for regulating knowledge dissemination among the remaining clients. Additionally, Chen *et al.* [179] introduced the BDFL framework, a mesh DFL framework specifically designed for autonomous vehicles. In this framework, a leader is randomly selected in each communication round, offering advantages such as increased privacy and security protection against Byzantine faults, as well as enhanced management through the leader's command issuance. In real-world scenarios, clients may face challenges where they lack knowledge about each other's statuses, leading to issues such as model version discrepancies and even system paralysis, such as in the case of an SPoF. Therefore, future research directions aim to ensure the smooth operation of the system by incorporating additional information or establishing contingency plans. These measures can help mitigate the impact of uncertainty and improve the reliability and robustness of the DFL framework.

VI. Conclusion

In this paper, we provided an extensive exploration of the DFL framework, covering communication protocols, network topologies, paradigm proposals, extension variants, challenges, and potential solutions. Our aim is to offer a comprehensive, well-defined, and systematic perspective that organizes and synthesizes the existing literature and definitions, thereby facilitating a comprehensive introduction to DFL for new researchers. Given that DFL is a rapidly evolving area, we established a solid theoretical foundation by defining and discussing five variants in this paper. This not only provides researchers with a comprehensive understanding of the field but also fosters the generation of new ideas and collaborations among peers.

It is important to note that our approach differs from traditional surveys, as we presented our own insights and innovative thinking on DFL. Moreover, this paper uncovers a considerable number of previously unexplored types within the DFL framework. For example, no existing studies have demonstrated the integration of the `Continual` paradigm with mesh network topology. Researchers might consider asynchronous DFL, where clients acquire data and learn at different times, thus benefiting from the dual advantages offered by the `Continual` paradigm and the mesh network topology, such as reduced communication overhead, personalization, and more. By considering diverse usage scenarios, we aim to stimulate and extend the research interest of other DFL practitioners, enabling them to adapt the framework to their specific needs.

References

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021.

[2] 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*. PMLR, 2017, pp. 1273–1282.

[3] 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," *arXiv preprint arXiv:1811.03604*, 2018.

[4] D. Ramage and S. Mazzocchi, "Federated analytics: Collaborative data science without data collection," *Google Research*, 2020.

[5] D. Wang, S. Shi, Y. Zhu, and Z. Han, "Federated analytics: Opportunities and challenges," *IEEE Network*, vol. 36, no. 1, pp. 151–158, 2021.

[6] A. R. Elkordy, Y. H. Ezzeldin, S. Han, S. Sharma, C. He, S. Mehrotra, S. Avestimehr *et al.*, "Federated analytics: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 12, no. 1, 2023.

[7] D. Chen, D. Wang, Y. Zhu, and Z. Han, "Digital twin for federated analytics using a bayesian approach," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 301–16 312, 2021.

[8] S. R. Pandey, M. N. Nguyen, T. N. Dang, N. H. Tran, K. Thar, Z. Han, and C. S. Hong, "Edge-assisted democratized learning toward federated analytics," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 572–588, 2021.

[9] Z. Wang, Y. Zhu, D. Wang, and Z. Han, "FedFPM: A unified federated analytics framework for collaborative frequent pattern mining," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 61–70.

[10] D. Froelicher, J. R. Troncoso-Pastoriza, J. L. Raisaro, M. A. Cuendet, J. S. Sousa, H. Cho, B. Berger, J. Fellay, and J.-P. Hubaux, "Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption," *Nature communications*, vol. 12, no. 1, p. 5910, 2021.

[11] Z. Wang, R. Gupta, K. Han, H. Wang, A. Ganlath, N. Ammar, and P. Tiwari, "Mobility digital twin: Concept, architecture, case study, and future challenges," *IEEE Internet Things J.*, Mar. 2022.

[12] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[13] Y. Li, B. Dang, W. Li, and Y. Zhang, "Glh-water: A large-scale dataset for global surface water detection in large-size very-high-resolution satellite imagery," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 20, 2024, pp. 22 213–22 221.

[14] Y. Li, J. Luo, Y. Zhang, Y. Tan, J.-G. Yu, and S. Bai, "Learning to holistically detect bridges from large-size vhr remote sensing imagery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[15] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nat. Mach. Intell.*, vol. 2, no. 6, pp. 305–311, Jun. 2020.

[16] A. Durrant, M. Markovic, D. Matthews, D. May, J. Enright, and G. Leontidis, "The role of cross-silo federated learning in facilitating data sharing in the agri-food sector," *Comput. Electron. Agric.*, vol. 193, p. 106648, Feb. 2022.

[17] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[18] L. Yuan, D.-J. Han, V. P. Chellapandi, S. H. Żak, and C. G. Brinton, "FedMFS: Federated multimodal fusion learning with selective modality communication," *arXiv preprint arXiv:2310.07048*, 2023.

[19] D. Shi, L. Li, R. Chen, P. Prakash, M. Pan, and Y. Fang, "Toward energy-efficient federated learning over 5g+ mobile devices," *IEEE Wireless Communications*, vol. 29, no. 5, pp. 44–51, 2022.

[20] B. Pfitzner, N. Steckhan, and B. Arnrich, "Federated learning in a medical context: a systematic literature review," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–31, 2021.

[21] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Comput. Surv. (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[22] J. Li, Y. Meng, L. Ma, S. Du, H. Zhu, Q. Pei, and X. Shen, "A federated learning based privacy-preserving smart healthcare system," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, 2021.

[23] J. C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing: Challenges and opportunities," *Sensors*, vol. 20, no. 21, p. 6230, 2020.

[24] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

[25] V. P. Chellapandi, Y. Nagaraj, J. Supplee, S. Hernandez-Gonzalez, H. Borhan, and S. H. Żak, "Predictive control of diesel oxidation catalysts with federated learning in connected vehicles," in *2024 Forum for Innovative Sustainable Transportation Systems (FISTS)*. IEEE, 2024, pp. 1–6.

[26] L. Sun and J. Wu, "A scalable and transferable federated learning system for classifying healthcare sensor data," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 866–877, 2022.

[27] Y.-W. Chu, D.-J. Han, and C. G. Brinton, "Only send what you need: Learning to communicate efficiently in federated multilingual machine translation," *arXiv preprint arXiv:2401.07456*, 2024.

[28] L. Yuan, D.-J. Han, S. Wang, D. Upadhyay, and C. G. Brinton, "Communication-efficient multimodal federated learning: Joint modality and client selection," *arXiv preprint arXiv:2401.16685*, 2024.

[29] J. Tan, Y. Li, S. A. Bartalev, B. Dang, W. Chen, Y. Zhang, and L. Yuan, "Bridging data islands: Geographic heterogeneity-aware federated learning for collaborative remote sensing semantic segmentation," *arXiv preprint arXiv:2404.09292*, 2024.

[30] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nat. Med.*, vol. 27, no. 10, pp. 1735–1743, Oct. 2021.

[31] S. Pati, U. Baid, B. Edwards, M. Sheller, S.-H. Wang, G. A. Reina, P. Foley, A. Gruzdev, D. Karkada, C. Davatzikos *et al.*, "Federated learning enables big data for rare cancer boundary detection," *Nat. Commun.*, vol. 13, no. 1, p. 7346, Dec. 2022.

[32] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. (TIST)*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[33] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.

[34] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, p. e2024789118, 2021.

[35] Z. Yang, Y. Shi, Y. Zhou, Z. Wang, and K. Yang, "Trustworthy federated learning via blockchain," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 92–109, 2022.

[36] M. Zhang, E. Wei, and R. Berry, "Faithful edge federated learning: Scalability and privacy," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3790–3804, 2021.

[37] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

[38] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[39] F. Lai, Y. Dai, S. Singapuram, J. Liu, X. Zhu, H. Madhyastha, and M. Chowdhury, "Fedscale: Benchmarking model and system performance of federated learning at scale," in *International Conference on Machine Learning*. PMLR, 2022, pp. 11 814–11 827.

[40] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, 2022.

[41] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[42] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

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

[44] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[45] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, D. Sánchez, A. Flanagan, and K. E. Tan, "Achieving security and privacy in federated learning systems: Survey, research challenges and future directions," *Engineering Applications of Artificial Intelligence*, vol. 106, p. 104468, 2021.

[46] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A sustainable incentive scheme for federated learning," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 58–69, 2020.

[47] S. Wan, J. Lu, P. Fan, Y. Shao, C. Peng, and K. B. Letaief, "Convergence analysis and system design for federated learning over wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3622–3639, 2021.

[48] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, "Robust federated learning in a heterogeneous environment," *arXiv preprint arXiv:1906.06629*, 2019.

[49] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[50] C. He, C. Tan, H. Tang, S. Qiu, and J. Liu, "Central server free federated learning over single-sided trust social networks," *arXiv preprint arXiv:1910.04956*, 2019.

[51] C. He, E. Ceyani, K. Balasubramanian, M. Annavaram, and S. Avestimehr, "Spreadgnn: Serverless multi-task federated learning for graph neural networks," *arXiv preprint arXiv:2106.02743*, 2021.

[52] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3723–3741, 2021.

[53] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, Jun. 2021.

[54] A. Lalitha, S. Shekhar, T. Javidi, and F. Koushanfar, "Fully decentralized federated learning," in *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

[55] Q. Li, Z. Wen, and B. He, "Practical federated gradient boosting decision trees," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 4642–4649.

[56] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[57] D. Monschein, J. A. P. Pérez, T. Piotrowski, Z. Nochta, O. P. Waldhorst, and C. Zirpins, "Towards a peer-to-peer federated machine learning environment for continuous authentication," in *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2021, pp. 1–6.

[58] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, and L. Barbieri, "Opportunities of federated learning in connected, cooperative, and automated industrial systems," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 16–21, Mar. 2021.

[59] Y. Shi, L. Shen, K. Wei, Y. Sun, B. Yuan, X. Wang, and D. Tao, "Improving the model consistency of decentralized federated learning," *arXiv preprint arXiv:2302.04083*, Feb. 2023.

[60] V. P. Chellapandi, A. Upadhyay, A. Hashemi, and S. H. Żak, "On the convergence of decentralized federated learning under imperfect information sharing," *IEEE Control Systems Letters*, 2023.

[61] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthc. Inform. Res.*, vol. 5, no. 1, pp. 1–19, Mar. 2021.

[62] 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," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[63] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[64] Y. Qu, H. Dai, Y. Zhuang, J. Chen, C. Dong, F. Wu, and S. Guo, "Decentralized federated learning for uav networks: Architecture, challenges, and opportunities," *IEEE Network*, vol. 35, no. 6, pp. 156–162, 2021.

[65] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[66] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–35, 2022.

[67] R. Gupta and T. Alam, "Survey on federated-learning approaches in distributed environment," *Wireless personal communications*, vol. 125, no. 2, pp. 1631–1652, 2022.

[68] L. Witt, M. Heyer, K. Toyoda, W. Samek, and D. Li, "Decentral and incentivized federated learning frameworks: A systematic literature review," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3642–3663, 2022.

[69] H. Gao, M. T. Thai, and J. Wu, "When decentralized optimization meets federated learning," *IEEE network*, 2023.

[70] E. Gabrielli, G. Pica, and G. Tolomei, "A survey on decentralized federated learning," *arXiv preprint arXiv:2308.04604*, 2023.

[71] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," *ACM Computing Surveys*, 2023.

[72] V. P. Chellapandi, L. Yuan, C. G. Brinton, S. H. Żak, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 119 – 137, November 2023.

[73] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[74] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, 2024.

[75] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–7, Sep. 2020.

[76] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[77] V. P. Chellapandi, L. Yuan, S. H. Żak, and Z. Wang, "A survey of federated learning for connected and automated vehicles," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 2485–2492.

[78] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[79] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[80] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[81] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, Apr. 2020.

[82] T. R. Gadekallu, Q.-V. Pham, T. Huynh-The, S. Bhattacharya, P. K. R. Maddikunta, and M. Liyanage, "Federated learning for big data: A survey on opportunities, applications, and future directions," *arXiv preprint arXiv:2110.04160*, 2021.

[83] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 586–19 597, 2020.

[84] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: clients clustering for better personalization," *World Wide Web*, vol. 26, no. 1, pp. 481–500, 2023.

[85] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 1–44, 2023.

[86] W. Fang, Y. Jiang, Y. Shi, Y. Zhou, W. Chen, and K. B. Letaief, "Over-the-air computation via reconfigurable intelligent surface," *IEEE transactions on communications*, vol. 69, no. 12, pp. 8612–8626, 2021.

[87] W. Fang, Z. Yu, Y. Jiang, Y. Shi, C. N. Jones, and Y. Zhou, "Communication-efficient stochastic zeroth-order optimization for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5058–5073, 2022.

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

[89] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Nov. 2019.

[90] G. Lan, X.-Y. Liu, Y. Zhang, and X. Wang, "Communication-efficient federated learning for resource-constrained edge devices," *IEEE Transactions on Machine Learning in Communications and Networking*, 2023.

[91] G. Lan, H. Wang, J. Anderson, C. Brinton, and V. Aggarwal, "Improved communication efficiency in federated natural policy gradient via admm-based gradient updates," *arXiv preprint arXiv:2310.19807*, 2023.

[92] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[93] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE journal on selected areas in communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[94] A. Imteaj, K. Mamun Ahmed, U. Thakker, S. Wang, J. Li, and M. H. Amini, "Federated learning for resource-constrained IoT devices: Panoramas and state of the art," *Federated and Transfer Learning*, pp. 7–27, 2022.

[95] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[96] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[97] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[98] W. Zhang, D. Yang, H. Peng, W. Wu, W. Quan, H. Zhang, and X. Shen, "Deep reinforcement learning based resource management for DNN inference in industrial IoT," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7605–7618, 2021.

[99] C. Ma, J. Li, M. Ding, H. H. Yang, F. Shu, T. Q. Quek, and H. V. Poor, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Mar. 2020.

[100] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[101] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *European Symposium on Research in Computer Security*. Springer, 2020, pp. 480–501.

[102] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021.

[103] Y. Zhang, D. Zeng, J. Luo, Z. Xu, and I. King, "A survey of trustworthy federated learning with perspectives on security, robustness, and privacy," *arXiv preprint arXiv:2302.10637*, 2023.

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

[105] 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. 10 700–10 714, Sep. 2019.

[106] D. Roschewitz, M.-A. Hartley, L. Corinzia, and M. Jaggi, "IFedAvg: Interpretable data-interoperability for federated learning," *arXiv preprint arXiv:2107.06580*, 2021.

[107] M. Salehi and E. Hossain, "Federated learning in unreliable and resource-constrained cellular wireless networks," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5136–5151, 2021.

[108] S. Ganguli, Z. Zhou, C. G. Brinton, and D. I. Inouye, "Fault-tolerant vertical federated learning on dynamic networks," *arXiv preprint arXiv:2312.16638*, 2023.

[109] S. Ganguli, A. Amalanshu, A. Ranjan, and D. I. Inouye, "Internet learning: Preliminary steps towards highly fault-tolerant learning on device networks," in *ICML Workshop on Localized Learning (LLW)*, 2023.

[110] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2019.

[111] Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5171–5183, Mar. 2020.

[112] W. Zhang, D. Yang, W. Wu, H. Peng, H. Zhang, and X. S. Shen, "Spectrum and computing resource management for federated learning in distributed industrial IoT," in *2021 IEEE International Conference*

on Communications Workshops (ICC Workshops). IEEE, 2021, pp. 1–6.

[113] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, 2021.

[114] D. Yang, W. Zhang, Q. Ye, C. Zhang, N. Zhang, C. Huang, H. Zhang, and X. Shen, "DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks," *IEEE Transactions on Mobile Computing*, 2023.

[115] 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. Surv. Tutor.*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.

[116] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "Edgefed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209 191–209 198, Nov. 2020.

[117] S. Wang, S. Hosseinalipour, V. Aggarwal, C. G. Brinton, D. J. Love, W. Su, and M. Chiang, "Towards cooperative federated learning over heterogeneous edge/fog networks," *IEEE Communications Magazine*, 2023.

[118] W. Fang, D.-J. Han, and C. G. Brinton, "Submodel partitioning in hierarchical federated learning: Algorithm design and convergence analysis," *arXiv preprint arXiv:2310.17890*, 2023.

[119] E. Chen, S. Wang, and C. G. Brinton, "Taming subnet-drift in D2D-enabled fog learning: A hierarchical gradient tracking approach," *arXiv preprint arXiv:2312.04728*, 2023.

[120] Y.-W. Chu, S. Hosseinalipour, E. Tenorio, L. Cruz, K. Douglas, A. Lan, and C. Brinton, "Mitigating biases in student performance prediction via attention-based personalized federated learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3033–3042.

[121] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, Mar. 2022.

[122] Y.-W. Chu, S. Hosseinalipour, E. Tenorio, L. Cruz, K. Douglas, A. Lan, and C. Brinton, "Multi-layer personalized federated learning for mitigating biases in student predictive analytics," *arXiv preprint arXiv:2212.02985*, 2022.

[123] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Apr. 2020.

[124] L. Yuan, L. Su, and Z. Wang, "Federated transfer-ordered-personalized learning for driver monitoring application," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18 292 – 18 301, May 2023.

[125] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2020.

[126] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8485–8493.

[127] D.-J. Han, D.-Y. Kim, M. Choi, C. G. Brinton, and J. Moon, "Splitgp: Achieving both generalization and personalization in federated learning," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[128] D.-J. Han, D.-Y. Kim, M. Choi, D. Nickel, J. Moon, M. Chiang, and C. G. Brinton, "Federated split learning with joint personalization-generalization for inference-stage optimization in wireless edge networks," *IEEE Transactions on Mobile Computing*, 2023.

[129] L. Qu, N. Tang, R. Zheng, Q. V. H. Nguyen, Z. Huang, Y. Shi, and H. Yin, "Semi-decentralized federated ego graph learning for recommendation," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 339–348.

[130] C. He, K. Balasubramanian, E. Ceyani, C. Yang, H. Xie, L. Sun, L. He, L. Yang, P. S. Yu, Y. Rong *et al.*, "FedGraphNN: A federated learning system and benchmark for graph neural networks," *arXiv preprint arXiv:2104.07145*, 2021.

[131] J. Baek, W. Jeong, J. Jin, J. Yoon, and S. J. Hwang, "Personalized subgraph federated learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 1396–1415.

[132] R. Liu, P. Xing, Z. Deng, A. Li, C. Guan, and H. Yu, "Federated graph neural networks: Overview, techniques, and challenges," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[133] S. Wagle, S. Hosseinalipour, N. Khosravan, M. Chiang, and C. G. Brinton, "Embedding alignment for unsupervised federated learning via smart data exchange," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 492–497.

[134] S. Wagle, A. B. Das, D. J. Love, and C. G. Brinton, "A reinforcement learning-based approach to graph discovery in D2D-enabled federated learning," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 225–230.

[135] S. Wagle, S. Hosseinalipour, N. Khosravan, and C. G. Brinton, "Unsupervised federated optimization at the edge: D2D-enabled learning without labels," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[136] G. Lan, D.-J. Han, A. Hashemi, V. Aggarwal, and C. G. Brinton, "Asynchronous federated reinforcement learning with policy gradient updates: Algorithm design and convergence analysis," *arXiv preprint arXiv:2404.08003*, 2024.

[137] Z.-L. Chang, S. Hosseinalipour, M. Chiang, and C. G. Brinton, "Asynchronous multi-model dynamic federated learning over wireless networks: Theory, modeling, and optimization," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[138] J. Bian and J. Xu, "Accelerating asynchronous federated learning convergence via opportunistic mobile relaying," *IEEE Transactions on Vehicular Technology*, 2024.

[139] X. Zhou, W. Liang, A. Kawai, K. Fueda, J. She, I. Kevin, and K. Wang, "Adaptive segmentation enhanced asynchronous federated learning for sustainable intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[140] W. Zhang, D. Yang, C. Zhang, Q. Ye, H. Zhang, and X. Shen, "$(\text{Com})^2$Net: A novel communication and computation integrated network architecture," *IEEE Network*, 2024.

[141] J. Chen, K. Li, and S. Y. Philip, "Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain," *IEEE Trans. Intell. Transp. Syst.*, Aug. 2021.

[142] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, Apr. 2018.

[143] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[144] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 2003, pp. 482–491.

[145] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3478–3487.

[146] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," *arXiv preprint arXiv:1908.07782*, 2019.

[147] A. Nedic, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Automat. Contr.*, vol. 56, no. 6, pp. 1337–1351, Sep. 2010.

[148] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, Feb. 2009.

[149] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Personalized and private peer-to-peer machine learning," in *International conference on artificial intelligence and statistics*. PMLR, 2018, pp. 473–481.

[150] S. Wang, D. Li, J. Geng, Y. Gu, and Y. Cheng, "Impact of network topology on the performance of DML: Theoretical analysis and practical factors," in *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2019, pp. 1729–1737.

[151] G. Neglia, G. Calbi, D. Towsley, and G. Vardoyan, "The role of network topology for distributed machine learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2350–2358.

[152] O. Marfoq, C. Xu, G. Neglia, and R. Vidal, "Throughput-optimal topology design for cross-silo federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 478–19 487, 2020.

[153] F. Malandrino and C. F. Chiasserini, "Federated learning at the network edge: When not all nodes are created equal," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 68–73, 2021.

[154] V. P. Chellapandi, A. Upadhyay, A. Hashemi, and S. H. Żak, "Decentralized federated learning: Model update tracking under imperfect information sharing," *arXiv preprint arXiv:2403.13247*, 2024.

[155] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Inf. Fusion*, vol. 58, pp. 52–68, Jun. 2020.

[156] 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.*, Feb. 2021.

[157] M. F. Criado, F. E. Casado, R. Iglesias, C. V. Regueiro, and S. Barro, "Non-IID data and continual learning processes in federated learning: A long road ahead," *Inf. Fusion*, vol. 88, pp. 263–280, Dec. 2022.

[158] A. Usmanova, F. Portet, P. Lalanda, and G. Vega, "A distillation-based approach integrating continual learning and federated learning for pervasive services," *arXiv preprint arXiv:2109.04197*, 2021.

[159] T. J. Park, K. Kumatani, and D. Dimitriadis, "Tackling dynamics in federated incremental learning with variational embedding rehearsal," *arXiv preprint arXiv:2110.09695*, 2021.

[160] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 073–12 086.

[161] R. Parasnis, S. Hosseinalipour, Y.-W. Chu, C. G. Brinton, and M. Chiang, "Connectivity-aware semi-decentralized federated learning over time-varying D2D networks," *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2023.

[162] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, Apr. 2013.

[163] 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. Am. Med. Inform. Assoc.*, vol. 25, no. 8, pp. 945–954, Aug. 2018.

[164] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2019, pp. 92–104.

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

[166] Y. Huang, C. Bert, S. Fischer, M. Schmidt, A. Dörfler, A. Maier, R. Fietkau, and F. Putz, "Continual learning for peer-to-peer federated learning: A study on automated brain metastasis identification," *arXiv preprint arXiv:2204.13591*, 2022.

[167] L. Yuan, Y. Ma, L. Su, and Z. Wang, "Peer-to-peer federated continual learning for naturalistic driving action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5249–5258.

[168] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 344–353.

[169] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[170] C. Pappas, D. Chatzopoulos, S. Lalis, and M. Vavalis, "Ipls: A framework for decentralized federated learning," in *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 2021, pp. 1–6.

[171] Y. Shi, Y. Zhou, and Y. Shi, "Over-the-air decentralized federated learning," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 455–460.

[172] S. Chen, D. Yu, Y. Zou, J. Yu, and X. Cheng, "Decentralized wireless federated learning with differential privacy," *IEEE Trans. Industr. Inform.*, vol. 18, no. 9, pp. 6273–6282, Jan. 2022.

[173] J. Wang, A. K. Sahu, G. Joshi, and S. Kar, "Matcha: A matching-based link scheduling strategy to speed up distributed optimization," *IEEE Trans. Signal Process.*, vol. 70, pp. 5208–5221, Nov. 2022.

[174] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, J. Wang *et al.*, "Vehicle-to-vehicle communications: readiness of V2V technology for application." United States. National Highway Traffic Safety Administration, Tech. Rep., Aug. 2014.

[175] 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.

[176] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, May 2020.

[177] S. R. Pokhrel and J. Choi, "A decentralized federated learning approach for connected autonomous vehicles," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2020, pp. 1–6.

[178] Z. Yu, J. Hu, G. Min, H. Xu, and J. Mills, "Proactive content caching for internet-of-vehicles based on peer-to-peer federated learning," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2020, pp. 601–608.

[179] J.-H. Chen, M.-R. Chen, G.-Q. Zeng, and J.-S. Weng, "BDFL: a byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8639–8652, Aug. 2021.

[180] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6g connected vehicles," *Veh. Commun.*, vol. 33, p. 100396, Jan. 2022.

[181] A. Nguyen, T. Do, M. Tran, B. X. Nguyen, C. Duong, T. Phan, E. Tjiputra, and Q. D. Tran, "Deep federated learning for autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1824–1830.

[182] D. Su, Y. Zhou, and L. Cui, "Boost decentralized federated learning in vehicular networks by diversifying data sources," in *2022 IEEE 30th International Conference on Network Protocols (ICNP)*. IEEE, 2022, pp. 1–11.

[183] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Netw.*, vol. 34, no. 3, pp. 50–56, Jun. 2020.

[184] B. C. Tedeschini, S. Savazzi, R. Stoklasa, L. Barbieri, I. Stathopoulos, M. Nicoli, and L. Serio, "Decentralized federated learning for healthcare networks: A case study on tumor segmentation," *IEEE Access*, vol. 10, pp. 8693–8708, Jan. 2022.

[185] T. Nguyen, M. Dakka, S. Diakiw, M. VerMilyea, M. Perugini, J. Hall, and D. Perugini, "A novel decentralized federated learning approach to train on globally distributed, poor quality, and protected private medical data," *Sci. Rep.*, vol. 12, no. 1, p. 8888, May 2022.

[186] S. Baghersalimi, T. Teijeiro, A. Aminifar, and D. Atienza, "Decentralized federated learning for epileptic seizures detection in low-power wearable systems," *IEEE Transactions on Mobile Computing*, 2023.

[187] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2964–2973, 2020.

[188] T. Ranathunga, A. McGibney, S. Rea, and S. Bharti, "Blockchain-based decentralized model aggregation for cross-silo federated learning in industry 4.0," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4449–4461, 2022.

[189] O. Friha, M. A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, and K.-K. R. Choo, "2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT," *Computers & Security*, vol. 127, p. 103097, 2023.

[190] W. Qiu, W. Ai, H. Chen, Q. Feng, and G. Tang, "Decentralized federated learning for industrial IoT with deep echo state networks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5849–5857, 2022.

[191] M. Du, H. Zheng, X. Feng, Y. Chen, and T. Zhao, "Decentralized federated learning with markov chain based consensus for industrial IoT networks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 6006–6015, 2022.

[192] F. Wilhelmi, E. Guerra, and P. Dini, "On the decentralization of blockchain-enabled asynchronous federated learning," *arXiv preprint arXiv:2205.10201*, 2022.

[193] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *International Conference on Learning Representations*, 2019.

[194] Y. Belal, A. Bellet, S. B. Mokhtar, and V. Nitu, "Pepper: Empowering user-centric recommender systems over gossip learning," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 3, pp. 1–27, Sep. 2022.

[195] Y. Xiao, Y. Ye, S. Huang, L. Hao, Z. Ma, M. Xiao, S. Mumtaz, and O. A. Dobre, "Fully decentralized federated learning-based on-board mission for uav swarm system," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3296–3300, 2021.

[196] Z. Yan and D. Li, "Convergence time optimization for decentralized federated learning with leo satellites via number control," *IEEE Transactions on Vehicular Technology*, 2023.

[197] Z. Zhai, Q. Wu, S. Yu, R. Li, F. Zhang, and X. Chen, "FedLEO: An offloading-assisted decentralized federated learning framework for

low earth orbit satellite networks," *IEEE Transactions on Mobile Computing*, 2023.

[198] D.-J. Han, S. Hosseinalipour, D. J. Love, M. Chiang, and C. G. Brinton, "Cooperative federated learning over ground-to-satellite integrated networks: Joint local computation and data offloading," *IEEE Journal on Selected Areas in Communications*, 2024.

[199] C. He, E. Ceyani, K. Balasubramanian, M. Annavaram, and S. Avestimehr, "Spreadgnn: Decentralized multi-task federated learning for graph neural networks on molecular data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6865–6873.

[200] J. Han, Y. Ma, and Y. Han, "Demystifying swarm learning: A new paradigm of blockchain-based decentralized federated learning," *arXiv preprint arXiv:2201.05286*, 2022.

[201] C. Che, X. Li, C. Chen, X. He, and Z. Zheng, "A decentralized federated learning framework via committee mechanism with convergence guarantee," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4783–4800, 2022.

[202] Y. Chen, L. Liang, and W. Gao, "DFedSN: Decentralized federated learning based on heterogeneous data in social networks," *World Wide Web*, vol. 26, no. 5, pp. 2545–2568, 2023.

[203] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[204] V. Kersic and M. Turkanovic, "A review on building blocks of decentralized artificial intelligence," *arXiv preprint arXiv:2402.02885*, 2024.

[205] H. Su, C. Xiang, and B. Ramesh, "Towards confidential chatbot conversations: A decentralised federated learning framework," *The Journal of The British Blockchain Association*, 2024.

[206] Z. Qin, D. Chen, B. Qian, B. Ding, Y. Li, and S. Deng, "Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes," *arXiv preprint arXiv:2312.06353*, 2023.

[207] Y. Gao, Z. Song, and J. Yin, "Gradientcoin: A peer-to-peer decentralized large language models," *arXiv preprint arXiv:2308.10502*, 2023.

[208] C. Chen, Z. Wu, Y. Lai, W. Ou, T. Liao, and Z. Zheng, "Challenges and remedies to privacy and security in aigc: Exploring the potential of privacy computing, blockchain, and beyond," *arXiv preprint arXiv:2306.00419*, 2023.

[209] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3043–3052.

[210] Z. Wang, Y. Hu, S. Yan, Z. Wang, R. Hou, and C. Wu, "Efficient ring-topology decentralized federated learning with deep generative models for medical data in ehealthcare systems," *Electronics*, vol. 11, no. 10, p. 1548, May 2022.

[211] T. Wink and Z. Nochta, "An approach for peer-to-peer federated learning," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2021, pp. 150–157.

[212] S. Wang, Y. Ruan, Y. Tu, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 2019–2032, 2021.

[213] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[214] S. Wang, R. Morabito, S. Hosseinalipour, M. Chiang, and C. G. Brinton, "Device sampling and resource optimization for federated learning in cooperative edge networks," *arXiv preprint arXiv:2311.04350*, 2023.

[215] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks," *IEEE Trans. Commun.*, vol. 25, no. 1, pp. 48–60, Jan. 1977.

[216] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via SGD over wireless D2D networks," in *2020 IEEE 21st international workshop on signal processing advances in wireless communications (SPAWC)*. IEEE, 2020, pp. 1–5.

[217] S. Kalra, J. Wen, J. C. Cresswell, M. Volkovs, and H. Tizhoosh, "Decentralized federated learning through proxy model sharing," *Nat. Commun.*, vol. 14, no. 1, p. 2899, May. 2023.

[218] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Trans. Mob. Comput.*, May 2022.

[219] Z. Tang, S. Shi, and X. Chu, "Communication-efficient decentralized learning with sparsification and adaptive peer selection," *arXiv preprint arXiv:2002.09692*, 2020.

[220] P. Zhou, Q. Lin, D. Loghin, B. C. Ooi, Y. Wu, and H. Yu, "Communication-efficient decentralized machine learning over heterogeneous networks," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 384–395.

[221] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[222] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity: Concepts, challenges, and future directions," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3501–3509, 2021.

[223] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8229–8249, 2022.

[224] X. Zhou, W. Liang, I. Kevin, K. Wang, Z. Yan, L. T. Yang, W. Wei, J. Ma, and Q. Jin, "Decentralized p2p federated learning for privacy-preserving and resilient mobile robotic systems," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 82–89, 2023.

[225] T.-T. Kuo and A. Pham, "Detecting model misconducts in decentralized healthcare federated learning," *Int. J. Med. Inform.*, vol. 158, p. 104658, Feb. 2022.

[226] L. Wang, X. Zhao, Z. Lu, L. Wang, and S. Zhang, "Enhancing privacy preservation and trustworthiness for decentralized federated learning," *Inf. Sci.*, vol. 628, pp. 449–468, May 2023.

[227] T.-T. Kuo and L. Ohno-Machado, "Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks," *arXiv preprint arXiv:1802.01746*, 2018.

[228] X. Chen, J. Ji, C. Luo, W. Liao, and P. Li, "When machine learning meets blockchain: A decentralized, privacy-preserving and secure design," in *2018 IEEE international conference on big data (big data)*. IEEE, 2018, pp. 1178–1187.

[229] M. Shayan, C. Fung, C. J. Yoon, and I. Beschastnikh, "Biscotti: A blockchain system for private and secure federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1513–1525, Dec. 2020.

[230] L. Yuan, Z. Wang, and C. G. Brinton, "Digital ethics in federated learning," *arXiv preprint arXiv:2310.03178*, 2023.

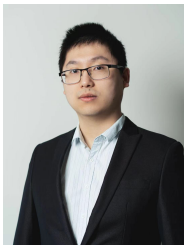
[231] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Nov. 2020.

Liangqi Yuan (S'22) received the B.E. degree from the Beijing Information Science and Technology University, Beijing, China, in 2020, and the M.Sc. degree from the Oakland University, Rochester, MI, USA, in 2022. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His research interests are in the areas of sensors, the internet of things, human–computer interaction, signal processing, and machine learning.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3407584

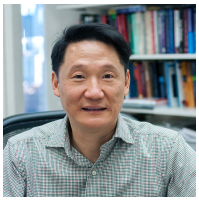
Ziran Wang (S'16-M'19) received his Ph.D. degree in Mechanical Engineering from the University of California, Riverside in 2019. He is a tenure-track assistant professor in the College of Engineering at Purdue University, where he leads the Purdue Digital Twin Lab. Prior to this, Dr. Wang was a principal researcher at Toyota Motor North America in Mountain View, California.

Dr. Wang serves as the founding chair of the IEEE Technical Committee on Internet of Things in Intelligent Transportation Systems, a member of three other IEEE technical committees, and a technical program committee member of multiple IEEE and ACM conferences. Dr. Wang is an associate of four academic journals, including IEEE Transactions on Intelligent Vehicles and IEEE Internet of Things Journal. His research achievements have been demonstrated at the Consumer Electronics Show (CES), and acknowledged by the U.S. Department of Transportation Dissertation Award, the IEEE "Shape the Future of ITS" 1st Prize Award, and five other best paper awards from IEEE and SAE. He is an author of five book chapters, 50+ refereed papers, and 50+ patent applications. His research focuses on autonomous driving, human-autonomy teaming, and digital twin.



Lichao Sun received the Ph.D. degree in computer science from the University of Illinois Chicago, Chicago, IL, USA, in 2020, under the supervision of Prof. Philip S. Yu.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA. He has published more than 45 research articles in top conferences and journals, such as CCS, USENIXSecurity, NeurIPS, KDD, ICLR, the Advancement of AI (AAAI), the International Joint Conference on AI (IJCAI), ACL, NAACL, TII, TNNLS, and TMC. His research interests include security and privacy in deep learning and data mining. He mainly focuses on artificial intelligence (AI) security and privacy, social networks, and NLP applications.



Philip S. Yu (Life Fellow, IEEE) received the Bachelor of Science degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1970, the Master of Science and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1973 and 1976, respectively, and the Master of Business Administration degree from New York University, New York, NY, USA, in 1980.

He is currently a Distinguished Professor of computer science with the University of Illinois Chicago (UIC), Chicago, IL, USA, and also holds the Wexler Chair in Information Technology. Before joining UIC, he was with IBM, USA, where he was the Manager of the Software Tools and Techniques Department, Watson Research Center. He has published more than 1200 papers in refereed journals and conferences. He holds or has applied for more than 300 U.S. patents. His research interest is on big data, including data mining, data stream, database, and privacy.

Dr. Yu is a fellow of the ACM. He was a recipient of the ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions to mining, fusion, and anonymization of big data, the IEEE Computer Society's 2013 Technical Achievement Award for pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining, and anonymization of big data, and the Research Contributions Award from IEEE International Conference on Data Mining (ICDM) in 2003 for his pioneering contributions to the field of data mining. He received the ICDM 2013 10-Year Highest-Impact Paper Award and the EDBT Test of Time Award in 2014. He was the Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data from 2011 to 2017 and IEEE Transactions on Knowledge and Data Engineering from 2001 to 2004.

Christopher G. Brinton (S'08, M'16, SM'20) is the Elmore Rising Star Assistant Professor of Electrical and Computer Engineering (ECE) at Purdue University. His research interest is at the intersection of networking, communications, and machine learning, specifically in fog/edge network intelligence, distributed machine learning, and data-driven wireless network optimization. Dr. Brinton is a recipient of the NSF CAREER Award, ONR Young Investigator Program (YIP) Award, DARPA Young Faculty Award (YFA), Intel Rising Star Faculty Award, and roughly $15M in sponsored research projects as a PI or co-PI. He has also been awarded Purdue College of Engineering Faculty Excellence Awards in Early Career Research, Early Career Teaching, and Online Learning. He currently serves as an Associate Editor for IEEE/ACM Transactions on Networking. Prior to joining Purdue, Dr. Brinton was the Associate Director of the EDGE Lab and a Lecturer of Electrical Engineering at Princeton University. He also co-founded Zoomi Inc., a big data startup company that has provided learning optimization to more than one million users worldwide and holds US Patents in machine learning for education. His book The Power of Networks: 6 Principles That Connect our Lives and associated Massive Open Online Courses (MOOCs) have reached over 400,000 students to date. Dr. Brinton received the PhD (with honors) and MS Degrees from Princeton in 2016 and 2013, respectively, both in Electrical Engineering.