Towards Federated Learning at the Far Edge: From Static Topologies to Generalized Network Graphs

Takumi Suimon
The University of Osaka
Japan
Email: t-suimon@ist.osaka-u.ac.jp

Elisa Rojas
Universidad de Alcalá
Spain
Email: elisa.rojas@uah.es

Yuki Koizumi
The University of Osaka
Japan
Email: ykoizumi@ist.osaka-u.ac.jp

Javier Diaz-Fuentes
Universidad de Alcalá
Spain
Email: j.diazf@edu.uah.es

David Carrascal
Universidad de Alcalá
Spain
Email: david.carrascal@uah.es

Junji Takemasa
The University of Osaka
Japan
Email: j-takemasa@ist.osaka-u.ac.jp

***Abstract*—Federated Learning (FL) is increasingly expected to operate in far-edge environments, where device mobility, intermittent connectivity, and resource heterogeneity challenge static and centralized architectures. This vision paper reviews the evolution of FL frameworks with a particular focus on the topological scenarios represented by generalized network graphs in the far edge. Our analysis indicates that most existing FL frameworks either cannot accommodate such topologies or do so only at the cost of substantial performance trade-offs, revealing a fundamental architectural limitation. We identify hierarchical decentralized architectures as a promising direction, and discuss the enabling standards and open challenges required for their further development. Our goal is to provide researchers with a concise overview and useful starting points for studying FL methods towards far-edge environments.**

***Index Terms*—Federated Learning, Far Edge, Generalized Network Graphs, Edge Computing**

I. INTRODUCTION

The meteoric rise of Artificial Intelligence (AI) research has catalyzed the evolution of Federated Learning (FL) as a dominant distributed machine learning paradigm. By allowing local data to remain on-device and only exchanging model updates, FL addresses critical contemporary challenges in data privacy and regulatory compliance [1].

However, as FL migrates toward the far edge, the traditional reliance on centralized orchestrators or static topologies is becoming a bottleneck for performance, resilience and scalability [2]. This limitation arises because conventional FL designs are generally not compatible with the network topologies specific to far-edge environments, namely *generalized network graphs*, which are characterized by high mobility and resource constraints [3]. Although Decentralized FL (DFL) alleviates this topological mismatch by removing the central orchestrator and allows more flexible topology, it often does so at the cost of training performance. Therefore, extending FL toward the far edge requires evaluating whether existing frameworks can preserve performance in far-edge environments.

To this end, this work first concisely analyzes the state of the art in FL frameworks using the Key Performance Indicators (KPIs) required in far-edge environments, with particular attention to the trade-off between training performance and topological versatility. Our analysis shows that most existing FL frameworks are constrained by this trade-off, and further suggests that hierarchical decentralized architectures supported by Device-to-Device (D2D) multi-hop communication offer a promising way to reconcile these competing requirements. We then examine emerging use cases and industry standards to assess how these architectural insights align with practical deployment conditions. On this basis, our survey identifies the critical challenges and open issues that must be resolved for FL to operate effectively at the far edge. The main contribution of this work is that it uniquely focuses on the far edge from the perspective of topologies, unlike previous surveys [1], [2], [4], [5]. This factor is critical for advanced 6G scenarios yet remains under-explored in current literature.

The remainder of the article continues with Section II, which surveys FL works in relation to far-edge environments. Section III delineates relevant standards and use cases that drive these architectural shifts. Building upon this analysis, Section IV outlines the open challenges required to realize these future applications and Section V concludes the work.

II. EVOLUTION OF FL FRAMEWORKS: TOWARDS GENERALIZED NETWORK GRAPHS

This section analyzes the evolution of FL frameworks in response to growing demand for far-edge deployment. We first motivate the shift from static topologies to generalized network graphs by examining far-edge characteristics and recent FL trends. We then define the KPIs for assessing the robustness of FL frameworks in far-edge environments and evaluate existing frameworks. Finally, we identify their architectural limitations and discuss the potential and challenges of hierarchical decentralized architectures such as HDFL at the far edge.

A. Embracing network generality: FL in far edge

1) The far edge as a driver of generalized network graphs in 6G: In the cloud-to-thing continuum, the far edge represents

ISBN 978-3-903176-82-9

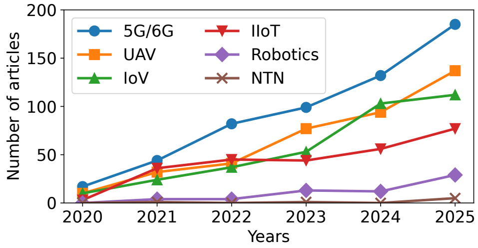

Fig. 1. Trends in keywords associated with FL use cases of far-edge scenarios (Survey date: April 2, 2026).

the localized processing layer situated at the extreme periphery of the network. It distinguishes itself from the near edge by operating on resource-constrained hardware—such as Internet of Things (IoT) sensors and smartphones—within the end-user's immediate environment to ensure real-time responsiveness and data sovereignty [6]. The evolution toward 6G networks introduces a paradigm shift toward pervasive intelligence, where decentralization, high mobility, and server independency are no longer optional features but core architectural requirements.

At the far edge, high device mobility and intermittent connectivity make it difficult to rely on a static, pre-defined topology. Instead, the network is better represented as a *generalized network graph*, in which both nodes and links may appear, disappear, or be reconfigured as the environment evolves. This abstraction is particularly important in sparse or remote settings, where persistent one-hop connectivity to a central orchestrator is often infeasible because of long latency or unstable radio conditions.

2) Research trends toward far-edge scenarios: To support the shift from static topologies to generalized network graphs, we examine research trends in FL for far-edge scenarios using Google Scholar. Based on current literature on far edge [6], [7], [8], we consider six representative use-case keywords: "5G/6G", "Unmanned Aerial Vehicle (UAV)", "Internet of Vehicles (IoV)", "Industrial IoT (IIoT)", "robotics", and "Non-Terrestrial Network (NTN)". Given the vastness of the literature, we restrict our analysis to the number of articles published between 2020 and 2025 whose titles specifically contain these keywords or their equivalents.

As shown in Fig. 1, all six keywords exhibit increasing trends, with especially strong growth for 5G/6G, UAV, and IoV. Since these use cases involve highly mobile devices such as smartphones, drones, and vehicles, their network structures are better modeled as generalized network graphs rather than static topologies. This trend reinforces the need to extend FL beyond static network assumptions in far-edge environments.

B. KPIs for evaluating FL in far edge

To evaluate existing FL frameworks, we define four KPIs for FL to operate effectively in far-edge environments. These comprise three established KPIs that capture the core performance under static topologies, together with *topological versatility*, a KPI introduced in this paper to assess whether that performance can be preserved under generalized network graphs. Taken together, these KPIs enable us to compare FL frameworks in terms of both their basic performance and their applicability to generalized network graphs.

1) Established KPIs: We evaluate FL frameworks using three established KPIs [8], which capture the fundamental performance of FL under static topologies. While they are generally important for FL, they become especially critical in far-edge environments.

Convergence Responsiveness: This KPI refers to the ability to drive the global model toward convergence within a small number of communication rounds. High convergence responsiveness not only leads to strong task performance but also reduces the computational and communication costs borne by individual nodes, since fewer communication rounds are required. This KPI is particularly important in far-edge environments, where nodes are subject to stringent computational and communication resource constraints.

Communication Efficiency: This KPI refers to the ability to avoid imposing a heavy communication burden on any particular node. One such case is when model updates from clients are handled by multiple aggregators, so that the communication burden is shared among them rather than concentrated on a single aggregator. This KPI is particularly important in far-edge environments, where nodes often operate under stringent bandwidth constraints.

Fault Tolerance: This KPI refers to the ability to maintain system operation despite the temporary or permanent failure or departure of arbitrary nodes. In other words, fault tolerance is high when the system does not rely on any single point of failure (SPoF). This KPI is particularly important in far-edge environments, where stable node operation cannot be assumed due to energy constraints and hardware limitations.

2) Topological Versatility: This KPI evaluates whether an FL framework can preserve the performance captured by the established KPIs even under generalized network graphs. It therefore serves as a binary determinant of the framework's applicability to far-edge environments.

To enable a clear assessment of whether an FL framework satisfies this KPI, we evaluate it based on its dependence on reachability. Specifically, an FL framework is said to achieve topological versatility if the system can operate without any node relying on one-hop reachability to a *particular* node. This criterion is motivated by the fact that, in far-edge environments, stable connectivity to a specific node cannot be guaranteed because of unstable links and high node mobility.

C. Evaluating FL frameworks

We compare existing FL frameworks and evaluate whether they can operate effectively in far-edge environments based on the KPIs defined above. Fig. 2 illustrates each framework, and Table I summarizes the evaluation.

1) CFL: Centralized Federated Learning (CFL) [9] is the foundational FL framework. It consists of a fixed server and mobile devices, thereby forming a star topology.

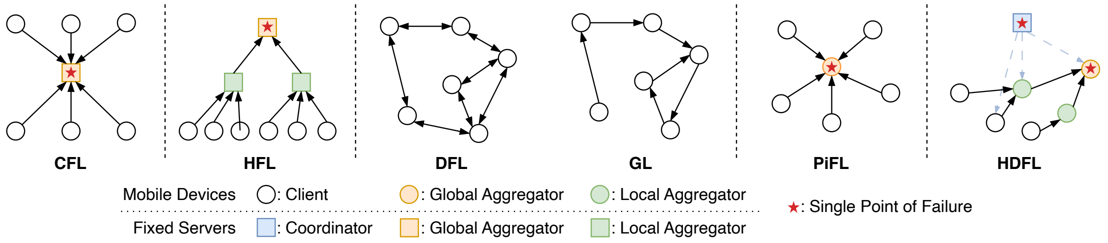

Fig. 2. Visual overview of FL frameworks and their associated topologies.

Overview: The server acts as the global aggregator, while the mobile devices act as clients. In each round, every client locally trains the model using its own dataset and sends the resulting model update to the global aggregator. The global aggregator then aggregates the received updates to construct a new global model and distributes it back to the clients. This process is iterated until the global model converges.

Evaluation: Under static topologies, CFL achieves high convergence responsiveness [10], because the updates from all clients are incorporated into the global model in each round, reducing the number of rounds required for convergence. In contrast, both communication efficiency and fault tolerance are low, because the communication burden is concentrated on the global aggregator [11], which also constitutes a SPoF [10].

Under generalized network graphs, CFL does not achieve topological versatility, because every client must maintain one-hop reachability to the global aggregator. With client mobility and intermittent connectivity, this requirement cannot always be satisfied, so updates from all clients are no longer incorporated into the global model in every round. Consequently, convergence responsiveness is compromised.

2) HFL: Hierarchical Federated Learning (HFL) [11], [12] is an extension of CFL designed to improve communication efficiency. It is deployed over a tree-topology network, where the leaf nodes are mobile devices, while the intermediate and root nodes are fixed servers.

Overview: The root server acts as the global aggregator, the intermediate servers as local aggregators, and the leaf mobile devices as clients. The key idea of HFL is to alleviate the communication burden on the global aggregator by distributing the aggregation process across multiple local aggregators. Specifically, each local aggregator aggregates updates from its child clients, which are assigned based on geographical proximity [12], and forwards the result to the global aggregator. The global aggregator then updates the global model using the updates of all clients, and broadcasts it back down the tree to the leaf clients.

Evaluation: Under static topologies, HFL preserves the high convergence responsiveness of CFL while improving communication efficiency. Specifically, global aggregators receive updates of all clients; meanwhile, the communication burden of the global aggregator is distributed across local aggregators. However, fault tolerance remains low, since the global aggregator still constitutes a SPoF [11].

Under generalized network graphs, HFL does not achieve topological versatility, because each local aggregator must maintain one-hop reachability to the global aggregator. With local-aggregator mobility, this requirement cannot always be satisfied, so updates from disconnected local aggregators are no longer incorporated into the global model, compromising convergence responsiveness. Reassigning those updates to other reachable local aggregators may partially mitigate this issue, but it concentrates communication on specific local aggregators, compromising communication efficiency. In other words, performance of HFL depends strongly on stable communication between local and global aggregators [13], which generalized network graphs cannot guarantee.

3) DFL/GL: Decentralized Federated Learning (DFL) [14] and Gossip Learning (GL) [15] were proposed to overcome the limited fault tolerance and communication efficiency of CFL. Unlike CFL, they are deployed in systems composed solely of mobile devices, without relying on any fixed servers.

Overview: The key idea of DFL and GL is that all mobile devices act as clients and operate symmetrically. Specifically, every client trains its local model and sends its update to geographically neighboring clients. The transmission mechanism differs between DFL and GL: in DFL methods (e.g., D-PSGD [14] and DFedAvg [16]), each client broadcasts its update to all neighbors, whereas in GL [15], each client sends its update only to one or more randomly selected neighbors. After receiving updates, each client aggregates them and updates its local model. This process is repeated until all clients' local models become identical (i.e., reaching consensus).

Evaluation: Under static topologies, DFL and GL improve communication efficiency and fault tolerance over CFL [17]. Communication efficiency is high because clients exchange updates only with neighboring clients, preventing any single client from bearing excessive communication overhead. This advantage is even greater in GL, where each client communicates with only one neighbor. Fault tolerance is also high, since all clients operate symmetrically and no client acts as a SPoF. The main drawback, however, is low convergence responsiveness due to the absence of a global aggregator [17]. In particular, an update generated by a client that is n hops away requires at least n rounds to be reflected in another client's local model. In GL, random transmissions can further increase the required rounds [18]. Consequently, information from local datasets propagates slowly across the system, requiring more rounds for convergence than CFL.

TABLE I
COMPARISON OF EXISTING FL FRAMEWORKS

Framework	Convergence Responsiveness	Communication Efficiency	Fault Tolerance	Topological Versatility
CFL	♦♦♦	♦◊◊	♦◊◊	✗
HFL	♦♦♦	♦♦♦	♦◊◊	✗
DFL/GL	♦◊◊	♦♦♦	♦♦♦	✓
PiFL	♦♦♦	♦◊◊	♦♦◊	✗
HDFL	♦♦♦	♦♦♦	♦◊◊	✓

Notation: ♦◊◊low, ♦♦◊medium, ♦♦♦high

Under generalized network graphs, DFL and GL achieve topological versatility because each client only needs one-hop reachability to *any* neighboring mobile clients. In other words, DFL and GL transition naturally to far-edge environments without substantially changing their behavior [19], since they do not rely on static or pre-defined topologies in the first place.

4) PiFL: Pivoting Federated Learning (PiFL[1]) [10], [20] was introduced to improve the convergence responsiveness of DFL and GL. Like DFL and GL, PiFL is deployed in systems composed solely of mobile devices.

Overview: The key idea of PiFL is to perform CFL-like aggregation without fixed servers by temporarily assigning the role of global aggregator to a mobile device. In each round, one device is selected as the global aggregator, while the others act as clients. This selection is performed autonomously based on a distributed consensus algorithm such as Raft [20]. Each client trains its local model and sends its update to the global aggregator. The global aggregator then aggregates the received updates to produce an updated global model and distributes it back to the clients. In the following round, a different device takes over the role of global aggregator.

Evaluation: Under static topologies, PiFL achieves high convergence responsiveness without fixed servers. That is, PiFL requires fewer rounds to converge because the global aggregator can collect updates from all clients in every round [10]. In contrast, introducing a global aggregator compromises the advantages of DFL and GL. First, fault tolerance is medium: lower than that of DFL and GL, but higher than that of CFL. This is because the global aggregator acts as a temporary SPoF during each round. Its role can be reassigned to another upon failure, which requires restarting the aggregation [21]. Second, communication efficiency is low, as the global aggregator must handle a significant communication burden by collecting updates from all clients.

Furthermore, PiFL lacks topological versatility under generalized network graphs because every client must maintain one-hop reachability to the selected global aggregator. As both clients and the aggregator may move, this condition cannot be guaranteed, so some updates may fail to reach the global model, degrading convergence responsiveness. Fault tolerance is also weakened because time-varying topologies make the communication conditions required for distributed consensus difficult to sustain [22]. In short, PiFL is fundamentally designed for full-mesh static topologies [23], limiting its applicability to far-edge environments.

5) HDFL: Hierarchical Decentralized Federated Learning (HDFL) [24], [18] is a framework designed to improve the convergence responsiveness of DFL and GL without compromising their communication efficiency. It is deployed in systems composed of a fixed server and mobile devices.

Overview: The key idea of HDFL is to perform HFL-like aggregation over a dynamically reconstructed tree formed among mobile devices. A fixed server acts as a coordinator, organizing devices into a tree based on their connectivity and hardware resources, where the root, intermediate, and leaf mobile devices serve as the global aggregator, local aggregators, and clients, respectively. Unlike HFL, both the global and local aggregators are implemented on mobile devices rather than fixed servers. In each round, local aggregators collect and aggregate updates from their child clients and forward the results to their parent node. The global aggregator then updates the global model by aggregating all received updates and disseminates the updated model back to the leaf nodes.

Evaluation: Under static topologies, HDFL preserves both high convergence responsiveness and high communication efficiency by hierarchical aggregation [18]. Specifically, while the communication burden is distributed across local aggregators, the global aggregator can receive updates from all clients in every round. Fault tolerance remains low because, although the global aggregator can be reassigned as in PiFL, the coordinator remains a persistent SPoF that is indispensable for constructing and maintaining the hierarchical topology.

Another key advantage of HDFL is that it achieves topological versatility under generalized network graphs. Each mobile device only requires one-hop reachability to one or more neighboring mobile devices to share its updates. While connectivity to the coordinator is still required, it does not need to be one-hop, as other mobile devices can act as relays.

D. Discussion

1) Architectural limitations in FL: The above analysis suggests that each KPI requires a distinct design choice:

- High convergence responsiveness requires a global aggregator that incorporates all client updates in each round.
- High communication efficiency requires that the aggregation is distributed across multiple local aggregators.
- High fault tolerance is best supported when all nodes play equivalent roles and no indispensable node exists.
- Topological versatility requires that the learning process depend only on one-hop reachability to arbitrary neighboring nodes, rather than to a particular fixed node.

These design requirements reveal a fundamental trade-off between convergence responsiveness and topological versatility. Achieving high convergence responsiveness generally requires a global aggregation point that collects updates from all clients in each round, but this in turn imposes one-hop reachability requirements on at least part of the aggregation hierarchy. Such requirements conflict with generalized network graphs, where nodes may reach only arbitrary neighbors. By

---

[1] PiFL is a name unique to this paper, as no official name exists. It is derived from the way the global aggregator *pivots* between mobile devices.

contrast, eliminating the global aggregator improves topological versatility, but at the cost of slower update propagation and reduced convergence responsiveness.

2) Potential and challenges of Hierarchical Decentralized Architectures: Hierarchical decentralized architectures, as exemplified by HDFL, are a promising direction for far-edge environments because they ease the trade-off between convergence responsiveness and topological versatility. This benefit comes from D2D multi-hop communication in every round. Specifically, D2D can replace the one-hop reachability requirement to the global aggregator with multi-hop reachability through arbitrary neighboring devices. Their hierarchical aggregation also improves communication efficiency.

However, hierarchical decentralized architectures face two major challenges. First, their deployment requires frequent D2D multi-hop communication, so their promise depends on realistic standards and supporting technologies. Second, fault tolerance remains limited because a coordinator for constructing optimal topologies becomes a persistent SPoF. A coordinator-free design might not be practical because topology optimization is essential in far-edge environments. In particular, without a coordinator, bottleneck links can lengthen each communication round and degrade convergence responsiveness, while an uneven communication burden reduces communication efficiency [25]. Assigning the coordinator role to a mobile device is also challenging, because doing so would require distributed consensus under generalized network graphs, much like selecting a global aggregator in PiFL.

III. REALIZING FAR-EDGE FL: STANDARDIZED ENABLERS AND INDUSTRIAL USE CASES

Building upon the comparative analysis of FL frameworks, it becomes evident that supporting generalized network graphs requires more than just algorithmic adaptation; it demands a synergy between decentralized learning logic and the underlying network infrastructure and driving use cases.

A. Standardization as a substrate for generalized graphs

The practical deployment of FL based on hierarchical decentralized architectures requires D2D multi-hop communication, and 3GPP Proximity Services (ProSe) and Sidelink provide key mechanisms in this direction. ProSe supports direct UE discovery and PC5 communication, while relay enhancements add functions for relay discovery, selection, and establishment [26]. In this way, HDFL could be realized by mapping its parent–child aggregation hierarchy onto relay-assisted D2D paths, so that model updates can still propagate through neighboring UEs even without direct one-hop connectivity. This makes ProSe and Sidelink a promising substrate for HDFL over generalized far-edge network graphs.

B. Vertical industry drivers and 6G Use Cases

The support for these technical advancements is further corroborated by industry-specific use cases: the 5G Alliance for Connected Industries and Automation (5G-ACIA) highlights IIoT scenarios that increasingly rely on 3GPP Sidelink for resilient local connectivity [27], while the 5G Automotive Association (5GAA) advocates for Vehicle-to-Network-to-Everything (V2N2X) communications to support high-mobility safety applications [28].

Expanding this vision, the one6G association [8] foresees distributed computational resources that extend from the core to deep-edge devices such as drones, smart factories, and autonomous vehicles, including use cases like mitigation of natural disasters or autonomous navigation. According to one6G, emerging AI and ML applications require a move away from centralized streaming toward multipoint-to-multipoint communication and in-network processing to handle the large complexity of data that exceeds traditional infrastructure capabilities. By advocating for a "Network-as-an-AI-Compute-Grid", one6G emphasizes that future networks must manage not only bandwidth but also computing and caching resources across the cloud-to-things continuum. This perspective validates the necessity of generalized network graphs for FL to enable continuous learning networks where mobile entities share models in real-time—addressing key KPIs such as data privacy, communication overhead, and network delays in highly heterogeneous settings.

IV. OPEN CHALLENGES AND RESEARCH OPPORTUNITIES

This section discusses the critical challenges and open issues for realizing FL based on hierarchical decentralized architectures at the far edge.

A. Distributed, near-optimal and responsive topology control

As argued in Section II-D2, high-performance FL over generalized network graphs requires (near-)optimal and responsive topology construction. However, reliance on a centralized coordinator is undesirable because it weakens fault tolerance and hinders rapid adaptation to network dynamics. Thus, a key challenge is to design fully distributed and autonomous topology-control algorithms. While distributed schemes like MuHoW [29] offer scalability for constrained devices [3], they remain insufficient for FL due to limitations in security and learning-related factors. Future work must evolve beyond conventional communication metrics to incorporate data and resource heterogeneity, ensuring it becomes fully FL-aware.

B. Resource-constrained heterogeneity

The far edge is characterized by extreme hardware heterogeneity, ranging from high-performance Multi-access Edge Computing (MEC) servers to energy-constrained Microcontroller Unit (MCU)-based sensors. Current frameworks often force a "one-size-fits-all" model architecture, which leads to underutilization of powerful nodes or the exclusion of weaker ones. Promising directions include heterogeneous model compression and partial-layer training, allowing each node to train only the model components that match its current resource availability. The key challenge is to realize such adaptive training in generalized network graphs without a coordinator for model assignment and distribution. An important target is enabling TinyML over these decentralized environments [30].

C. Security and privacy

From a security perspective, FL should be designed under a zero-trust assumption [31] because no permanently trusted node exists to provide participation control. A promising direction is therefore to establish trust through lightweight distributed-ledger mechanisms. From a privacy perspective, secure aggregation (SecAgg) protocols should be reconsidered for mobility-aware far-edge environments. Existing non-interactive SecAgg generally rely on fixed roles such as servers or committees [32]. These assumptions are unsuitable for generalized network graphs, where such roles may change frequently or disappear altogether, and SecAgg must therefore be redesigned without relying on persistent role assignments.

V. CONCLUSION

In this paper, we surveyed FL frameworks for far-edge environments through the lens of generalized network graphs. Our analysis shows that, despite rapid recent trends of far-edge demands, existing FL frameworks still cannot fully accommodate far-edge environments. While hierarchical decentralized architectures are a promising direction, they remain incomplete. To help close this gap, we discussed KPI trade-offs, standardization trends, industry use cases, and future research directions toward making FL at the far edge practical.

ACKNOWLEDGMENT

This work was supported by the following projects and institutions: project TUCAN6-CM (TEC-2024/COM-460), funded by the Comunidad de Madrid under ORDEN 5696/2024; project VERANO (CM/DEMG/2024-038), funded through the CM-UAH collaboration agreement; and JSPS KAKENHI Grant Number 24K02930.

REFERENCES

[1] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized Federated Learning: A Survey on Security and Privacy," *IEEE Trans. Big Data*, vol. 10, no. 02, 2024.

[2] E. T. Martínez 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 Commun. Surv. Tutor.*, vol. 25, no. 4, 2023.

[3] C. Bormann, M. Ersue, A. Keränen, and C. Gomez, "Terminology for Constrained-Node Networks," Internet Engineering Task Force, Internet-Draft draft-ietf-iotops-7228bis-05, Mar. 2026, work in Progress.

[4] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," *ACM Comput. Surv.*, vol. 56, no. 10, pp. 1–41, 2024.

[5] Q. Duan, J. Huang, S. Hu, R. Deng, Z. Lu, and S. Yu, "Combining Federated Learning and Edge Computing Toward Ubiquitous Intelligence in 6G Network: Challenges, Recent Advances, and Future Directions," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 4, 2023.

[6] G. Baldoni, L. Cominardi, M. Groshev, A. de la Oliva, and A. Corsaro, "Managing the Far-Edge: Are Today's Centralized Solutions a Good Fit?" *IEEE Consum. Electron. Mag.*, vol. 12, no. 3, 2023.

[7] A. Hazra, M. Adhikari, S. Nandy, K. Doulani, and V. G. Menon, "Federated-learning-aided next-generation edge networks for intelligent services," *IEEE Netw.*, vol. 36, no. 3, pp. 56–64, 2022.

[8] one6G Association, "6G Technology Overview," White Paper, 2025. [Online]. Available: https://one6g.org/download/6324/

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *AISTATS*, 2017.

[10] A. Dhasade, A.-M. Kermarrec, E. Lavoie, J. Pouwelse, R. Sharma, and M. de Vos, "Practical federated learning without a server," in *ACM EuroMLSys*, 2025.

[11] X. Liu, Y. Zhou, D. Wu, M. Hu, M. Chen, M. Guizani, and Q. Z. Sheng, "CPFedAvg: Enhancing hierarchical federated learning via optimized local aggregation and parameter mixing," *IEEE/ACM Trans. Netw.*, vol. 33, no. 3, 2025.

[12] K. Fan, W. Chen, J. Li, Q. Wu, M. Ding, X. Han, K. Wei, and X. Deng, "Mobility in hierarchical federated learning: Resource allocation and convergence analysis," *IEEE Trans. Cogn. Commun. Netw.*, vol. 12, pp. 2737–2752, 2025.

[13] X. Chen, G. Xu, X. Xu, H. Jiang, Z. Tian, and T. Ma, "Multicenter hierarchical federated learning with fault-tolerance mechanisms for resilient edge computing networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 47–61, 2025.

[14] 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," in *NeurIPS*, 2017.

[15] Z. Tang, S. Shi, B. Li, and X. Chu, "GossipFL: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 3, 2023.

[16] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, 2023.

[17] B. Li, W. Gao, X. Deng, J. Xie, Z. Xiong, M. Siew, B. Guo, S. Mao, and Z. Han, "Decentralized federated learning over time-varying and heterogeneous mobile computing networks," *IEEE Trans. Mob. Comput.*, vol. 25, no. 5, pp. 6688–6704, 2026.

[18] S. Abed, N. Sabar, and A. Mahmood, "T-FLASH: Topology-flexible latency-aware scheduling for hierarchical decentralized federated learning," *Inf. Fusion*, no. 103858, 2025.

[19] Y. Chen, C. Wu, Y. Lin, Z. Du, J. Tang, S. Djahel, and L. Zhong, "HDFL: A hierarchical decentralized federated learning framework for dynamic and heterogeneous IoV environments," in *IEEE INFOCOM WKSHPS*, 2026.

[20] R. Dautov and E. J. Husom, "Raft protocol for fault tolerance and self-recovery in federated learning," in *IEEE/ACM SEAMS*, 2024.

[21] Y. Yahata, K. Sugiura, and H. Matsutani, "A scalable secure fault tolerant aggregation for P2P federated learning," in *IEEE IPDPSW*, 2024.

[22] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *Int. J. Parallel Emergent Distrib. Syst.*, vol. 27, no. 5, pp. 387–408, 2012.

[23] Y. Lu, Z. Yu, and N. Suri, "Privacy-preserving decentralized federated learning over time-varying communication graph," *ACM Trans. Priv. Secur.*, vol. 26, no. 3, 2023.

[24] J. A. Rincon, C. Carrascosa, G. Lucca, and C. Marco-Detchart, "Heterogeneous communication in decentralized federated learning," in *IDEAL*, 2025.

[25] M. Wu, M. Boban, and F. Dressler, "Hierarchical federated learning in device-to-device networks with learning-topology co-optimization," *IEEE Trans. Mob. Comput.*, 2026, early access.

[26] P. Mach and Z. Becvar, "Device-to-device relaying: Optimization, performance perspectives, and open challenges towards 6G networks," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 3, pp. 1336–1393, 2022.

[27] 5G-ACIA, "Using 5g sidelink in industrial factory applications," Tech. Rep., 2023.

[28] 5G Automotive Association (5GAA), "Vehicle-to-Network-to-Everything (V2N2X) Communications: Architecture, Solution Blueprint, and Use Case Implementation Examples," 5G Automotive Association (5GAA), Tech. Rep., Jun 2025.

[29] J. Alvarez-Horcajo, I. Martinez-Yelmo, E. Rojas, J. A. Carral, and V. Noci-Luna, "MuHoW: Distributed protocol for resource sharing in collaborative edge-computing networks," *Comput. Netw.*, vol. 242, no. 110243, 2024.

[30] P. K. Myakala, P. Naayini, and S. Kamatala, "A survey on federated learning for TinyML: challenges, techniques, and future directions," *Partn. Univers. Int. Innov. J.*, vol. 3, no. 2, 2025.

[31] L. Wang, Z. Lu, M. Li, J. Wang, K. Gai, and X. Chen, "DeSA: Decentralized secure aggregation for federated learning in zero-trust D2D networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 21, pp. 1987–2001, 2026.

[32] J. Bell-Clark, A. Gascón, B. Li, M. Raykova, and P. Schoppmann, "Willow: Secure aggregation with one-shot clients," in *Crypto*, 2025.