This article has been accepted for publication in IEEE Communications Surveys & Tutorials. 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/COMST.2023.3315746

Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges

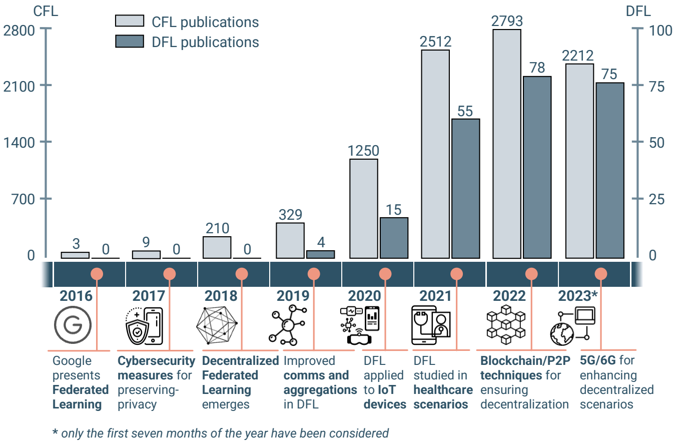
Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán, *Member, IEEE*

***Abstract*—In recent years, Federated Learning (FL) has gained relevance in training collaborative models without sharing sensitive data. Since its birth, Centralized FL (CFL) has been the most common approach in the literature, where a central entity creates a global model. However, a centralized approach leads to increased latency due to bottlenecks, heightened vulnerability to system failures, and trustworthiness concerns affecting the entity responsible for the global model creation. Decentralized Federated Learning (DFL) emerged to address these concerns by promoting decentralized model aggregation and minimizing reliance on centralized architectures. However, despite the work done in DFL, the literature has not (i) studied the main aspects differentiating DFL and CFL; (ii) analyzed DFL frameworks to create and evaluate new solutions; and (iii) reviewed application scenarios using DFL. Thus, this article identifies and analyzes the main fundamentals of DFL in terms of federation architectures, topologies, communication mechanisms, security approaches, and key performance indicators. Additionally, the paper at hand explores existing mechanisms to optimize critical DFL fundamentals. Then, the most relevant features of the current DFL frameworks are reviewed and compared. After that, it analyzes the most used DFL application scenarios, identifying solutions based on the fundamentals and frameworks previously defined. Finally, the evolution of existing DFL solutions is studied to provide a list of trends, lessons learned, and open challenges.**

***Index Terms*—Decentralized Federated Learning, Communication Mechanisms, Security and Privacy, Key Performance Indicators, Frameworks, Application Scenarios.**

I. Introduction

Artificial Intelligence (AI) and, in particular, Machine Learning (ML), will be one of the techniques that benefit from the vast amount of data expected in the coming years. However, data originating from a massive number of Internet of Things (IoT) devices are commonly stored in a distributed manner for a wide range of scenarios such as smart grids [1], remote health monitoring [2], or Internet of Vehicles (IoV) [3]. In such cases, data collection in central entities, as traditionally done in ML, is often infeasible or impractical due to limited communication resources, data privacy concerns, or country regulations. In 2016, Federated Learning (FL) [4] solved this issue by allowing entities (also known as participants, clients, or nodes of a federation) to train collaborative models without sharing training data. FL can be classified into two categories according to how federated models are created: centralized, known as Centralized Federated Learning (CFL), and decentralized, known as Decentralized Federated Learning (DFL). Nowadays, CFL is the predominant FL approach. It considers a central orchestration server to create and distribute a global model to the rest of the participants or clients. More in detail, clients train their models with local data, then send the local model parameters to a central server, where a global model is created by aggregating and combining the parameters of the individual model. Fig. 1 shows the timeline since Google introduced FL and how new features and approaches for decentralized scenarios have emerged. As can be seen, the growth for CFL and DFL is remarkable.



** only the first seven months of the year have been considered*

Fig. 1: DFL Timeline (source: Web of Science).

DFL, also known as Serverless or Distributed FL, emerged in 2018 to distribute the aggregation of model parameters between the neighboring participants [5]. The operation of DFL focuses on transmitting fast updates computed locally by each node (e.g., model parameters or gradients) and metadata (e.g., activations functions in neural networks) to

This work has been partially supported by *(a)* 21629/FPI/21, Seneca Foundation - Science and Technology Agency of the Region of Murcia (Spain), *(b)* the Swiss Federal Office for Defense Procurement (armasuisse) with the DEFENDIS and CyberForce projects, and *(c)* the University of Zürich UZH. (Corresponding author: Enrique Tomás Martínez Beltrán.)

Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Manuel Gil Pérez, and Gregorio Martínez Pérez are with the Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain (e-mail:enriquetomas@um.es; mqp@um.es; pedromiguel.sanchez@um.es; slopez@um.es; mgilperez@um.es; gregorio@um.es).

Alberto Huertas Celdrán is with the Communication Systems Group, Department of Informatics (IFI), University of Zurich, 8050 Zürich, Switzerland (e-mail: huertas@ifi.uzh.ch).

Gérôme Bovet is with the Cyber-Defence Campus, Armasuisse Science and Technology, 3602 Thun, Switzerland (e-mail: gerome.bovet@armasuisse.ch).

the rest of the federation nodes. Therefore, compared to CFL, DFL improves the limitations of having a single point of failure, trust dependencies, and bottlenecks at the server node [6]. Firstly, DFL improves fault tolerance because nodes constantly update their knowledge about available nodes or those that have ceased communicating [7]. This enhances the robustness of the network and mitigates the risk of a single point of failure, as it happens in CFL. Secondly, DFL improves trust issues by distributing trust between the federation nodes, which determines the trust of the entire federation, drastically reducing the single point of attack. Thirdly, DFL reduces the network bottleneck issue by allowing more evenly distributed communication and workload among the nodes, thus reducing the chances of congestion or delays in the overall network performance. Furthermore, DFL provides self-scaling federations, as new nodes can join the network and start communicating with other nodes, considering factors such as device throughput or computational capacity. Although the number of connections may decrease due to the limited number of nodes each node can share data with, the network remains tightly connected [4]. Finally, DFL enables more efficient use of resources by distributing the computing power required to aggregate the model parameters among all participating nodes, rather than relying on a single parameter server under centralized control [8].

Despite the benefits provided by DFL, it also introduces new challenges in areas such as communication overhead, training optimization, and trustworthy AI. Depending on how the model aggregation is distributed across the network, some DFL topologies may face increased communication overhead [9]. In these cases, careful design and optimization of the communication protocols, client selection strategies, and trust mechanisms can help address these limitations. Trust plays a critical role in these tasks, as it influences the decision of which clients to share and aggregate model parameters with. Additionally, the significant amount of model parameters exchanged from the network edge to the data centers carries the risk of saturating the backbone. As a result, using DFL necessitates reevaluating the infrastructure from a new perspective, as it reduces the need for centralized infrastructure while aiming to improve network performance and trustworthiness. In this context, trust becomes essential in addressing the emerging challenges of DFL, such as client selection and parameter sharing. These challenges warrant further exploration and research, as the paper at hand does.

To explore the research field of DFL in an organized fashion, the paper follows the methodological steps indicated in Fig. 2. First, it is essential to identify and study the key aspects that differentiate DFL from CFL (step 1 in Fig. 2). For this purpose, the literature has highlighted some elements such as (i) federation architecture, in charge of modeling the decentralized scenario based on participants, roles, decentralization of the nodes, and distribution of data features; (ii) network topology, defining the associations between the nodes of the federation [10]; (iii) communication mechanisms, orchestrating the exchange of model parameters within the federation [11]; and (iv) security and privacy, studying possible cyberattacks and the countermeasures to preserve data privacy and models robustness [12]. In addition to the above fundamentals, identifying Key Performance Indicators (KPIs) is important for assessing DFL performance [13]. Three perspectives can be explored in this context: nodes, communications, and models. The first focuses on evaluating the heterogeneity and dynamism of nodes in DFL, the second on the efficiency of inter-node communications for data exchange, and the third on the performance of ML/DL models for solving collaborative tasks. Finally, mechanisms to optimize KPIs are also essential to ensure the efficiency and multi-scenario adaptability of the approach [14]. In this context, this article expands upon the existing literature by comprehensively identifying and describing the DFL fundamentals, KPIs, and optimization mechanisms.

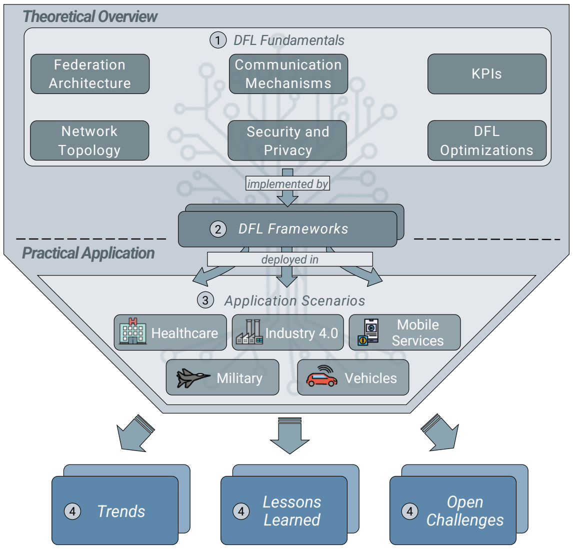


Fig. 2: Methodological steps and storyline of this work.

The aforementioned fundamentals are crucial for developing and comparing frameworks capable of training DFL-based models (step 2 in Fig. 2). These fundamentals are particularly important because they guide frameworks on managing model parameter exchanges efficiently, establishing federations with network participants, implementing effective aggregation mechanisms for local models, and fostering more reliable and trustworthy models. By considering these fundamentals, DFL frameworks can ensure better performance, security, scalability, and trustworthiness across a wide range of use cases [15]. Therefore, the article at hand also aims to support researchers working in decentralized systems using federated approaches with a review of the most relevant features of existing frameworks.

Then, application scenarios of DFL should be analyzed according to the fundamentals and frameworks they implement (step 3 in Fig. 2). At this point, it is important to identify strengths and weaknesses, and enables more informed decisions for selecting and deploying suitable approaches in real-world scenarios. Nowadays, the most used scenarios are: (i) healthcare, favoring the decentralization of clinical records and collaborative diagnosis [16]; (ii) Industry 4.0, improving the efficiency of automated industrial systems

[17]; (iii) mobile services, decreasing response times and increasing the bandwidth of constraints devices [18]; (iv) military, creating robust networks between Unmanned Aerial Vehicles (UAVs) and protecting them with cyber defense mechanisms [19]; and (v) vehicles, ensuring high mobility and local storage management [20]. Subsequently, each scenario is explored by studying the proposed solutions according to the previous fundamentals. Finally, all the aspects highlighted above (fundamentals, frameworks, and application scenarios) are analyzed with a view into the past, present, and future to extract trends, lessons learned, and challenges in the field of DFL. This is the last step of the followed methodology (step 4 in Fig. 2), which also shares the same structure and storyline as the document at hand.

II. Motivation and Contributions

DFL is an encouraging field of research that has repeatedly been reported as challenging in several review articles in recent years. While there are numerous surveys on FL, most focus primarily on the CFL-based approach and provide limited coverage of DFL. Therefore, to the best of our knowledge, this is the first survey providing a comprehensive literature review of DFL. The increase in the number of papers using DFL in different scenarios: healthcare [2], [21], Industry 4.0 [17], mobile services [18], military [19], and vehicles [22] motivate the need to identify the fundamental elements of DFL and review the work done to date.

Table I compares the most relevant surveys analyzing both CFL and DFL approaches. Regarding CFL, several works have attracted immense interest from academia and industry [37]. In this sense, in 2020, Lim *et al.* [23] reviewed the use of CFL in mobile networks using different end-user applications, while in 2021, researchers such as Nguyen *et al.* [24] and Khan *et al.* [25] examined the application of FL in more complex scenarios involving different heterogeneous IoT devices. Regarding security and privacy, Mothukuri *et al.* [26] defined secure protocols between server and participants, while Boobalan *et al.* [27] provided solid CFL fundamentals. Other articles, such as [28], reviewed the applicability in specific medical scenarios, discussing sensitive data privacy mechanisms. Finally, Witt *et al.* [29] detailed the frameworks most commonly used.

In contrast, most of the works covering DFL approaches focus on analyzing the use of Distributed Ledger Technology (DLT) technologies (e.g., Blockchain) [30]. Similarly, other authors explored the applicability of DFL in particular scenarios such as IoV [31], wireless communications [32], or UAV devices [33]. Moreover, the article by Wu *et al.* [34] provided a comprehensive survey of optimized DFL models and algorithms focusing on network topologies. The authors introduced various topologies with different participants and potential optimizations for them. Additionally, Chen *et al.* [35] analyzed the advances in FL, including aspects such as communications, privacy, and security requirements in DFL. Also, they provided solutions for collaborative training using more generalized algorithms. Further research has also been conducted on the frameworks that enable DFL. Witt *et al.* [36] proposed a set of solutions that address privacy and security issues in DFL. The authors also provided reward systems based on smart contracts to incentivize honest customer participation in the federated process. These latter works address the performance of DFL in selected application scenarios and detail critical elements of superiority over CFL approaches.

Despite the contributions of previous works, as illustrated in Table I, none addresses a literature review of DFL. In addition, no previous research studies the fundamentals of the emerging DFL approach. Similarly, recent work does not review DFL solutions from a fundamentals perspective as elements of analysis and comparison for different application scenarios. These solutions are often deployed on frameworks capable of managing the proposed federated approach [31], [38]. However, no study highlights which frameworks are feasible for this new decentralized approach nor collects the latest developments. Within this context, and adhering to the storyline outlined in Section I, it becomes essential to address the following research questions:

- *Q1. What are the fundamental aspects of DFL?* Depending on the federation architecture of the nodes, network topology, communication mechanisms between nodes, or security techniques, variations in the DFL approach may occur. Similarly, approach-based performance metrics and optimization techniques are needed. However, solutions in the literature ignore details about these elements and how they are combined.
- *Q2. What DFL frameworks exist, and what fundamentals do they provide?* There is no study reviewing existing frameworks able to build DFL solutions, analyzing their characteristics, and defining which application scenarios can be utilized.
- *Q3. Which are the main characteristics of the most relevant scenarios of DFL?* It is necessary to analyze and compare the previous DFL fundamentals and frameworks to solve the problems motivated by each application scenario. In addition, it is also necessary to detect the characteristics and limitations posed by existing solutions. The literature has not studied these approaches from a broad perspective to have a complete view.
- *Q4. What trends, lessons learned, and challenges have emerged in DFL?* To establish the guidelines for future research, it is critical to describe how DFL has evolved over the last years and what trends and open challenges are in the field.

To answer the previous research questions and provide readers with an up-to-date vision of DFL, the main contributions of this manuscript are:

- An analysis of the fundamentals that compose DFL: (i) federation architecture, (ii) network topology, (iii) communication mechanisms, (iv) security and privacy, (v) KPIs, and (vi) techniques to optimize the above KPIs, paying attention to the studies and applications that address each fundamental (answering *Q1* in Section III).
- A description of the main open-source frameworks to create and manage DFL solutions. This description

Table I: Comparison of surveys analyzing CFL and DFL.

Ref.	Year	FL Approach	Device Types / Area	Fundamentals	Frameworks	Application Scenarios	Key Performance Indicators	Focus and Solution Categorization
[23]	2020	CFL	Mobile networks	✓	✓	✗	✓	• A survey on the integration of FL and edge computing. • The applications of FL in IoT networks and services have not been explored and discussed.
[24]	2021	CFL	IoT devices	✗	✗	✓	✗	• A survey on FL in an IoT scenario, reviewing data distribution, ML models, privacy mechanism, and communication architecture. • The study has not discussed other approaches or frameworks capable of managing the scenario.
[25]	2021	CFL	IoT devices	✓	✗	✓	✓	• A systematic review of FL in IoT scenarios, detailing the application areas and limitations. • The study only considers CFL architectures without discussing the disadvantages compared to other approaches.
[26]	2021	CFL	Security Privacy	✗	✓	✓	✗	• A survey of FL security and privacy, defining secure protocols. • The analysis of secure measures using different FL approaches is missing.
[27]	2022	CFL	FL baselines	✓	✓	?	?	• A survey on the FL baselines with a basic introduction to definitions and architectures. • The application scenarios of FL have not been discussed.
[28]	2022	CFL	Healthcare	?	✗	✓	✗	• An overview of the use of FL in Healthcare scenarios. • Limited overview of techniques for deploying realistic scenarios.
[29]	2022	CFL	Framework review	✓	✓	✗	✗	• An overview of frameworks for deploying DFL-based architectures. • Lack of identification of application scenarios.
[30]	2022	DFL	DLT	✗	✗	✗	?	• A description of the challenges and applications of Blockchain. • The paper only focuses on the Blockchain technology to ensure node decentralization.
[31]	2022	DFL	IoV	✗	✓	✓	?	• A brief description of FL in fog radio access networks. • The applications of FL in IoT networks have not been presented.
[32]	2022	DFL	Wireless communications	✗	✗	✓	✓	• A review of the techniques for adapting FL to distributed environments. • The paper only reviews techniques without giving specific application scenarios or frameworks.
[33]	2022	DFL	UAV devices	✗	✗	✓	✓	• A brief survey on the application of FL in UAV networks. • Other domains, such as smart city or IoT devices, are not considered.
[34]	2023	DFL	DFL baselines	?	✗	?	✓	• A study of optimized DFL models and algorithms focusing on network topologies. • Limited review regarding federation architectures, device heterogeneity, and frameworks.
[35]	2023	DFL	DFL baselines	?	✗	✓	✓	• A comparison of CFL and DFL federation architectures regarding topologies, privacy, and security. • The study does not have enough information about DFL, including algorithms, optimizations, or metrics for evaluating its performance.
[36]	2023	DFL	Framework review	?	✓	✗	✗	• A systematic review of DFL frameworks, highlighting differences, limitations, and future research directions. • The study focuses only on Blockchain-related frameworks and participant reward techniques.
This work	2023	DFL	Healthcare Industry 4.0 Mobile services Military Vehicles	✓	✓	✓	✓	• A survey on the baselines and fundamentals of DFL, generating a novel taxonomy in the literature. • The paper explores the application of DFL solutions and frameworks in various areas of interest.

✓ fully addressed, ? partially addressed, ✗ not addressed by the work

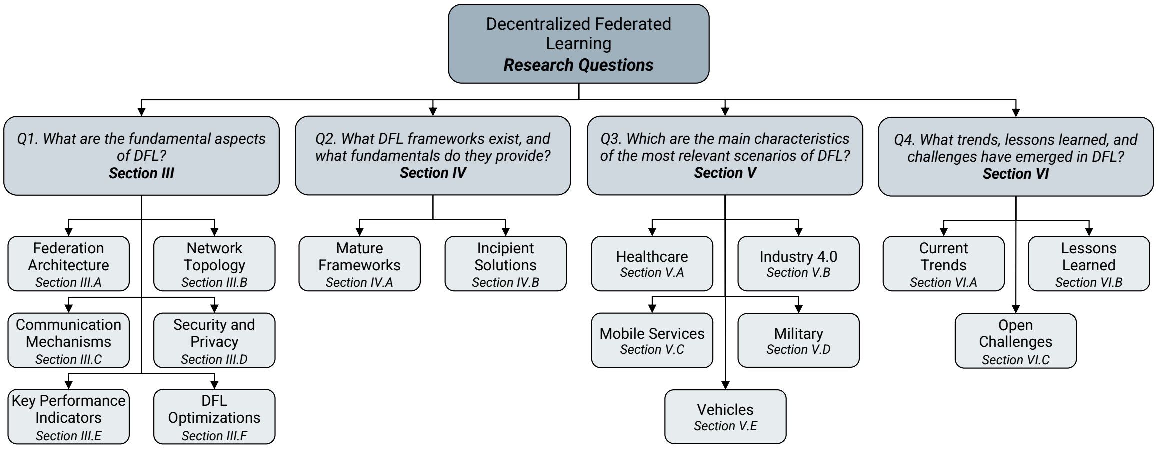


Fig. 3: Research questions and sections answering them.

is divided into mature literature solutions that have been redesigned to offer decentralized architectures and incipient solutions that address specific scenarios (answering *Q2* in Section IV).

- A comprehensive review and comparison of the characteristics, advantages, and limitations of the most relevant solutions in the literature on the most used application scenarios: (i) healthcare, (ii) Industry 4.0, (iii) mobile services, (iv) military, and (v) vehicles (answering *Q3* in Section V).
- A set of current trends, lessons learned, and future challenges drawn from DFL works and frameworks

reviewed (answering *Q4* in Section VI).

Fig. 3 shows where and how the above questions are addressed in the paper at hand, acting as a table of contents. In particular, Section III analyzes the fundamentals of DFL in terms of architectures, topologies, communications, security, performance indicators, and techniques for optimizing scenarios. Section IV examines the main open-source frameworks that support the deployment of DFL with the previously defined fundamentals. After that, Section V describes and compares the leading solutions found in the state of the art, analyzing deployment considerations, the robustness of the proposed solution, and the results obtained. Section VI draws a set of lessons learned, current trends, and future challenges in the research area. Finally, Section VII provides an insight into the conclusions extracted from the work.

III. Fundamentals and Taxonomy

The fundamentals of DFL establish the groundwork for analyzing existing frameworks and solutions in the literature. Following the storyline presented in Fig. 2, this section responds to "*Q1. What are the fundamental aspects of DFL?*," identifying and analyzing the following fundamentals: (i) federation architectures, (ii) network topology, (iii) communication mechanisms, (iv) security and privacy, (v) KPIs, and (vi) techniques to optimize the previous metrics. These fundamentals are detailed in Fig. 4, showing the subcategories that compose them and the resulting taxonomy. Additionally, Table II details the above taxonomy with a detailed definition of the various fundamentals and the literature solutions that address them.

A. Federation Architecture

DFL solutions require an architecture that allows effective collaboration and communication between network participants. In this sense, this work analyzes the fundamentals based on the federation type in the communication, differentiating organizations or data centers from other devices. Subsequently, DFL architectures can also be categorized based on the roles of the participants. In this classification, each participant is assigned a specific task within the federated learning process. Furthermore, the decentralization schema differentiates between a decentralized, semi-decentralized, and centralized architecture. The semi-decentralized approach can be seen as a hybrid between centralized and decentralized architectures, offering a balance of both. Finally, the definition of DFL architectures is based on the data distribution during federation. In the following sections, each category will be discussed in greater detail.

1) Federation Type: The literature categorizes DFL architectures according to the type of federation, which is based on the participating nodes: cross-silo and cross-device. The differences lie in the number of participants and the amount of data stored in each one [39].

In cross-silo DFL, the nodes are organizations or data centers [40]. There is usually a relatively small number of nodes (<100), each with a large amount of data (about millions of samples), perhaps distributed and aggregated from consumers in different businesses. At the same time, nodes use consistent, robust, and scalable computing over time. Likewise, these nodes have a high performance in the network, avoiding points of failure during communications between nodes. For example, it is applied in different hospitals by training a federated model for tumor classification while keeping their Positron Emission Tomography (PET) images locally [116].

In cross-device DFL, the number of nodes is relatively large (>100), where each node has a relatively small amount of data (about thousands of samples) and limited computational power [41]. Due to concerns regarding power consumption, individual devices cannot be asked to perform complex training tasks. Furthermore, nodes could periodically disconnect from the network so that the network dropout rate would increase considerably, negatively impacting DFL performance. The nodes are usually on-edge devices or robots like UAVs [42], where communications between devices are often weak if not kept in a close coverage radius.

2) Participant Role: Network nodes may have one or more roles that define their behavior and operation in DFL architectures. In particular, the existing roles are trainer, aggregator, proxy, and idle. A trainer node aims to train a local model with its local dataset and transmit the parameters to its neighboring nodes [43]. The trainer node expects to receive the parameters of the updated federated model to incorporate it back into the local model. In contrast, the node with the aggregator role is responsible for obtaining the parameters from the neighboring nodes, aggregating them in the global

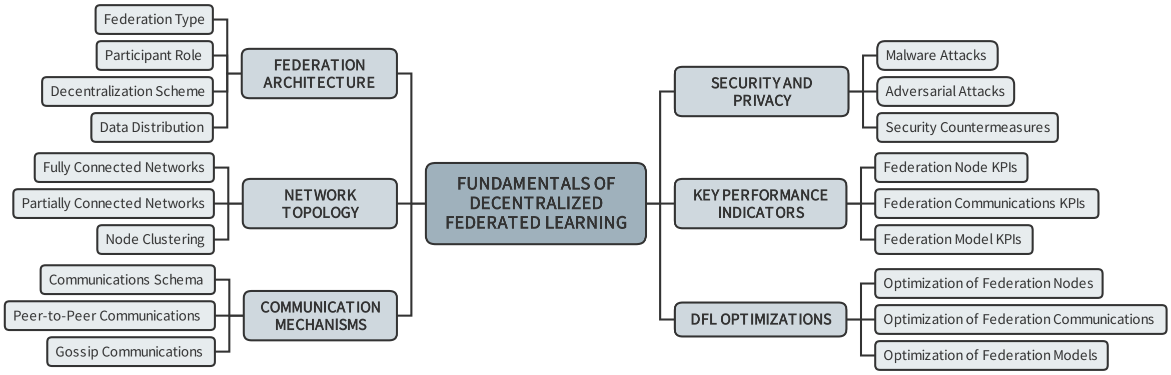


Fig. 4: Taxonomy with the fundamental aspects of DFL.

Table II: Comparison of solutions addressing different DFL fundamentals.

Taxonomy	Fundamentals	Definition and Solutions Analyzed in the Literature	References
Federation Architecture	Federation Type	• Federation from data silos or data centers	[39], [39], [40]
		• Federation from different types of devices, such as mobile phones or IoT devices	[2], [41], [42]
	Participant Role	• Identify the functionality of participants during federation	[43], [44]
	Decentralization Schema	• Techniques for enabling DFL while maintaining model convergence and data privacy	[6], [45], [46]
		• Balance the power and leadership in SDFL while maintaining the benefits of DFL	[14], [47]
		• Combine centralized and decentralized elements to optimize FL performance and scalability	[48]
	Data Distribution	• Solutions for HFL, VFL, and TFL partitioning the dataset	[42], [49]
		• Explore ways to sample data to ensure better distribution across nodes	[50]
Network Topology	Fully Connected Networks	• Define the federated networks with participants connected to each other	[9], [51]
		• Measure of robustness, flexibility, fault tolerance, cost, or security	[10], [52]
	Partially Connected Networks	• Define the classification in star, ring, or random structured networks	[53], [54]
		• Measure of robustness, flexibility, fault tolerance, cost or security	[55], [56]
	Node Clustering	• Define and classify the federated networks based on participants clusters	[44], [57]
		• Measure of robustness, flexibility, fault tolerance, cost or security	[58]
Communication Mechanisms	Communications Scheme	• Data exchange where all participants transmit their models at the same time	[11], [59]
		• Update participants models independently at different times	[60], [61]
	Peer-to-Peer Communications	• Methods for designing P2P topologies in DFL	[62]–[64]
		• Develop techniques for disseminating data and models in P2P networks	[65]–[68]
	Gossip Communications	• Methods to design gossip protocols in DFL	[69], [70]
		• Strategies to select nodes and optimize the efficiency in gossip-based communication	[71]–[74]
Security and Privacy	Malware Attacks	• Determine the impacts of malware threats in decentralized scenarios	[75]–[77]
		• Identify the risks associated with fully connected, star-structured, and random networks	[12], [49], [56]
	Adversarial Attacks	• Identify adversarial attacks affecting decentralized architectures during collaborative federation	[49], [78], [79]
		• Compare the adversarial attacks between different FL approaches and their adaptive capacity	[80], [81]
	Security Countermeasures	• Secure techniques to protect the transmission of model parameters between participants	[4], [82]–[84]
		• Encryption mechanisms while preserving reliability between participants	[45], [83], [85]
Key Performance Indicators	Federation Nodes KPIs	• Measure the node performance considering the heterogeneity and dynamism of the network	[13], [86]
		• Quantify the resource capabilities and node mobility of participating nodes in DFL scenarios	[7], [87]
	Federation Communications KPIs	• Measure the reliability of model parameter exchanges between participants in DFL	[10], [88]
		• Evaluate communications flexibility and overhead in DFL	[7], [89]
	Federation Models KPIs	• Measure the performance by solving different tasks or using multiple datasets for benchmarking	[13], [90], [91]
		• Mechanisms to ensure the trustworthiness of federated models	[92]–[96]
DFL Optimizations	Optimization of Federation Nodes	• Optimal methods for selecting nodes to transmit model parameters	[97], [98]
		• Optimize algorithms to suit the heterogeneous node resources and capabilities	[26], [58], [99]
	Optimization of Federation Communications	• Efficient distribution schemes that minimize the number of network exchanges	[71], [100]–[102]
		• Compression techniques to reduce the amount of data that needs to be exchanged	[103]–[107]
	Optimization of Federation Models	• Explore efficient ML models to increase model performance and reliability during federation	[88], [108]–[111]
		• Use of meta-learning, multitasking, and federated distillation in federation tasks	[91], [112]–[115]

model, and transmitting them to the neighboring nodes. In specific network topologies (see Section III-B), aggregators are not directly reachable by other nodes, and proxy nodes are needed. This role is intended to relay the received model parameters to neighboring nodes, allowing different nodes or network topologies to be interconnected [44]. However, a participant may not have any of the above roles, being idle in the network and not participating in the federation.

3) Decentralization Schema: The federation architecture can be affected by the decentralization level maintained between participants, with three different approaches emerging (see Fig. 5): DFL, Semi-Decentralized Federated Learning (SDFL), and CFL. In DFL, participants perform four steps independently: local model training, parameter exchange, local model aggregation, and parameter exchange again. In SDFL, participants perform the first two steps, while an aggregator participant handles the third step and transfers leadership for the aggregation functionality (step 5). In CFL, a central server handles parameter aggregation (step 3), with the rest of the network receiving and updating their local models accordingly (steps 4 and 5).

In DFL architectures, the network is fully autonomous, managing communications with other nodes and aggregating the model parameters independently of the other nodes. Each network node can select one or more nodes to transmit data, defining bidirectional communications between them. Furthermore, decentralization can be assumed to be fixed or dynamic, as the interconnections between nodes can change over time. Unlike the node-server architecture, the absence of a centralized server in DFL allows for the relaxation of the synchronous model update. Different solutions in the literature make several assumptions about network connectivity, particularly considering that each node is connected to all other nodes in the network or only to a set of neighboring nodes. Therefore, issues such as fixed or dynamic topology arise and the use of directed or undirected graphs [46]. A fixed topology provides stability and predictability but limits adaptability, while a dynamic topology allows flexibility but requires additional overhead to maintain connectivity. In contrast, directed graphs model asymmetric relationships, and undirected graphs are more flexible for modeling symmetric relationships between participants.

In contrast to the previous approach, SDFL architectures maintain an aggregator role that rotates among the participants belonging to the network. The aggregator transmits the leadership (aggregation functionality) periodically during the federation, selecting the neighboring node randomly or based on its performance, such as network, computational, or power capacity [47], [48]. The leadership in SDFL architectures plays an essential role in determining which node is responsible for collecting the model updates and computing the next-generation model by aggregation. Also, the choice of leadership selection mechanism can significantly impact the performance, efficiency, and robustness of the system.

Finally, in CFL-based architectures, data decentralization is orchestrated by a single aggregator. This central node aggregates the model parameters shared periodically by each trainer and creates a robust central model distributed among the nodes [6]. In this approach, the aggregating participant is fixed during federation, with the central node

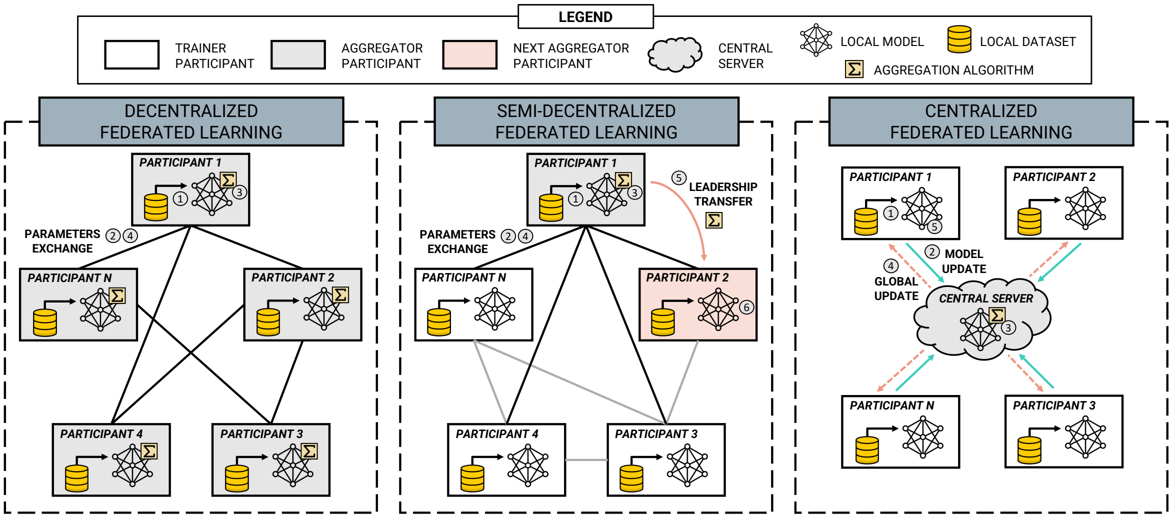


Fig. 5: Common approaches based on DFL architecture decentralization. The numbers inside the circles represent the training lifecycle.

being responsible for aggregating model parameters for each trainer in the network. Additionally, the CFL architecture makes it possible to ignore problems related to topology and device performance by not having decentralized aggregators. To improve this situation, techniques such as Multiparty Computation (MPC) can guarantee the immutability of model updates and coordination between participants directly, as could be the case for healthcare institutions [45].

4) Data Distribution: The DFL approach, like CFL, has different configurations depending on the properties of the distributed data. Data can be Independent and Identically Distributed (IID) when the federation participants behave similarly. Therefore, the data distribution does not fluctuate ($x^i \sim D$), and the events associated with the data points are independent, i.e., they are not connected to each other in any way ($\nexists j \; p(x^i, x^j) = p(x^i) \; p(x^j)$), where p represents the joint probability distribution of the data points. However, in some application scenarios, nodes are usually heterogeneous. Hence, the data often have a non-IID format, where data vary in quality, diversity, and quantity in the network, thus increasing the complexity of modeling, analysis, and evaluation. The problem of heterogeneous data distribution poses a challenge for DFL. It can cause local model parameters to converge at different stationary points for different participants, making global convergence difficult. In addition, there might be multiple aggregated models during the federation where nodes are connected irregularly, leading to different global models receiving data with varying distributions. Addressing this challenge involves determining how to handle these various models. One option presented in the literature is to continue aggregating them at a higher level [117], while another option is to accept the presence of different federated models based on the network topology and characteristics of the nodes [118]. By considering these alternatives, DFL can better address the issues arising from non-IID data distributions.

Finally, depending on the nature of the problem to be solved and the organization of data (features and samples) among nodes, DFL is usually divided into three types: Horizontal Federated Learning (HFL) [42], Vertical Federated Learning (VFL) [49], and Transfer Federated Learning (TFL) [50]. HFL is the most commonly employed method in DFL, as it involves sample federation and is applicable when there are many overlapping features and few overlapping nodes, typically associated with cross-device scenarios. In contrast, VFL and TFL require more in-depth analysis and comparison, as they are more complex to adapt and deploy in application scenarios that take advantage of their unique data organization. VFL focuses on feature binding when there are many overlapping nodes and few overlapping features, while TFL is considered when there is a limited feature and sample intersection between nodes. The goal of TFL is to build efficient models for specific applications in cases where data are sparse, which makes it another relevant aspect of DFL strategies.

The successful transition from CFL to DFL involves the adaptation of commonly used CFL datasets. Techniques like decentralized sparse partitioning prove instrumental in aligning these datasets with the unique characteristics of DFL [38]. These techniques consider the aggregation algorithms, participants' roles, and the task to be solved. Similarly, the distinct data distribution across nodes in DFL could be addressed using federated sampling techniques, which help maintain data diversity while ensuring statistical accuracy [119]. These adaptations make it feasible to use CFL datasets in DFL scenarios efficiently, leading to optimized learning processes and improved performance of DFL implementations.

B. Network Topology

In DFL, the network topology describes the organization of participants and determines their communications. Therefore, the importance lies in the impact on convergence, generalization, overhead, and robustness of the DFL approach. There are three network topologies used in DFL: fully connected networks, partially connected, and node

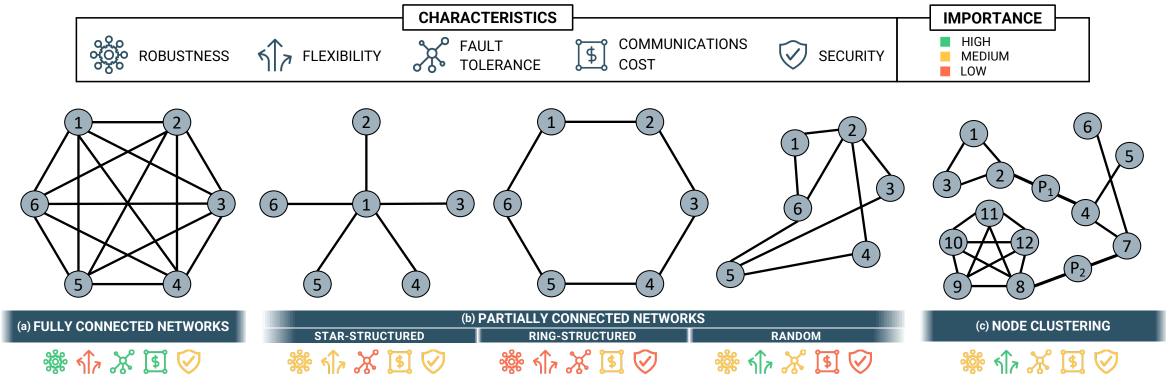


Fig. 6: DFL network topologies. (a) Fully connected networks, (b) Partially connected networks (star-structured network, ring-structured network, random network), (c) Node clustering. P_n indicates the node acting as a proxy.

clustering. Fig. 6 shows these network topologies and their characteristics. In particular, the topologies are evaluated depending on the importance (high, medium, low) of robustness, flexibility, fault tolerance, communications cost, and security.

1) Fully Connected Networks: This network topology maintains direct links between all pairs of nodes. When a new node is added to the system, a link must be added to each node (see Fig. 6a) [10]. Thus, the communication cost is high and grows with the number of nodes in the network. Moreover, adding new nodes increases the complexity of managing connections for each node, resulting in low flexibility. Despite the high communication cost and low flexibility, this topology is highly reliable and robust, as the network can still function when a few nodes or links fail [51].

2) Partially Connected Networks: In some network topologies, participating nodes may only have direct links to some nodes in the network, regardless of the shape of the topology. Therefore, the basic transmission cost for each node is lower than in fully connected networks since the model parameters are not transmitted to all nodes. As a result, communication cost may be higher than in fully connected topologies since the model parameters transmitted may go through several intermediate trainer participants until they reach the aggregating participant, resulting in a longer delay [55]. This work divides partially connected networks into star-structured, ring-structured, and random.

In a star-structured network, one of the nodes acts as a proxy participant and enables the federation communication with all other nodes (see the first subfigure of Fig. 6b). When a new participant is added, only one link is needed to connect them to the central node. Consequently, the communication cost grows linearly as the number of nodes increases, and the resources of the node limit network scalability. However, since all communications between non-central nodes must pass through the central node, it becomes a potential bottleneck, reducing flexibility. Additionally, the network has low fault tolerance, as the failure of the central node disrupts the communication between all nodes. Some DFL solutions use temporary connections in star topologies, but establishing and terminating these connections adds significant overhead.

In a ring-structured network, nodes are connected circularly (see the second subfigure of Fig. 6b). The communication cost grows linearly as the number of nodes increases, as each node only maintains two fixed links, regardless of network size. This contributes to medium flexibility. However, as the number of nodes grows, transmission delays for model parameters also increase. Ring networks can be unidirectional, where nodes transmit model parameters to only one neighbor, or bidirectional, where nodes send parameters to both neighbors, providing higher reliability and fault tolerance than unidirectional rings.

Sometimes a defined topology is not supported by the federation, and nodes generate connections following heuristics based on the proximity or computational capacity of nearby neighbors. Due to intermittent connections in DFL, the network maintains a dynamic node structure, creating random network topologies over time [53]. In this sense, some topologies use expander graphs and deterministic local optimization algorithms to reduce communication overhead in these networks. The third subfigure of Fig. 6b represents a network topology with six random nodes. Specifically, it describes a network based on the Erdös–Rényi model, where links between nodes are established randomly [54]. This type of network offers high flexibility and moderate fault tolerance but can result in higher communication costs and variable security levels.

3) Node Clustering: Recent publications have identified another DFL network topology for creating hierarchical clusters with models adapted to the distribution of nodes. In this regard, the literature addresses two approaches: similarity-based clusters [120] and proxy-based clusters [44]. In the former, clusters are determined by the similarity of the local model parameters of the nodes. As a result, each cluster is more individualized for the nodes that compose it, obtaining a homogeneous performance between nearby nodes. In contrast, proxy-based clusters aim to use nodes that interconnect different topologies, forwarding model parameters. In this regard, proxy nodes have to be evaluated in terms of performance since they transmit data from one cluster to another, generating an important bottleneck in the overall architecture [58]. The disadvantage of this clustering method implies that all nodes are initially linked to a coordinating node, which is not feasible in all application scenarios.

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. 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/COMST.2023.3315746

Furthermore, since nodes within a cluster share similar data distributions, individually trained cluster models may be less generic and robust than models exposed to global data distributions [57]. Fig. 6c shows the usage of proxy nodes to interconnect topologies in node clustering.

C. Communication Mechanisms

This fundamental defines the procedures that enable DFL communications between network nodes. First, the communications scheme in DFL, based on synchronous, asynchronous, and semi-synchronous communications, is detailed. Then, this subsection presents two different decentralized communications mechanisms: peer-to-peer (P2P) [62] and gossip [121].

1) Communications Scheme: The communications schemes determine the behavior of the nodes when transmitting or receiving model parameters from neighbors and then aggregating them. Depending on the type of scenario and the objective pursued, DFL can be supported by synchronous, asynchronous, or semi-synchronous communications. The first alternative requires each node to perform local training, consisting of multiple steps, usually expressed in epochs. An epoch is defined as a complete iteration through the entire local dataset. After completing local training, all nodes acting as trainers send the parameters of their local models to all their neighboring nodes and receive the new model they have to integrate. This training technique, known as synchronization points, is repeated for several federation rounds. In this sense, the nodes will have to wait for the end of the round to transmit their model parameters again. Therefore, a drawback of this scheme is its slow convergence due to waiting for slow or idle participants. To address this problem, over-selection techniques could increase participant selection but also lead to resource waste and selection bias [11].

In asynchronous communications, no synchronization point exists, and nodes can transmit and receive model parameters independently of the other federation participants. Because there is no idle time for participating nodes, asynchronous protocols have a faster convergence speed [61]. However, they cause higher communication costs and lower generalization due to staleness [60]. It is also necessary to detect stragglers (i.e., slow devices) in the federation, delaying data propagation through the network topology. In the same way, aggregators need aggregation policies adapted to the number of model parameters received, as they vary during federation rounds. Nevertheless, asynchronous communications are well suited for cross-device DFL scenarios, in which nodes have varying computational capability and intermittent availability [59].

Finally, semi-synchronous communications provide a balance between resource usage and communication costs. In this schema, each node trains locally until a predefined synchronization point is reached. Consequently, nodes with varying processing capacities and data volumes perform different numbers of epochs. The batch serves as the basic computation unit, allowing for finer control over when a node contributes to the federation. Some studies, such as [60], introduce thresholds for each device that compare the changes in local model parameters with available local resources to determine the benefits of aggregation at each round.

2) Peer-to-Peer Communications: Traditionally, the deployment of overlay networks has been used mainly for P2P message exchange, online social networks, and routing infrastructures. DFL leverages P2P message exchange networks for local model parameter exchanges at each node. However, unlike traditional P2P networks, DFL deals with a dynamic and heterogeneous topology, where participants often change their location or role in the federation [62]. In this regard, selecting d random neighbors to exchange information is not always feasible. The main drawback of DFL is that nodes cannot randomly choose d neighbors among existing nodes since there is no central coordinator to determine these associations [67]. However, each node can determine neighbors if global information is given when the topology is created, and neighbor discovery mechanisms are implemented during federation [64]. Other alternatives are to employ distributed Delaunay triangulation networks for wireless sensor networks or regular random topologies for data center and memory interconnection networks [122].

3) Gossip Communications: Gossip communications enable asynchronous transmissions of model parameters between network nodes. This type of communication relies on the P2P sampling service with the objective that nodes in the federation communicate and exchange data [71]. The comparison of the DFL approach using gossip communications versus the CFL approach has been extensively studied, with superior performance obtained in the former when a better understanding of model parameters is employed [69]. However, this mechanism maintains a restrictive functionality where communication occurs between close neighboring nodes. To address this limitation, additional processes are presented to determine the location of a node using random walks [123] and node selection techniques, relying on the heuristics of the distance between nodes or underlying device features [72]. For instance, local models can be updated with partially received parameters and be optimized by mixing weights according to the matrix of communication link reliability [89]. Finally, some application scenarios of gossip communications in DFL are providing accurate recommendations (e.g., location logs, movie ratings) [72] or in Distributed Online Social Networks (DOSN) where users can directly communicate only with their immediate neighbors in a social graph [73].

D. Security and Privacy

Decentralized scenarios may be more susceptible to attacks than CFL scenarios as the number of participants grows within the federation. Nevertheless, the level of vulnerability largely depends on the network topology. For example, in a fully connected network, where each participant has numerous connections, the impact of an attacker might be less severe than in a centralized CFL scenario since the attacker can only compromise a limited portion of the network. Nonetheless, decentralized scenarios still face increased risk due to the numerous intermittent and often weak connections between the participants.

In this sense, DFL can be subject to various attacks, including malware deployment and adversarial attacks, which can affect federation models and decentralized data, similar to the CFL approach [76]. While the former exploits intrinsic DFL fundamentals in the architecture or communications to affect the federation adversely, the latter aims to infer federated models and local node data, manipulate preconditions or destroy the model. Notable among these attacks are model inversion or membership inference, which can potentially infer the raw data by accessing the model [49]. Despite these potential attacks, countermeasures have been adopted to address them, preserve data and model privacy, and develop recovery plans to minimize harm and disruption. In particular, this paper presents two main approaches adopted in current DFL scenarios for data protection: cryptographic methods and differential privacy.

1) Malware Attacks: Malware, also known as "malicious software", is a critical issue in the DFL approach due to the fundamentals based on decentralized architectures without a server coordinating the scenario. Malicious entities with high propagation through the network can affect behavior and interfere with its proper functioning. This issue differs in impact compared to more traditional approaches such as CFL, mainly because DFL does not have a centralized entity to coordinate the cooperation of participants in the federation. Thus, malware can be created and deployed directly among new network participants. Due to the number of connections to neighboring nodes, fully connected networks, star-structured networks, and random networks are the most affected in terms of security. While the first ones maintain communication links between each pair of nodes, the others establish links with either a central node or random nodes that could be malicious or dishonest.

Regarding communications between nodes, malware also has a significant impact due to the high number of exchanges in DFL. The exchanged messages are obtained by neighboring participants, extracted, and processed for aggregation or forwarding in the case of proxy participants. However, the information obtained may have been modified during the exchange, adding malicious data using rootkits or including the participant in a botnet [75], [76]. In this way, the node would be infected and could spread easily and quickly among the federation participants, especially when there are many links between neighbors.

2) Adversarial Attacks: DFL must address adversarial attacks caused by malicious nodes participating in the federation. These attacks, inherited from traditional CFL operations, include poisoning attacks on HFL and VFL approaches [49]. Furthermore, DFL is prone to poisoning attacks for the following reasons: (i) DFL scenarios involve numerous participants, increasing the likelihood of faulty behavior from one or more nodes; (ii) local training data and training processes of participants are hidden from other nodes in the network, making it impossible to verify the authenticity of updates sent by participants; and (iii) multiple participants may generate vastly different local updates. Consequently, secure aggregation protocols might be negatively affected by aggregation performed by malicious nodes, rendering it infeasible to audit local updates [56]. In this sense, the paper at hand analyzes such attacks on DFL solutions to assess their impact on the network.

Adversarial attacks can be classified based on the timing of the attack (training or evaluation time) and frequency (one-shot or multiple) [124]. Among training-time attacks, Byzantine attacks are prevalent. In these attacks, malicious nodes disrupt model training by transmitting poisoned, corrupted, or fake model updates to other nodes in the network. Since malicious nodes in DFL keep their data and aggregations private, they can poison their model and distribute their parameters without consequences. These attacks can be executed in a one-shot manner each round or multiple times in asynchronous communication scenarios. Similarly, Sybil attacks create multiple fake pseudonym nodes that flood DFL networks. It is difficult to mitigate these threats since no central authority can verify node identities and distinguish between pseudonym and non-pseudonym nodes. Using pseudonyms, an attacker can launch DDoS attacks and disrupt the DFL network. The attacker can command multiple pseudonyms to flood a targeted node with pairing requests, rendering it unable to respond to and communicate with legitimate nodes [80]. Additionally, an adversary can corrupt the reputation system of decentralized applications. Nodes could obtain a false high reputation, being forced to participate in the network and being queried by other nodes, thus degrading their performance.

Other attacks attempt to infer DFL communications with a high attack frequency. One of these threats is Eclipse attacks, which enable attackers to gain partial or full control of a participant by manipulating its connections with neighboring nodes [78]. In such attacks, the attacker nodes control many connections, which allows them to "eclipse" the victim, rendering messages that should reach it useless. The attackers could gain complete control of all traffic to the victim if all node connections belong to them [79]. Free Rider attacks are also a type of attack that takes advantage of DFL without contributing useful data to it during the training process [125]. Finally, in evasion attacks, malicious nodes mimic participation in DFL by generating misleading updates, crafting inputs that manipulate the model into producing faulty collaborative models.

3) Security Countermeasures: The use of countermeasures is essential in the DFL approach to prevent malicious participation and preserve data privacy. DFL establishes a federation by sharing model parameters, such as gradients, weights, or meta-level information among participants. However, exchanging gradients or meta-level information during distributed training can reveal information about the training data and model updates, making exchanging model weights preferable for privacy preservation [81].

Additional countermeasures comprise several techniques inherited from CFL architectures, as reported in [12]. Most of them are employed during decentralized federation to obfuscate updates at the cost of reducing the accuracy of each local model, relying on traditional Differential Privacy (DP) relaxations to inject less noise. However, this technique may degrade the performance of constrained participants, leading to suboptimal results in the federation. In DFL, each participant needs to perform a significant computation to add or suppress noise per set of model parameters transmitted on each communication link, making it more challenging to adapt than CFL [82]. Other techniques, such as data augmentation and obfuscation, can prevent sample reconstruction in local training. However, model parameters shared among neighbors would be equally exposed to malicious nodes. A promising solution is to combine DP mechanisms with secure aggregation and additive Homomorphic Encryption (HE), depending on the application scenario and required security level, and deploy them on a larger number of federation participants. In other cases, it is preferable to rely on anomaly detection mechanisms [126] or model misbehavior detection [90] to detect malicious actors that may deteriorate DFL performance. While the former focuses on the data supplied to the model, the latter analyzes the internal performance of the model depending on the task. In these cases, detection could involve the deployment of mitigation to reduce the severity of a cyberattack. Recent research has further augmented defensive capabilities by integrating Moving Target Defense (MTD) techniques. Specifically, random neighbor selection and dynamic IP/port switching offer enhanced resilience against communication-based threats [83]. Moreover, anomaly detection has significantly improved by incorporating time series analysis techniques [84].

Finally, cryptographic methods, such as Secure Multiparty Computation (SMC), are widely used in DFL to preserve the privacy of exchanges between neighboring nodes [45]. Federation participants must encrypt their messages before sending them, operate on them, and decrypt the encrypted output to obtain the final result. Despite the security provided by SMC, nodes can experience a significant computational overhead, similar to previous DP techniques. To address this issue, new approaches have emerged to provide adequate security performance while maintaining efficient resource usage. One such approach is based on deploying a Trusted Execution Environment (TEE) [77], such as Intel SGX processors, which can protect the code and data loaded inside it. Each node in the network can use this environment to increase its trustworthiness while allowing model parameters to be securely added during federation. In addition to cryptographic mechanisms and secure environments, DFL also requires mechanisms that guarantee the decentralization of the architecture while ensuring inter-participant reliability. DLT technologies, such as Blockchain, provide an ideal solution for this requirement by ensuring the immutability of exchanged models [68]. It is accomplished through a P2P consensus scheme governing the network, where the model parameters of each participant can only be updated through consensus. In this sense, a novel solution uses a lightweight Blockchain with a unique consensus protocol and storage strategy, significantly reducing overhead while guaranteeing privacy. This approach optimizes performance by mitigating slower device impacts and maximizing storage utilization [127]. At the same time, a transparent and immutable reward system ensures the trust of the participants by detecting malicious activities.

E. Key Performance Indicators

KPIs are necessary to measure the efficiency and suitability of the activities and components of DFL architectures. It is essential to know these metrics to detect and resolve federation, communications, or security issues more quickly, reducing the frequency and duration of incidents while increasing efficiency [86]. In this sense, KPIs are utilized for the proper evaluation of the main components of the DFL approach: (i) the federation nodes, detailing the evaluation in terms of mobility and resource capabilities; (ii) federation communications, defining the flexibility and overhead of the deployed architecture; and (iii) the federation models, reviewing their performance and trustworthiness in solving distributed ML tasks (see Fig. 7).

1) Federation Nodes KPIs: DFL must have KPIs to evaluate the participating nodes of the federation. Thus, the heterogeneity and dynamism of the nodes in DFL scenarios are relevant factors adding complexity to the network. In this sense, *resource capabilities* and *node mobility* are identified as promising KPIs to determine node performance. The *resource capabilities* KPI comprises computational and network capacity.

- *Computing capacity*. It provides internal participant resource indicators for local training computation and model parameter aggregation during federation. Some metrics related to this category are CPU/GPU, disk, or RAM usage. One of the aspects of computational capacity is the autonomy of the node to address the tasks. In this sense, time, cycles, and temperature are considered [87].
- *Network capacity*. It determines the node ability to manage communications with one or more neighbors. In this regard, the following metrics are important to analyze: the number of bytes transmitted and received, the number of packets transmitted/received, the bandwidth, or the number of listening/established sockets [7].

In addition to *resource capabilities*, DFL presents the *node mobility* KPI to evaluate the dynamism of the network. Therefore, nodes can be easily affected in communications with other nodes, generating intermittent inputs and outputs in the network. The *node mobility* KPI is determined by node traceability in both the network and local.

- *Network traceability*. It monitors the state of each node and traces its mobility in the network. For this purpose, this metric computes the edges of nearby participating nodes to determine their link with other participants in the federation, which denote the number of near neighbors. In this way, the network can extract information about the position, direction, and absolute velocity of each node [13].
- *Local traceability*. It allows determining the DFL network to contribute by monitoring local interfaces and their characteristics. Thus, it is possible to determine the mobility intent. Metrics such as the number of open ports or network usage during federation are identified [128]. Network traceability metrics usually complement them in the evaluation of node performance.

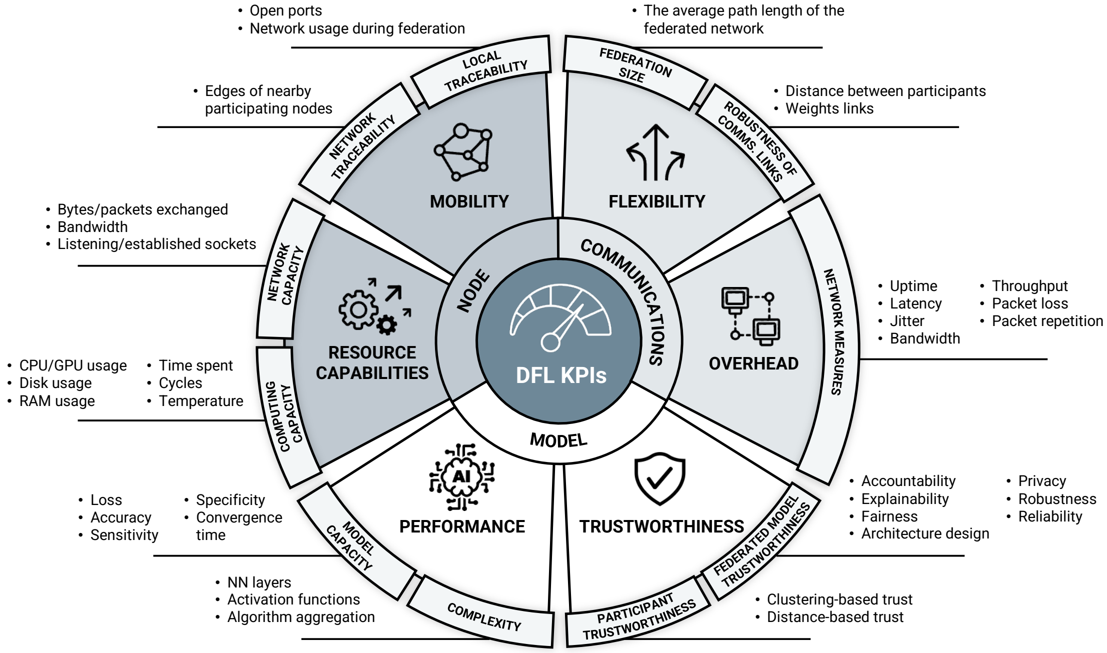


Fig. 7: Key Performance Indicators for DFL.

2) Federation Communications KPIs: Model parameter exchanges between nodes in a DFL approach might experience instability due to asynchronous communications, leading to nodes processing and exchanging data at varying rates. Consequently, this could result in delays and inconsistencies in the learning process, as some nodes may handle outdated information. Furthermore, the aggregation process is complicated by heterogeneous aggregator devices with diverse computational capabilities and communication protocols. Such heterogeneity can cause inconsistencies across different network segments, ultimately having a negative impact on the accuracy of the models for a group of participants. Moreover, asymmetric topologies may cause uneven data distributions or processing capabilities among nodes, posing challenges in attaining a uniformly distributed learning process. In this sense, *communications flexibility* and *network overhead* are essential for optimal balance. Regarding the *communications flexibility*, the DFL approach is evaluated according to the following elements.

- *Federation size*. It provides an evaluation of the predisposition of the architecture to exchange model parameters between nodes. This metric accurately calculates the average path length of the network, representing a stable number strongly affected by the movement of the nodes and, thus, the flexibility of the network. Reduced network size improves parameter transmission performance by reducing interference [10].
- *Robustness of communication links*. It determines the strength of the links with the other neighbors participating in the federation. Some metrics that allow its evaluation are the distance between nodes, reactance, and reliability. Link weights are often used to support evaluations, with high weights being maintained for links with a high percentage of correctly received transmissions [89].

In the context of DFL, several scenarios require decentralized technologies, which can face significant performance issues. One of the main challenges arises from the communication-intensive nature of DFL applications, which require frequent model parameter transmissions involving large amounts of data. As the federation progresses or the model becomes more complex, this communication becomes even more intensive, leading to high *network overhead*, where a high overhead can lead to applications failing to meet response time, quality, or resource utilization requirements. *Network overhead* is evaluated from the following metrics [7].

- *Uptime*, often known as availability, reflects the time of use of the federated network since it became available to participants.
- *Latency* measures the time a data packet travels between network nodes.
- *Jitter* indicates the variability of a latency time and measures the consistency of a network transfer rate. Very low jitter is preferred when a real-time reaction is required (e.g., warning of enemy UAVs).
- *Bandwidth and throughput*. While the first indicates the quantity of data a network path is anticipated to support or successfully convey from one point to another at a specific time, the second describes the amount of data transported between nodes inside a network path.
- *Packet loss* can denote congestion, low bandwidth, and interference. It refers to packets transferred from one device to another but fail to arrive at their destination. Packet loss is the ratio of packets received at the destination to packets sent from the source.
- *Packet repetition*. When a data packet is successfully sent but does not reach its destination, it must be retransmitted. Retransmissions occur in all networks; however, they are more common in wireless networks due to low signal

strength, concealed nodes, and interference.

Both *communications flexibility* and *network overhead* are mainly limited to the model parameters exchanges during federation for each K step. Assuming M is the parameter size of the model, there would be $N-1$ times communications in a round, being the average overhead $M \cdot (N-1)$. Thus, K could reduce the communication frequency, decreasing the performance of DFL. A summary of the communication complexity comparison with CFL architectures is shown in Table III, where [ ] denotes the standard rounding function. The table shows that the total volume of data transferred per DFL round is comparable across all three architectures. However, the communication overhead per node is different among decentralized architectures. Specifically, SDFL has the potential to achieve better performance in node communication overhead, which could benefit system bandwidth utilization and increase robustness. Overall, it is important to consider the trade-off between communication overhead and other features when designing decentralized architectures with FL.

Table III: Comparison of the flexibility of federated communications between DFL, SDFL, and CFL.

Federated Architecture	Communication Times/Round	Node Overhead	Total Transferred Data Volume per Round (MB)
DFL	1	$M \cdot (N-1)$	$M \cdot (N-1)^2$
SDFL	$\left[\frac{N-1}{2}\right]$	$2 \cdot M$	$2 \cdot M \cdot N \left[\frac{N-1}{2}\right]$ *
CFL	1	$M \cdot N$	$M \cdot N$

* It considers the additional communication required for leadership transmission, which is not present in the DFL and CFL architectures.

3) Federation Models KPIs: The *performance* of the federated model in solving the task for which it has been created is critical. In other words, the performance indicates the success of DFL in generating accurate predictions or solving the intended problem. The following elements are used for measurement [90].

- *Model capacity*. It is based on comparing the model predictions with known values of the dependent variable in a dataset. Some relevant metrics are model loss, accuracy, sensitivity, specificity, and the time associated with the convergence of the model.
- *Complexity*. It is influenced by both the functions of the model trying to learn and the nature of the training data. Some metrics are the number of layers in the NN, activation functions, and algorithm aggregation implemented by the federated model.

In addition to the above elements, federation models must be *trustworthy* to ensure the technology is used to contribute productively to the task. If they are not trustworthy, nodes will not collaborate with these systems. The problem increases when model parameters are deliberately distributed among nodes in a topology. Therefore, it is necessary to determine the confidence of the participants and the federated models to ensure trustworthiness, where robust privacy and security measures significantly contribute to this [92], [93].

- *Participant trustworthiness*. It defines the confidence level of neighboring participants based on the model parameters exchanged. There are two main metrics, clustering-based trust, which determines the federation nodes trustworthiness relying on clusters of participating nodes; and distance-based trust, which is determined based on the distance between participants and the transmission time of the model parameters.
- *Federated model trustworthiness*. It provides confidence in the local models of each participant in the federation. Normally it is calculated in the aggregator nodes before supplying the federated model to the rest of the participants. The metrics are divided into several pillars: accountability, analyzing the quality of the model life cycle (client registry, anomaly monitoring); explainability, providing clarification of the decision rationale (algorithm class, model size, DFL configuration); fairness, ensuring impartial and just decisions without discrimination of protected groups (participation rate, discrimination index); architecture design, creating a baseline behavior for the management of participants in the federation (data augmentation, regularized local loss); privacy, protecting federated model data (perturbation, anonymization, entropy); robustness, checking the resilience against adversarial inputs (confidence score, loss sensitivity); and reliability, predicting when an asset will fail or deteriorate (client dropout rate, data provenance, data quality).
- *Security and privacy*. The preservation of security and privacy is paramount in DFL, even while allowing effective model training [95]. DP is a widely used method that includes a privacy budget parameter, denoted as epsilon (ϵ), serving as a metric to quantify the maximum amount of privacy loss allowed. This budget diminishes with each data-sharing instance, with smaller budgets enhancing privacy but potentially impeding data utility. Another critical parameter in DP is delta (δ), which represents the probability of information inadvertently being leaked. Other privacy-preserving methods, such as HE and anonymization, also have vital performance indicators. In HE, the degree of noise introduced can be a significant parameter. Anonymization techniques, such as k-anonymity and l-diversity, use k and l values as performance metrics, ensuring privacy in DFL by making it difficult to re-identify models within a federation [96].

F. DFL Optimizations

Despite the advantages of DFL, a series of optimization mechanisms are necessary to guarantee the full efficiency of the approach. In this sense, DFL includes three main elements to optimize: federation nodes, federation communications, and federation models (see Fig. 8).

1) Optimization of Federation Nodes: An important optimization for improving node *resources capabilities* and *mobility* in the federation is the selection of nodes to transmit the model parameters. Node selection is divided into three different strategies.

- *Sequential selection of nodes*. Participants are selected at the beginning of the federation process, which iterates over time [93].

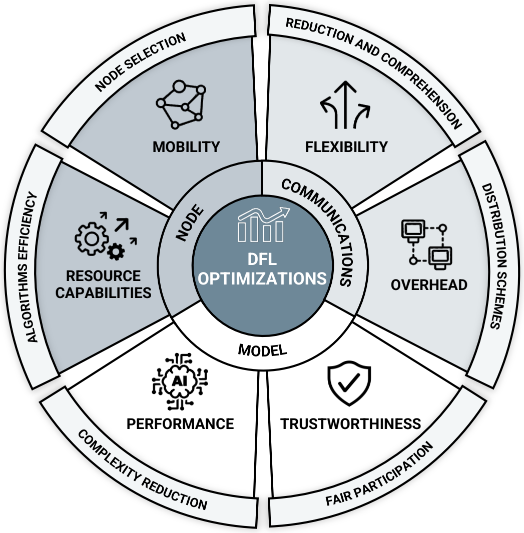


Fig. 8: DFL optimizations.

- *Random selection of nodes*. It excludes resource capabilities or network constraints from the participants. In this sense, a subset of nodes is obtained in a fixed time of the federated process to send the model parameters.
- *Scheme schedule selection of nodes*. This selection strategy requires a node to act as an orchestrator of the communication links and the characteristics of each neighboring node. Furthermore, the controller can select participants based on a preassigned probability calculated according to participation in the federation [4].

Additionally, it is relevant to optimize traditional aggregation algorithms such as Federated Averaging (FedAvg) to make it more suitable for the node resource capabilities [4]. FedAvg is a widely accepted heuristic algorithm used as a baseline for star-structured topologies due to its simplicity and empirical effectiveness. In this case, the aggregator node obtains the model parameters of the neighboring nodes, aggregating the average of the updates and computing the resulting global model. It is also worth noting that while in CFL, the FedAvg algorithm has been widely accepted as a baseline, in DFL, no algorithm has risen among the others. Customizations of FedAvg and new techniques focused on decentralized scenarios have emerged in the literature intending to adapt to DFL solutions.

- *Decentralized Stochastic Gradient Descent (DSGD)*. It provides optimization for a set of objective functions distributed over a network. Each node maintains estimates of the optimal optimization step on information concerning its cost function and exchanges these estimates directly with the other nodes in the network [129].
- *FedPGA*. It defines a partial gradient exchange mechanism that leverages node-to-node bandwidth to speed up communication time. At the same time, it reduces the convergence rate by adapting the step size of the stable gradient descent direction [97].
- *Dynamic Average Consensus-based FL (DACFL)*. It transforms the aggregation of the DFL model into a local training procedure as a discrete time series, whose convergence guarantees convex objectives [130]. In the algorithm, nodes compute and update a local solution to a subproblem and share the solution with neighboring nodes. Moreover, the authors demonstrate that the algorithm adapts to dynamic networks and heterogeneous scenarios by tuning a node-specific local parameter based on the node resources [5].
- *Split Learning (SL)*. This method, called SplitNN [99], enables collaborative training of NNs without sharing raw private data, but emphasizes the suitability for DFL architectures. It employs split models instead of full model replication in a DFL architecture. The training participants hold replications of the shallower layers up to a certain layer (i.e., the cut layer), while a specific node holds the deeper layers.
- *Decentralized FedAvg with Momentum (DFedAvgM)*. It uses SGD with momentum to train the local models of the federation participants, communicating only with their neighbors in an undirected graph. It provides a better convergence rate and communication efficiency than DSGD [131].
- *DeceFL*. It focuses on neighbor communications to improve efficiency for both convex and non-convex loss functions. It applies to a wide range of real-world medical and industrial applications [132].

2) Optimization of Federation Communications: Optimizing network flexibility and overhead is crucial for addressing performance issues caused by continuous data exchanges between participants. Concerning *network flexibility*, it is necessary to have techniques that reduce the number of exchanges in the network without losing functionality during federation [133]. Several studies have proposed reducing the number of bits transferred for each worker update through data compression. However, these studies do not consider the impact of data loss from compression results, which could impair the learning accuracy and affect convergence [103].

In the literature, different distributed optimization schemes aim to maintain acceptable convergence rates in terms of recurring iterations and device computation time. Some examples are Decentralized Gradient Descent (DGD) [104], decentralized Alternating Direction Method of Multipliers (ADMM) [105], EXTRA [106], and ADMM based on Jacobi-Proximal [107]. A recent contribution includes a method that applies the Lloyd-Max algorithm to DFL to minimize quantization distortion, with the resulting LM-DFL algorithm able to adjust quantization levels adaptively [102]. Other communication optimization mechanisms use an incremental learning method to reduce costs by activating and linking agents while keeping other nodes and links inactive. These include Random Walk ADMM (WADMM) [134], Parallel Random Walk ADMM (PW-ADMM) [123] and Walk Proximal Gradient (WPG) [135] which are commonly used in increment-based approaches. Recent research has introduced mechanisms that aim to optimize the balance between improving the quality of the model and saving communication resources, which can result in a more sustainable federation policy [100]. These mechanisms typically offer an independent selection of participants and fragments of the NN to be transmitted, providing a promising alternative to traditional optimizations. One such technique is FL-EOCD, leveraging D2D communications and overlapped clustering to enable decentralized aggregation, thereby reducing overall energy consumption and latency while maintaining convergence rates [58]. Further approaches, such as [136], implement latency optimization strategies by incorporating Blockchain. The latency optimization is achieved by strategically managing data offloading decisions, power transmission, bandwidth allocation, and computational resources, resulting in a highly responsive system.

Regarding optimizing *network overhead*, the DFL approach presents limitations in communication bandwidth between nodes [71]. To address this issue, researchers have explored different distribution schemes, such as quantization and sparsification. For example, Quantized Sparse SGD (QSGD) [109] and quantized ADMM [137] were proposed to reduce network overhead when exchanging model parameters. Furthermore, the Qsparse-local-SGD algorithm [110] includes a combination of aggressive sparsification with quantization and local computation along with error compensation. However, these methods may decrease accuracy to achieve lower communication costs [74]. With adequate communication infrastructure, decentralized learning can achieve accuracies like its centralized counterpart. In this sense, the use of advanced technologies such as 5G [138] or 6G [139] can help reduce communication limitations, improving network bandwidth and enable faster and more efficient data exchange. Other studies, such as [140], have shown that decentralized SGD can outperform centralized approaches in low-bandwidth networks. However, these studies do not consider typical federated settings, such as non-IID data among participants, limiting the applicability of their results.

3) Optimization of Federation Models: It is necessary to optimize federated models to avoid high processing demand on the most constrained participants. In this sense, optimizations are responsible for increasing the efficiency of the models without sacrificing performance and trustworthiness.

In DFL, nodes often aim to train a state-of-the-art ML model on a specified task. To increase *model performance* and *trustworthiness*, researchers have developed new models and improved existing solutions. The most popular ML model is a NN, which achieves state-of-the-art results in many tasks, such as image classification. Many DFL studies are based on Stochastic Gradient Descent (SGD), which can be used to train NNs [110]. Another widely used algorithm is a decision tree, which is more efficient to train and easier to interpret than NNs. In particular, a tree-based DFL is designed for federated training of single or multiple decision trees, such as Gradient Boosting Decision Trees (GBDTs) and Random Forests (RFs). GBDTs have been especially popular recently due to their superior performance in many classification and regression tasks [26]. Besides NNs and trees, linear models such as Linear Regression (LR), Logistic Regression, or Support Vector Machine (SVM) are classic and easy-to-use models. To enhance the consistency of models generated by each federation participant, algorithms such as DFedSAM utilize gradient perturbation to produce local flat models with uniformly low loss values [111]. In non-IID scenarios where adaptive training models can be challenging on resource-constrained devices, these models can be particularly useful [108]. Furthermore, techniques such as SwarmSGD jointly leverage non-blocking communication, quantization, and local steps, allowing for better compensation by training on more samples but with a lower quantization level of model parameters. This technique is effective in heterogeneous node data distributions and random topologies [88].

In addition to the efficient models mentioned above, new techniques have emerged in recent years to improve the efficiency and performance of DFL. These techniques include meta-learning, multitasking, and federated distillation. Meta-learning aims to improve the efficiency and performance of the model generated at each node by creating less complex models capable of performing tasks in parallel. It uses information from the activities performed to better adapt to new tasks [112]. As a result, meta-learners generalize better when trained with a more extensive set of practical tasks and more data for each task. In this sense, the learner can acquire knowledge and processing capacity distributed among agents, adapting its model to the best-performing ones in the network. Meta-learning often involves multitasking when ML models are trained to perform multiple tasks simultaneously. These models optimize metrics for each task and generate a corresponding output [91]. Finally, federated distillation, first introduced in 2015 as knowledge distillation, aims to optimize federated models by performing model compression. Specifically, it is a procedure to train a model using a second NN as a reference instead of learning directly from the ground truth of the data [141]. In this way, it allows several advantages over more complex models: (i) more information can be extracted from a single sample, (ii) training can be performed with fewer examples, and (iii) labeled data are not needed. All this offers less computation time and energy consumption in the federated node [113]. A decentralized version of this approach has been proposed in [114], where instead of having a pre-trained model to imitate it, each model considers the aggregated knowledge of the remaining participants' models.

Finally, the most relevant optimization of federated models for increasing *trustworthiness* is using DLT techniques, providing reliability during model parameter updates in the federation. They also facilitate the fair participation of nodes during the federation. In such trust-based relationships, there are two entities: the trustor and trustee node. For each observation from the trustee, the trustor determines whether it is an outlier, while the trustee node validates the contributions provided by the trustor. To maintain the reputation of the participating devices and ensure that all participants make active and honest contributions to the model, the trustor node i can generate a smart contract with the parameters of the local model, which is validated by a set of trustee nodes

j [142]. DLT techniques also provide an added explanation and robustness to federated models. The immutability of the initialization and update of federated models is guaranteed by DLT, maintaining a chronological log in the Blockchain. Moreover, the fair participation of nodes in the federation is further ensured by DLT.

IV. Open-Source Frameworks for DFL

Building upon the storyline presented in Fig. 2, this section responds to "*Q2. What DFL frameworks exist, and what fundamentals do they provide?*" In this context, frameworks serve as the bridge between the "Theoretical Overview" with its fundamentals and "Practical Application" with its application scenarios. This section introduces the most relevant existing open-source frameworks for DFL. First, it discusses the most mature and highly customized frameworks supported by large companies. Then, it focuses on those more incipient with limited functionality but notable research advances. Table IV compares the main aspects of the frameworks analyzed. The double horizontal line separates the mature frameworks from the incipient ones.

A. Mature Frameworks

TensorFlow Federated (TFF) [143], developed by Google, provides the building blocks for FL based on TensorFlow. It includes three key components: models, federated computation builders, and datasets. TFF can be deployed on multiple machines, creating a rotating aggregator. Currently, TFF does not consider any adversaries during FL training nor provides privacy mechanisms. Other frameworks, such as PySyft [144], provide interfaces for developers to implement their training algorithm. While TFF is based on TensorFlow, PySyft can work well with PyTorch and TensorFlow. This framework provides multiple optional privacy mechanisms, including secure multi-party computation and differential privacy. Moreover, it can be deployed on a single machine or multiple machines, where the communication between nodes is performed using the WebSocket API. For privacy preservation, SecureBoost [56] builds multi-part reinforcement trees with an encryption strategy between different nodes. This solution allows for different feature sets corresponding to a vertically partitioned dataset. One of the advantages of SecureBoost is that it provides remarkable accuracy across different topology architectures without revealing information about each private data provider.

Other frameworks, such as FederatedScope [145] and FedML [146], allow lower-level management of the deployed architecture. The first one employs an event-driven architecture to provide great flexibility in describing node behaviors. It allows learning objectives and backends to be added and coordinated in an FL life cycle with synchronous or asynchronous training strategies. In addition, it includes privacy protection, attack simulation, and automatic tuning. In addition, FedML offers a platform for FL benchmarking based on PyTorch and utilizes the Message Passing Interface (MPI) protocol, gRPC, or MQTT for communication. This framework comprises two main modules: FedML-core, which implements the training engine and the distributed communication infrastructures, and FedML-API, which is built on top of FedML-core and provides training models, datasets, and FL algorithms. Finally, FedML supports three computing paradigms: standalone simulation, distributed computing, and on-device training. Along the same lines, LEAF [147] provides only benchmark functionality of the deployed architectures. In this sense, it uses distributed datasets and partitioning mechanisms that other frameworks can leverage to complement its functionality.

B. Incipient Solutions

BrainTorrent [148] is an FL framework without a central server explicitly aimed at medical applications. With no need for a central authority to oversee the training process, BrainTorrent provides highly dynamic P2P communications where each medical center communicates directly, also offering robust training of participants through asynchronous updates achieving performance similar to on-device model training. Although the framework focuses on image segmentation, it can be extended to other data and models. Similarly, Scatterbrained [43] facilitates the construction of both CFL and DFL systems. In the latter, it can create a customized communication system based on ZeroMQ, corresponding to a communications library based on UDP. Nodes can have a specific role: (i) leech, in which nodes listen for broadcasts but do not share data; (ii) offline, where nodes do not listen or broadcast data; (iii) peer, where participants listen and broadcast data; and (iv) seeding, where nodes broadcast but do not listen to incoming data. In addition, it allows the use of different ML frameworks such as TensorFlow, Scikit-learn, or PyTorch.

Other studies present DFL models based on the InterPlanetary File System (IPFS) [149]. In this sense, the authors deployed a framework called IPLS, whose nodes are interconnected and perform federated model training based on the operation of one or more network participants. Any participant can initiate the federated training process or join an existing one. This solution shows robustness in the face of dynamic nodes and intermittent connectivity, ensuring limited resource usage and convergence of the federated model. Similarly, TrustFed [142] uses IPFS and Blockchain smart contracts to maintain the reputation of participants in a centralized or decentralized network. In this way, compelling participants to make active and honest contributions to the federated model. In contrast, other frameworks such as FLoBC [150] deploy a Blockchain mechanism to ensure the decentralization of nodes. In this sense, the framework compares and contrasts the effects of the trainer-validator relationship, reward-penalty policy, and model synchronization schemes on overall system performance. The authors demonstrate that using this framework enables the deployment of DFL systems with performance similar to more centralized architectures. Similarly, BLADE-FL [151] uses Blockchain to mitigate vulnerabilities during federation. It offers robustness against malfunctions, distrustful connections, and external threats by having each client compete to generate a block

Table IV: Comparison of mature and incipient DFL frameworks.

Reference	OS	Federation Architecture		Aggregation Algorithms	Communication Protocol	Security and Privacy	Data Type	Scenario	Benchmarking
		Participant Type	Aggregator Node						
TFF [143]	Linux MacOS	Cross-silo	Centralized Decentralized	Median FedAvg FedProx	gRPC	✓	Time series Images	Simulation	✓
PySyft [144]	Windows Linux MacOS Mobile	Cross-silo Cross-device	Centralized Decentralized	FedAvg	Websockets	✓	Images	Simulation Real	✓
SecureBoost [56]	Linux MacOS	Cross-silo	Centralized Decentralized	FedAvg GBDT	gRPC	✓	Time series	Simulation	✗
FederatedScope [145]	Windows Linux MacOS Mobile	Cross-silo Cross-device	Centralized	FedAvg FedOpt	gRPC	✓	Time series Images	Simulation Real	✓
FedML [146]	Linux MacOS	Cross-silo	Centralized Decentralized	FedAvg FedOpt FedNova	gRPC MPI MQTT	✓	Time series Images	Simulation Real	✓
LEAF [147]	Linux MacOS	Cross-silo	Centralized Decentralized	-	-	-	Time series Images	Simulation	✓
BrainTorrent [148]	Windows Linux MacOS	Cross-device	Decentralized	FedAvg	*N/S*	✗	Time series	Simulation Real	✗
Scatterbrained [43]	Windows Linux MacOS	Cross-device	Centralized Decentralized	FedAvg	ZeroMQ	✗	Time series Images	Simulation	✗
IPLS [149]	Windows Linux MacOS	Cross-device	Decentralized	FedAvg	P2P	✓	Time series Images	Simulation Real	✓
TrustFed [142]	Windows Linux MacOS	Cross-device	Centralized Decentralized	FedAvg	P2P	✓	Time series Images	Simulation	✗
FLoBC [150]	Windows Linux MacOS	Cross-device	Decentralized	FedAvg	HTTP (REST API)	✗	Time series Images	Simulation Real	✓
BLADE-FL [151]	Windows Linux	Cross-device	Decentralized	Custom algorithm	P2P	✓	Images	Simulation	✓
DISCO [15]	Windows Linux MacOS Mobile	Cross-device	Centralized Decentralized	Custom FedAvg	peer.js	✓	Time series Images	Simulation	✗
CMFL [152]	Windows Linux MacOS	Cross-device	Decentralized	Median Trimmed Mean Krum Multi-Krum	P2P	✓	Time series Images	Simulation	✓
DeFL [40]	Windows Linux MacOS	Cross-device	Decentralized	Custom algorithm	*N/S*	✗	Time series Images	Simulation Real	✓
FL-SEC [12]	Windows Linux MacOS	Cross-device	Decentralized	FedAvg	*N/S*	✓	Time series	Simulation	✓
DisPFL [38]	Windows Linux MacOS	Cross-device	Decentralized	FedAvg Ditto FOMO Sub-FedAvg	P2P	✗	Time series Images	Simulation Real	✗
GossipFL [70]	Windows Linux MacOS	Cross-device	Decentralized	FedAvg S-FedAvg D-PSGD CHOCO-SGD	P2P	✗	Images	Simulation	✓
Fedstellar [153]	Windows Linux MacOS	Cross-silo Cross-device	Centralized Decentralized Semi-Decentralized	FedAvg Krum TrimmedMean Median	P2P HTTP (REST API)	✓	Time series Images	Simulation Real	✓

N/S (Not Specified) by the authors. The double horizontal line indicates the separation between mature and incipient frameworks

before local training in the next federation round. The system also addresses training deficiency caused by lazy clients and optimizes resource allocation.

In addition to the above solutions, some frameworks explore new applicability technologies to generate DFL scenarios. DISCO [15] is a framework presented by the EPFL Machine Learning and Optimization Laboratory that creates an easy-to-use platform that allows non-specialists to participate in collaborative learning. This platform is based on Javascript, supporting arbitrary DL tasks and model architectures via TF.js and relying on P2P communications. Additionally, DISCO provides mechanisms to ensure communication efficiency, privacy preservation, fault tolerance, and deployment customization. Moreover, frameworks seek to improve the robustness of their privacy techniques. In this case, [152] presented a serverless framework named Committee Mechanism-based Federated Learning (CMFL), orchestrated by a committee system deploying selection strategies of model parameters during federation. In contrast to the other frameworks, this one implements several aggregation algorithms such as median, trimmed mean, Krum, and Multi-Krum, evaluating them in terms of robustness and efficiency. Regarding the security and privacy of the scenarios, [12] and [15] provide frameworks capable of ensuring the integrity and confidentiality of the communications and data exchanged. Both alternatives present custom aggregation algorithms to process image and time series data.

Finally, other frameworks in the literature allow deploying cross-silo-based solutions, creating robust topologies of nodes that exchange information using P2P networks. In this case, [40] removed the central server by aggregating the weights at each participating node, and the weights of only the current training round are maintained and synchronized among all nodes. This framework has been evaluated on two widely adopted public datasets, CIFAR-10 [154], and Sentiment140 [155], demonstrating convergence against more common threats with minimal accuracy loss, achieving up to 100x reduction in storage overhead and up to 12x reduction in network overhead compared to state-of-the-art DFL approaches. Similar to the previous solution, the authors of [38] allow further savings in communication and computational costs. In this sense, the authors proposed a decentralized sparse training technique, where each local model only maintains a fixed number of active parameters throughout the local training and P2P communication process. They employ aggregation algorithms such as Ditto, FOMO, and Sub-FedAvg to enhance the framework efficiency. Furthermore, the authors demonstrated that the framework allows easy adaptation to heterogeneous local clients with different computational complexities. Similarly, GossipFL [70] presents a novel approach to address communication challenges in IoT scenarios characterized by intermittent connections between participants. The framework utilizes sparsification techniques and a unique gossip matrix generation method, employing algorithms such as S-FedAvg, D-PSGD, and CHOCO-SGD to optimize communication traffic while preserving convergence. The efficacy is demonstrated across various datasets, such as MNIST and CIFAR10, achieving model accuracies of up to 98%. Building on these advancements, Fedstellar [153] offers a versatile platform for DFL training that addresses common limitations in existing frameworks. It efficiently manages heterogeneous topologies and adapts the federation to physical or virtual deployments. In a real-world cyberattack detection employing single-board devices and a fully connected network, Fedstellar obtained a 91% F1 score. Furthermore, in a virtualized scenario using the MNIST dataset, the platform delivered a compelling F1 score of 98% with DFL and 97.3% with SDFL while reducing model convergence training time by 32% relative to CFL.

V. Application scenarios

Following the storyline presented in Fig. 2, this section responds to "*Q3. Which are the main characteristics of the most relevant scenarios of DFL?*" In particular, this section provides a review of existing works classified according to the application scenario: (i) healthcare, (ii) Industry 4.0, (iii) mobile services, (iv) military, and (v) vehicles. These scenarios combine most of the literature work, thus giving a great overview of current DFL applicability. For each scenario, the following aspects are highlighted: the goal, the participants during federation, the ML algorithm used, and the aggregation method employed.

Beyond the previously discussed scenarios, DFL is also relevant in emerging areas such as smart cities and metaverse, where it enhances urban planning and service provision, all while facilitating privacy-preserving data sharing between real and virtual entities [156]. In the realm of swarm robotics, DFL enables asynchronous knowledge sharing and task coordination [157]. In the energy and utilities sector, DFL supports predictive maintenance and demand forecasting, which contributes to operational efficiency across companies [158]. Finally, the telecommunications sector benefits from DFL in optimizing network performance and anomaly detection, allowing network providers to share insights without compromising proprietary information [159]. However, the challenges of these domains need to be further explored to determine the scope of DFL.

A. Healthcare

Cooperation between research institutes, hospitals, and federal agencies is essential in modern healthcare systems to improve healthcare standards. However, transferring patient information between these entities can be challenging due to regulations like the General Data Protection Regulation (GDPR) or European Health Data Space (EHDS). Through DFL, each hospital or institute only needs to share the parameters of the local model to obtain an accurate diagnosis model, thereby preserving patients' privacy. Notably, the successful application of DFL-based approaches in the healthcare sector is evident from recent studies, as demonstrated in the relevant articles presented in Table V.

The first study to present the privacy issues of server use in healthcare is [160]. The authors noted the challenges in achieving a trade-off between global learning of the model and local information from each data source due to the large number of data sources with varying amounts and properties of data. To address this, they proposed a new strategy called Federated-Autonomous Deep Learning (FADL), where the first layer of the NN model is trained in a federated way using data from all sources. In contrast, the other layers of the model are trained locally in each data source. Their approach demonstrated an accuracy similar to the centralized analysis and outperformed regular FL for distributed electronic health records.

In the same way as the previous work, Huang *et al.* [128] proposed a community-based FL algorithm to predict mortality and hospital stay time. Electronic medical records were clustered into communities inside each hospital based on standard medical aspects. Each cluster learned and shared a particular ML model customized for each community rather than a general global model shared among all hospitals. In this work, a participant with an aggregator role was in charge of converting patients' drug features into privacy-preserving representations. The authors of [162] introduced an algorithm to perform local updates during several iterations, enabling communication between nodes and improving communication efficiency for DFL. This approach was tested on simulations with real-world electronic medical records, demonstrating its effectiveness in Alzheimer's disease detection compared to

Table V: Comparison of DFL solutions focused on healthcare.

Ref.	Year	Goal	Federation Architecture	Topology	Scenario	ML Model	Aggregation Algorithm	Local Model Aggregation	Client Selection	Dataset
[160]	2018	Improve the accuracy of the CFL approximation by training parts of the model in a decentralized way and others using specific data sources	Cross-silo	Centralized Decentralized	Simulation	Three-layered NN	Custom algorithm	Synchronous	Sequential	eICU-CRD [161]
[128]	2019	Decrease the cost of communications between hospitals and the aggregator by using a clinical community data clustering algorithm for disease detection	Cross-silo	Centralized Decentralized	Simulation	K-Means	FedAvg	Synchronous	Sequential	eICU-CRD [161]
[162]	2020	Improve the communication efficiency for fully DFL over a graph without loss of optimality of the solutions	Cross-device	Decentralized	Real	Custom model	Custom algorithm	Synchronous Asynchronous	Sequential	Private
[163]	2021	Deep FL framework for decentralized healthcare systems that maintain user privacy	Cross-device	Decentralized	Simulation	Custom model	Custom algorithm	Synchronous	Scheduling	Private
[164]	2021	Facilitate the integration of any medical data from any data owner worldwide without violating privacy laws	Cross-silo	Decentralized	Real	Custom model	Custom aggregator	Asynchronous	Scheduling	PBMC transcriptome [165] Chest X-ray [166]
[167]	2022	Secure valuable hospital biomedical data useful for clinical research organizations, use of Blockchain and smart contracts	Cross-device	Decentralized	Simulation	CNN	FedAVG	Asynchronous	Sequential	Private
[16]	2022	Privacy and protection of medical data exchanged between healthcare entities	Cross-silo	Decentralized	Real	ResNet18 ResNet50 DenseNet121	Custom algorithm	Synchronous	Sequential	ImageNet [168] Private
[98]	2022	Detect brain tumor segmentation, using representative clinical datasets	Cross-device	Centralized Decentralized	Real	U-Net	CFA FedPer	Synchronous Asynchronous	Random	BRATS [169] Private
[90]	2022	Detect misconduct model on specific sites from any learning iteration	Cross-device	Centralized Decentralized	Simulation	GloreChain	FedAvg	Asynchronous	Sequential	Cancer Biomarker [170]
[2]	2022	Collaboratively training between healthcare devices for creating a robust model without raw data exchange	Cross-device	Decentralized	Simulation	3D CNN	Custom FedAvg (RingAVG)	Synchronous Asynchronous	Random	COVID-19 CT [171] Eye Disease [172] Skin Cancer [173]
[21]	2022	Ring topology for DGM to ensure security and privacy in medical scenarios	Cross-device	Decentralized	Simulation	GAN	FedAvg	Asynchronous	Sequential	MNIST [174]
[175]	2022	Exploit concepts from experience repetition and adversarial generation research	Cross-device	Centralized Decentralized	Simulation	GAN	FedAvg FedProx FedBN	Synchronous Asynchronous	Random	X-ray data [176], [177] HAM10000 [178]
[179]	2023	Privacy-preserving decentralized approach while supporting efficient communications	Cross-silo	Decentralized	Simulation	*N/S*	Custom algorithm	Synchronous	Random	MNIST [174]

classical methods. Similarly, Elayan *et al.* [163] proposed a framework to train on skin lesion images using smartphones. They further utilized TFL to overcome the need for large, labeled data. They also proposed an automated training data-acquiring process and evaluated the algorithm on skin diseases, leveraging TL techniques to address the problem of a lack of healthcare data in generating DL models. Moreover, Warnat-Herresthal *et al.* [164] followed a similar approach to ensure the privacy of medical data exchanged between cross-silos.

The advancement of clinical research and new scientific discoveries relies on enabling data mobility, including medical records. In this context, various schemes have been proposed to exchange medical records using Blockchain and smart contracts to ensure appropriate data privacy and protection. Salim and Park [167] compared the results of their decentralized Convolutional Neural Network (CNN) model for Electronic Health Records (EHR) with a private Interplanetary File System (IPFS) and found promising results in accuracy, sensitivity, and specificity, similar to the traditional centralized model. In contrast, Nguyen *et al.* [16] discussed the use of federated distillation to decrease model complexity, using a private dataset called The Noisy Blind consisting of 1198 original images collected from four clinics. Their solution provided superior performance to traditional centralized training when nodes comprise low-quality data, which is common in healthcare. Similarly, the authors of [98] focused on medical image prediction for brain tumor segmentation, enabling the collaboration of multiple institutions by sharing their local computational models and training a U-Net model. This approach reached a target Dice Similarity Coefficient (DSC) above 85% using private data from patients' devices (e.g., heart rate, blood oxygen saturation).

B. Industry 4.0

The use of federated and decentralized techniques has increased considerably in Industry 4.0 to overcome the lack of communication between devices and sensors without needing a central node. As shown in Table VI, several recent studies have shown promising results in applying DFL-based approaches in industrial settings.

In [8], the authors discussed the opportunities and applications of DFL in automated and networked industrial systems supported by Device-to-Device (D2D) communications. The authors demonstrated the integration of the DFL approach into the sensing-decision-action loop, improving knowledge discovery operations. The system evaluation showed a convergence of the collaboration model in cross-device environments with minimal synchronous and asynchronous communications. Following the same direction, the authors of [181] applied cognitive computing in a simulated industrial scenario. This technique models the reasoning process of the human brain, having a progressive welcome in Industry 4.0 automation. This work implemented a DFL scenario supported by Blockchain and a Distributed Approximate Newton (DANE) aggregation method with advanced checks and random node selection. The results of an extensive evaluation and assessment showed accuracy values between 0.74 and 0.82. Ma, Wang, and Li [52] defined Q-DFL for a quantized selection of network nodes. The proposal contained two phases: (i) a local model was trained with the SGD algorithm on each IIoT device, and then the parameters of the quantized model were exchanged among neighboring nodes; and (ii) a consensus mechanism was designed to ensure that the local models converged to the same global model. Subsequent simulation for the MobileNet model revealed its performance trade-off between system

Table VI: Comparison of DFL solutions focused on Industry 4.0.

Ref.	Year	Goal	Federation Architecture	Topology	Scenario	ML Model	Aggregation Algorithm	Local Model Aggregation	Client Selection	Dataset
[8]	2021	Explore emerging FL opportunities using next-generation networked and autonomous industrial systems (e.g., robots, vehicles, drones)	Cross-device	Centralized Decentralized	Real	DNN	FedAvg	Synchronous Asynchronous	Random	MIMO radar [180]
[181]	2021	Decentralized paradigm for big data-based cognitive computing	Cross-device	Decentralized	Simulation	CNN	DANE	Synchronous	Random	CIFAR-10 [154]
[52]	2021	A method for D2D network in industrial IoT devices for improving the convergence of communications	Cross-device	Decentralized	Real	MobileNet	FedAvg (consensus)	Asynchronous	Random	MNIST [174]
[126]	2021	Anomaly detection ML models in non-IID scenarios, including mechanisms to locally rebalance training datasets	Cross-device	Decentralized	Simulation	Four-layered FCNN	P2PK-SMOTE	Synchronous Asynchronous	Random	N-BaIoT [182]
[183]	2022	A method for decentralized anomaly detection using neural networks, offering a comparison with traditional federated architectures	Cross-device	Decentralized	Simulation	Custom model	FedAvg	Asynchronous	Sequential	IoT23 [184]
[185]	2022	A BoEI framework that integrates an innovative DFL for cyberattack detection in IIoT	Cross-device	Centralized Decentralized	Simulation	Custom model	Fed-Trust	Synchronous Asynchronous	Random	TON_IoT [186] LITNET-2020 [187]
[93]	2022	Secure application approach to seal and sign asynchronous and synchronous FL collaborative tasks	Cross-silo Cross-device	Decentralized	Real	Custom model	Custom algorithm	Asynchronous	Sequential	Private
[59]	2022	Generic decentralized FL framework for training distributed models in inherently heterogeneous IoT environments	Cross-device	Decentralized	Real	Custom model	Custom algorithm	Synchronous Asynchronous	Random	Private
[188]	2022	Cooperative machine learning organized by physically close nodes	Cross-device	Decentralized	Simulation	Two-layered FCNN	WAFL	Synchronous Asynchronous	Random	MNIST [174]
[17]	2022	User-defined privacy-preserving framework for industrial metaverses	Cross-silo Cross-device	Decentralized	Simulation	*N/S*	Custom FedAvg	Asynchronous	Scheduling	Private
[189]	2023	A method that leverages Blockchain to improve model quality and reduce latency	Cross-silo	Decentralized	Simulation	Two-layered MLP	FedAvg	Asynchronous	Random	MNIST [174]
[138]	2023	Cooperative use of Blockchain and FL while preserving privacy using 5G	Cross-silo	Decentralized	Simulation	*N/S*	Custom algorithm	Synchronous	Random	CIFAR-10 [154] FEMNIST [147]

N/S (Not Specified) by the authors

information flow consumption, time delay, and energy cost.

Cybersecurity is also an important consideration in Industry 4.0, and several studies have investigated the use of DFL for anomaly and attack detection, such as the work published in [126]. The authors developed anomaly detection ML models for non-IID scenarios and included mechanisms to locally rebalance the training datasets through the synthetic generation of data points from the minority class. Using different metrics to evaluate the model performance on the N-BaIoT dataset, they obtained a convergence close to 75% of False Negative Rate (FNR) and recall. Another study, conducted by Lian and Su [183], discussed a similar approach with the IoT23 dataset [184], obtaining a model convergence 30% faster than in the previous study with an increase of only 3% in the overhead of the participants.

Apart from ensuring the reliability of IIoT and the connectivity of industrial objects, Abdel-Basset, Moustafa, and Hawash [185] proposed a Blockchain-orchestrated Edge Intelligence (BoEI) framework for cyberattack detection in IIoT. The framework uses a temporal convolutional generative network to enable semi-supervised learning from semi-labeled data. The authors used TON_IoT [186] and LITNET-2020 [187] to validate the robustness and efficiency of the framework over different cyberattack detection approaches. Several studies have also addressed the synchronous or asynchronous operation of nodes within the network topology. In this regard, Pinyoanuntapong *et al.* [59] implemented a solution to mitigate high communication congestion in decentralized networks, leading to robust distributed model training in heterogeneous IoT environments where stragglers (i.e., slow devices) are commonplace due to varying computation and network speeds of IoT devices. Similarly, Ochiai *et al.* [188] proposed a similar approach for ad-hoc wireless environments, achieving an average accuracy of 95% for the test IID dataset compared to on-device training, which obtained 84.7%. Mothukuri *et al.* [93] proposed a Blockchain-based approach and Hyperledger Fabric with a gamification component for integrating a secure application to seal and sign asynchronous and synchronous DFL collaborative tasks. Their results demonstrated that security enhancements improve the FL process using a custom aggregation algorithm and asynchronous communications. Finally, Kang *et al.* [17] enhanced privacy in the industrial metaverse by using a DFL framework with cross-chain power for decentralized, secure, and privacy-preserving data training in physical and virtual spaces. The framework uses a hierarchical Blockchain architecture with the main chain and multiple sub-chains. Ranathunga *et al.* [189] introduced a similar cross-silo architecture using a network of aggregators to optimize model quality, minimize convergence time, and improve system performance in various industrial applications. Similarly, Singh, Yang, and Park [138] proposed an approach for Industry 5.0. This scheme allowed local learning updates within the different departments of the industry using 5G. The validation outcomes showed an accuracy of 93.5% in a 50% active node.

C. Mobile Services

The proliferation of interconnected mobile devices and the development of advanced intelligent models have facilitated the emergence of decentralized approaches that offer increased efficiency and optimal training of models involved in various everyday tasks. To this end, DFL-based approaches have gained significant attention in mobile services, as demonstrated by the literature in Table VII.

Most research in this field has focused on leveraging edge computing to bring processing and data storage closer to the source of the request, thereby improving response times and conserving bandwidth on constrained devices. In this context,

Table VII: Comparison of DFL solutions focused on mobile services.

Ref.	Year	Goal	Federation Architecture	Topology	Scenario	ML Model	Aggregation Algorithm	Local Model Aggregation	Client Selection	Dataset
[7]	2021	Reduce the number of directly connected end devices, avoiding the overhead of unnecessary local updates	Cross-silo Cross-device	Centralized Decentralized	Real	CNN MLP	FedAvg	Synchronous Asynchronous	Random	MNIST [174]
[190]	2021	Novel approach based on continuous authentication of users trained by an organized peer-to-peer federation involving different organizations	Cross-device	Centralized Decentralized	Simulation	Custom model	FedAvg	Asynchronous	Scheduling	Private
[191]	2021	Blockchain-based FL scheme to strengthen communication security and data privacy protection in DITENs	Cross-silo Cross-device	Centralized Decentralized	Simulation	DNN	Custom algorithm	Synchronous Asynchronous	Random	MNIST [174] FMNIST [192]
[193]	2022	Simulation tool to capture the decentralized and asynchronous nature of FL operation in conjunction with DTL techniques	Cross-device	Decentralized	Simulation	U-Net	FedAvg	Asynchronous	Sequential	BRATS [169] Private
[68]	2022	Reputation scheme to balance the weights between trust values of parameters and bid prices	Cross-device	Centralized Decentralized	Simulation	Four-layered FNN	FedAvg	Synchronous	Scheduling	MNIST [174]
[18]	2022	Novel distributed validation weighting scheme to evaluate the performance of a mobile node in the federation versus a distributed validation set	Cross-device	Centralized Decentralized	Simulation	Two-layered CNN	Custom algorithm	Synchronous Asynchronous	Sequential	CIFAR-10 [154] CIFAR-100 [154] EMNIST [194]
[72]	2022	Decentralized recommender system based on the principles of gossip learning	Cross-device	Decentralized	Simulation	GMF PRME-G	FedAvg FedFast Reptile	Synchronous	Scheduling	Foursquare-NYC [195] Gowalla-NYC [196] MovieLens [197]
[73]	2022	Gossip-based system to maximize social awareness among nodes, improving reliability and latency	Cross-device	Decentralized	Simulation	*N/S*	Custom algorithm	Asynchronous	Sequential	Facebook [198]
[199]	2023	Federated GNN supporting supervised and unsupervised learning with security measures	Cross-device	Centralized Decentralized	Simulation	GCN GAT	Custom algorithm	Synchronous	Squential	Facebook [198] LastFM [200]
[201]	2023	Privacy-preserving P2P learning for asynchronous and collaborative tasks.	Cross-device	Centralized Decentralized	Simulation	MobileNetv2 BLSTM	Custom algorithm	Synchronous Asynchronous	Random	CIFAR-10 [154] Avito [202] IMDb [203]
[204]	2023	Personalized and fully decentralized FL algorithm using knowledge distillation	Cross-device	Decentralized	Simulation	Three-layered NN CNN	Custom algorithm	Synchronous Asynchronous	Random	IoT identification [205] EMNIST [194]
[206]	2023	Decentralized and privacy-preserving FL for improved performance and scalability	Cross-device	Decentralized	Simulation	CNN	FedAvg	Synchronous	Random	MNIST [174]

N/S (Not Specified) by the authors

Wang *et al.* [7] proposed a novel solution that deployed mobile edge nodes at various network locations to act as communication hubs between the cloud and end devices. This approach effectively sidesteps the latency associated with high server concurrency. Moreover, the proposed method filters unnecessary models and communications using cosine similarity. Experimental results showed that the proposed scheme reduces the number of local updates by 60% compared to CFL and increases the convergence speed of the evaluated model by 10.3%. Additionally, Monschein *et al.* [190] addressed the challenge of acquiring large amounts of data to train powerful ML models, which is often an overwhelming task for a single organization. The authors proposed a methodology combining the establishment of continuous user-based authentication with federated and decentralized data governance.

As in the previous scenarios, DLT is recurrently used to ensure communications in decentralized scenarios. In this sense, Lu *et al.* [191] conducted a theoretical analysis to enhance communication security and protect data privacy in Digital Twin Edge Networks (DITENs). The proposed scheme demonstrated significant improvements in communication efficiency and data security for IoT applications, as confirmed by numerical results. Wilhelmi, Guerra, and Dini [193] studied the impact of ledger inconsistencies on DFL performance. The study employed a reliable simulation tool that captures the decentralized and asynchronous nature of Blockchain operation. Qi *et al.* [68] proposed a hybrid Blockchain-based incentive mechanism that addresses similar challenges. The authors leveraged smart contracts and dynamic reputation score calculation of each DFL participant. In contrast to previous studies, performance is evaluated against malicious or non-honest nodes. In edge computing settings where P2P communication is commonly used for exchanging model parameters between nodes, an efficient algorithm named CoCo was introduced in [18]. This algorithm integrates topology construction optimization and model compression to accelerate DFL. Extensive simulation results show that CoCo achieves a ten-fold speedup and reduces the communication cost by 50% on average compared to existing DFL baselines.

The rise of decentralized scenarios has enabled new applications such as recommender systems and social networks. In particular, Belal *et al.* [72] introduced PEPPER, a decentralized recommender system based on gossip learning principles. The system extracts relevant content for users to aid them in their daily activities, such as finding relevant places to visit, content to consume, or items to buy. In PEPPER, users gossip about model updates and aggregate them asynchronously. By conducting experiments on three real datasets implementing two use cases, location registration recommendation and movie recommendation, the authors demonstrated that their solution converges up to 42% faster than other decentralized solutions. Furthermore, Khelghatdoust and Mahdavi [73] proposed a Decentralized Online Social Network (DOSN) based on gossip to address the privacy and scalability issues of centralized social networks. The authors observed almost a 30% reduction in search latency and a 10% improvement in communication reliability through trusted contacts. These results demonstrate the potential of decentralized approaches in providing efficient solutions for modern applications such as recommender systems and social networks. To improve the previous scenarios, Pan *et al.* [199] utilized a tree constructor to improve representation capability given the limited structural information and a Monte

Table VIII: Comparison of DFL solutions related to the military scenario.

Ref.	Year	Goal	Federation Architecture	Topology	Scenario	ML Model	Aggregation Algorithm	Local Model Aggregation	Client Selection	Dataset
[208]	2020	Federated learning-based security architecture for jamming attack detection for FANET	Cross-device	Centralized Decentralized	Simulation	Three-layered NN	FedAvg	Asynchronous	Random	CRAWDAD [209] NS3 FANET [210]
[211]	2020	Distributed defense solution for sustainable society using the features of Blockchain technology and federated learning	Cross-device	Centralized Decentralized	Simulation	CNN	FedAvg	Synchronous	Random	Custom dataset
[51]	2021	DFL solution with a fully connected network between battlefield devices employing a method of random walks and alternate directions during federation	Cross-device	Decentralized	Real	ELM	ISPW-ADMM	Asynchronous	Random	Private
[212]	2021	Novel architecture that allows FL within UAV networks without a central entity	Cross-device	Decentralized	Simulation	CNN	FedAvg	Asynchronous	Random	*N/S*
[19]	2021	Secure framework for UAV-assisted mobile crowdsensing to promote high-quality model sharing	Cross-device	Decentralized	Simulation	Custom model	Custom FedAvg	Synchronous	Scheduling	*N/S*
[65]	2021	Precoding and decoding strategies for D2D communication	Cross-device	Decentralized	Real	Custom model	FedAvg	Synchronous	Scheduling	BRATS [169]
[213]	2022	Novel integration of Blockchain and FL in UAV edge computing networks and associated challenges and solutions	Cross-device	Decentralized	Simulation	*N/S*	Custom FedAVG	Asynchronous	Sequential	*N/S*
[42]	2022	Cross-domain authentication of UAVs using multi-signature smart contracts	Cross-device	Decentralized	Simulation	CNN	Custom algorithm	Synchronous Asynchronous	Random	EMNIST [194]
[214]	2023	Resource-efficient framework for mmWave aerial-terrestrial integrated networks using UAVs	Cross-device	Decentralized	Simulation	*N/S*	Custom algorithm	Asynchronous	Scheduling	MNIST [174] CIFAR-10 [154]
[215]	2023	Feasibility and adaptability of DFL in maritime transportation	Cross-device	Centralized Decentralized	Simulation	Fully-connected ANN	Custom algorithm	Synchronous	Random	Private

N/S (Not Specified) by the authors

Carlo Markov Chain-based algorithm to mitigate workload imbalance caused by degree heterogeneity. Finally, it is worth mentioning that other studies have explored the use of innovative techniques in DFL, such as P2P with knowledge distillation, for various applications such as social network scenarios or IoT device identification [201], [204], [206].

D. Military

The military scenario has also yielded numerous improvements in DFL communications between different devices on the battlefield. A clear example is the European Future Combat Air System (FCAS) program [207], the largest European armament effort since World War II, where unmanned aircraft belong to a decentralized and collaborative system during combat missions. Table VIII highlights the most relevant items in the military scenario where DFL-based approaches are applied.

Flying Ad-hoc Network (FANET) is a type of network that applies DFL as a promising way to train collaborative and decentralized models. One challenge in this network is the evasion of malicious attacks, such as jamming. In jamming attacks, adversaries disrupt the victim network communication by generating interference at the receiver side, making it difficult for the intended signals to be correctly received and processed. For this problem, Mowla *et al.* [208] proposed security measures to prevent node jamming by creating a client group prioritization technique leveraging the Dempster-Shafer theory and using asynchronous communications to exchange model parameters. The results provided an accuracy of 82.01% for the CRAWDAD dataset and 89.73% for the NS3 FANET dataset. Additionally, Sharma, Park, and Cho [211] discussed using DFL in the Internet of Battlefield Things (IoBT) as a defense system employing AI to strengthen the armed forces. The authors rely on the characteristics of Blockchain technology to obtain high accuracy and a random device selection. A custom dataset consisting of several drone detection datasets [216] and images [217] is used to evaluate the effectiveness of the proposed model, obtaining an accuracy rate of 99% at the fog layer. Xiao *et al.* [51] presented an Inexact Parallel Random Walk Alternate Direction Multiplier Method (ISPW-ADMM) applied to fully connected networks. Then, a UAV communication scenario was deployed and compared with traditional methods such as W-ADMM, PW-ADMM, DGD, and distributed-ADMM (D-ADMM). Qu *et al.* [212] proposed a similar solution, where the authors provided three critical considerations: (i) the average loss achieved by DFL is similar to the one obtained in a centralized approach; (ii) for each UAV, the loss value after 60 rounds is similar to the achieved in the centralized approach; and (iii) the training latency is always more negligible in DFL since it does not need to broadcast the model parameters.

Alternative solutions have been proposed to address the issue of secure DFL for UAV-assisted Mobile Crowdsensing (MC). For instance, Wand *et al.* [19] introduced a DFL approach that utilizes Blockchain to securely exchange local model updates and verify contributions without a central node in a cross-device scenario. In the same way, Shi *et al.* [65] proposed a consensus phase based on additive noise at each iteration of the algorithm, which enhanced the robustness of the solution to changes in wireless network topology. The solution converged linearly in a binary classification using MNIST. Furthermore, Zhu *et al.* [213] used the same technology among multiple untrusted parties with anonymous, immutable, and distributed characteristics, achieving similar results in performance for UAV intelligent edge computing networks to the previous study. Another study by Feng *et al.* [42] followed a similar approach, utilizing HFL and non-IID data in a fully connected network. The authors implemented cross-domain UAV authentication through multi-signature smart contracts, with global model updates computed using these smart contracts instead of a centralized server. Recently, decentralized model dissemination has emerged as a promising approach for DFL in mmWave aerial-terrestrial integrated networks. To this end, Al-Abiad *et al.* [214] presented an

Table IX: Comparison of DFL solutions dealing with vehicular scenarios.

Ref.	Year	Goal	Federation Architecture	Topology	Scenario	ML Model	Aggregation Algorithm	Local Model Aggregation	Client Selection	Dataset
[20]	2020	Proactive caching scheme where vehicles are trained from sparse data, mitigating privacy risks	Cross-device	Decentralized	Real	CF-VAE	FedAvg	Asynchronous	Random	MovieLens [197]
[219]	2020	Asynchronous FL-based approach for efficient and secure data sharing	Cross-device	Decentralized	Simulation	Custom NN	FedAvg	Asynchronous	Scheduling	*N/S*
[220]	2020	Autonomous Blockchain-based powered learning design for a vehicular communication network	Cross-device	Decentralized	Simulation	*N/S*	oVML	Synchronous Asynchronous	Random	*N/S*
[22]	2021	Novel method proposed by DFL with P2P communication between autonomous vehicles	Cross-device	Centralized Decentralized	Simulation	CNN	Custom algorithm	Synchronous	Schematic	MNIST [174] KITTI [221]
[3]	2022	DFL system to increase road user and object classification capability based on Lidar data	Cross-device	Centralized Decentralized	Simulation	PointNet ML	Custom FedAvg	Synchronous	Random	nuScenes [222]
[223]	2023	Reinforcement Learning with FL agent for predictive QoS in teleoperated driving scenarios	Cross-device	Decentralized	Simulation	DNN	Custom algorithm	Asynchronous	Random	KITTI [221]
[224]	2023	Blockchain-enhanced DFL for efficient and privacy-protective data sharing	Cross-device	Decentralized	Simulation	CNN	FedAvg	Asynchronous	Scheduling	FMNIST [192]

N/S (Not Specified) by the authors

algorithm that makes use of UAVs as local model aggregators through UAV-to-UAV communications and reduces the energy consumption of DFL using Radio Resource Management (RRM) under the constraints of over-the-air learning latency. In maritime environments, optimizing transportation focusing on certain performance metrics may lead to non-convex problems due to the large number and heterogeneity of network nodes and vessels. To tackle this issue, Giannopoulos *et al.* [215] presented and analyzed various use cases in these scenarios and demonstrated the superiority of DFL over traditional ML approaches using datasets from an enterprise specializing in the maritime industry.

E. Vehicles

In recent years, concerns regarding road safety have significantly increased, prompting significant efforts towards developing automated solutions to detect distractions while driving and to alert vehicles on the road intelligently [218]. In this sense, decentralized solutions and FL have emerged as promising approaches in this sector, as illustrated by the latest vehicle solutions highlighted in Table IX.

The unique characteristics of Vehicle-to-Everything (V2X) communication, including the high mobility of vehicles and limited storage capacity of nodes, present particular communication and processing challenges that can be addressed through DFL. Building on recent advances in ML, Yu *et al.* [20] proposed a proactive caching scheme based on SDFL and P2P communications. In this solution, a vehicle acts as a parameter server to aggregate the updated global model from peers instead of an edge node. Experimental results show that the solution outperforms typical baselines, gaining efficiency and autonomy. Furthermore, Blockchain supports data integrity when transmitting model parameters, thus addressing privacy concerns when sharing private data between vehicles. This mechanism has been utilized in several studies, such as [219] and [220]. While the former used deep Reinforcement Learning (RL) to select the participating nodes of DFL, thereby improving the efficiency of the data-sharing process, the latter minimized the system delay by exploiting the channel dynamics.

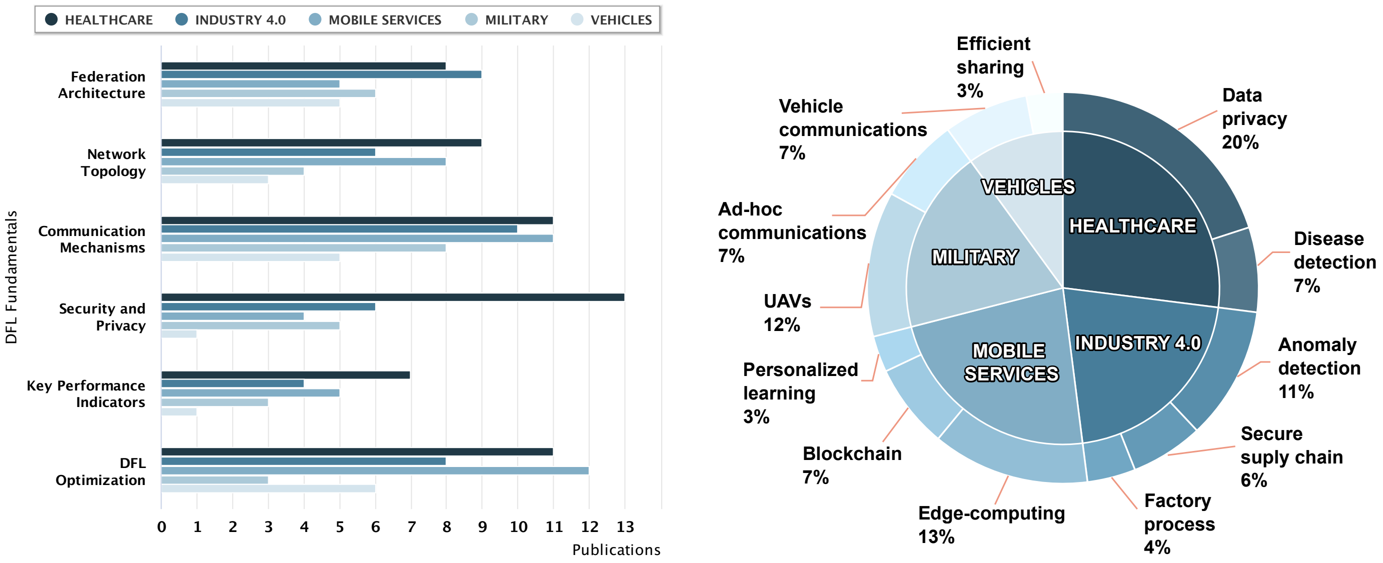
Finally, other studies consider new ways to optimize resources to present a solution comparable to server-based architectures. Chen *et al.* [22] proposed a novel approach to DFL based on a P2P network designed to be resilient to byzantine faults. The authors evaluated their method on the MNIST and KITTI datasets, assessing the performance of the proposed P2P approach in terms of accuracy, convergence speed, and fault tolerance and comparing it to traditional server-based methods. Similarly, Barbieri *et al.* [3] investigated the use of DFL methods to improve the classification of road users (such as pedestrians, cyclists, and vehicles) and objects based on Lidar data. The authors simulated a realistic V2X network using the collective perception service to share PointNet model parameters. The results showed a low latency training compared to existing distributed ML approaches. Another study that investigated the use of DFL in vehicular networks is presented in [223]. In this work, the authors focused on ensuring the Quality of Service (QoS) for communication between vehicles and remote drivers in remote driving scenarios. Specifically, they proposed using Predictive Quality of Service (PQoS) to predict and react to unanticipated degradation of the QoS. To implement PQoS in vehicular networks, the authors designed an RL agent to identify the optimal compression level for sending automotive data under low latency and reliability constraints. Finally, Hu et al. [224] integrated Blockchain into a fully connected vehicle topology. The authors demonstrated robust privacy protection and system reliability without significant latency, ensuring efficient and accurate data exchange, even with data of varying quality.

VI. Trends, Lessons Learned, and Open Challenges

Based on the different aspects of DFL analyzed through research questions Q1-Q3, and following the storyline presented in Fig. 2, this section responds to "*Q4. What trends, lessons learned, and challenges have emerged in DFL?*" To provide a clear structure, each trend, lesson, and challenge is related to a specific category, such as *[Fund.]* for fundamentals, *[Fram.]* for frameworks, and *[Sce.]* for application scenarios.

A. Current Trends

The main trends in current work, based on the evolution of recent solutions, are as follows (see Fig. 9).



(a) Fundamentals per application scenario.

(b) Main objectives per application scenario.

Fig. 9: Distribution of DFL publications.

***[Fund.]* Federation architectures, network topologies, and communication mechanisms are extensively studied**. Most works analyze the communications between network nodes, the network topologies (mostly fully connected), and the architecture it supports, with DFL being the main approach addressed. Fig. 9a shows the trend of the above fundamentals in different application scenarios.

***[Fund.]* Fully connected network topologies are widely applied to DFL scenarios**. Network topologies where all nodes are connected to each other are the most commonly used in DFL. Its versatility and simplicity of construction have made it a usable approach in most application scenarios. Table II shows how about 50% of the papers analyzed in the network topology fundamental belong to this approach.

***[Fund.]* The optimization of communications is predominant in recent work on DFL**. The inherent functionality of DFL requires communication optimization techniques between federation participants. In this sense, Table II shows how more than 65% of the analyzed solutions address complexity reduction in model parameter exchanges. The healthcare and mobile services are the most optimized application scenarios in terms of communications (see Fig. 9a).

***[Fram.]* Most of the frameworks are adapted to cross-device environments**. The reviewed frameworks offer DFL for deploying simulated or real federated scenarios with limited devices (see Table IV). They also typically provide models fed by images or timelines, aggregation algorithms, and various communication technologies between participants.

***[Sce.]* DFL is widely used in healthcare scenarios**. DFL meets its goals in healthcare data analytics, with a significant impact on the accuracy of medicine and, ultimately, improved treatment and diagnosis (see Fig. 9b). Electronic medical records, medical imaging, disease detection, and collaborative drug discovery are the most significant use cases for DFL.

***[Sce.]* DFL gains momentum in mobile services and Industry 4.0**. The high performance offered by DFL has led to the increasing relevance of both application scenarios. In mobile services, DFL improves personalization and recommendation systems in edge devices while protecting user privacy. In Industry 4.0, DFL enables collaboration between multiple organizations without sharing sensitive data, reducing costs, and improving efficiency.

***[Sce.]* Anomaly detection and deployment on UAVs are prominent trends**. Fig. 9b shows how both topics maintain remarkable applicability in different application scenarios. Specifically, anomaly detection represents the 11% of works documented in industrial scenarios, while the use of UAVs represents the 12% of works in military scenarios.

B. Lessons Learned

After reviewing and analyzing the state of the art, the following lessons have been learned:

***[Fund.]* The use of specific aggregation algorithms for DFL is still limited**. Although FedAvg is widely used in existing frameworks, applications in certain DFL scenarios need to obtain similar results. Studies of different application scenarios (see Table V, VI, VII, VIII, and IX) have opted for customizations of the algorithm to adapt it to the peculiarities of the federation models.

***[Fund.]* Limited analysis of improving decentralized systems with DFL**. Decentralized systems face challenges such as consistency and coordination between nodes. In the literature, there are not enough studies comparing and evaluating DFL enhancement in these systems. There is a need to determine the resilience, robustness, and overall security provided by this approach that reduces dependency on a server.

***[Fram.]* There is a limited number of solutions providing realistic federation benchmarks**. As more DFL architectures are developed, it is important to have a benchmark with representative datasets and workloads to evaluate existing systems and direct future development. Although many DFL frameworks provide benchmarks (see Table IV), no single benchmark has been widely used in all studies reviewed. In addition, existing benchmarks often ignore metrics such as system efficiency, reliability, or architecture robustness. The evaluation of model performance on non-IID datasets and system security needs further research.

***[Fram.]* There is no consensus on frameworks in the literature for deploying DFL architectures**. The frameworks analyzed in Table IV show heterogeneous characteristics, mostly adapted to validation scenarios. Therefore, there are limited open-source DFL frameworks with sufficient maturity to be network, node, and data agnostic.

***[Sce.]* The military and vehicular scenarios are complex application scenarios for deploying DFL solutions**. The solutions presented in the military (see Table VIII) and vehicles scenarios (see Table IX) lack the robustness to be put into practice, so they usually recreate simulated deployments to adapt nodes and models. In addition, they present limited bandwidth, unstable network connections, and high-security requirements in simulations. Therefore, strategies such as edge computing, hybrid approaches, and Blockchain technology are used to reduce complexity.

***[Sce.]* There is a lack of literature using unsupervised learning in DFL architectures**. The literature has not addressed the applicability of unsupervised learning in DFL-based scenarios where federation participants do not operate locally labeled data. Table VI, VII, VIII, and IX indicate that the literature only defines solutions using supervised ML models such as MLP or CNN for classification tasks.

C. Open Challenges

Based on the current state of the art, the following points represent the main challenges that future DFL solutions might consider. In addition, Table X summarizes the open challenges and their future developments, indicating the importance of their application in the future.

***[Fund.]* Improve the scalability of the solution when the number of participants in the federation increases**. The development of algorithms that can dynamically select participants based on their availability, network connection, and trustworthiness would help to ensure that the federation remains resilient to dropouts and the unavailability of participants. In addition, exploring new techniques for compressing and aggregating models across multiple participants can reduce communication and computation overhead, particularly for large-scale federations. Finally, including personalized local model learning can improve training while minimizing the impact of low network dependability and client availability.

***[Fund.]* Improve the cybersecurity mechanisms depending on the participant and the application scenario**. The security of the nodes participating in DFL and their ability to detect and prevent attacks or threats in heterogeneous scenarios are crucial to the success of this approach. A promising solution would be to design an architecture that treats participants differently according to their privacy restrictions, allowing for a more personalized and secure approach. This is particularly relevant for application scenarios such as social media applications, industrial IoT, and Blockchain, where user privacy protection is critical.

Table X: Open challenges in DFL and future developments.

Challenge	Future Developments
Fundamentals *[Fund.]*	
Scalability of DFL with increasing participants (!!!)	• Dynamic participant selection • Personalized local model learning
Cybersecurity mechanisms for a secure DFL (!!!)	• Detect attacks in DFL scenarios • Different treatment based on privacy
Trustworthiness among federation participants (!!!)	• Maintain trust policies • Prevent dishonest behavior
Homogeneous node participation (!!)	• Quantization and gradient compression • Use of SDFL
Address participant mobility in DFL scenarios (!!)	• Topology-aware node reconfiguration • Resilient synchronization methods
Study of adversarial attacks (!!)	• Identify the techniques and their impacts • Compare against traditional approaches
Explore the use of Reinforcement Learning (!!)	• Optimize the federated model performance • Improve the selection of participants
DFL standardization efforts (!!)	• Promote comprehensive DFL standards • Involve standard-setting bodies (ISO, IEEE)
5G and 6G technologies for communications (!)	• Network slicing utilization • 5G/6G-integrated edge computing
Frameworks *[Fram.]*	
Modular, scalable, and efficient frameworks (!!!)	• Implement and manage DFL fundamentals • Application in practical scenarios
Heterogeneous datasets in decentralized participants (!!)	• Data preprocessing and normalization • Advanced data augmentation techniques
Dynamic scheduling of federated network (!)	• Adaptable federation architecture • Resilient algorithms
Application scenarios *[Sce.]*	
Exploration of new DFL application scenarios (!!)	• Evaluate DFL in smart city technologies • Combine AI, IoT, and DFL for testing tools

! low importance, !! high importance, !!! critical

***[Fund.]* Enhance trustworthiness among federation participants in DFL approaches**. To ensure the accuracy and reliability of the federated model, participating nodes must establish trust policies with one another. These policies allow participants to aggregate model parameters based on reputation and performance history only from those they trust. By doing so, participants can avoid dishonest behavior and maintain the integrity of the FL process, thereby encouraging the participation of other nodes in the network.

***[Fund.]* Ensure the homogeneous participation of the constrained nodes in the federation**. It is challenging to guarantee the participation of nodes with limited autonomy or bandwidth. Some solutions could use quantization methods and gradient compression techniques to reduce communication overhead. Another option is to consider other federation architectures, such as SDFL, where nodes with limited resources can participate in the federation while still maintaining some level of centralization.

***[Fund.]* Address participant mobility**. The dynamic nature of DFL, where participants may join or leave the federation at any time, poses challenges in maintaining learning stability and reliability. Future DFL solutions need to manage mobility and develop strategies to handle the continuous flow of participants while ensuring efficient learning.

***[Fund.]* In-depth study of adversarial attacks applied in DFL approaches**. The literature on cyberattacks affecting DFL is limited, with relatively little attention given to adversarial attacks and their impact on DFL performance. Thus, evaluating their impact compared to other traditional approaches, such as CFL, can help understand the vulnerabilities of DFL better and inform the development of more robust security measures.

[Fund.] **Explore the use of RL**. The literature regarding RL applied in decentralized collaborative scenarios is scarce. This learning approach can optimize the performance of the federated model by dynamically allocating resources and adjusting the learning process. Additionally, it can be used to optimize the selection of participants in the federation, as well as to dynamically adjust the communication and aggregation parameters based on the performance of the participants.

[Fund.] **Research the use of 5G/6G networks to improve DFL communications**. The rise of 5G and subsequent 6G networks presents a significant potential to enhance communications performance by providing faster and more reliable communication between nodes. Thus, exploring techniques for efficient data aggregation and transmission is necessary using ultra-low latency and high bandwidth capabilities. Moreover, researchers could examine the possibility of utilizing network slicing to allocate dedicated resources for DFL tasks and integrate edge computing within these networks. Furthermore, with 5G/6G-powered DFL, a deep focus on the security and integrity of P2P communications between edge devices becomes critical. Given the potential vulnerability of these networks to novel security threats, dedicated research on defense mechanisms is essential to safeguard secure and trusted P2P exchanges. To this end, it is necessary to deploy appropriate defense mechanisms, such as encryption, SMPC, or Blockchain technologies.

[Fram.] **Create modular, scalable, and efficient frameworks for diverse application scenarios**. The literature has not addressed an agnostic federation for different solutions in terms of DFL fundamentals. Therefore, it is necessary to design and implement a robust and reliable solution to generate realistic scenarios, considering elements such as model creation, data storage, framework usage, model parameter exchange, and model aggregation. The framework should focus on efficiency, considering communication overhead, computational resources, and storage requirements.

[Fram.] **Handle heterogeneous datasets in decentralized participants**. Participants may possess varying sizes, distributions, and quality data, impacting the model performance. Addressing the challenge of dividing or adapting datasets to account for these differences is crucial to achieving optimal performance in DFL scenarios. It involves creating methods for data preprocessing, normalization, and augmentation that can accommodate the distinct characteristics and variations inherent in the datasets of each participating node.

[Fram.] **Adapt the dynamic scheduling of the federated network to the application scenario**. Due to the instability of the application scenario, the number of participants may not be fixed during the learning process in DFL. Therefore, it is necessary to develop algorithms that can dynamically adjust the number of participants in the federation based on changes in inputs or outputs from the nodes. This would require building resilient algorithms that handle low network dependability and client availability. In addition, exploring the use of adaptive FL techniques that can adjust the level of centralization or decentralization in response to changes in the application scenario could be beneficial.

[Sce.] **Explore new DFL application scenarios**. DFL can be effectively applied in a multitude of application scenarios. For instance, DFL can help to increase the intelligence of robotic and autonomous systems, making them more efficient and reliable. In smart city technologies, DFL can exchange local model parameters between devices, enhancing overall performance. Additionally, DFL can be evaluated using Digital Twin (DT) technology, combining AI, IoT, and DFL to create a simulated environment where tools can be tested in real time without exposing people or facilities to unknown risks.

[Fund.] **Explore standardization activities for DFL**. It is necessary to have comprehensive standards to ensure uniformity, interoperability, and trustworthiness across various implementations. These standards should address key elements like privacy-preserving methods, security features, data quality standards, model training processes, and performance indicators. Standardizing DFL could encourage consistency in research outcomes and establish a universal framework for assessing DFL methodologies.

VII. Conclusion and Future Work

This work studies the evolution of DFL in recent years, providing the basic and distinctive fundamentals versus traditional federated architectures, its current application scenarios, and the frameworks that manage the deployment of DFL architectures. In this context, the present work has answered the following research questions.

Q1. What are the fundamental aspects of DFL? Section III studies the main aspects of DFL in the most current and representative works in the literature, clarifying the differences with CFL. The analysis provides a set of fundamentals related to federation architectures, network topologies, communication mechanisms, and security techniques. It also proposes KPIs to evaluate the approach and mechanisms to optimize DFL. With all this, the work proposes the first taxonomy that defines and details the fundamentals of DFL.

Q2. What DFL frameworks exist, and what fundamentals do they provide? Section IV discusses the main frameworks currently used to deploy DFL scenarios. This section shows the most mature frameworks in FL versus the incipient solutions that favor the inclusion of DFL. Thus, this paper highlights the prominence of current solutions and the DFL fundamentals contemplated to generate robust solutions to the scenario.

Q3. Which are the main characteristics of the most relevant scenarios of DFL? Section V describes, analyzes, and compares the most relevant and recent solutions according to their application scenario. The predominant application scenarios are healthcare, mobile services, and Industry 4.0. Also, this work determines that fields such as military or vehicles have grown significantly in recent years. Regarding fundamentals, decentralized cross-device architectures in fully connected network topologies are predominant. Optimization mechanisms are applied to the above aspects, mainly to communications and aggregation algorithms.

Q4. What trends, lessons learned, and challenges have emerged in DFL? Lessons learned, current trends, and future challenges have been documented in Section VI.

It details how to create more sophisticated federation architectures, topologies, optimizations, and their applicability in different scenarios. At the same time, certain limitations in the recent literature are also detailed. For instance, the scarce comparison between DFL scenarios or the lack of frameworks allowing their deployment. In addition, handling heterogeneous datasets, cyberattacks, or using 5G/6G in communications have not yet been robustly applied, opening avenues for future research.

As future work, it is planned to design and implement scalable solutions capable of generating heterogeneous scenarios. To achieve this, it is necessary to create solutions using agnostic data types employing performance data from the devices acting as nodes. These solutions will be integrated with ML/DL techniques, communication optimization, and data aggregation techniques that preserve security and privacy while guaranteeing network and node performance capabilities. Furthermore, these solutions must offer a range of configurable options based on various federation architectures, including CFL, DFL, or SDFL, as well as decentralized technologies such as Blockchain. Finally, it is considered to define and build sets of metrics to complement the functionality of the solution and the evaluation of the system.

REFERENCES

[1] H. Liu and W. Wu, "Federated reinforcement learning for decentralized voltage control in distribution networks," *IEEE Transactions on Smart Grid*, vol. 13, no. 5, pp. 3840–3843, 2022.

[2] Z. Lian *et al.*, "DEEP-FEL: Decentralized, efficient and privacy-enhanced federated edge learning for healthcare cyber physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3558–3569, 2022.

[3] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6G connected vehicles," *Vehicular Communications*, vol. 33, p. 100396, 2022.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[5] L. He, A. Bian, and M. Jaggi, "COLA: Decentralized Linear Learning," in *32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Curran Associates Inc., 2018, p. 4541–4551.

[6] A. Hard, K. Partridge, R. Mathews, and S. Augenstein, "Jointly learning from decentralized (federated) and centralized data to mitigate distribution shift," in *Proceedings of NeurIPS Workshop on Distribution Shifts*, 2021.

[7] T. Wang, Y. Liu, X. Zheng, H.-N. Dai, W. Jia, and M. Xie, "Edge-based communication optimization for distributed federated learning," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.

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

[9] A. Bellet, A.-M. Kermarrec, and E. Lavoie, "D-Cliques: Compensating for data heterogeneity with topology in decentralized federated learning," in *41st International Symposium on Reliable Distributed Systems proceedings*, 2022, p. 11. [Online]. Available: http://infoscience.epfl.ch/record/294834

[10] E. Georgatos, C. Mavrokefalidis, and K. Berberidis, "Efficient fully distributed federated learning with adaptive local links," *arXiv preprint arXiv:2203.12281*, 2022.

[11] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia, Eds., vol. 1, 2019, pp. 374–388.

[12] Y. Qu, C. Xu, L. Gao, Y. Xiang, and S. Yu, "FL-SEC: Privacy-preserving decentralized federated learning using SignSGD for the Internet of Artificially Intelligent Things," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 85–90, 2022.

[13] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, "System optimization of federated learning networks with a constrained latency," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 1095–1100, 2022.

[14] B. Wang, J. Fang, H. Li, X. Yuan, and Q. Ling, "Confederated learning: Federated learning with decentralized edge servers," *arXiv preprint arXiv:2205.14905*, 2022.

[15] EPFL - Machine Learning and Optimization Laboratory, "DISCO - Distributed Collaborative Machine Learning," GitHub, 2022. [Online]. Available: https://github.com/epfml/disco

[16] T. V. Nguyen *et al.*, "A novel decentralized federated learning approach to train on globally distributed, poor quality, and protected private medical data," *Scientific Reports*, vol. 12, no. 1, p. 8888, 2022.

[17] J. Kang *et al.*, "Blockchain-based federated learning for industrial metaverses: Incentive scheme with optimal AoI," in *IEEE International Conference on Blockchain (Blockchain)*. IEEE Computer Society, 2022, pp. 71–78.

[18] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.

[19] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for uav-assisted crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1055–1069, 2021.

[20] 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 *IEEE 26th International Conference on Parallel and Distributed Systems*, 2020, pp. 601–608.

[21] 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, 2022.

[22] 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 Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8639–8652, 2021.

[23] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[24] 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.

[25] 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.

[26] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[27] P. Boobalan *et al.*, "Fusion of federated learning and industrial Internet of Things: A survey," *Computer Networks*, vol. 212, p. 109048, 2022.

[28] M. Joshi, A. Pal, and M. Sankarasubbu, "Federated learning for healthcare domain - pipeline, applications and challenges," *ACM Trans. Comput. Healthcare*, 2022.

[29] L. Witt, M. Heyer, K. Toyoda, W. Samek, and D. Li, "Decentral and incentivized federated learning frameworks: A systematic literature review," *arXiv preprint arXiv:2205.07855*, 2022.

[30] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Comput. Surv.*, 2022.

[31] M. Billah, S. T. Mehedi, A. Anwar, Z. Rahman, and R. Islam, "A systematic literature review on blockchain enabled federated learning framework for Internet of Vehicles," *arXiv preprint arXiv:2203.05192*, 2022.

[32] R. Gupta and T. Alam, "Survey on federated-learning approaches in distributed environment," *Wireless Personal Communications*, 2022.

[33] D. Saraswat *et al.*, "Blockchain-based federated learning in UAVs beyond 5G networks: A solution taxonomy and future directions," *IEEE Access*, vol. 10, pp. 33 154–33 182, 2022.

[34] J. Wu, S. Drew, F. Dong, Z. Zhu, and J. Zhou, "Topology-aware federated learning in edge computing: A comprehensive survey," *arXiv preprint arXiv:2302.02573*, 2023.

[35] H. Chen, H. Wang, D. Jin, and Y. Li, "Advancements in federated learning: Models, methods, and privacy," *arXiv preprint arXiv:2302.11466*, 2023.

[36] 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, 2023.

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

[38] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, "DisPFL: Towards communication-efficient personalized federated learning via decentralized sparse training," *arXiv preprint arXiv:2206.00187*, 2022.

[39] O. Marfoq, C. Xu, G. Neglia, and R. Vidal, "Throughput-optimal topology design for cross-silo federated learning," in *34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Curran Associates Inc., 2020.

[40] J. Han, Y. Han, G. Huang, and Y. Ma, "DeFL: Decentralized weight aggregation for cross-silo federated learning," *arXiv preprint arXiv:2208.00848*, 2022.

[41] S. P. Karimireddy *et al.*, "Breaking the centralized barrier for cross-device federated learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 28 663–28 676.

[42] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3582–3592, 2022.

[43] M. Wilt, J. K. Matelsky, and A. S. Gearhart, "Scatterbrained: A flexible and expandable pattern for decentralized machine learning," *arXiv preprint arXiv:2112.07718*, 2021.

[44] S. Kalra, J. Wen, J. C. Cresswell, M. Volkovs, and H. R. Tizhoosh, "ProxyFL: Decentralized federated learning through proxy model sharing," *arXiv preprint arXiv:2111.11343*, 2021.

[45] R. Kanagavelu *et al.*, "Ce-fed: Communication efficient multi-party computation enabled federated learning," *Array*, vol. 15, p. 100207, 2022.

[46] R. Xu and Y. Chen, "μDFL: A secure microchained decentralized federated learning fabric atop IoT networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2677–2688, 2022.

[47] M. Yemini, R. Saha, E. Ozfatura, D. Gündüz, and A. J. Goldsmith, "Semi-decentralized federated learning with collaborative relaying," in *IEEE International Symposium on Information Theory*, 2022, pp. 1471–1476.

[48] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative D2D local model aggregations," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3851–3869, 2021.

[49] P. M. Sánchez Sánchez *et al.*, "Analyzing the Robustness of Decentralized Horizontal and Vertical Federated Learning Architectures in a Non-IID Scenario," *arXiv preprint arXiv:2210.11061*, 2022.

[50] C. Li, G. Li, and P. K. Varshney, "Decentralized federated learning via mutual knowledge transfer," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1136–1147, 2022.

[51] Y. Xiao *et al.*, "Fully decentralized federated learning-based on-board mission for uav swarm system," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3296–3300, 2021.

[52] T. Ma, H. Wang, and C. Li, "Quantized distributed federated learning for industrial internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3027–3036, 2023.

[53] Y.-T. Chow, W. Shi, T. Wu, and W. Yin, "Expander graph and communication-efficient decentralized optimization," in *50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 1715–1720.

[54] Y. Hua, K. Miller, A. L. Bertozzi, C. Qian, and B. Wang, "Efficient and reliable overlay networks for decentralized federated learning," *SIAM Journal on Applied Mathematics*, vol. 82, no. 4, pp. 1558–1586, 2022.

[55] T. Vogels, H. Hendrikx, and M. Jaggi, "Beyond spectral gap: The role of the topology in decentralized learning," *arXiv preprint arXiv:2206.03093*, 2022.

[56] K. Cheng *et al.*, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, pp. 87–98, 2021.

[57] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *International Joint Conference on Neural Networks*, 2020, pp. 1–9.

[58] M. S. Al-Abiad, M. Obeed, M. J. Hossain, and A. Chaaban, "Decentralized aggregation for energy-efficient federated learning via d2d communications," *IEEE Transactions on Communications*, vol. 71, no. 6, pp. 3333–3351, 2023.

[59] P. Pinyoanuntapong, W. H. Huff, M. Lee, C. Chen, and P. Wang, "Toward scalable and robust AIoT via decentralized federated learning," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 30–35, 2022.

[60] S. Zehtabi, S. Hosseinalipour, and C. G. Brinton, "Decentralized event-triggered federated learning with heterogeneous communication thresholds," *arXiv preprint arXiv:2204.03726*, 2022.

[61] J. Cao, Z. Lian, W. Liu, Z. Zhu, and C. Ji, "HADFL: Heterogeneity-aware decentralized federated learning framework," in *58th ACM/IEEE Design Automation Conference (DAC)*, 2021, p. 1–6.

[62] Z. Chen, W. Liao, P. Tian, Q. Wang , and W. Yu, "A fairness-aware peer-to-peer decentralized learning framework with heterogeneous devices," *Future Internet*, vol. 14, no. 5, 2022.

[63] Y. Zhou, Q. Chen, Z. Wang, D. Xiao, and J. Chen, "Communication-efficient cluster federated learning in large-scale peer-to-peer networks," *arXiv preprint arXiv:2204.03843*, 2022.

[64] X. Wang, A. Lalitha, T. Javidi, and F. Koushanfar, "Peer-to-peer variational federated learning over arbitrary graphs," *IEEE Journal on Selected Areas in Information Theory*, pp. 1–1, 2022.

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

[66] Z. Li *et al.*, "Mining latent relationships among clients: Peer-to-peer federated learning with adaptive neighbor matching," *arXiv preprint arXiv:2203.12285*, 2022.

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

[68] M. Qi, Z. Wang, S. Chen, and Y. Xiang, "A hybrid incentive mechanism for decentralized federated learning," *Distrib. Ledger Technol.*, 2022.

[69] I. Hegedus, G. Danner, and M. Jelasity, "Decentralized learning works: An empirical comparison of gossip learning and federated learning," *Journal of Parallel and Distributed Computing*, vol. 148, pp. 109–124, 2021.

[70] Z. Tang, S. Shi, B. Li, and X. Chu, "Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 909–922, 2023.

[71] J. Jiang, L. Hu, C. Hu, J. Liu, and Z. Wang, "BACombo: Bandwidth-aware decentralized federated learning," *Electronics*, vol. 9, no. 3, 2020.

[72] 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, 2022.

[73] M. Khelghatdoust and M. Mahdavi, "A socially-aware, privacy-preserving, and scalable federated learning protocol for distributed online social networks," in *Advanced Information Networking and Applications*. Springer International Publishing, 2022, pp. 192–203.

[74] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *36th International Conference on Machine Learning*, vol. 97. PMLR, 2019, pp. 3479–3487.

[75] J. Yin, X. Cui, and K. Li, "A reputation-based resilient and recoverable P2P botnet," in *IEEE Second International Conference on Data Science in Cyberspace (DSC)*, 2017, pp. 275–282.

[76] R. S. Rawat, M. Diwakar, and P. Verma, "Zeroaccess botnet investigation and analysis," *International Journal of Information Technology*, vol. 13, no. 5, pp. 2091–2099, 2021.

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

[78] B. Alangot, D. Reijsbergen, S. Venugopalan, P. Szalachowski, and K. S. Yeo, "Decentralized and lightweight approach to detect eclipse attacks on proof of work blockchains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1659–1672, 2021.

[79] B. Niu, Y. Chen, Z. Wang, F. Li, B. Wang, and H. Li, "Eclipse: Preserving differential location privacy against long-term observation attacks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 125–138, 2022.

[80] J. Verbraeken, M. de Vos, and J. Pouwelse, "Bristle: Decentralized federated learning in byzantine, non-i.i.d. environments," *arXiv preprint arXiv:2110.11006*, 2021.

[81] G. Lu, Z. Xiong, R. Li, and W. Li, "Decentralized federated learning: A defense against gradient inversion attack," in *Wireless Internet*, 2023, pp. 44–56.

[82] S. Chen, D. Yu, Y. Zou, J. Yu, and X. Cheng, "Decentralized wireless federated learning with differential privacy," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6273–6282, 2022.

[83] E. T. Martínez Beltrán *et al.*, "Mitigating Communications Threats in Decentralized Federated Learning through Moving Target Defense," *arXiv preprint arXiv:2307.11730*, 2023.

[84] Á. L. Perales Gómez, E. T. Martínez Beltrán, P. M. Sánchez Sánchez, and A. Huertas Celdrán, "TemporalFED: Detecting Cyberattacks in Industrial Time-Series Data Using Decentralized Federated Learning," *arXiv preprint arXiv:2308.03554*, 2023.

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

[86] B. Verhaelen, F. Mayer, S. Peukert, and G. Lanza, "A comprehensive KPI network for the performance measurement and management in global production networks," *Production Engineering*, vol. 15, no. 5, pp. 635–650, 2021.

[87] Y. Qu *et al.*, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020.

[88] G. Nadiradze, A. Sabour, P. Davies, S. Li, and D. Alistarh, "Asynchronous decentralized sgd with quantized and local updates," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6829–6842, 2021.

[89] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 487–500, 2022.

[90] T.-T. Kuo and A. Pham, "Detecting model misconducts in decentralized healthcare federated learning," *International Journal of Medical Informatics*, vol. 158, p. 104658, 2022.

[91] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, "Decentralized collaborative learning of personalized models over networks," *arXiv preprint arXiv:1610.05202*, 2016.

[92] A. Gholami, N. Torkzaban, and J. S. Baras, "Trusted decentralized federated learning," in *IEEE 19th Annual Consumer Communications & Networking Conference*, 2022, pp. 1–6.

[93] V. Mothukuri, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and K.-K. R. Choo, "FabricFL: Blockchain-in-the-loop federated learning for trusted decentralized systems," *IEEE Systems Journal*, vol. 16, no. 3, pp. 3711–3722, 2022.

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

[95] P. M. Sánchez Sánchez, A. Huertas Celdrán, N. Xie, G. Bovet, G. Martínez Pérez, and B. Stiller, "Federatedtrust: A solution for trustworthy federated learning," *arXiv preprint arXiv:2302.09844*, 2023.

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

[97] J. Jiang and L. Hu, "Decentralised federated learning with adaptive partial gradient aggregation," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 230–236, 2020.

[98] B. Camajori Tedeschini *et al.*, "Decentralized federated learning for healthcare networks: A case study on tumor segmentation," *IEEE Access*, vol. 10, pp. 8693–8708, 2022.

[99] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, 2018.

[100] L. Barbieri, S. Savazzi, and M. Nicoli, "A layer selection optimizer for communication-efficient decentralized federated deep learning," *IEEE Access*, vol. 11, pp. 22 155–22 173, 2023.

[101] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 131–143, 2022.

[102] W. Liu, L. Chen, Y. Chen, and W. Wang, "Communication-efficient design for quantized decentralized federated learning," *arXiv preprint arXiv:2303.08423*, 2023.

[103] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Curran Associates Inc., 2018, p. 7663–7673.

[104] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[105] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[106] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[107] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o(1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[108] S. Wu, D. Huang, and H. Wang, "Network gradient descent algorithm for decentralized federated learning," *Journal of Business & Economic Statistics*, pp. 1–13, 2022.

[109] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30. Curran Associates, Inc., 2017.

[110] D. Basu, D. Data, C. Karakus, and S. Diggavi, *Qsparse-Local-SGD: Distributed SGD with Quantization, Sparsification, and Local Computations*. Curran Associates Inc., 2019.

[111] Y. Shi *et al.*, "Improving the model consistency of decentralized federated learning," *arXiv preprint arXiv:2302.04083*, 2023.

[112] X. Li, Y. Li, J. Wang, C. Chen, L. Yang, and Z. Zheng, "Decentralized federated meta-learning framework for few-shot multitask learning," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8490–8522, 2022.

[113] R. Anil, G. Pereyra, A. T. Passos, R. Ormandi, G. Dahl, and G. Hinton, "Large scale distributed neural network training through online distillation," in *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[114] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "Federated knowledge distillation," *arXiv preprint arXiv:2011.02367*, 2020.

[115] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature Communications*, vol. 13, no. 1, p. 2032, 2022.

[116] I. Shiri *et al.*, "Decentralized distributed multi-institutional pet image segmentation using a federated deep learning framework," *Clinical Nuclear Medicine*, vol. 47, no. 7, 2022.

[117] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[118] H. Gao, M. T. Thai, and J. Wu, "When decentralized optimization meets federated learning," *IEEE Network*, vol. In press, pp. 1–7, 2023.

[119] Y. Shi *et al.*, "Towards more suitable personalization in federated learning via decentralized partial model training," *arXiv preprint arXiv:2305.15157*, 2023.

[120] M. Duan *et al.*, "FedGroup: Efficient federated learning via decomposed similarity-based clustering," in *IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*, 2021, pp. 228–237.

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

[122] A. Schwab and J. Lunze, "A distributed algorithm to maintain a proximity communication network among mobile agents using the delaunay triangulation," *European Journal of Control*, vol. 60, pp. 125–134, 2021.

[123] Y. Ye, H. Chen, Z. Ma, and M. Xiao, "Decentralized consensus optimization based on parallel random walk," *IEEE Communications Letters*, vol. 24, no. 2, pp. 391–395, 2020.

[124] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Information Fusion*, vol. 90, pp. 148–173, 2023.

[125] J. Wang, X. Chang, R. J. Rodrìguez, and Y. Wang, "Assessing anonymous and selfish free-rider attacks in federated learning," in *IEEE Symposium on Computers and Communications*, 2022, pp. 1–6.

[126] H. Wang, L. Muñoz González, D. Eklund, and S. Raza, "Non-IID data re-balancing at IoT edge with peer-to-peer federated learning for anomaly detection," in *14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '21. Association for Computing Machinery, 2021, p. 153–163.

[127] X. Ma and D. Xu, "Torr: A lightweight blockchain for decentralized federated learning," *IEEE Internet of Things Journal*, vol. In press, pp. 1–1, 2023.

[128] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of Biomedical Informatics*, vol. 99, p. 103291, 2019.

[129] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[130] Z. Chen, D. Li, J. Zhu, and S. Zhang, "DACFL: Dynamic average consensus-based federated learning in decentralized sensors network," *Sensors*, vol. 22, no. 9, 2022.

[131] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2023.

[132] Y. Yuan *et al.*, "Decefl: a principled fully decentralized federated learning framework," *National Science Open*, vol. 2, no. 1, p. 20220043, 2023.

[133] S. Minsker and N. Strawn, "Distributed statistical estimation and rates of convergence in normal approximation," *arXiv preprint arXiv:1704.02658*, 2017.

[134] X. Mao, K. Yuan, Y. Hu, Y. Gu, A. H. Sayed, and W. Yin, "Walkman: A communication-efficient random-walk algorithm for decentralized optimization," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2513–2528, 2020.

[135] X. Mao, Y. Gu, and W. Yin, "Walk proximal gradient: An energy-efficient algorithm for consensus optimization," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2048–2060, 2019.

[136] D. C. Nguyen, S. Hosseinalipour, D. J. Love, P. N. Pathirana, and C. G. Brinton, "Latency optimization for blockchain-empowered federated learning in multi-server edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3373–3390, 2022.

[137] S. Zhu and B. Chen, "Quantized consensus by the ADMM: Probabilistic versus deterministic quantizers," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1700–1713, 2016.

[138] S. K. Singh, L. T. Yang, and J. H. Park, "Fusionfedblock: Fusion of blockchain and federated learning to preserve privacy in industry 5.0," *Information Fusion*, vol. 90, pp. 233–240, 2023.

[139] I. A. Ridhawi, S. Otoum, and M. Aloqaily, "Decentralized zero-trust framework for digital twin-based 6g," *arXiv preprint arXiv:2302.03107*, 2023.

[140] 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 *31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Curran Associates Inc., 2017, p. 5336–5346.

[141] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[142] M. H. Rehman, A. M. Dirir, K. Salah, E. Damiani, and D. Svetinovic, "TrustFed: A framework for fair and trustworthy cross-device federated learning in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8485–8494, 2021.

[143] Google, "Tensorflow federated," TensorFlow, 2022. [Online]. Available: https://www.tensorflow.org/federated

[144] A. Ziller *et al.*, *PySyft: A Library for Easy Federated Learning*. Springer International Publishing, 2021, pp. 111–139.

[145] Y. Xie *et al.*, "FederatedScope: A flexible federated learning platform for heterogeneity," *arXiv preprint arXiv:2204.05011*, 2022.

[146] C. He *et al.*, "FedML: A research library and benchmark for federated machine learning," *Advances in Neural Information Processing Systems, Best Paper Award at Federate Learning Workshop*, 2020.

[147] S. Caldas *et al.*, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[148] 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.

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

[150] M. Ghanem, F. Dawoud, H. Gamal, E. Soliman, H. Sharara, and T. El-Batt, "FLoBC: A decentralized blockchain-based federated learning framework," *arXiv preprint arXiv:2112.11873*, 2021.

[151] J. Li *et al.*, "Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2401–2415, 2022.

[152] 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.

[153] E. T. Martínez Beltrán *et al.*, "Fedstellar: A Platform for Decentralized Federated Learning," *arXiv preprint arXiv:2306.09750*, 2023.

[154] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[155] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," 2009. [Online]. Available: https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf

[156] D. Gurung, S. R. Pokhrel, and G. Li, "Decentralized quantum federated learning for metaverse: Analysis, design and implementation," *arXiv preprint arXiv:2306.11297*, 2023.

[157] X. Zhou *et al.*, "Decentralized p2p federated learning for privacy-preserving and resilient mobile robotic systems," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 82–89, 2023.

[158] Q. Liu *et al.*, "Asynchronous decentralized federated learning for collaborative fault diagnosis of pv stations," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1680–1696, 2022.

[159] M. Nakıp, B. C. Gül, and E. Gelenbe, "Decentralized online federated g-network learning for lightweight intrusion detection," *arXiv preprint arXiv:2306.13029*, 2023.

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

[161] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU collaborative research database, a freely available multi-center database for critical care research," *Scientific Data*, vol. 5, no. 1, p. 180178, 2018.

[162] S. Lu, Y. Zhang, and Y. Wang, "Decentralized federated learning for electronic health records," in *54th Annual Conference on Information Sciences and Systems*, 2020, pp. 1–5.

[163] H. Elayan, M. Aloqaily, and M. Guizani, "Deep federated learning for IoT-based decentralized healthcare systems," in *International Wireless Communications and Mobile Computing*, 2021, pp. 105–109.

[164] S. Warnat-Herresthal *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[165] N. C. for Biotechnology Information, "Geo dataset browser," www.ncbi.nlm.nih.gov, 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/sites/GDSbrowser/

[166] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, *ChestX-ray: Hospital-Scale Chest X-ray Database and Benchmarks on Weakly Supervised Classification and Localization of Common Thorax Diseases*. Springer International Publishing, 2019, pp. 369–392.

[167] M. M. Salim and J. H. Park, "Federated learning-based secure electronic health record sharing scheme in medical informatics," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 617–624, 2023.

[168] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[169] B. H. Menze *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[170] K. H. Zou, A. Liu, A. I. Bandos, L. Ohno-Machado, and H. E. Rockette, *Statistical Evaluation of Diagnostic Performance*, 1st ed. CRC, 2011.

[171] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "COVID-CT-Dataset: A CT Scan Dataset about COVID-19," *arXiv preprint arXiv:2003.13865*, 2020.

[172] A. Dos, "Predicting human eye diseases," Kaggle.com, 2021. [Online]. Available: https://www.kaggle.com/datasets/fabianogalaxy/dataset-with-catarats-images

[173] M. Jennings, "Skin-cancer-identification," GitHub, 2021. [Online]. Available: https://github.com/Matt-Jennings-GitHub/Skin-Cancer-Identification

[174] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[175] M. Pennisi *et al.*, "Decentralized distributed learning with privacy-preserving data synthesis," *arXiv preprint arXiv:2206.10048*, 2022.

[176] S. Candemir *et al.*, "Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration," *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, pp. 577–590, 2014.

[177] S. Jaeger *et al.*, "Automatic tuberculosis screening using chest radiographs," *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, pp. 233–245, 2014.

[178] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, no. 1, p. 180161, 2018.

[179] Y. Tian, S. Wang, J. Xiong, R. Bi, Z. Zhou, and M. Z. A. Bhuiyan, "Robust and privacy-preserving decentralized deep federated learning training: Focusing on digital healthcare applications," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. In press, pp. 1–12, 2023.

[180] S. Savazzi, "Federated learning: mmwave mimo radar dataset for testing," *IEEE Dataport*, 2020.

[181] 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, 2021.

[182] Y. Meidan *et al.*, "N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[183] Z. Lian and C. Su, "Decentralized federated learning for Internet of Things anomaly detection," in *ACM on Asia Conference on Computer and Communications Security*. Association for Computing Machinery, 2022, p. 1249–1251.

[184] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic," 2020.

[185] M. Abdel-Basset, N. Moustafa, and H. Hawash, "Privacy-preserved cyberattack detection in Industrial Edge of Things (IEoT): A blockchain-orchestrated federated learning approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7920–7934, 2022.

[186] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[187] R. Damasevicius *et al.*, "LITNET-2020: An annotated real-world network flow dataset for network intrusion detection," *Electronics*, vol. 9, no. 5, 2020.

[188] H. Ochiai, Y. Sun, Q. Jin, N. Wongwiwatchai, and H. Esaki, "Wireless ad hoc federated learning: A fully distributed cooperative machine learning," *arXiv preprint arXiv:2205.11779*, 2022.

[189] 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, 2023.

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

[191] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2276–2288, 2021.

[192] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

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

[194] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[195] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[196] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, p. 1082–1090.

[197] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, 2015.

[198] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-Scale attributed node embedding," *Journal of Complex Networks*, vol. 9, no. 2, 2021.

[199] Q. Pan, Y. Zhu, and L. Chu, "Lumos: Heterogeneity-aware federated graph learning over decentralized devices," *arXiv preprint arXiv:2303.00492*, 2023.

[200] B. Rozemberczki and R. Sarkar, "Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models," in *Association for Computing Machinery*, 2020, p. 1325–1334.

[201] I. Arapakis, P. Papadopoulos, K. Katevas, and D. Perino, "P4l: Privacy preserving peer-to-peer learning for infrastructureless setups," *arXiv preprint arXiv:2302.13438*, 2023.

[202] "Avito context ad clicks," Kaggle.com, 2016. [Online]. Available: https://www.kaggle.com/c/avito-context-ad-clicks

[203] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, p. 142–150.

[204] E. Jeong and M. Kountouris, "Personalized decentralized federated learning with knowledge distillation," *arXiv preprint arXiv:2302.12156*, 2023.

[205] "Iot device identification," Kaggle.com, 2021. [Online]. Available: https://www.kaggle.com/datasets/fanbyprinciple/iot-device-identification

[206] A. Salama, A. Stergioulis, A. M. Hayajneh, S. A. R. Zaidi, D. McLernon, and I. Robertson, "Decentralized federated learning over slotted aloha wireless mesh networking," *IEEE Access*, vol. 11, pp. 18 326–18 342, 2023.

[207] W. Koch, "On digital ethics for artificial intelligence in hybrid military operations," 2021. [Online]. Available: https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/STO-MP-IST-190/MP-IST-190-22.pdf

[208] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Federated learning-based cognitive detection of jamming attack in flying ad-hoc network," *IEEE Access*, vol. 8, pp. 4338–4350, 2020.

[209] O. Puñal, C. Pereira, A. Aguiar, and J. Gross, "CRAWDAD dataset uportorwthaachen/vanetjamming2014 (v. 2014-05-12)," 2014.

[210] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Springer Berlin Heidelberg, 2010, pp. 15–34.

[211] P. K. Sharma, J. H. Park, and K. Cho, "Blockchain and federated learning-based distributed computing defence framework for sustainable society," *Sustainable Cities and Society*, vol. 59, p. 102220, 2020.

[212] Y. Qu *et al.*, "Decentralized federated learning for UAV networks: Architecture, challenges, and opportunities," *IEEE Network*, vol. 35, no. 6, pp. 156–162, 2021.

[213] C. Zhu, X. Zhu, J. Ren, and T. Qin, "Blockchain-enabled federated learning for UAV edge computing network: Issues and solutions," *IEEE Access*, vol. 10, pp. 56 591–56 610, 2022.

[214] M. S. Al-Abiad, M. Hassan, and M. Hossain, "Decentralized model dissemination empowered federated learning in mmwave aerial-terrestrial integrated networks," *arXiv preprint arXiv:2303.00032*, 2023.

[215] A. Giannopoulos, P. Gkonis, P. Bithas, N. Nomikos, G. Ntroulias, and P. Trakadas, "Federated learning for maritime environments:use cases, experimental results, and open issues," *TechRxiv preprint techrxiv.22133549.v1*, 2023.

[216] C. Reiser, "creiser/drone-detection," GitHub, 2022. [Online]. Available: https://github.com/creiser/drone-detection

[217] R. Fisher, "Cvonline: Image databases," 2019. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm

[218] E. T. Martínez Beltrán, M. Quiles Pérez, S. López Bernal, G. Martínez Pérez, and A. Huertas Celdrán, "SAFECAR: A brain–computer interface and intelligent framework to detect drivers' distractions," *Expert Systems with Applications*, vol. 203, p. 117402, 2022.

[219] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[220] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[221] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[222] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[223] F. Bragato *et al.*, "Towards decentralized predictive quality of service in next-generation vehicular networks," *arXiv preprint arXiv:2302.11268*, 2023.

[224] X. Hu, R. Li, Y. Ning, K. Ota, and L. Wang, "A data sharing scheme based on federated learning in iov," *IEEE Transactions on Vehicular Technology*, vol. In press, pp. 1–13, 2023.