Revista Eletrônica de Iniciação Científica em Computação, 2026, 24:1
ISSN: 3085-8461 • doi: 10.5753/reic.2026.8089

Research Paper

Distributed Machine Learning on Edge Computing: A Survey of Challenges and Techniques

Luis Henrique Lopes Teodoro [Universidade Tecnológica Federal do Paraná | *luisteodoro@alunos.utfpr.edu.br*]
Daniel Fernando Pigatto [Universidade Tecnológica Federal do Paraná | *pigatto@utfpr.edu.br*]
Ana Cristina Barreiras Kochem Vendramin [Universidade Tecnológica Federal do Paraná | *criskochem@utfpr.edu.br*]
Juliana de Santi [Universidade Tecnológica Federal do Paraná | *jsanti@utfpr.edu.br*]

Departamento Acadêmico de Informática (DAINF), Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba, Paraná, 80230-000, Brazil.

Abstract. This survey examines the dynamic field of Distributed Machine Learning (DML) in the context of Edge Computing (EC). It analyzes prevailing architectures, implementation, identifies critical challenges, and synthesizes proposed mitigation techniques within resource-constrained edge environments. The study delineates Edge-Only and Cloud-Edge architectures, as well as Federated Learning (FL) implementation, highlighting their characteristics and suitability for diverse applications within DML. It thoroughly examines fundamental challenges, including resource limitations, energy efficiency, communication overhead, data privacy, failure resilience, and data heterogeneity. By exploring recent strategies, the survey provides a comprehensive overview of current solutions and promising future research directions for optimizing DML deployment at the network edge.

Keywords: Distributed Machine Learning, Federated Learning, Edge Computing, Edge Intelligence, Resource-Constrained Environments

Received: 21 April 2026 • **Accepted:** 08 July 2026 • **Published:** 10 July 2026

1 Introduction

The evolution of Artificial Intelligence (AI) and Machine Learning (ML), coupled with the proliferation of Internet of Things (IoT) devices, has led to an exponential increase in data generation and model complexity. This has resulted in large volumes of distributed data and has exposed scalability, privacy, and communication limitations of traditional centralized ML paradigms.

In response to these challenges, Distributed Machine Learning (DML) has emerged as a critical paradigm that enables scalable, efficient, and privacy-aware training by distributing ML tasks across multiple computational nodes [Ramírez-Gordillo *et al*., 2026].

By leveraging decentralized computational resources, DML supports collaborative model training across heterogeneous infrastructures, particularly within Edge Computing (EC) environments. This paradigm shift, often referred to as edge intelligence, enables the execution of ML algorithms closer to data sources, reducing dependence on cloud-centric infrastructures [Filho *et al*., 2022].

Processing data locally helps mitigate limitations associated with cloud-based approaches, including high latency, bandwidth constraints, and privacy concerns. In addition, local processing enables real-time decision-making and enhanced data privacy, which are crucial for applications such as industrial automation, surveillance, intelligent transportation systems, and healthcare [Oliveira *et al*., 2024].

However, deploying DML in EC environments remains challenging due to limited computational, memory, and energy resources available on edge devices, which constrain data processing and the execution of ML models [Oliveira *et al*., 2024; Cajas Ordóñez *et al*., 2025; Rajassekharan, 2025].

Other critical challenges include ensuring energy efficiency without compromising model accuracy, managing communication overhead, addressing data privacy and security concerns, handling device failures, and coping with the heterogeneity and variable quality of data generated at the edge [Xia *et al*., 2021; Oliveira *et al*., 2024]. These challenges highlight the need for a comprehensive understanding of existing approaches and emerging techniques to enable efficient and reliable distributed learning in edge environments.

This survey offers a comprehensive analysis of the challenges and techniques in DML on EC, examining architectures, implementations, and strategies that enable the efficient and effective operation of ML models within resource-constrained edge environments. Unlike existing surveys that focus predominantly on Federated Learning (FL) [Xia *et al*., 2021] or address edge ML deployment without systematically linking architectures to mitigation strategies [Rajassekharan, 2025; Cajas Ordóñez *et al*., 2025], this work provides a structured analysis that (i) covers Edge-Only, Cloud-Edge, and FL architectural variants within a unified DML perspective; (ii) explicitly maps identified challenges to concrete mitigation techniques; and (iii) offers a qualitative comparative assessment of those techniques across multiple challenge dimensions. This positioning bridges the gap between taxonomy-focused surveys and technique-oriented reviews, providing a consolidated reference for researchers and practitioners working on DML deployment at the edge.

This paper is organized as follows. Section *Literature Review* presents a review of the relevant literature. Section *Research Methodology* describes the survey methodology. Section *Architectures, Implementation and Challenges* discusses architectures, implementation aspects, and challenges of DML in EC environments. Section *Techniques and Trends*

reviews recent techniques proposed to address these challenges. Finally, Section *Conclusion* summarizes the main findings and highlights directions for future research.

2 Literature Review

Building on the foundational understanding of DML in EC, recent research has extensively explored strategies for deploying ML models in resource-constrained distributed environments.

Filho *et al*. [2022] presents a systematic literature review of techniques for deploying ML/Deep Learning (DL) on edge devices, categorizing approaches into nine groups, including FL, model partitioning, edge-only, and model compression. For instance, Edge-Only approaches execute computations entirely on edge devices, reducing communication overhead and improving privacy by minimizing data transfer. Model compression, in turn, enables the deployment of neural networks on resource-constrained devices by reducing model size and complexity. Despite these advances, the authors highlight that the seamless adaptation of pre-trained models to edge environments remains an open challenge.

Cajas Ordóñez *et al*. [2025] provides a comprehensive overview of intelligent edge ML, covering optimization strategies such as model compression, FL, and edge-oriented ML Operations (MLOps) frameworks. The work also highlights applications across multiple domains and identifies open research challenges, including multimodal deployment, concept drift adaptation, and sustainable edge AI systems.

Focusing on practical applications, Rajassekharan [2025] surveys ML techniques for edge environments across domains such as IoT, smart manufacturing, autonomous systems, and healthcare. The study discusses deployment challenges and optimization techniques that support real-world EC implementations.

From a distributed learning perspective, Tu *et al*. [2025] provides an overview of DML in edge computing, covering distributed architectures, parallelism patterns, communication mechanisms, and model aggregation strategies. It also analyzes issues such as communication overhead, data heterogeneity, privacy preservation, and resource management.

Complementarily, Ramírez-Gordillo *et al*. [2026] proposes a unified taxonomy and formal foundations to systematize the fragmented research landscape. The study categorizes DML approaches according to distribution topologies, aggregation strategies, application domains, and system-level challenges, while also formalizing key concepts related to model and data partitioning. Additionally, the authors highlight FL as a hybrid and privacy-preserving extension of DML and identify open research challenges that shape emerging distributed learning paradigms in decentralized environments.

Among DML approaches, FL has emerged as one of the most prominent solutions for EC due to its ability to preserve data privacy. Wu *et al*. [2024] emphasize FL's role in enabling collaborative ML tasks among decentralized devices while adhering to privacy regulations, such as General Data Protection Regulation (GDPR) [European Union, 2016], ensuring that data remains locally stored at the edge.

This privacy-preserving capability is further supported by Guo *et al*. [2022], who note that distributed learning frameworks enable mobile devices to protect user privacy by not uploading full information to centralized points, thereby simultaneously reducing communication overhead and offering a flexible system design through multi-module cooperation. Likewise, Wang *et al*. [2022] highlight FL's dual benefit of preventing personal privacy exposure while efficiently utilizing computational resources available at the network edge.

Further analyses by Abreha *et al*. [2022] delve into the adaptation of FL in EC environments, particularly how Deep Neural Networks (DNN) are configured, often involving data pre-processing near edge devices or deploying distributed DNN models. Their analysis extends to the inherent challenges of FL implementation in edge networks, including communication efficiency, managing heterogeneity, security, privacy preservation, and meeting service pricing requirements. Complementarily, Ma *et al*. [2021a] reinforce that the shift of computation to the network edge, enabled by EC, streamlines local data processing and actively promotes the adoption of FL.

Despite these advancements and the promising strategies for bringing ML/DL to the edge, the effective deployment of DML, particularly FL, in EC environments, continues to face substantial challenges related to computational constraints, energy efficiency, communication efficiency, data privacy and security, and data quality, which are discussed in Section 4.

These challenges highlight the complexity of DML in EC and reveal important opportunities for improving distributed learning in resource-constrained and heterogeneous environments [Cajas Ordóñez *et al*., 2025; Rajassekharan, 2025; Ramírez-Gordillo *et al*., 2026]. However, existing approaches still struggle to jointly address resource limitations, communication overhead, and data heterogeneity while maintaining scalability and efficiency, motivating the development of more adaptive and effective edge learning mechanisms.

3 Research Methodology

This survey was conducted following a transparent and auditable methodology aimed at minimizing bias and ensuring comprehensive coverage of the literature on DML in EC. The review process consisted of two main phases: *Planning the Review* and *Conducting the Review*. The planning phase defined the scope, research questions, and search strategy, while the conducting phase included the execution of the search, study selection, data extraction, and synthesis of relevant studies.

3.1 Planning the Review

The planning phase established the foundation for the entire review, ensuring clarity, focus, and a structured approach to address the research objectives. Specifically, this phase involved defining the research questions, establishing inclusion and exclusion criteria, and developing a comprehensive search strategy to identify relevant studies.

The research questions guided the study, aiding in the selection of articles and the development of survey sections. The inclusion and exclusion criteria ensured that only pertinent and high-quality studies were considered, while the search strategy aimed to systematically identify a broad range of literature from various academic databases, capturing diverse perspectives and recent advancements.

3.1.1 Formulation of Research Questions

Formulating research questions is a foundational and critical step in any literature review, as these questions serve to guide all subsequent stages of the review process. They are instrumental in defining the scope, directing the search for relevant literature, and ultimately in determining, interpreting, and evaluating the research findings. The primary objective in formulating these questions is to ensure they can be comprehensively answered through the rigorous review of primary works retrieved from the literature.

To establish a clear direction and focus for this survey, a thorough process was undertaken to define the research questions. This process involved identifying all pertinent subjects within DML in EC and considering emerging areas not yet extensively covered by existing surveys. The resulting research questions were pivotal in guiding the search, selection, and synthesis processes, and ultimately defined this review paper's objective: to explore the current state of DML in EC, encompassing its inherent challenges and proposed techniques.

To systematically address this objective, the survey is guided by three research questions (RQ): (RQ-1) What are the current architectures utilized in DML within EC environments? (RQ-2) What are the primary technical challenges and limitations encountered when implementing DML models? and (RQ-3) What are the emerging techniques proposed to mitigate these challenges in EC environments?

These precisely defined RQ's were crucial in structuring the entire survey. Specifically, the section 4 was meticulously crafted to provide comprehensive answers to RQ-1 and RQ-2, offering an in-depth analysis of the prevailing architectures, implementation, and the complex limitations faced in DML on EC. Concurrently, Section 5 was wholly dedicated to addressing RQ-3, detailing the various strategies and algorithms developed to overcome the identified challenges. This direct mapping ensures that the survey remains focused, coherent, and effectively addresses its stated objectives.

3.1.2 Search Strategy

The search strategy was formulated using the Boolean formula: 'A' AND 'B'. Preposition 'A' comprised an OR combination of terms related to DML, specifically "distributed machine learning", "federated learning", or "decentralized machine learning". Conversely, preposition 'B' consisted of an OR combination of terms pertinent to EC, such as "edge computing", "fog computing", or "edge intelligence" resulting in the final query: *('distributed machine learning' OR 'federated learning' OR 'decentralized machine learning') AND ('edge computing' OR 'fog computing' OR 'edge intelligence')*.

The search was conducted across four academic databases: IEEE Xplore, ACM Digital Library, Scopus, and Web of Science. The initial query returned 77 candidate studies. After applying the inclusion and exclusion criteria, retaining only peer-reviewed journal articles and conference papers in English, published between 2021 and 2026, that directly addressed DML in EC and presented empirical findings or novel theoretical contributions, 40 studies were selected for full-text review. Of these, 22 were ultimately included in the survey as primary references directly informing the analysis of architectures, challenges, and mitigation techniques. Reviews, editorials, books, and articles that did not directly address DML or EC were excluded. Additionally, studies other than surveys were excluded if they did not present empirical findings or novel theoretical contributions.

4 Architectures, Implementation and Challenges

DML in EC employs architectural approaches to process data closer to its source, mitigating latency, bandwidth, and privacy challenges. These architectures define how computational tasks are allocated, coordinated, and executed across the edge–cloud continuum.

This section delves into two prominent physical architectures for implementing DML in edge environments: the **Edge-Only** architecture, characterized by computations occurring exclusively on local edge devices, and the **Cloud-Edge** architecture, which distributes tasks between edge and cloud resources to optimize for latency, bandwidth, and computational demands.

In addition to these architectural models, this section introduces **Federated Learning (FL)**, a widely adopted distributed learning approach that can operate within both architectures. FL enables collaborative model training across decentralized devices without transferring raw data to a central location, thereby reducing privacy risks and communication overhead in edge deployments.

4.1 Edge-Only

The Edge-Only architecture is a paradigm in which ML computations, including training and inference, are executed exclusively on local edge devices or within the edge network. This approach brings computation closer to the data source, addressing limitations of cloud-centric models, such as high communication latency, bandwidth constraints, and privacy risks associated with transmitting sensitive data to centralized cloud servers [Oliveira *et al*., 2024; Rajassekharan, 2025].

Local data processing enable real-time decision-making while enhancing data privacy, making them well suited for applications such as industrial automation, surveillance, and intelligent transportation systems. In this setting, the cloud server, when present, plays a limited role, typically restricted to data storage or occasional high-level aggregation without direct access to raw data. This design keeps sensitive information localized, reduces dependence on external networks, and minimizes communication latency.

However, implementing this architecture presents several significant challenges [Rajassekharan, 2025; Filho *et al*., 2022]. One of the main obstacles is the **limited** computational power, memory, and energy **resources** available on edge devices. These constraints complicate the efficient execution of complex ML algorithms and may restrict the availability of large datasets or the processing capacity required for sophisticated model training.

Energy efficiency is another critical concern. Edge devices often operate under strict power limitations, creating an inherent trade-off between the complexity of processing tasks and the quality of the decisions or inferences made.

In addition, edge environments are characterized by significant **device heterogeneity**. Nodes may differ in terms of computational power, memory, and energy resources, com-

plicating the design of learning models that can generalize effectively across diverse devices and data sources.

Moreover, the intrinsic possibility of edge **device failures** demands the development of distributed systems that can robustly overcome such situations, maintaining stability and reliability even if devices or connectivity fail.

Given its focus on on-device intelligence and minimized cloud reliance, the Edge-Only architecture serves as a foundational topology for specific DML methodologies. For instance, its principles are extensively applied in Decentralized FL (DFL), which will be further explored in the subsection 4.3. This approach enforces direct P2P collaboration among client nodes to train a shared model, thereby enhancing robustness and autonomy by eliminating the single point of failure inherent in central servers [Wu *et al*., 2024] and offering a practical, consolidated approach for on-device learning.

4.2 Cloud-Edge

The Cloud-Edge architecture integrates edge devices with centralized cloud systems, forming a hybrid infrastructure [Duan *et al*., 2023]. This paradigm strategically distributes ML tasks across various tiers, leveraging the centralized cloud for coordination and extensive resources while enabling localized processing at the edge. This hybrid approach aims to mitigate limitations of purely edge-only or cloud-centric models, such as high latency, bandwidth constraints, and privacy concerns [Abreha *et al*., 2022; Duan *et al*., 2023].

However, the effective deployment of distributed ML in cloud-edge environments faces significant challenges stemming from its hybrid and distributed nature [Xia *et al*., 2021; Duan *et al*., 2023; Rahmani *et al*., 2024]. Although this architecture aims to reduce **latency** by processing data closer to its source, communication between edge and cloud components, particularly when handling large datasets, can still be constrained by **limited bandwidth**, thereby affecting the performance of real-time applications [Abreha *et al*., 2022].

Another important challenge concerns **data privacy and security**. Implementing robust privacy and security measures across this distributed, hybrid infrastructure is complex, introducing multiple potential vulnerabilities to malicious attacks or data leakage [Abreha *et al*., 2022].

Within this architecture, multi-tiered networks are typically employed, featuring a central cloud-based server that coordinates the hierarchy by aggregating updates and distributing improved models to lower tiers [Duan *et al*., 2023]. This structure enables a balanced distribution of computational tasks and data flow, supporting both localized processing at edge nodes and global model consistency.

Such an arrangement naturally supports multi-tier FL, in which learning tasks are strategically distributed, allowing edge nodes to perform local processing and intermediate aggregation before transmitting updates to a central FL server. Next section further discusses how these hierarchical configurations enable collaborative model training, particularly in approaches such as Hierarchical FL, where a central orchestrator coordinates clients across multiple tiers.

4.3 Federated Learning

Federated Learning is a DML implementation that facilitates collaborative ML model training across numerous distributed client devices without centralizing raw data. This paradigm emerged to address critical concerns, primarily data privacy by keeping sensitive information localized, and to significantly reduce bandwidth requirements through minimized data movement [Guo *et al*., 2022; Abreha *et al*., 2022; Wu *et al*., 2024; Hasan *et al*., 2024; Riedel *et al*., 2024].

Beyond privacy preservation, FL supports decentralized learning while complying with regulations such as GDPR. It can also reduce training and inference times and facilitate collaborative model development through large-scale data "crowdsourcing" [Xia *et al*., 2021; Guo *et al*., 2022; Wu *et al*., 2024]. This section explores the core principles of FL, its implementation variants, and the challenges within DML.

At its core, FL involves entities collaboratively training a shared global model [Abreha *et al*., 2022]. This is achieved by iteratively updating local models using their own private data and subsequently sending only model parameters or gradients to a server for aggregation, rather than the raw data itself [Rahmani *et al*., 2024]. This process protects privacy and decreases communication overhead [Guo *et al*., 2022].

Although FL often involves a central server orchestrating the training process, novel approaches featuring fully decentralized, P2P interactions have also emerged [Duan *et al*., 2023; Wu *et al*., 2024]. These decentralized configurations eliminate the single point of failure and can further enhance privacy by removing reliance on a central aggregator [Duan *et al*., 2023; Wu *et al*., 2024].

As a result, FL can be implemented through various topological configurations, including centralized setups with a cloud server, decentralized ones dominated by edge devices, and hierarchical configurations, each suited to different application requirements [Wu *et al*., 2024].

4.3.1 Centralized FL

Centralized Federated Learning (CFL) is the most common FL implementation, characterized by a central server that orchestrates the entire training process [Wu *et al*., 2024; Duan *et al*., 2023]. It typically forms a star topology, with the central server at its core and client devices acting as the nodes.

Within this framework, the software on each participating FL device comprises three core components: an application process, an example store, and an FL runtime. The application collects local data and provides it to the FL runtime via an Application Programming Interface (API). This data is then stored in an "example store" for subsequent model training and evaluation [Riedel *et al*., 2024].

The FL server, often implemented as a cloud-based distributed service, plays a crucial role in managing communication protocols. It takes into account factors such as security, device connectivity, and availability. In each training round, the server selects a subset of clients from a potentially large pool, specifies the computations these clients must perform according to an FL plan, and orchestrates the aggregation and distribution of the updated models [Duan *et al*., 2023].

The CFL implementation involves a series of sequential steps: **initialization**, **local training**, **model update transmission**, **global model aggregation**, and **global model distribution** as illustrated in Figure 1. This iterative process, shown in Figure 2, continues until the desired model performance is achieved or a predetermined number of communication

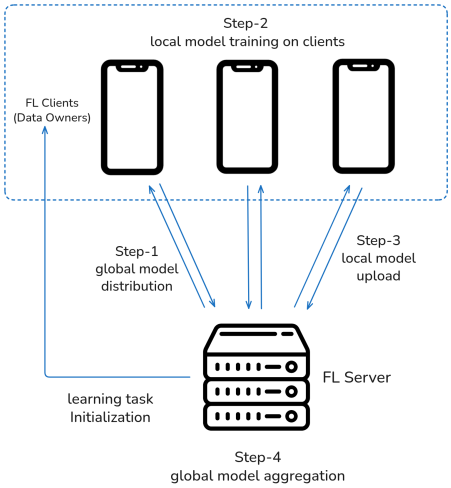

Figure 1. The system architecture of FL.

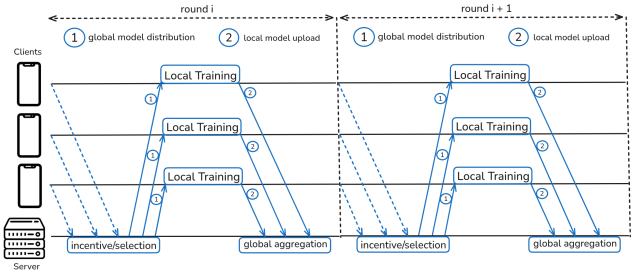

Figure 2. The server-client interactions in each round of the FL process.

rounds is completed [Filho *et al.*, 2022]. Each execution round, comprises the following steps:

1. **Initialization:** The process begins with an FL server determining an ML model to be trained and initializing it. This initial global model is then distributed to a selected subset of participating client devices [Duan *et al.*, 2023];
2. **Local Training:** Subsequently, upon receiving the global model, each selected client downloads the current model parameters and trains the model locally using its own private dataset. This training occurs entirely on the device, ensuring that raw data never leaves the local environment [Duan *et al.*, 2023];
3. **Model Update Transmission:** Following local training, only the updated model parameters or gradients are securely sent from the client devices back to the central server [Guo *et al.*, 2022]. These updates are often sent via encrypted communication [Wu *et al.*, 2024];
4. **Global Model Aggregation:** Once these updates are received, the FL server aggregates the locally trained model parameters from multiple clients to generate an improved version of the global model [Duan *et al.*, 2023];
5. **Global Model Distribution:** Finally, the newly aggregated global model is then distributed back to the client devices, either to the same subset or a new one, for the next round of local training [Riedel *et al.*, 2024].

However, challenges associated with cloud-edge architectures arise, notably high communication costs between edge devices and the central server, and the potential for unreliable device connectivity, which underscore the need for robust FL protocols and architectures [Wu *et al.*, 2024]. Additionally,

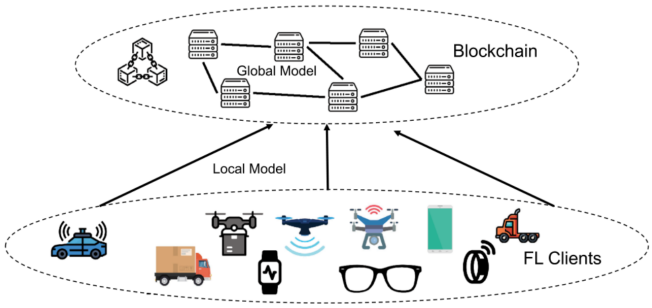

Figure 3. FL with blockchain as a distributed ledger to increase the availability of the global model. Reproduced from Wu *et al.* [2024] under the CC BY 4.0 license.

since each training round requires all devices to send their updates, some resource-constrained devices can significantly hinder the process. The heterogeneity of data distributions across edge nodes further poses significant hurdles for convergence and model performance [Ma *et al.*, 2021b; Brecko *et al.*, 2022].

4.3.2 *Decentralized FL*

DFL eliminates the need for a central server, enabling client nodes to collaborate directly through P2P interactions to train a shared model [Duan *et al.*, 2023; Wu *et al.*, 2024]. This direct collaboration inherently enhances robustness, providing greater fault tolerance and autonomy by removing the single point of failure that a central aggregator presents.

In DFL, nodes collaborate directly, with participating data owners synchronizing their local model training to achieve a global consensus model [Brecko *et al.*, 2022; Duan *et al.*, 2023]. This P2P approach removes the need for a central server: nodes either start with an initial model or exchange models among neighboring peers. After local training, clients directly share model parameters or gradients with adjacent devices [Duan *et al.*, 2023], and each participant fuses its own trained model with the received updates, iteratively converging toward a global consensus across the network.

To further enhance integrity and traceability within DFL framework, blockchain technology can be employed to record and validate local model updates, as illustrated in Fig. 3. This decentralized aggregation reduces the risk of single-point failures and distributes the responsibility of global model aggregation across the network [Ren and Lee, 2025].

Despite its advantages, the deployment of DFL presents several significant challenges. While general concerns, such as communication efficiency and device heterogeneity, are common, DFL's serverless implementation introduces unique complexities. A primary challenge, for instance, lies in achieving timely and stable convergence of local models to a reliable global consensus without a central coordinating entity. This is particularly challenging given the inherent device heterogeneity and non-independent and identically distributed (non-IID) data distributions, which can lead to local model divergence [Xia *et al.*, 2021; Brecko *et al.*, 2022].

Security and trust management also become more complex in fully decentralized environments. Although DFL eliminates vulnerabilities associated with centralized servers, malicious peers may still attempt to manipulate the training process or inject compromised model updates. Consequently,

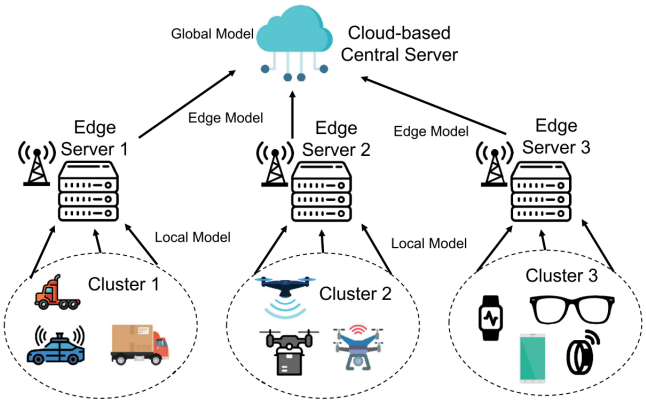

Figure 4. Hierarchical FL. Reproduced from Wu *et al*. [2024] under the CC BY 4.0 license.

robust mechanisms are required to ensure the integrity, reliability, and trustworthiness of P2P interactions [Duan *et al*., 2023; Filho *et al*., 2022].

4.3.3 Hierarchical FL

Hierarchical FL (HFL) employs a multi-tier aggregation structure for collaborative model training [Chen *et al*., 2024]. Edge devices train local models and send their updates to nearby edge servers, which perform intermediate aggregation before forwarding the results to a central server for global aggregation [Chen *et al*., 2024; Wu *et al*., 2024]. Typically, resource-constrained devices communicate with more powerful edge servers, such as those deployed at cellular base stations, as illustrated in Figure 4 [Wu *et al*., 2024].

By aggregating updates closer to the data source, HFL reduces communication overhead, alleviates bottlenecks between edge devices and the cloud, and improves scalability, with time costs growing logarithmically rather than linearly [Wu *et al*., 2024]. However, unlike fully decentralized approaches, HFL still relies on a central server for the final aggregation step [Chen *et al*., 2024].

4.3.4 Applications of FL

The broad applicability of FL arises from its fundamental ability to maintain sensitive data locally and to minimize bandwidth consumption by only transmitting model parameters or gradients [Xia *et al*., 2021; Hasan *et al*., 2024].

One significant area where FL demonstrates utility is in **industrial automation**, particularly in the context of Industry 5.0 and IoT applications, where it facilitates intelligent control and predictive maintenance by allowing models to learn from sensor data across numerous industrial devices without centralizing proprietary information [Brecko *et al*., 2022].

In **surveillance**, FL enables privacy-preserving analysis of data from sources such as video cameras used in transport to monitor traffic in smart cities, supporting anomaly detection and security monitoring while ensuring raw footage remains local [Brecko *et al*., 2022].

Furthermore, FL is increasingly applied in **vehicular networks and intelligent transportation systems**, which falls under the broader category of transportation applications [Brecko *et al*., 2022]. For instance, an autonomous vehicle service provider can leverage FL to jointly train models for tasks such as Traffic Sign Recognition, Blind Spot Monitoring, and Pedestrian Detection without requiring consumers to share their raw local training data [Zhang *et al*., 2022]. This extends to end-to-end autonomous driving scenarios [Yu and Li, 2021]. FL supports collaborative learning among connected vehicles for traffic prediction, route optimization, and autonomous driving functionalities, all while safeguarding the privacy of individual vehicle data [Hasan *et al*., 2024].

Within the **healthcare sector**, FL effectively addresses stringent privacy regulations by enabling medical institutions to collaboratively train robust AI models on patient data without the direct sharing of sensitive personal health information, leading to advancements in diagnostics and personalized treatments [Wu *et al*., 2024].

Beyond these specific sectors, FL is widely implemented in scenarios involving **mobile phones and various IoT devices**, also listed as a general application area [Brecko *et al*., 2022]. In these contexts, it powers on-device intelligence for applications such as predictive text, voice assistance, and personalized recommendations, while simultaneously upholding user data privacy and reducing reliance on constant cloud connectivity [Hasan *et al*., 2024].

The inherent flexibility of FL, which encompasses both centralized, decentralized and hierarchical architectural configurations, permits tailored deployments to align with specific application requirements, thereby ensuring that advanced ML capabilities are brought closer to the data sources efficiently and securely [Wu *et al*., 2024].

4.4 Architectural Trade-offs and Deployment Considerations

No single DML architecture is universally optimal. Edge-Only architectures are better suited for latency-sensitive and privacy-critical applications, but are constrained by the limited computational and energy resources available at edge devices. Cloud-Edge architectures provide greater scalability and support for computationally intensive workloads through cloud assistance; however, they remain dependent on network connectivity and may introduce additional communication delays. Similarly, centralized FL simplifies coordination and model aggregation but can create bottlenecks and single points of failure. In contrast, decentralized FL improves resilience and eliminates centralized dependencies at the cost of increased coordination complexity. Hierarchical FL represents a compromise between these approaches by reducing communication overhead and improving scalability while maintaining partial centralized coordination. Consequently, the choice of architecture should be guided by application requirements, including latency constraints, privacy demands, computational resources, and scalability needs.

5 Techniques and Trends

Advancements in DML on EC represent a complex and rapidly evolving field, necessitating a careful balance of computational efficiency, data privacy, and model accuracy within resource-constrained edge environments. While FL has become the dominant paradigm in recent literature due to its privacy-preserving properties, other DML strategies, including model compression, task offloading, and decentralized optimization, also play a significant role in enabling efficient edge intelligence, as discussed in the context of Edge-Only

and Cloud-Edge architectures in Section 4. The techniques surveyed here address challenges primarily within FL-based implementations, given their prevalence in the literature, but their underlying principles extend to broader DML deployments. Besides reviewing existing solutions, this section highlights the main gaps and open research issues in the field. Table 1 provides an overview of the surveyed techniques and the challenges they address.

For clarity, the discussion is organized into three major challenge categories: (i) resource limitations, energy efficiency, and communication overhead; (ii) data quality and heterogeneity; and (iii) data privacy and security. Each category reviews representative techniques and concludes with a discussion of the remaining open challenges.

5.1 Resource Limitations, Energy Efficiency, and Communication Overhead

Addressing resource limitations, enhancing energy efficiency, and reducing communication overhead are critical interconnected challenges in FL implementation. Existing solutions tackle these issues from different perspectives, ranging from communication and computation optimization to dynamic resource management strategies. The following subsections first discuss techniques that reduce communication and computation costs, then present approaches based on adaptive training and resource allocation, and finally summarize the remaining open challenges.

5.1.1 Optimizing Communication and Computation for Resource and Energy Efficiency

A primary strategy involves minimizing the data transmitted and processed on resource-constrained edge devices, which inherently contributes to both resource and energy efficiency. Techniques like **FedLFP** exemplify this by transmitting low-dimensional prototypes instead of full model parameters, thereby minimizing resource usage [Sun *et al*., 2025] and implicitly reducing the volume of transmitted data. Similarly, the **Robust and Communication-Efficient Federated Learning (RCFL)** framework significantly reduces data transmission requirements by employing a privacy encoding mechanism that represents gradient elements with a single bit. This reduction in communication overhead directly translates to lower energy consumption, allowing for more frequent model updates without overburdening edge devices [Zhou *et al*., 2025a].

The **Privacy-Preserving Federated Learning** (**PFLF**) framework also contributes to reduced communication overhead and resource strain on clients by utilizing a flexible arrangement and participation mechanism, directing data to nearby edge servers for initial aggregation before reaching the application server [Zhou *et al*., 2022].

Another approach consists of reducing the amount of information exchanged during training. For instance, **NestFL** slashes communication burdens by transmitting updates only for compact subnetworks, rather than entire models, which also optimizes computational overhead by tailoring subnetworks to client-specific device runtime resources. Likewise, Federated Slimmable Neural Networks (**FedSNN**) strategically distributes different width configurations of SNN to devices based on their individual capacities, prioritizing lightweight models for low-resource devices. This approach reduces network traffic consumption and the total number of training epochs required for model convergence, optimizing both communication and local training resources [Xu *et al*., 2024].

Security mechanisms can also affect resource consumption. To avoid the high computational and communication costs associated with complex cryptographic primitives, such as Homomorphic Encryption and Multi-Party Computation, the Lightweight and Secure Federated Learning (**LSFL**) scheme adopts a lightweight design that ensures security, requiring only three communications between the server and participants for secure collaborative training [Zhang *et al*., 2022].

5.1.2 Efficient Training Strategies and Dynamic Resource Management

Other approaches focus on more efficient training strategies and dynamic resource allocation to conserve energy and manage communication effectively. The Differential Privacy-Semi-Asynchronous Federated Learning (**DP-SAFL**) framework and the hierarchical Blockchain-enabled Semi-asynchronous FL **BSFL** architecture use semi-asynchronous FL to balance training efficiency with global model stability, reducing round durations and lowering resource consumption [He *et al*., 2025; Ren and Lee, 2025].

Adaptive Staleness-aware Momentum Asynchronous Federated Learning (**ASMAFL**) is specifically designed for resource and energy efficiency, accelerating model training and reducing energy costs through optimal staleness-aware parameters and a three-stage training strategy. It also handles asynchronous settings and variable wireless parameters, coordinating devices to arrive simultaneously for aggregation, thereby mitigating the "straggler problem" and improving communication efficiency [Qiao *et al*., 2024].

Similarly, a semi-asynchronous federated learning **FedSA** mechanism mitigates the straggler effect by efficiently selecting the optimal number of participating workers. It balances the frequent model transfers of asynchronous FL with the waiting times of synchronous FL, all within a constrained communication budget [Ma *et al*., 2021a].

Dynamic resource allocation and offloading schemes are also crucial. The Hierarchical Federated Unlearning (**HierFUN**) framework dynamically determines the appropriate number of clusters based on available device resources, allowing in-cluster training to maintain performance with limited capacity. It further enhances communication efficiency by denying direct communication between the central server and cluster heads during training and adapts a timeout retransmission mechanism for unstable networks [Ma *et al*., 2024]. The **Dynamic Distributed Compression** algorithm empowers edge nodes to independently optimize compression factors and training data samples based on their CPU/bandwidth resources. It improves communication efficiency by allowing edge nodes to use dynamic, non-identical compression factors and leverages a Broadcast-Based Gossip protocol to limit traffic and reduce delayed model exchanges [Asheralieva *et al*., 2025]. Resource-Efficient Federated Learning with Hierarchical Aggregation (**RFL-HA**) optimizes cluster formation and hierarchical aggregation to reduce both communication and computation resources [Wang *et al*., 2021].

Table 1. Challenges Addressed by Each Technique.

Technique	Resource Limitations	Energy Efficiency	Communication Efficiency	Data Quality	Privacy/ Security
FedLFP [Sun *et al*., 2025]			✓	✓	✓
DP-SAFL [He *et al*., 2025]	✓		✓		✓
RCFL [Zhou *et al*., 2025a]	✓	✓	✓	✓	✓
NestFL [Zhou *et al*., 2025b]	✓		✓	✓	✓
BSFL [Ren and Lee, 2025]	✓		✓		✓
Dynamic Distributed Compression [Asheralieva *et al*., 2025]	✓		✓		
ASMAFL [Qiao *et al*., 2024]	✓	✓	✓	✓	
Multicenter Hierarchical FL [Chen *et al*., 2024]	✓		✓		✓
MGWFL [Zhao *et al*., 2024]			✓	✓	✓
STAR [Liu *et al*., 2024]	✓		✓	✓	
Hier-FUN [Ma *et al*., 2024]	✓		✓	✓	✓
FedSNN [Xu *et al*., 2024]	✓		✓	✓	
PFLF [Zhou *et al*., 2022]			✓		✓
LSFL [Zhang *et al*., 2022]	✓		✓		✓
LiMPO [uz Zaman *et al*., 2022]	✓	✓	✓		
FedSTN [Yuan *et al*., 2022]	✓		✓		✓
CoCo [Wang *et al*., 2022]	✓		✓	✓	
RFL-HA [Wang *et al*., 2021]	✓		✓		
FedSA [Ma *et al*., 2021a]	✓		✓	✓	
Adaptive Batch Size with Scaled Learning Rate Algorithm [Ma *et al*., 2021b]	✓		✓		
AAFL [Liu *et al*., 2021a]	✓		✓		
Blockchain-Enabled Asynchronous FL [Liu *et al*., 2021b]	✓		✓		✓

For mobile scenarios, the Lightweight Mobility Prediction and Offloading (**LiMPO**) framework offloads compute-intensive tasks to predicted user locations, using a multi-objective genetic algorithm for server selection to optimize Multi-access Edge Computing/Mobile Edge Computing server resource utilization. This framework jointly optimizes energy consumption and reduces latency by considering user mobility in offloading decisions [uz Zaman *et al*., 2022]. **FedSTN** distributes models among EC servers to alleviate computational burden on central Base Stations and mitigate communication overload and delays [Yuan *et al*., 2022].

Addressing battery life, the **Adaptive Batch Size with Scaled Learning Rate Algorithm** targets IoT devices by dynamically adjusting batch sizes to reduce waiting times, thereby saving energy and improving training efficiency [Ma *et al*., 2021b]. This technique, along with the Adaptive Asynchronous FL (**AAFL**), alleviates the synchronization barrier, enabling faster training by aggregating local updates from a fraction of edge nodes as they arrive, adapting fraction values based on real-time system conditions [Liu *et al*., 2021a]. **STAR** enhances communication efficiency by reducing bandwidth consumption by approximately 40% and communication costs by 25% to 49.7% through its progressive model growth and pseudo-labeling [Liu *et al*., 2024]. **CoCo** addresses dynamic network conditions and limited bandwidth by constructing a P2P network topology and determining different compression ratios for workers [Wang *et al*., 2022].

5.1.3 Remaining Challenges

Despite recent advances, explicit energy optimization remains an open challenge, as most approaches improve energy efficiency only indirectly through reductions in computation or communication costs. Future research should explore adaptive privacy mechanisms and more efficient privacy encoding strategies that balance privacy and model utility [Zhou *et al*., 2025a].

Additional trade-offs also persist. Blockchain-based FL can improve integrity and traceability but may increase communication overhead [Liu *et al*., 2021b]. Similarly, network management remains complex even when aided by deep reinforcement learning, as demonstrated by **AAFL** [Liu *et al*., 2021a]. Furthermore, **CoCo** shows that excessive compression or overly sparse network topologies may reduce communication costs at the expense of training performance [Wang *et al*., 2022].

5.2 Data Quality and Heterogeneity

Data quality and data heterogeneity are pervasive challenges in FL, encompassing non-IID data distributions, varied model scales, and the presence of low-quality prototypes. Effectively addressing these issues is crucial for the performance and reliability of intelligent systems in dynamic edge environments. This subsection reviews approaches for dealing with data quality and heterogeneity in FL, particularly non-IID data distributions, followed by a discussion of the remaining challenges.

5.2.1 Strategies for Non-IID Data and Personalized Model Learning

Many techniques in FL focus on adapting model training and aggregation mechanisms to explicitly account for divergent local data distributions and to enable personalized model learning. For instance, **FedLFP** is specifically designed to handle heterogeneous data distributions by enabling collaborative training of personalized models. It ensures effective global knowledge transfer through unsupervised clustering, incorporating confidence scores and sample counts for weighted clustering to minimize the impact of low-quality prototypes [Sun *et al*., 2025]. Similarly, **RCFL** tackles non-IID data by utilizing the sign of gradients to update the global model, and it enhances privacy budget allocation to perturb gradient descent, which helps mitigate overfitting [Zhou *et al*., 2025a].

NestFL also learns personalized models for each client, addressing Non-IID data distributions through cross-training mechanisms that maintain consistent decision boundaries and mitigate class-level data representation variability. This approach employs a weighted aggregation technique that adaptively assigns weights proportional to each node's contribution to maximize personalization preservation.

Further strategies focus on addressing data heterogeneity and improving training on Non-IID data. **Multi-Granularity Weighted Federated Learning** (**MGWFL**) targets heterogeneous clients by combining a distance-based FL mechanism for similar client types with an attention-weighted FL mechanism for different client types, facilitating knowledge transfer and correcting gradients during local updates [Zhao *et al*., 2024]. **FedSNN** addresses data heterogeneity by optimizing model aggregation to learn from diverse datasets, assigning higher weights to devices contributing more data and ensuring that each SNN width is trained on a dataset approximating the global distribution [Xu *et al*., 2024]. To further improve training accuracy on Non-IID data, **FedSA** deploys adaptive learning rates for workers based on their relative participation frequency [Ma *et al*., 2021a]. Finally, **CoCo** introduces a "consensus distance" metric to quantify the discrepancy between each local model and the average of all local models, guiding fine-grained operations that adapt to varying data distributions across workers [Wang *et al*., 2022].

Other approaches contribute to managing data quality and heterogeneity. **ASMAFL** addresses the negative impact of non-IID data on minimizing the global loss function by integrating staleness-aware parameters into a unified momentum gradient descent framework [Qiao *et al*., 2024]. **STAR** enhances semi-supervised FL by ensuring high-quality pseudo-labeled data through a predefined confidence threshold, which helps mitigate performance degradation caused by a scarcity of pseudo-labeled data in edge clients and ensures slower degradation in test accuracies even with varying proportions of unlabeled data. It also progressively increases pseudo-labeled data [Liu *et al*., 2024]. In the context of **Hier-FUN**, hierarchical clustering techniques expedite both learning and unlearning processes by grouping devices with complementary data distributions, thereby mitigating challenges posed by Non-IID data by constraining the influence sphere of target devices through in-cluster training [Ma *et al*., 2024].

5.2.2 Remaining Challenges

Several challenges persist in fully addressing data quality and heterogeneity in FL. For **FedLFP**, further validation on a wider range of real-world Mobile Edge Computing datasets is required to assess its applicability to more diverse and complex non-IID data distributions [Sun *et al*., 2025]. In the case of **Hier-FUN**, while clustering devices based on data distribution can facilitate efficient model training, it might compromise the model's generalization performance, especially when dealing with non-IID data [Ma *et al*., 2024]. **MGWFL** exhibits a slight performance decrease during its initial guidance phase and necessitates more iterations to surpass the performance of initially pre-trained models, indicating a potential trade-off in early training stages [Zhao *et al*., 2024].

The underlying complexity of theoretical analysis for asynchronous structures remains a general issue for research, including for **DP-SAFL**, where synchronous FL analysis is not directly applicable [He *et al*., 2025]. Furthermore, optimizing for non-IID data remains a significant challenge for blockchain-enabled FL due to the inherent heterogeneity and imbalance of real-world data distributions [Liu *et al*., 2021b]. A key concern in semi-supervised learning using **STAR** is that not all pseudo-labels generated by model predictions are perfectly accurate. Reducing the error rate of these pseudo-labels is critical to prevent the propagation of errors throughout subsequent training stages. Additionally, increasing model depth can degrade training performance if the amount of labeled data is limited [Liu *et al*., 2024].

5.3 Data Privacy and Security

Data privacy and security are essential in FL, where sensitive data stays on local devices and only model updates are shared. Several techniques aim to protect user information and maintain training integrity by preventing data leakage and supporting decentralized control. This subsection discuss differential privacy approaches, data obfuscation techniques, decentralized architectures, cryptographic and unlearning mechanisms, robustness against malicious actors, and finally summarize the remaining open challenges.

5.3.1 Differential Privacy and Noise Perturbation

One approach to preserving privacy involves introducing controlled noise into model parameters, making it difficult to infer individual data points. **DP-SAFL** incorporates dual ϵ-differential privacy (DP) by adding Gaussian noise to both local model parameters uploaded by mobile devices and global parameters broadcast by the edge server, ensuring privacy protection in both uplink and downlink channels and mitigating parameter leakage [He *et al*., 2025]. Similarly, the **PFLF** framework provides a global privacy-preserving mechanism that guarantees ϵ-DP across the entire training process, where each client adds artificial noise to model parameters before transmission to maintain strong privacy guarantees [Zhou *et al*., 2022]. While primarily designed to address heterogeneity, **MGWFL** also benefits from FL's inherent privacy properties by not requiring raw data sharing and can be further combined with DP mechanisms to strengthen protection through noisy model updates [Zhao *et al*., 2024].

5.3.2 Advanced Data Obfuscation and Encoding Strategies

Other techniques employ sophisticated methods to obscure or encode sensitive information during the FL process. **FedLFP** enhances data privacy by utilizing label-free prototypes instead of sharing high-dimensional model parameters or class labels, preventing the leakage of sensitive information. This approach is critical in applications such as malware classification, where prototype clustering also contributes to safeguarding client data privacy [Sun *et al*., 2025].

Meanwhile, **RCFL** offers a robust defense against Multiple Privacy Leakage Attacks (MPLA) by implementing a global privacy protection mechanism and an innovative privacy encoding strategy. This dual approach acts as an obfuscation layer, ensuring differential privacy across multiple data releases and anonymizing gradient information, thereby reducing the MPLA success rate from 88.56% to 42.57% compared to other defense mechanisms [Zhou *et al*., 2025a].

NestFL strengthens privacy by only communicating subnetworks that represent a small portion of the local model between devices and the central server. This practice hides partial information about local data by pruning certain gradients on the device, thereby significantly reducing the exposure of sensitive information. Its cross-training mechanism further mitigates privacy risks by avoiding the transmission of additional data distribution information from heterogeneous clients.

5.3.3 Decentralized Architectures for Enhanced Security

Moving away from centralized aggregators can eliminate single points of failure and increase overall system resilience. **BSFL**, a hierarchical blockchain-based semi-asynchronous FL architecture, implements a decentralized approach using blockchain, providing an alternative to a central aggregator. This design mitigates single-point-of-failure risks and enhances security through blockchain's decentralization, immutability, and inherent protection features [Ren and Lee, 2025]. Similarly, the **Blockchain-Enabled Asynchronous FL** approach uses blockchain to establish a decentralized global model convergence environment. Model updates are verified via a consensus algorithm and stored on public ledgers, eliminating the need for a third party and protection against cyberattacks such as poisoning [Liu *et al*., 2021b].

Another approach to improving resilience and security is **Multicenter Hierarchical FL**, which reduces reliance on central parameter servers by deploying a distributed network of aggregation centers at the edge. This multicenter design decreases dependence on individual Mobile EC servers, improving the reliability and security of model training in edge intelligence environments [Chen *et al*., 2024].

Beyond security benefits, these decentralized and hierarchical designs also contribute to failure resilience. By eliminating central aggregators or distributing aggregation across multiple nodes, approaches such as BSFL, DFL, and Multicenter Hierarchical FL reduce the impact of device failures. Asynchronous strategies such as ASMAFL and FedSA further tolerate stragglers and dropped devices without halting the training process. Explicit fault-tolerance mechanisms, such as checkpoint-based recovery and formal failure detection protocols, remain an open research direction.

5.3.4 Cryptographic and Unlearning Mechanisms

Other techniques employ cryptographic methods or specific unlearning mechanisms to further enhance privacy. **LSFL** protects participant data privacy by using a secret sharing method where local training results are uploaded in partial form to two non-colluding honest-but-curious servers. This ensures that neither server can deduce participants' private information without relying on complex cryptographic algorithms, thus offering a computationally lightweight approach suitable for resource-constrained edge nodes [Zhang *et al*., 2022]. **FedSTN** integrates FL to address privacy concerns in urban traffic flow prediction by employing an Attentive Mechanism Federated Network module that shares short-term spatio-temporal hidden information using an additive homomorphic encryption approach, particularly based on Vertical FL, which allows distributed training while safeguarding data privacy [Yuan *et al*., 2022].

Beyond protecting data during training, some approaches focus on removing previously learned information. **Hier-FUN** enhances data privacy by selectively removing the contributions of specific devices from a global model, aligning with regulations like GDPR and California Consumer Privacy Act (CCPA), which emphasize the "right to be forgotten." It efficiently unlearns a device's impact without requiring retraining from scratch, and its strategic clustering and in-cluster training restrict the influence of target devices [Ma *et al*., 2024].

5.3.5 Robustness Against Malicious Actors

Protecting against malicious clients, such as those attempting model poisoning or data inference attacks, is another critical aspect of security. **LSFL** specifically guarantees Byzantine robustness, ensuring correct model training even in the presence of malicious or non-cooperative participants. It employs a Lightweight Byzantine-Robustness Two-Server Secure Aggregation protocol and includes a reward and penalty mechanism to distinguish and manage benign versus malicious participants, thereby preventing them from impacting model accuracy [Zhang *et al*., 2022]. **BSFL** incorporates a dual validation mechanism to protect against model poisoning attacks. This includes verification at the Directed Acyclic Graph layer and further scrutiny by sub-chain validators who recheck authenticity and evaluate model performance, thereby preserving stable convergence and accuracy against malicious nodes [Ren and Lee, 2025]. As mentioned, the **Blockchain-Enabled Asynchronous Federated Learning** also provides security against cyberattacks such as poisoning attacks through its decentralized and verified update mechanism [Liu *et al*., 2021b].

5.3.6 Remaining Challenges

Despite advancements, several challenges persist in ensuring robust data privacy and security in FL. Primarily, achieving an appropriate balance between privacy preservation and model utility remains a significant ongoing challenge. For instance, differential privacy mechanisms, while crucial, introduce noise that can "destroy the data utility" if excessive [He *et al*., 2025]. This addition of noise can lead to poor convergence, especially as the number of iterations increases for DP-enabled models compared to non-DP models [Zhou *et al*., 2022]. Furthermore, the introduction of random Differential Privacy noise may increase overall noise levels, potentially impacting model convergence. Therefore, future research is essential to develop adaptive privacy mechanisms that can dynamically manage this trade-off and to further optimize privacy encoding strategies [Zhou *et al*., 2025a].

While security mechanisms enhance protection, they can introduce overheads that impact scalability. For example, the performance of **FedSTN** can degrade with an increasing number of Edge Computing Servers [Yuan *et al*., 2022]. Integrating blockchain, as seen in **Blockchain-Enabled Asynchronous Federated Learning**, may introduce additional communication costs [Liu *et al*., 2021b]. Furthermore, scalability assessments for ultra-large networks are still needed for **BSFL** [Ren and Lee, 2025]. **Multicenter Hierarchical FL** also notes that its scalability in extremely large and heterogeneous networks, especially concerning privacy requirements, requires further investigation [Chen *et al*., 2024].

Existing defenses against malicious actors, such as **RCFL**'s protection against Multiple Privacy Leakage Attacks, still leave room for improvement, as the success rate for such attacks is reduced but not entirely eliminated [Zhou *et al*., 2025a]. There is a need to expand threat models to address more sophisticated adversarial attacks, such as model inversion and membership inference [Ren and Lee, 2025]. Further research is also required to address more advanced privacy issues in blockchain-enabled FL, potentially utilizing techniques such as differential privacy [Liu *et al*., 2021b].

Some techniques, like **NestFL**, explicitly state that the effectiveness of their privacy preservation mechanisms requires further evaluation in future work. Similarly, **ASMAFL**'s future work will concentrate on addressing security vulnerabilities and privacy leakage concerns, indicating that privacy is not fully addressed in its current iteration [Qiao *et al*., 2024].

6 Conclusion

This survey addressed three research questions. For RQ-1, two primary physical architectures were identified, Edge-Only and Cloud-Edge, alongside FL as a widely adopted distributed learning implementation in centralized, decentralized, and hierarchical configurations. For RQ-2, five critical challenge categories were identified: resource limitations, energy efficiency, communication overhead, data privacy and security, and data heterogeneity, with failure resilience emerging implicitly through decentralized and asynchronous designs. For RQ-3, a broad set of mitigation techniques was synthesized, including communication-efficient strategies (FedLFP, NestFL, RCFL), asynchronous training mechanisms (ASMAFL, FedSA, AAFL), personalized learning approaches (MGWFL, FedSNN), and privacy-preserving designs based on differential privacy, data obfuscation, and decentralized architectures — most of which simultaneously address two or three challenge dimensions.

Despite significant advancements, persistent challenges remain, particularly in balancing privacy preservation with model utility, improving scalability in large and heterogeneous networks, and enhancing robustness against sophisticated adversarial attacks. These areas highlight important future research directions for developing robust, scalable, and privacy-preserving solutions for ubiquitous edge intelligence.

Declarations

Authors' Contributions

L.H. was responsible for the conceptualization, investigation, and writing of the survey. Other authors acted as supervisors and contributed through review and editing of the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The data will be made available upon request.

References

Abreha, H. G., Hayajneh, M., and Serhani, M. A. (2022). Federated learning in edge computing: A systematic survey. *Sensors*, 22:450–450. DOI: 10.3390/s22020450.

Asheralieva, A., Niyato, D., and Wei, X. (2025). Dynamic distributed model compression for efficient decentralized federated learning and incentive provisioning in edge computing networks. *IEEE Transactions on Mobile Computing*, 24:6293–6314. DOI: 10.1109/tmc.2025.3543295.

Brecko, A., Kajáti, E., Koziorek, J., and Zolotová, I. (2022). Federated learning for edge computing: A survey. *Applied Sciences*, 12:9124–9124. DOI: 10.3390/app12189124.

Cajas Ordóñez, S. A., Samanta, J., Suárez-Cetrulo, A. L., and Carbajo, R. S. (2025). Intelligent edge computing and machine learning: A survey of optimization and applications. *Future Internet*, 17(9). DOI: 10.3390/fi17090417.

Chen, X., Xu, G., Xu, X., Jiang, H., Tian, Z., and Ma, T. (2024). Multicenter hierarchical federated learning with fault-tolerance mechanisms for resilient edge computing networks. *IEEE Trans. on Neural Net. and Learning Systems*, 36:47–61. DOI: 10.1109/tnnls.2024.3362974.

Duan, Q., Huang, J., Hu, S., Deng, R., Lu, Z., and Yu, S. (2023). Combining federated learning and edge computing toward ubiquitous intelligence in 6g network: Challenges, recent advances, and future directions. *IEEE Communications Surveys & Tutorials*, 25:2892–2950. DOI: 10.1109/comst.2023.3316615.

European Union (2016). General data protection regulation (GDPR). `https://gdpr-info.eu/`. Accessed: 2026-04-01.

Filho, C. P., Marques, E. L., Chang, V., dos Santos, L., Bernardini, F., Pires, P. F., Ochi, L. S., and Delicato, F. C. (2022). A systematic literature review on distributed machine learning in edge computing. *Sensors*, 22:2665–2665. DOI: 10.3390/s22072665.

Guo, Y., Zhao, R., Lai, S., Fan, L., Lei, X., and Karagiannidis, G. K. (2022). Distributed machine learning for multiuser mobile edge computing systems. *IEEE J. Sel. Top. Signal Process.*, 16:460–473. DOI: 10.1109/jstsp.2022.3140660.

Hasan, M. K., Jahan, N., Nazri, M. Z. A., Islam, S., Khan, M. A., Alzahrani, A. I., Alalwan, N., and Nam, Y. (2024). Federated learning for computational offloading and resource management of vehicular edge computing in 6g-v2x network. *IEEE Transactions on Consumer Electronics*, 70:3827–3847. DOI: 10.1109/tce.2024.3357530.

He, C., Guo, S., Liu, G., and Zhang, W. (2025). Dp-safl: Semi-asynchronous federated learning with differential privacy in heterogeneous edge computing. *Computer Networks*, 267:111346–111346. DOI: 10.1016/j.comnet.2025.111346.

Liu, J., Liu, J., Xu, H., Liao, Y., Yao, Z., Chen, M., and Qian, C. (2024). Enhancing semi-supervised federated learning with progressive training in heterogeneous edge computing. *IEEE Transactions on Mobile Computing*, 24:2315–2330. DOI: 10.1109/tmc.2024.3492140.

Liu, J., Xu, H., Wang, L., Xu, Y., Qian, C., Huang, J., and Huang, H. (2021a). Adaptive asynchronous federated learning in resource-constrained edge computing. *IEEE Transactions on Mobile Computing*, 22:674–690. DOI: 10.1109/tmc.2021.3096846.

Liu, Y., Qu, Y., Xu, C., Hao, Z., and Gu, B. (2021b). Blockchain-enabled asynchronous federated learning

in edge computing. *Sensors*, 21:3335–3335. DOI: 10.3390/s21103335.

Ma, Q., Xu, Y., Xu, H., Jiang, Z., Huang, L., and Huang, H. (2021a). Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing. *IEEE J. Sel. Areas Commun.*, 39:3654–3672. DOI: 10.1109/jsac.2021.3118435.

Ma, Z., Tu, H., Zhou, L., PengLi, Ji, Yan, X., Xu, H., Wang, Z., and Chen, S. (2024). Hier-fun: Hierarchical federated learning and unlearning in heterogeneous edge computing. *IEEE Internet of Things Journal*, 12:8653–8668. DOI: 10.1109/jiot.2024.3502666.

Ma, Z., Xu, Y., Xu, H., Meng, Z., Huang, L., and Xue, Y. (2021b). Adaptive batch size for federated learning in resource-constrained edge computing. *IEEE Trans. Mobile Comput.*, 22:37–53. DOI: 10.1109/tmc.2021.3075291.

Oliveira, F., Costa, D. G., Assis, F., and Silva, I. (2024). Internet of intelligent things: A convergence of embedded systems, edge computing and machine learning. *Internet of Things*, 26:101153–101153. DOI: 10.1016/j.iot.2024.101153.

Qiao, D., Guo, S., Zhao, J., Le, J., Zhou, P., Li, M., and Chen, X. (2024). Asmafl: Adaptive staleness-aware momentum asynchronous federated learning in edge computing. *IEEE Trans. Mobile Comput.*, 24:3390–3406. DOI: 10.1109/tmc.2024.3510135.

Rahmani, A. M., Alsubai, S., Alanazi, A., Alqahtani, A., Zaidi, M., and Hosseinzadeh, M. (2024). The role of mobile edge computing in advancing federated learning algorithms and techniques: A systematic review of applications, challenges, and future directions. *Computers & Electrical Engineering*, 120:109812–109812. DOI: 10.1016/j.compeleceng.2024.109812.

Rajassekharan, D. (2025). Survey on Applications, Techniques and Challenges of Machine Learning for Edge Environments. *Journal of Soft Computing Paradigm*, 7(4):331–345. DOI: 10.36548/jscp.2025.4.002.

Ramírez-Gordillo, T., Pujol, F. A., and Mora, H. (2026). Unpacking distributed machine learning: A unified taxonomy, formal foundations, and the rise of emerging paradigms. *Neurocomputing*, 673:132828. DOI: https://doi.org/10.1016/j.neucom.2026.132828.

Ren, S. and Lee, C. (2025). A hierarchical blockchain architecture for federated learning in edge computing networks. *J. Supercomputing*, 81. DOI: 10.1007/s11227-025-07262-2.

Riedel, P., Schick, L., von Schwerin, R., Reichert, M., Schaudt, D., and Hafner, A. (2024). Comparative analysis of open-source federated learning frameworks - a literature-based survey and review. *International Journal of Machine Learning and Cybernetics*, 15:5257–5278. DOI: 10.1007/s13042-024-02234-z.

Sun, Y., Pan, S., Sun, A., Fu, Z., Long, S., and Li, Z. (2025). Fedlfp: Communication-efficient personalized federated learning on non-iid data in mobile edge computing environments. *IEEE Transactions on Mobile Computing*, 24:8811–8823. DOI: 10.1109/tmc.2025.3558406.

Tu, J., Yang, L., and Cao, J. (2025). Distributed machine learning in edge computing: Challenges, solutions and future directions. *ACM Comput. Surv.*, 57(5). DOI: 10.1145/3708495.

uz Zaman, S. K., Jehangiri, A. I., Maqsood, T., Haq, N. U., Umar, A. I., Shuja, J., Ahmad, Z., Dhaou, I. B., and Alsharekh, M. F. (2022). Limpo: lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Cluster Computing*, 26:99–117. DOI: 10.1007/s10586-021-03518-7.

Wang, L., Xu, Y., Xu, H., Chen, M., and Huang, L. (2022). Accelerating decentralized federated learning in heterogeneous edge computing. *IEEE Transactions on Mobile Computing*, pages 1–1. DOI: 10.1109/tmc.2022.3178378.

Wang, Z., Xu, H., Liu, J., Huang, H., Qiao, C., and Zhao, Y. (2021). Resource-efficient federated learning with hierarchical aggregation in edge computing. *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10. DOI: 10.1109/infocom42981.2021.9488756.

Wu, J., Dong, F., Leung, H., Zhu, Z., Zhou, J., and Drew, S. (2024). Topology-aware federated learning in edge computing: A comprehensive survey. *ACM Computing Surveys*, 56:1–41. DOI: 10.1145/3659205.

Xia, Q., Ye, W., Tao, Z., Wu, J., and Li, Q. (2021). A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1:100008–100008. DOI: 10.1016/j.hcc.2021.100008.

Xu, Y., Liao, Y., Xu, H., Wang, Z., Wang, L., Liu, J., and Chen, Q. (2024). Fedsnn: Training slimmable neural network with federated learning in edge computing. *EEE/ACM Trans. Netw.*, pages 1–16. DOI: 10.1109/tnet.2024.3487582.

Yu, R. and Li, P. (2021). Toward resource-efficient federated learning in mobile edge computing. *IEEE Network*, 35:148–155. DOI: 10.1109/mnet.011.2000295.

Yuan, X., Chen, J., Yang, J., Zhang, N., Yang, T., Han, T., and Taherkordi, A. (2022). Fedstn: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.*, 24:8738–8748. DOI: 10.1109/tits.2022.3157056.

Zhang, Z., Wu, L., Ma, C., Li, J., Wang, J., Wang, Q., and Yu, S. (2022). Lsfl: A lightweight and secure federated learning scheme for edge computing. *IEEE Trans. Inf. Forensics Secur.*, 18:365–379. DOI: 10.1109/tifs.2022.3221899.

Zhao, Y., Qiu, C., Cai, S., Liu, Z., Wang, Y., Wang, X., and Hu, Q. (2024). Multi-granularity weighted federated learning for heterogeneous edge computing. *IEEE Trans. Serv. Comput.*, 18:270–287. DOI: 10.1109/tsc.2024.3495532.

Zhou, H., Dai, H., Yang, G., and Xiang, Y. (2025a). Robust Federated Learning for Privacy Preservation and Efficiency in Edge Computing. *IEEE Transactions on Services Computing*, 18(03):1739–1752. DOI: 10.1109/TSC.2025.3562359.

Zhou, H., Yang, G., Dai, H., and Liu, G. (2022). PFLF: privacy-preserving federated learning framework for edge computing. *IEEE Trans. Inf. Forensics Secur.*, 17:1905–1918. DOI: 10.1109/tifs.2022.3174394.

Zhou, X., Hu, Y., Jia, Q., and Xie, R. (2025b). Nestfl: Enhancing federated learning through nested multicapacity model pruning in heterogeneous edge computing. *IEEE Internet of Things Journal*, 12:27435–27449. DOI: 10.1109/jiot.2025.3562633.