frontiers | Frontiers in Computer Science

TYPE Review
PUBLISHED 04 November 2025
DOI 10.3389/fcomp.2025.1617597

Check for updates

OPEN ACCESS

EDITED BY
Rashid Ibrahim Mehmood,
Islamic University of Madinah, Saudi Arabia

REVIEWED BY
N. Nasurudeen Ahamed,
United Arab Emirates University,
United Arab Emirates
Jinyang Huang,
Hefei University of Technology, China
Shan Wei Chen,
Baoji University of Arts and Sciences, China

*CORRESPONDENCE
Lakshan Cooray
✉ lakshan.20221470@iit.ac.lk

RECEIVED 24 April 2025
ACCEPTED 09 October 2025
PUBLISHED 04 November 2025

CITATION
Cooray L, Sendanayake J, Vithanaarachchi P and Priyadarshana YHPP (2025) Deep federated learning: a systematic review of methods, applications, and challenges. *Front. Comput. Sci.* 7:1617597.
doi: 10.3389/fcomp.2025.1617597

Deep federated learning: a systematic review of methods, applications, and challenges

Lakshan Cooray[1]*, Janaka Sendanayake[1], Pramuditha Vithanaarachchi[1] and Y. H. P. P. Priyadarshana[2]

[1]School of Computing, Informatics Institute of Technology, Colombo, Sri Lanka, [2]Ubiquitous and Personal Computing Lab, Kyoto University of Advanced Science (KUAS), Kyoto, Japan

Federated Learning (FL) represents a paradigm shift in machine learning, enabling collaborative model training on decentralized data while preserving user privacy. However, the transition from theory to real-world application is impeded by significant challenges, including high communication costs, statistical and system heterogeneity and persistent privacy vulnerabilities. These barriers critically limit the performance, scalability and security of FL systems. This paper provides a systematic review of the state-of-the-art solutions developed to address these fundamental obstacles. The review analyzes core methodological advancements, including advanced model aggregation methods, techniques to enhance communication efficiency such as model compression and decentralized training and strategies to combat statistical heterogeneity arising from non-IID data. Furthermore, it delves into emerging paradigms like Federated Meta-Learning and Federated Reinforcement Learning, alongside advanced architectural models such as hierarchical and blockchain-based systems. The practical impact of these advancements is contextualized through a review of key application domains, including healthcare, vehicular networks and the Internet of Things. A benchmark analysis is presented to assess the practical efficacy of these diverse techniques. In conclusion, this work synthesizes the critical trade-offs inherent in FL systems and highlights key directions for future research, offering a comprehensive guide for researchers and practitioners in this rapidly evolving field.

KEYWORDS

federated learning, model aggregation, communication efficiency, statistical heterogeneity, privacy, meta-learning, reinforcement learning

1 Introduction

During the last decade, machine learning and deep learning related technologies have gained significant prominence in a variety of disciplines due to their development and problem-solving capabilities. In addition, these deep learning-based algorithms has made a huge impact on computational research and real-world applications due to their ability to reveal hidden patterns of information and non-linearity of data. As a result, deep learning and machine learning based technologies have started being used in various domains such as medicine, banking, manufacturing, agriculture etc. Even though it has contributed to considerable development in the above-mentioned fields, it has started to raise concerns about ensuring the privacy and security of data that are generated through millions of devices during the process of distributed deep learning. This has occurred due to the possibility of having data leakage in distributed systems since traditional machine learning and deep learning based models are usually trained in centralized data centers (Qi et al., 2024). Therefore, data generated through local devices has to be sent to the central servers to train the required models. The availability and generation of extensive datasets through millions of devices has made remarkable progress in the context of distributed deep learning. Nevertheless, when these centralized data centers, which store vast amounts of data received through local devices, are attacked, the damage will be limitless. Consequently, sensitive and confidential data presents significant privacy problems by limiting efforts to centralize data gathering, model training and the inferencing process (Zhang et al., 2023).

To address these challenges, edge computing has gained global attention, driven by the proliferation of edge devices like smartphones, IoT sensors and wearables that generate enormous volumes of data daily (Li T. et al., 2020). Federated Learning (FL), introduced by Google in 2016, has emerged as a promising solution, enhancing privacy by training models locally on edge devices (McMahan et al., 2017). FL employs a collaborative architecture where a central server maintains a global model, while local devices update their models using device-specific data. These updated models are periodically sent back to the server for aggregation, typically by combining parameters like weights and biases. Decentralized architectures have also been introduced, enabling direct communication between local devices, alongside blockchain-based FL systems (Qi et al., 2024). This foundational client-server process is illustrated in Figure 1, which depicts the core workflow of federated learning. In this architecture, local devices first train their models using their own private data. These locally trained models (specifically, their updated weights) are then sent to a central server. The server aggregates these weights for example, by using a weighted average to create a new, improved global model. This updated global model is then broadcast back to the devices and the cycle repeats. This iterative process preserves data privacy, as the raw data never leaves the local device, while collaboratively improving the global model's performance.

This distributed learning paradigm ensures data remains on local devices, mitigating privacy risks. However, FL faces challenges such as communication efficiency, system heterogeneity due to varied device capabilities, statistical heterogeneity arising from non-iid data and transmission delays during model aggregation. Additionally, concerns persist about data pattern leakage through shared model parameters, which may lead to the exposure of information via explainable AI techniques (Saifullah et al., 2024). Despite these challenges, FL has significantly advanced edge computing-based deep learning approaches and is widely applied in medicine, autonomous vehicles, IoT and traffic management.

Existing surveys on Federated Learning have predominantly addressed isolated aspects, such as privacy preservation or aggregation algorithms, without offering a comprehensive integration across multiple dimensions. There remains a critical need for a holistic overview that synthesizes advancements in aggregation methodologies, communication efficiency, privacy-preserving mechanisms and strategies for managing both statistical and system heterogeneity.

In addition, current research in federated learning is fragmented. Benchmarking practices are often inconsistent, and applications remain focused on a limited range of domains. Emerging directions such as federated meta-learning, federated reinforcement learning, and blockchain-based frameworks are still in their early stages and lack comprehensive study. In addition, deployment challenges such as client dropout, unreliable networks, and the balance between privacy and efficiency have not been systematically addressed. These gaps underline the need for a systematic review that consolidates technical advances and examines their interaction under practical constraints.

This study makes the following key contributions:

- Provides a structured and systematic review of Federated Learning (FL) research from 2018 to 2025, covering core challenges such as communication efficiency, statistical heterogeneity, system heterogeneity and privacy preservation.
- Summarizes state-of-the-art model aggregation methods, communication-efficient techniques (e.g., compression and pruning) and strategies for mitigating device and data heterogeneity.
- Examines emerging paradigms including Federated Meta-Learning and Federated Reinforcement Learning, as well as advanced architectures such as hierarchical and blockchain-based FL.
- Reviews robust privacy-preserving mechanisms (e.g., differential privacy, secure aggregation and homomorphic encryption) that are critical for real-world deployment.
- Highlights the practical applications of FL across domains such as healthcare, IoT, autonomous vehicles and mobile computing, demonstrating its real-world impact and implementation challenges.

In summary, this review synthesizes existing research, identifies open challenges and outlines future directions for building scalable, secure and efficient FL systems.

Firstly, Section 2 will introduce the methodology, outlining the PRISMA framework, search strategy, inclusion and exclusion criteria and the guiding research questions. Section 3 will provide a comprehensive review of core methodological advancements in Federated Learning (FL), beginning with model aggregation algorithms and continuing with strategies for communication efficiency, statistical and system heterogeneity and privacy-preserving mechanisms. The section will further examine emerging paradigms such as Federated Meta-Learning and Federated Reinforcement Learning, as well as advanced architectures including hierarchical and blockchain-based systems. Following this, Section 4 will review the applications of FL across diverse domains such as healthcare, vehicular networks and mobile computing, highlighting real-world implementations and challenges. Next, Section 5 will present benchmarking and performance analyses of state-of-the-art methods, consolidating experimental evidence from past studies. Finally, Section 6 will conclude the paper by synthesizing the key findings, identifying current limitations and outlining future research directions.

2 Methodology

2.1 PRISMA diagram

This review follows the PRISMA guidelines to maintain methodological transparency and reproducibility throughout the

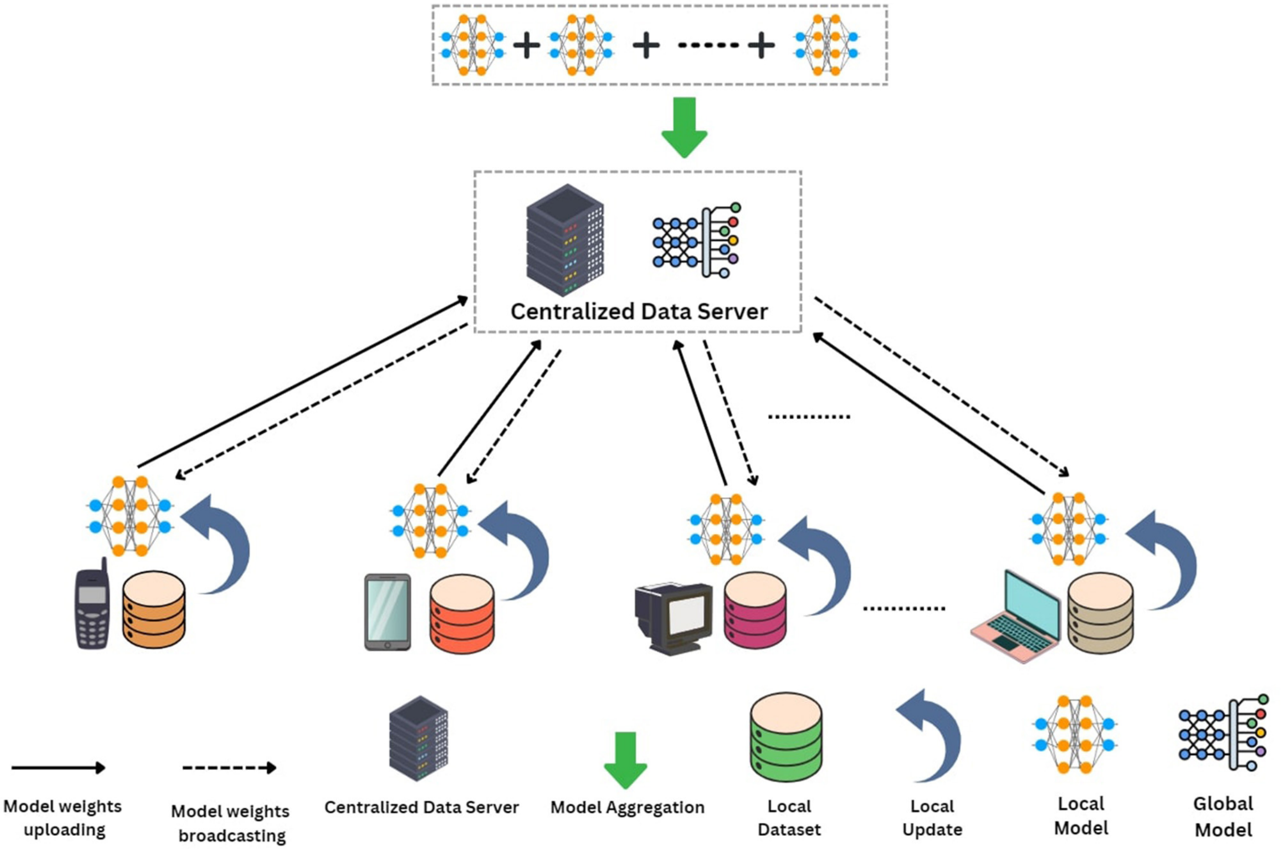

FIGURE 1
Federated learning basic architecture where devices train local models, send their model weights to a centralized server for aggregation and receive updated global models. This ensures data privacy while improving the global model iteratively.

research process. As illustrated in Figure 2, PRISMA provides a structured and widely accepted framework that strengthens the reliability and scientific rigor of systematic reviews (Tugwell and Tovey, 2021).

2.2 Search strategy

The literature search was conducted across multiple databases, including *IEEE Xplore*, *ACM Digital Library*, *SpringerLink*, *arXiv*, *Semantic Scholar* and *Google Scholar*. Search queries combined keywords such as "Federated Learning," "Model Aggregation," "Communication Efficiency," "Statistical Heterogeneity," "System Heterogeneity," "Privacy Preservation," "Differential Privacy," "Homomorphic Encryption," "Federated Meta-Learning," "Federated Reinforcement Learning," "Hierarchical Federated Learning," "Blockchain-based Federated Learning," and "Applications of Federated Learning."

To ensure that only relevant and high-quality studies were included, the following inclusion and exclusion criteria were applied:

Inclusion criteria:

- Papers that apply or propose Federated Learning methods addressing model aggregation, communication efficiency, statistical heterogeneity, system heterogeneity, or privacy.
- Studies that describe or evaluate emerging paradigms such as Federated Meta-Learning, Federated Reinforcement Learning, or blockchain/hierarchical FL.
- Research reporting real-world FL applications in domains such as healthcare, IoT, autonomous vehicles, or mobile systems.
- Survey or review papers that critically assess existing Federated Learning approaches.

Exclusion criteria:

- Papers unrelated to Federated Learning or decentralized training.
- Non-scientific sources (blogs, tutorials, opinion articles) without verifiable research contributions.
- Duplicate versions of the same paper across multiple repositories.

The review focused on peer-reviewed journals and conference proceedings mainly published between 2018 and 2025. The strategy ensured broad coverage across both academic and preprint repositories, though it remained subject to limitations such as subscription access and keyword specificity. A summary

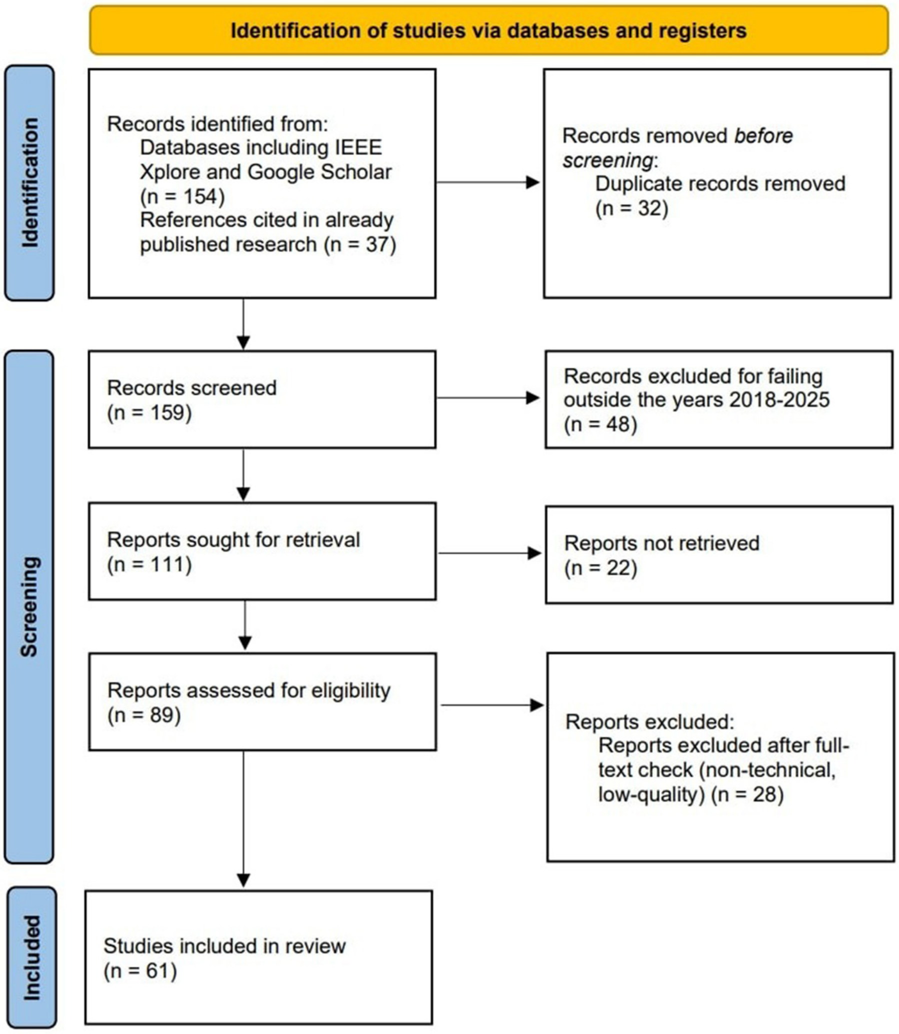
Identification of studies via databases and registers
Identification
Records identified from:
Databases including IEEE Xplore and Google Scholar (n = 154)
References cited in already published research (n = 37)
Records removed *before screening*:
Duplicate records removed (n = 32)
Screening
Records screened (n = 159)
Records excluded for failing outside the years 2018-2025 (n = 48)
Reports sought for retrieval (n = 111)
Reports not retrieved (n = 22)
Reports assessed for eligibility (n = 89)
Reports excluded:
Reports excluded after full-text check (non-technical, low-quality) (n = 28)
Included
Studies included in review (n = 61)

FIGURE 2
PRISMA diagram.

of the major federated learning areas and representative techniques is provided in Table 1 to give a holistic overview of the field, and these categories will be discussed in detail in the following sections. Additionally, as illustrated in Figure 3, the evolution of Federated Learning from 2016 to 2025 reflects a clear trajectory from foundational aggregation algorithms toward advanced personalization, meta-learning, and emerging paradigms such as reinforcement learning based federated frameworks.

2.3 Research questions

To guide the review process, the study specifically addressed the following research questions (RQ):

1. RQ1: How have recent advancements in deep learning-based model aggregation and compression techniques mitigated the communication inefficiencies that challenge FL?
2. RQ2: To what extent have the proposed solutions for statistical heterogeneity improved the performance and generalizability of FL models across diverse client device datasets?
3. RQ3: What strategies have been developed to address the challenges posed by system heterogeneity within FL environments?
4. RQ4: What privacy-preserving techniques have been introduced to reduce the risk of data leakage through model weights during the model aggregation process?

3 Methodological advancements

3.1 Model aggregation

This section examines the model aggregation techniques employed in FL, which serve as a fundamental component of the FL architecture. As outlined in the introduction, model aggregation in FL refers to the process of merging the updated weights of locally trained models with the global model

TABLE 1 Major federated learning areas and techniques discussed.

Category	Paper/technique	Key details/innovations
Aggregation	McMahan et al. (2017), FedAvg	Classical averaging of local updates weighted by dataset size. Performs well in IID settings but is less stable under non-IID data distributions.
	Sahu et al. (2018), FedProx	Adds a proximal term to local objectives, helping to stabilize convergence when clients are heterogeneous.
	Wang et al. (2020), FedNova	Normalizes client updates based on the number of local steps, reducing objective inconsistency across devices.
	Karimireddy et al. (2021), SCAFFOLD	Uses control variates on both server and clients to counter client drift, improving performance under non-IID distributions.
	Li et al. (2021), MOON	Incorporates a model-contrastive loss to align local and global representations, reducing divergence between clients.
	Niu and Wei (2023), FedHybrid	Hybrid approach combining FedAvg, FedProx, and SCAFFOLD, leveraging averaging, proximal regularization, and drift correction.
	Palihawadana et al. (2022), FedSim	Aggregation guided by gradient similarity, enabling implicit clustering of clients without data sharing.
	Kairouz et al. (2021), FedGPA	Prototype-based personalized aggregation that decouples feature extractor and classifier for better personalization.
	Ding et al. (2025), FEDMWAD	Multi-weight aggregation with anomaly detection, down-weighting outlier updates using median similarity measures.
Communication	Zhang et al. (2024), EdgeFL	Fully decentralized, peer-to-peer FL where clients can join asynchronously, reducing reliance on a central server.
	Shah and Lau (2023), Compression	Introduces gradient sparsification and binary masks to lower communication costs while preserving accuracy.
	Jiang et al. (2024), FedMp	Employs structured pruning with adaptive ratios based on client capability, improving speed in heterogeneous systems.
Statistical, system heterogeneity	Zhao et al. (2018), Data sharing	Mitigates non-IID effects by sharing a small portion of global data across all clients.
	Liu L. et al. (2020), Hier-FAVG	Introduces a client-edge-cloud hierarchy that reduces communication and energy costs in large-scale deployments.
	Sattler et al. (2019), CFL	Groups clients into clusters based on gradient similarity, then trains separate models for each cluster.
	Reisizadeh et al. (2020), FLANP	Allows progressive participation of slower clients to handle stragglers without stalling training.
	Luo et al. (2021), Adaptive sampling	Adjusts client selection probabilities according to gradient utility and computation speed.
	Li et al. (2024), FedLGA	Approximates missing gradients from resource-limited clients using Taylor expansion techniques.
	Menegatti et al. (2023), Hierarchical FL	Uses intermediate aggregation at edge servers to reduce bandwidth consumption and mitigate stragglers.
Meta-learning	Chen et al. (2019), FedMeta	Adapts meta-learning methods such as MAML and Meta-SGD to FL, enabling faster personalization under non-IID settings.
	Liu X. et al. (2024), Survey on FedMeta	Reviews extensions including ADMM-based methods, collaborative FL strategies, and wireless-specific optimizations.
Federated RL	Qi et al. (2021), FedRL	Proposes reward-aware aggregation and knowledge distillation to address challenges in averaging policies across agents.
	Zhuo et al. (2020), VFRL	Vertical FL setting for reinforcement learning, enabling secure gradient exchange between agents with complementary features.
Privacy	Wei et al. (2020), DP in FL	Applies Gaussian noise to model updates to achieve differential privacy, balancing accuracy with privacy budget.
	Zhang et al. (2020), BatchCrypt HE	Implements low-bit homomorphic encryption for efficient training with minimal accuracy drop.
	Fang and Qian (2021), Paillier FMLP	Optimizes homomorphic encryption to reduce the encryption and decryption overhead.
Architectures	Liu L. et al. (2020), Hierarchical FL	Multi-tier aggregation structure (client, edge, cloud) to alleviate communication bottlenecks.
	Briggs et al. (2020), Clustered HFL	Extends hierarchical FL with clustering for specialized models.
	Liu et al. (2024b), FINCH	Uses hierarchical neural architecture search within clustered clients.
	Liu et al. (2024a), FedMigr	Introduces reinforcement learning to guide model migration between clients.

(Continued)

TABLE 1 (Continued)

Category	Paper/technique	Key details/innovations
	Qu et al. (2020), Nguyen et al. (2021), and Chai et al. (2021), Blockchain FL	Combines FL with blockchain frameworks (FLchain, FL-Block, IoV-blockchain) for auditability and decentralization.
Applications	Gecer and Garbinato (2024), Autonomous driving	Demonstrates FL for perception models in autonomous driving, with emphasis on privacy-preserving training.
	Shaheen et al. (2022), Smart cities/IoT	Explores FL applications for transport, governance, and healthcare in smart cities.
	Diba et al. (2025), Deployment challenges	Examines resource constraints of IoT devices and their impact on FL training.
	Rjoub et al. (2025), IoT energy constraints	Analyzes energy-aware FL systems under limited device budgets.

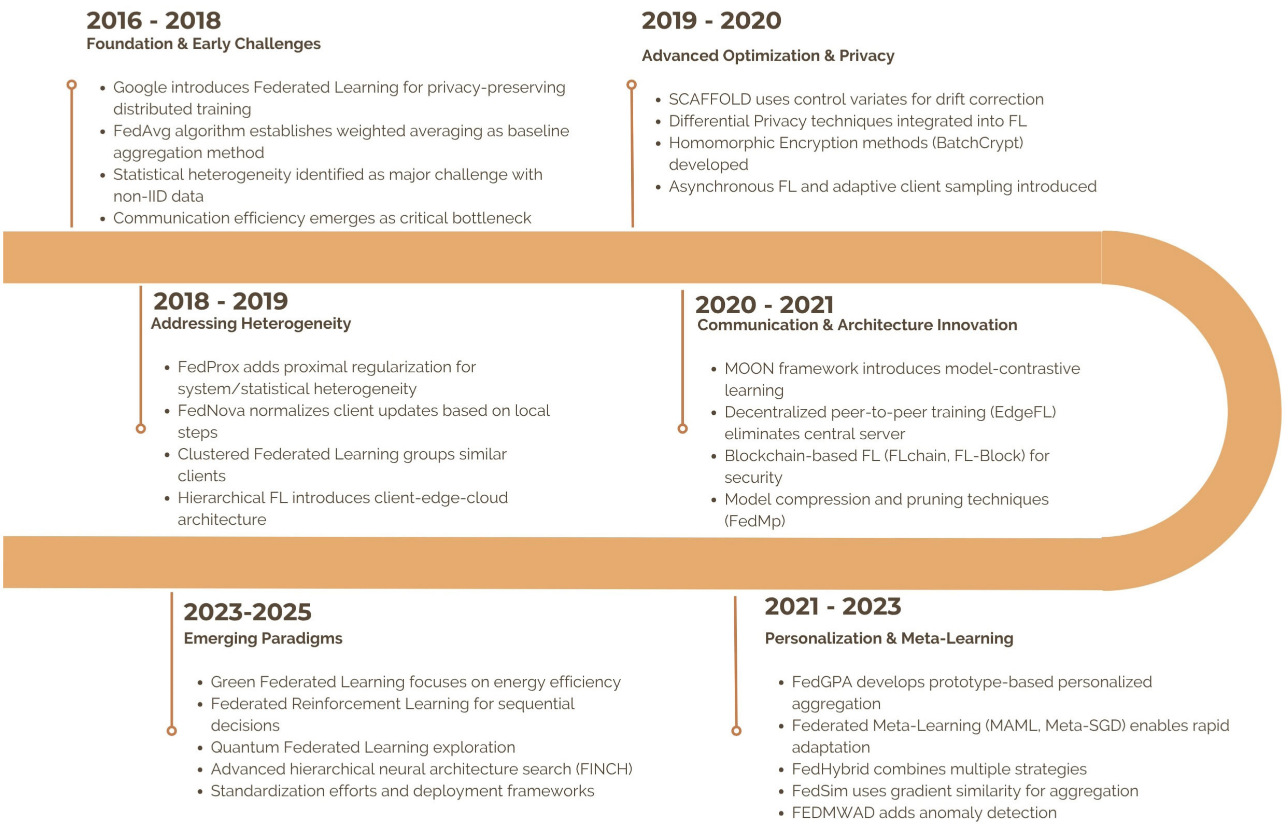
2016 - 2018
Foundation & Early Challenges
• Google introduces Federated Learning for privacy-preserving distributed training
• FedAvg algorithm establishes weighted averaging as baseline aggregation method
• Statistical heterogeneity identified as major challenge with non-IID data
• Communication efficiency emerges as critical bottleneck
2019 - 2020
Advanced Optimization & Privacy
• SCAFFOLD uses control variates for drift correction
• Differential Privacy techniques integrated into FL
• Homomorphic Encryption methods (BatchCrypt) developed
• Asynchronous FL and adaptive client sampling introduced
2018 - 2019
Addressing Heterogeneity
• FedProx adds proximal regularization for system/statistical heterogeneity
• FedNova normalizes client updates based on local steps
• Clustered Federated Learning groups similar clients
• Hierarchical FL introduces client-edge-cloud architecture
2020 - 2021
Communication & Architecture Innovation
• MOON framework introduces model-contrastive learning
• Decentralized peer-to-peer training (EdgeFL) eliminates central server
• Blockchain-based FL (FLchain, FL-Block) for security
• Model compression and pruning techniques (FedMp)
2023-2025
Emerging Paradigms
• Green Federated Learning focuses on energy efficiency
• Federated Reinforcement Learning for sequential decisions
• Quantum Federated Learning exploration
• Advanced hierarchical neural architecture search (FINCH)
• Standardization efforts and deployment frameworks
2021 - 2023
Personalization & Meta-Learning
• FedGPA develops prototype-based personalized aggregation
• Federated Meta-Learning (MAML, Meta-SGD) enables rapid adaptation
• FedHybrid combines multiple strategies
• FedSim uses gradient similarity for aggregation
• FEDMWAD adds anomaly detection

FIGURE 3
Federated learning: technological evolution (2016–2025).

maintained on a centralized server during each communication round. This process forms the backbone of the basic FL architecture. The aggregation is primarily based on the weight and gradient values generated by models trained on local devices. This section delves into significant deep learning-based model aggregation algorithms introduced in prior research, highlighting the advantages and limitations associated with each approach.

3.1.1 Federated averaging (FedAvg)

This primary aggregation method, introduced by McMahan et al. (2017) when FL was first introduced, uses an aggregation algorithm to update the weights of the deep neural network stored on the centralized server by selecting a fraction of clients from the total devices in each communication round. This approach uses a weighted averaging technique, considering the dataset size of each client device, which allows devices with larger datasets to have a greater impact on model updates. Additionally, it allows multiple local updates on each device before each communication round, improving communication efficiency by reducing the number of rounds needed for model convergence.

$$\boldsymbol{w}_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \boldsymbol{w}_k^{t+E} \tag{1}$$

The formula above shows how global updates are done in the FedAvg process. The total weight w_{t+1} in each communication round is calculated by summing the weighted average of each client device's weights for E number of epochs. Here, n_k represents the size of the client dataset, while n is the total dataset size across all clients. This aggregation method has been tested on the MNIST and CIFAR-10 datasets, using around 100 client devices with IID datasets. The MNIST dataset achieved 99.44% accuracy after 300 communication rounds and CIFAR-10 reached over 85% after 2,000 rounds. However, another research done by Zhao et al. (2018) shows a significant accuracy drop of up to 55% when FedAvg is applied to non-IID data due to data heterogeneity and model parameter divergence. They proposed an alternative method where a small portion (5%) of global data is shared with local devices, leading to a 30% accuracy improvement over FedAvg when trained with non-IID data.

3.1.2 FedProx algorithm

Since Fed Average uses a fixed number of epochs (local updates) for all client devices, which lacks convergence guarantees in realistic scenarios with non-IID data. Additionally, it can drop straggler clients (devices that cannot complete local updates due to processing speed variations), leading to biased global model updates. Since FedAvg includes multiple local updates per communication round, there is a risk of prioritizing local model updates over the globally aggregated model. To address both statistical and system heterogeneity issues in FedAvg, Sahu et al. (2018) introduced FedProx algorithm, which adds a proximal term for local model updates.

$$\min_{\boldsymbol{w}} h_k(\boldsymbol{w}; \boldsymbol{w}^t) = F_k(\boldsymbol{w}) + \frac{\mu}{2} \|\boldsymbol{w} - \boldsymbol{w}^t\|^2 \tag{2}$$

The above function shows that this approach adds a proximal term to the original loss function based on the clients' $F_k(w)$. The proximal term $\frac{\mu}{2}\|w - w^t\|^2$ calculates the squared Euclidean distance between the local model parameters (w) and the global model parameters (w^t). The hyperparameter μ serves as the regularization parameter to prevent the local model from deviating too far from the global model. By adjusting μ between 0 and 1, it helps maintain model stability despite deviations in local device model weights. The proximal function enables local model updates to bring model parameters closer to those of the global model without requiring a fixed number of epochs for local training. It also allows local models to aggregate with the global model after varying numbers of local updates, maintaining system heterogeneity. Practical experiments with synthetic and publicly available datasets (MNIST, FEMNIST, Shakespeare) were done using about 1,000 devices, with 0%, 50% and 90% of devices as stragglers. Straggler devices were created by assigning varying numbers of epochs and adjusting the μ value. Results showed that the FedProx algorithm achieved significantly lower loss values in fewer communication rounds than FedAvg, both with $\mu = 0$ and $\mu > 0$ conditions. Even though, it performs well with non-IID data the proximal term parameter μ has to be adjusted to find the optimal μ value. While FedProx effectively mitigates issues from system heterogeneity, its primary limitation is the introduction of the hyperparameter μ. This parameter requires careful, problem-specific tuning and an improper value can either fail to correct for heterogeneity or slow down convergence.

3.1.3 FedNova algorithm

The FedProx aggregation method was introduced to address system and statistical heterogeneity issues in FedAvg by incorporating a proximal term that aligns local model parameters with the global model. However, the need to calculate the Euclidean distance between model weights results in slower convergence and does not fully resolve objective inconsistency. To overcome this limitation, the "FedNova" approach, introduced by Wang et al. (2020), mitigates slower convergence by normalizing client model parameters and weight updates based on the number of local steps performed. Instead of averaging all local updates, FedNova averages normalized local gradients, ensuring that each client's contribution to the centralized model update is proportional to the true global objective, regardless of the number of local updates done before global communication.

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta \sum_{i=1}^{m} p_i \frac{\boldsymbol{g}_i^{(t)}}{\tau_i} \tag{3}$$

The formula above illustrates the normalization process in the FedNova algorithm. It shows that the global model $x^{(t+1)}$ is updated by subtracting a weighted sum of normalized client gradients from the current global model $x^{(t)}$. Each client's gradient $g_i^{(t)}$ is normalized by dividing by the total number of local updates τ_i and then weighted by p_i, representing the client's relative importance based on dataset size. The learning rate η scales the overall update. This ensures that each client's contribution to the global model is proportional to its actual progress, eliminating bias from varying local update numbers. Evaluations with synthetic and real-world data showed significant improvements over FedAvg and FedProx, achieving 6%–9% higher test accuracy on non-IID CIFAR-10 dataset with 30 clients. However, the approach does not fully address client dropout during training and requires tuning the normalization factor p_i for optimized results, raising concerns about communication efficiency in the aggregation process. The main cost of FedNova's improved handling of objective inconsistency is its implementation complexity. The normalization step requires tracking local update counts for each client, which can be difficult in asynchronous settings or when clients frequently drop out.

3.1.4 SCAFFOLD algorithm

Traditional federated learning algorithms like FedAvg struggle with statistical heterogeneity, where data distributions vary significantly across clients (non-IID data). This leads to a problem called "client-drift," where each client's local model moves toward its own local optimum, diverging from the true global objective. This drift makes convergence unstable, slow and can even cause the model to diverge, especially when clients perform many local updates before aggregation. To address this fundamental issue, the Stochastic Controlled Averaging (SCAFFOLD) algorithm introduces a drift-correction mechanism using control variates, a technique from variance reduction. The core idea is to estimate the client-drift and actively correct for it during local training (Karimireddy et al., 2021).

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta_l\big(g_i(\boldsymbol{y}_i) + \boldsymbol{c}_i - \boldsymbol{c}\big) \tag{4}$$

The formula above shows the corrected local update rule for client i, where the standard local gradient $g_i(y_i)$ is adjusted by the term $(c - c_i)$. SCAFFOLD maintains a state variable on the server (server control variate, c) and one for each client (client control variate, c_i). The server's variate approximates the global update direction, while the client's variate approximates its local update direction. The difference between them represents an estimate of the client-drift. By subtracting the client's update direction (c_i) and adding the server's global update direction (c), the update is steered away from the client's local optimum and back toward the global optimum, effectively synchronizing the clients' progress.

SCAFFOLD was evaluated on simulated data designed to highlight heterogeneity and on the real-world EMNIST dataset. The results show that SCAFFOLD consistently outperforms FedAvg and FedProx, especially in highly heterogeneous settings. Unlike FedAvg, which often performs worse with more local steps on non-IID data, SCAFFOLD's performance robustly improves, confirming its ability to handle client-drift. It achieves the same accuracy in significantly fewer communication rounds. However, SCAFFOLD introduces two main trade-offs. Firstly, it requires stateful clients, as each client must store its control variate c_i between communication rounds. Secondly, it doubles the upload communication cost because clients must send both the model update and the control variate update to the server in each round, which can impact communication efficiency in bandwidth-constrained environments.

3.1.5 MOON algorithm

As discussed above, traditional aggregation models like FedAvg struggle with non-IID data distributions across clients. To address this with fewer calculations, the Model-Contrastive Federated Learning (MOON) framework was introduced by Li et al. (2021). The primary goal of MOON is to reduce the distance between data representations learned by the local and global models during training. This framework consists of three components: a base encoder to extract weight representations as vectors, a projection head to map these representations into a fixed dimension and an output layer for predictions.

$$\ell_{\mathrm{con}} = -\log\left(\frac{\exp\left(\frac{\mathrm{sim}(z,z_{\mathrm{glob}})}{\tau}\right)}{\exp\left(\frac{\mathrm{sim}(z,z_{\mathrm{glob}})}{\tau}\right) + \exp\left(\frac{\mathrm{sim}(z,z_{\mathrm{prev}})}{\tau}\right)}\right) \tag{5}$$

The loss function above aims to bring the local device model's vector representation (z) closer to the global model's vector representation (z_{glob}), while distancing it from the previous state (z_{prev}) before updating the local model for each communication round. This is achieved by maximizing the numerator $\exp(\mathrm{sim}(z, z_{\mathrm{glob}})/\tau)$ relative to the denominator. As the similarity between the local and global model vector representations increases, the loss decreases. The temperature parameter (τ) regulates the alignment's intensity. Despite this alignment, MOON still uses the same weighted averaging method for local model aggregation as FedAvg. Evaluations on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets show that MOON algorithm demonstrates faster convergence and better communication efficiency. For example, MOON achieved 69.1% accuracy on CIFAR-10 dataset in 27 communication rounds, while FedAvg reached 66.3% accuracy after 100 rounds. Overall, MOON outperforms FedAvg by 2.6% on average across all datasets. However, MOON also has drawbacks, including higher computational costs, sensitivity to hyperparameters and limited applicability to non-vision tasks. MOON's performance gains come at a significant computational cost. The contrastive loss calculation requires storing previous model representations (zprev) and performing extra forward passes, increasing both memory and processing requirements on client devices, making it less suitable for highly resource-constrained environments.

3.1.6 FedHybrid algorithm

Standard Federated Learning algorithms like FedAvg perform well on IID data but struggle with convergence and accuracy in more realistic non-IID settings. While subsequent methods like FedProx and FedScaffold were developed to address specific challenges data heterogeneity and client drift, respectively they each tackle only a part of the problem. This leaves a gap for a more comprehensive solution that can simultaneously handle multiple sources of instability in non-IID environments. To bridge this gap, the FedHybrid algorithm was introduced as a novel aggregation strategy that unifies the strengths of three foundational FL methods: FedAvg, FedProx and FedScaffold (Niu and Wei, 2023).

$$\boldsymbol{w}_g^{(t+1)} = \sum_{i=1}^{m} \alpha_i \Big[(1-\mu)\boldsymbol{w}_i + \mu \boldsymbol{w}_g + (\boldsymbol{c}_i - \boldsymbol{c}_g)\Big] \tag{6}$$

The formula above illustrates the FedHybrid aggregation process, where the global model $w_g^{(t+1)}$ is updated using normalized accuracy-based weights α_i that give more influence to better-performing clients. The algorithm combines three key components: the proximal term μ that prevents local models from deviating too far from the global model, the local model weights w_i and control variates $(c_i - c_g)$ that correct for client drift by tracking the difference between local and global control variables. During local training, clients incorporate a proximal term $\frac{\mu}{2}\|w - w_g\|^2$ into their loss function to mitigate statistical heterogeneity, while control variates reduce variance between local updates and the global objective.

Evaluations on MNIST and CIFAR-10 datasets with non-IID distribution across 100 clients showed that FedHybrid significantly outperformed existing methods, achieving 94.12% accuracy on MNIST and 93.52% on CIFAR-10 with faster convergence rates. However, the primary limitation of FedHybrid is the increased complexity and number of hyperparameters that require careful tuning, including the proximal term weight μ and learning rates for control variates. The algorithm's computational overhead from managing multiple components simultaneously may impact communication efficiency and its evaluation was limited to image classification datasets, raising questions about generalizability to other domains.

3.1.7 FedSim algorithm

Traditional aggregation algorithms, such as the foundational FedAvg (McMahan et al., 2017), treat all participating client models equally during the aggregation process. While methods like FedProx (Sahu et al., 2018) address heterogeneity by allowing for variable local updates, they still do not explicitly account for potential gradient similarities between subsets of client models. This can result in reduced stability of the global model in highly heterogeneous settings, leading to significant performance fluctuations across communication rounds. To address this issue, researchers have explored methods for clustering local devices that produce similar data distributions. However, sharing local device data to train a model that clusters such devices would violate the data privacy principles inherent in FL. To bridge this gap, Palihawadana et al. (2022) introduced the FedSim aggregation method, a similarity-guided model aggregation strategy that leverages client gradient similarities without compromising privacy. Firstly, the system will cluster the client devices into random clusters using an algorithm like k-means in the first communication round, taking into account the data sharing constraints of FL. Then the selected clients for a communication round are clustered based on the local weight gradients, which are calculated based on the error value between the existing global model. Subsequently, the respective selected clients for communication will update their gradients using an optimizer like stochastic gradient descent (SGD). As the fourth step, the locally updated model weights are aggregated with their respective cluster model by using the weighted averaging technique. The weights are considered based on the sample data set sizes included in each client device, respective to the whole cluster size. Finally, the global aggregation has been done using the updated weights of each specialized cluster model produced by the selected client devices in each communication round. FedSim was evaluated using popular datasets such as MNIST, FEMNIST, Fed-MEX, Fed-Goodreads and synthetic data. The results demonstrated that FedSim achieved higher accuracy and required fewer communication rounds compared to both FedAvg and FedProx algorithms. For example, on the MNIST dataset, FedSim outperformed FedAvg by 11.69% and FedProx by 17.25%, with comparatively fewer communication rounds. However, despite employing Principal Component Analysis (PCA) for dimensionality reduction during the clustering process, FedSim exhibited higher computational costs, resulting in longer communication round times compared to FedAvg and FedProx, highlighting the limitations in communication efficiency associated with this approach.

3.1.8 FedGPA algorithm

As discussed above, traditional Federated Learning approaches like FedAvg experience significant performance drops when dealing with non-uniform data, where different clients have varying data patterns. To address this challenge, Personalized Federated Learning methods have been developed, but they typically focus on either improving client-side model adjustments or server-side combination strategies separately. The FedGPA (Federated Learning with Global Personalized Aggregation) algorithm was introduced to create a more comprehensive framework that simultaneously enhances local training and employs a sophisticated, personalized combination mechanism on the server (Kairouz et al., 2021). The core of FedGPA algorithm lies in separating the model into a feature extractor (for learning generalizable representations) and a classifier (for personalization) and applying distinct optimization and combination strategies to each.

To prevent local models from drifting too far from the global objective due to local data bias, FedGPA introduces a regularization term during local training. This term aligns the client's local data representations with the global data distribution using prototypes (the average feature vector for each data class). The local loss function for client i is modified as:

$$L_i = L_s(w_f, w_c; D_i) + R_i(w_f; C) \tag{7}$$

where L_s is the standard supervised loss and R_i is the prototype-based regularization loss. The regularization term measures the distance between local and global class prototypes. Local prototypes are calculated as the average of feature vectors for each class on a specific client, while global prototypes are computed on the server by taking a weighted average of all clients' local prototypes. A hyperparameter balances the supervised loss and the regularization term. This alignment encourages the local feature extractor to learn representations that are consistent with the global data structure.

FedGPA computes unique combination weights for each client, personalizing both the feature extractor and the classifier. For the feature extractor, the combination weight (the weight client i assigns to client j's model) is calculated based on both prototype similarity and sample size. The prototype similarity represents the inverse of the distance between the prototype sets of different clients, serving as a similarity score. The sample size represents the number of data samples on each client. A hyperparameter balances the influence of prototype similarity and sample size. For the classifier, the combination weights are determined by solving an optimization problem that considers both client similarity matrix and the variability of features within each client's data.

Evaluations on five benchmark datasets (FMNIST, EMNIST, CIFAR-10, CIFAR-100, and CINIC-10) under three different non-uniform data scenarios demonstrated that FedGPA consistently achieved higher test accuracy compared to state-of-the-art baselines like FedProx and SCAFFOLD aggregation methods. However, the primary limitation of FedGPA is its increased computational and communication overhead. The server needs to receive prototypes from each client, compute global prototypes and solve an optimization problem for each client's classifier in every round, making each communication round more time-consuming than simpler algorithms like FedAvg or FedProx, highlighting a trade-off between model accuracy and system efficiency.

3.1.9 FEDMWAD algorithm

While algorithms like FedProx and FedNova address system and statistical heterogeneity, they largely operate under the assumption that all client contributions are benign. This leaves them vulnerable to performance degradation from outlier clients, whose model updates may be anomalous due to extreme non-IID data, system faults, or even malicious intent (e.g., data poisoning attacks). Such anomalous updates can destabilize the global model and slow down or prevent convergence. To address this vulnerability, the Federated Multi-Weighted Aggregation with Anomaly Detection (FEDMWAD) algorithm was proposed by Ding et al. (2025). FEDMWAD introduces a server-side validation mechanism that identifies and down-weights anomalous client updates before aggregation.

$$\boldsymbol{w}_{t+1} = \sum_{k=1}^{K} \alpha_k \frac{n_k}{N} \boldsymbol{w}_{t+1}^{k}, \quad \alpha_k = f(\mathrm{sim}(\Delta \boldsymbol{w}_k, \Delta \boldsymbol{w}_{\mathrm{median}})) \tag{8}$$

The formula above demonstrates FEDMWAD's dynamic re-weighting mechanism, where the global model update w_{t+1} is computed as a weighted average with an additional anomaly-based weighting factor α_k. The server first computes a reference update using the geometric median of all received client model updates $\Delta w_{\mathrm{median}}$, which is robust to outliers. It then measures the similarity between each client's update Δw_k and this reference using metrics such as cosine similarity. The function f maps this similarity score to the weight α_k, assigning lower weights to clients whose updates deviate significantly from the norm and higher weights to those in consensus. This process effectively filters out or minimizes the impact of potentially harmful updates, making the aggregation more robust against anomalous client behavior.

In evaluations on CIFAR-10 and synthetically-generated datasets with simulated label-flipping attacks, FEDMWAD demonstrated significant improvements in model stability and final accuracy. In scenarios with 20% malicious clients, FEDMWAD achieved final accuracy 8% higher than FedAvg and 5% higher than FedProx, while also showing much lower variance in accuracy across communication rounds. However, the primary drawback of FEDMWAD is the increased computational overhead on the central server, as it must collect all updates before computing the median and similarity scores, which increases the time per communication round. Furthermore, the algorithm introduces new hyperparameters, such as the choice of similarity metric and the sensitivity of the weighting function f, which require careful tuning for optimal performance and may impact the algorithm's practical deployment in diverse federated learning environments.

3.2 Federated meta-learning

Federated Meta-Learning (FedMeta) represents an advanced paradigm within federated learning, specifically designed to address the inherent challenges of statistical and systemic heterogeneity across clients. This approach shifts the learning objective from the direct optimization of a model's parameters to the optimization of the learning process itself. Consequently, the framework enables the distributed system to "learn how to learn" efficiently across decentralized devices.

The fundamental divergence between FedMeta and traditional federated learning algorithms, such as Federated Averaging (FedAvg), lies in their core objective and the nature of the information exchanged (Chen et al., 2019). Whereas conventional federated learning aims to train a single, robust global model through the iterative aggregation of model weight updates, FedMeta's objective is to derive an adaptive algorithm, or a "meta-learner." Instead of exchanging full model parameters, clients in the FedMeta framework share this parameterized algorithm, which they then rapidly adapt to their specific local tasks. This distinction allows FedMeta to more effectively accommodate non-IID data distributions, often yielding significant gains in communication efficiency and convergence speed.

The operational framework of FedMeta, as proposed by Chen et al. (2019), is predicated on a bi-level optimization structure. A central server first distributes a parameterized algorithm the meta-learner to a subset of participating clients. Each client then utilizes its local dataset to execute a "fast adaptation" step, which typically involves a limited number of gradient descent iterations. During this inner update, the client fine-tunes the meta-learner's parameters to align with its unique local data distribution. Following this local adaptation, clients transmit concise performance feedback, such as the loss incurred on a local test set or the gradients derived from this loss, rather than their fully adapted models. The server aggregates this feedback to perform an outer update, which refines the meta-learner algorithm. This iterative process is designed to enhance the meta-learner's capacity to facilitate rapid and effective model adaptation for future tasks. The efficacy of FedMeta is significantly informed by the principles of prominent meta-learning algorithms, notably Model-Agnostic Meta-Learning (MAML) and Meta-SGD (Liu X. et al., 2024).

3.2.1 Model-agnostic meta-learning (MAML)

The central objective of MAML is to identify an initial set of model parameters, denoted as θ, that can be rapidly fine-tuned for new tasks with a minimal number of gradient descent steps. The optimization process is structured into two distinct stages: a client-side inner update for local adaptation and a server-side outer update for meta-optimization.

For the inner update, each client i partitions its local data into a support set ($D_S^{T_i}$) for training and a query set ($D_Q^{T_i}$) for evaluation. Beginning with the shared parameters θ from the server, the client performs one or more gradient descent steps on its support set. This local training yields a set of task-specific adapted parameters, ϕ_i. The formulation for a single gradient step is:

$$\boldsymbol{\phi}_i = \boldsymbol{\theta} - \alpha \, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{T_i}\left(f_{\boldsymbol{\theta}}, D_S^{T_i}\right) \tag{9}$$

In this formulation, f_θ represents the model parameterized by θ, and $\mathcal{L}_{T_i}(f_\theta, D_S^{T_i})$ is the loss function measuring the model's performance on the support set for its local task T_i. The term ∇_θ denotes the gradient of this loss with respect to the parameters θ. The update rule adjusts θ in the direction opposite to the gradient, scaled by a local learning rate α, to produce the adapted parameters ϕ_i.

The process then transitions to the server-side outer update. After deriving their adapted parameters ϕ_i, clients evaluate the performance of the adapted model, f_{ϕ_i}, on their respective query sets. The resulting performance feedback is transmitted to the central server, which aggregates it to update the global initial parameters θ. This meta-optimization step aims to refine θ such that future inner updates result in superior performance on the query sets. The outer update is expressed as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \, \nabla_{\boldsymbol{\theta}} \sum_{T_i} \mathcal{L}_{T_i}\left(f_{\boldsymbol{\phi}_i}, D_Q^{T_i}\right) \tag{10}$$

Here, $\sum_{T_i} \mathcal{L}_{T_i}(f_{\phi_i}, D_Q^{T_i})$ is the aggregated loss of the adapted models across the query sets of all participating clients. The meta-learning rate β scales the update to the initial parameters θ. A critical aspect of this update is that the gradient is taken with respect to θ. Since ϕ_i is a function of θ, this calculation necessitates computing gradients through the inner update step, thus requiring second-order derivatives (gradients of gradients). MAML, therefore, learns an initialization that is highly sensitive to task-specific fine-tuning.

3.2.2 Meta-SGD

Meta-SGD extends the MAML framework by learning not only an optimal set of initial parameters but also a corresponding per-parameter learning rate for the inner adaptation process. Instead of employing a single scalar learning rate α, Meta-SGD learns a vector of learning rates, α_{vec}, where each element corresponds to a specific model parameter.

The client-side inner update is modified to incorporate these adaptive learning rates. Upon receiving both the initial parameters θ and the learning rate vector α_{vec} from the server, each client performs a local adaptation on its support set. The update for each parameter in θ is now scaled by its unique learning rate from α_{vec}. The local adaptation is guided by the following formula:

$$\boldsymbol{\phi}_i = \boldsymbol{\theta} - \boldsymbol{\alpha}_{\text{vec}} \odot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{T_i}\left(f_{\boldsymbol{\theta}}, D_S^{T_i}\right) \tag{11}$$

The symbol $\odot$ denotes the Hadamard product, signifying an element-wise multiplication between the learning rate vector and the gradient vector. This mechanism allows for a more flexible adaptation, where different parameters can be updated at different rates according to their learned sensitivity.

Subsequently, the server-side outer update performs a joint meta-optimization for both the initial parameters θ and the learning rate vector α_{vec}. The server aggregates the query set performance from clients and computes gradients with respect to both meta-learned components. The resulting updates are:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta_1 \, \nabla_{\boldsymbol{\theta}} \sum_{T_i} \mathcal{L}_{T_i}\left(f_{\boldsymbol{\phi}_i}, D_Q^{T_i}\right) \tag{12}$$

$$\boldsymbol{\alpha}_{\text{vec}} \leftarrow \boldsymbol{\alpha}_{\text{vec}} - \beta_2 \, \nabla_{\boldsymbol{\alpha}_{\text{vec}}} \sum_{T_i} \mathcal{L}_{T_i}\left(f_{\boldsymbol{\phi}_i}, D_Q^{T_i}\right) \tag{13}$$

Distinct meta-learning rates, β_1 and β_2, are used for updating θ and α_{vec}, respectively. Similar to MAML, this outer update requires the computation of second-order derivatives, as the query loss is a function of the adapted parameters ϕ_i, which in turn depend on both θ and α_{vec}. By learning both an optimal initialization and individualized learning rates, Meta-SGD provides a more powerful and flexible meta-learner, often leading to enhanced convergence and robustness in heterogeneous federated settings.

The theoretical underpinnings of these Federated Meta-Learning frameworks translate into tangible performance improvements when empirically evaluated against baseline federated optimization strategies. The research conducted by Chen et al. (2019) offers a rigorous comparative analysis against Federated Averaging (FedAvg), revealing the practical efficacy of the FedMeta framework in environments characterized by statistical heterogeneity. A primary and significant finding is the superior accuracy and accelerated convergence exhibited by FedMeta. This advantage is clearly quantified in Table 2, which details the performance on the LEAF benchmark. On the FEMNIST image classification task, the FedMeta variants consistently achieve accuracies around 90%, representing a substantial improvement over the roughly 77% reached by FedAvg. This trend holds for the natural language processing tasks as well, with Meta-SGD reaching 80.94% accuracy on Sent140 compared to FedAvg's 73.38%. Performance graphs from the study further illustrate that MAML and Meta-SGD not only converge to this considerably higher final accuracy plateau but also exhibit a steeper initial learning curve, indicating that they reach a state of high performance in significantly fewer communication rounds. This performance differential is a direct consequence of the two frameworks' contrasting approaches to handling non-IID data. FedAvg aims to learn a single global model by averaging client updates, a process that inherently seeks a consensus model that often represents a poor compromise in heterogeneous settings. In stark contrast, FedMeta learns a meta-model that serves as an optimized initialization, explicitly trained to be adaptable and thus providing a far more effective starting point for rapid personalization on diverse local data.

Furthermore, this accelerated convergence directly addresses the principal bottleneck in most federated learning applications: communication overhead. The research highlights that FedMeta provides a compelling solution for mitigating this challenge, demonstrating a reduction in the total required communication cost by a factor of 2.82–4.33 times when compared to FedAvg. This efficiency stems from the fact that the meta-learner generalizes more effectively, necessitating less frequent and prolonged iterative refinement between the server and clients to achieve a desired performance threshold. This reduction in communication,

TABLE 2 Comparison of performances of FedAvg and FedMeta learning on LEAF benchmark datasets.

Paper	Method used	Model used	Evaluated dataset	Support fraction	Evaluations	#Comm. rounds	Device count
Chen et al. (2019)	FedAvg	CNN	FEMNIST	20%	76.79%	2,000	1,068
				50%	75.44%		
				90%	77.05%		
		Stacked LSTM	Shakespeare	20%	40.76%	400	528
				50%	42.01%		
				90%	40.85%		
		LSTM classifier	Sent140	20%	71.53%	400	3,790
				50%	72.29%		
				90%	73.38%		
Chen et al. (2019)	FedMeta (MAML)	CNN	FEMNIST	20%	88.46%	2,000	1,068
				50%	89.77%		
				90%	89.31%		
		Stacked LSTM	Shakespeare	20%	46.06%	400	528
				50%	46.29%		
				90%	46.49%		
		LSTM classifier	Sent140	20%	76.37%	400	3,790
				50%	78.63%		
				90%	79.53%		
Chen et al. (2019)	FedMeta (Meta-SGD)	CNN	FEMNIST	20%	89.26%	2,000	1,068
				50%	90.28%		
				90%	89.31%		
		Stacked LSTM	Shakespeare	20%	44.72%	400	528
				50%	45.24%		
				90%	46.25%		
		LSTM classifier	Sent140	20%	77.24%	400	3,790
				50%	79.38%		
				90%	80.94%		

however, is accompanied by an increase in local computational demands. The bi-level optimization at the core of MAML and Meta-SGD requires the computation of second-order derivatives, a "meta-gradient" calculation that is more expensive for client devices than the standard first-order gradient descent in FedAvg. This presents a critical design trade-off: FedMeta is demonstrably superior in communication-constrained or high-latency networks, whereas FedAvg may remain a viable option if client devices have severely limited computational capacity.

Beyond aggregate metrics, the equity of performance distribution across the client population which is a concept known as fairness, is a vital consideration where FedMeta shows a distinct advantage. The empirical evidence strongly suggests that FedMeta fosters a more equitable performance distribution, as observed through kernel density estimations of final client accuracies. The distributions for MAML and Meta-SGD are significantly more centered and exhibit lower variance, indicating that a large majority of clients achieve a similarly high level of accuracy. Conversely, the distribution for FedAvg is typically wider and flatter, signifying high performance variance and the presence of poorly performing clients. This enhancement in fairness is directly attributable to the superior generalization capabilities of the meta-learning objective. The framework is explicitly designed to produce a meta-learner that is readily adaptable across a diverse set of tasks, capturing a more fundamental and transferable representation of the network's collective knowledge. As a result, FedMeta can effectively personalize models even for outlier clients with unique data distributions or limited data, directly addressing a key failure mode of the single-global-model approach and thereby reducing performance disparity.

3.2.3 Advancements and applications in wireless networks

While the MAML and Meta-SGD frameworks form the bedrock of FedMeta, the research landscape has since expanded to address more complex optimization challenges and deployment scenarios, as surveyed by Liu X. et al. (2024). These advancements aim to improve computational efficiency and knowledge transfer within the unique constraints of wireless networks. One notable evolution is Collaborative FedMeta, which formalizes a platform-based learning framework where a set of "source" edge nodes collaboratively trains a meta-model for subsequent transfer and rapid adaptation at a "target" edge node. To tackle the high computational cost of bi-level optimization, researchers have also proposed ADMM-FedMeta, which leverages the Alternating Direction Method of Multipliers (ADMM) to decompose the meta-optimization into parallelizable sub-problems, enhancing scalability for large-scale networks.

The theoretical advantages of FedMeta translate into tangible solutions for critical operational challenges within wireless communications. The framework moves beyond general model training to actively optimize network performance and resource management. For instance, in a massive wireless network, FedMeta enables intelligent device selection schemes. By rigorously analyzing the contribution of each device to the global loss reduction, the system can prioritize participants whose updates will most effectively improve the meta-learner, accelerating convergence while minimizing communication latency and energy consumption. This is particularly crucial for battery-constrained devices such as IoT sensors, where energy efficiency is paramount. Specialized frameworks, such as a "meta-backward" algorithm, have been developed to learn a meta-model with low computation and communication energy overhead, making sophisticated on-device intelligence viable for resource-constrained hardware.

Despite its significant promise, the widespread deployment of FedMeta faces several open challenges that constitute the future research frontier. While FedMeta inherits the privacy benefits of FL, the shared meta-learner may be vulnerable to novel attack vectors, and further investigation into its specific privacy guarantees is essential. Moreover, as networks scale to billions of devices, the overhead of a single meta-learner may become prohibitive. This has motivated research into Multi-Model FedMeta, where devices are clustered to train specialized meta-models in a hierarchical fashion. Perhaps most critically, a research gap exists in the theoretical analysis of FedMeta under realistic wireless conditions. Characterizing its convergence properties in the presence of channel noise, transmission errors, and device dropout is vital for designing robust algorithms that can perform reliably in the unpredictable environments of future 6G networks.

3.3 Federated reinforcement learning

Federated Reinforcement Learning (FRL) represents a paradigm that combines the privacy-preserving architecture of Federated Learning (FL) with the sequential decision-making framework of Reinforcement Learning (RL). This approach enables multiple agents to collaboratively learn optimal strategies from their experiences without sharing sensitive raw data.

The fundamental distinction from traditional FL lies in the learning objective and data characteristics. While conventional FL focuses on supervised tasks with the goal of minimizing a predictive loss function over static, labeled datasets, FRL aims to discover a policy π_θ parameterized by θ that maximizes the expected cumulative reward $J(\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$, where R_t represents the reward at time t and $\gamma \in [0, 1]$ is the discount factor. This objective is learned from dynamic streams of experience tuples rather than fixed datasets.

The transition from minimizing static prediction errors to maximizing dynamic behavioral rewards introduces a fundamental challenge known as the *policy-averaging problem* (Qi et al., 2021). Traditional Federated Averaging (FedAvg), which computes weighted parameter averages $\theta_{global} = \sum_{i=1}^{N} \frac{n_i}{n}\theta_i$, where n_i is the number of samples from client i and $n = \sum_{i=1}^{N} n_i$, can be detrimental when applied to RL policies. Averaging policies from agents trained in disparate but valid environments may result in incoherent global policies.

FRL methodologies are categorized into two primary structures based on data distribution characteristics. Horizontal FRL (HFRL) applies when agents share identical state and action spaces but possess different experiences. Research in this domain focuses on developing intelligent aggregation mechanisms beyond simple parameter averaging. Key approaches include performance-based weighting, where agents with higher reported rewards receive greater influence in the aggregation process and knowledge distillation, where a central model learns to emulate successful agent decisions rather than blending internal parameters.

Vertical FRL (VFRL) addresses scenarios where agents observe different, complementary features of the same environment (Zhuo et al., 2020). Since agents operate on distinct feature spaces, direct policy averaging is not applicable. Instead, VFRL employs sophisticated protocols involving secure exchange of intermediate computations, such as encrypted gradients or feature representations, with a central coordinator.

The synergy between RL and FL extends beyond FRL applications. RL techniques are increasingly employed to address operational challenges within traditional FL systems. Central servers can deploy RL agents to optimize client selection by observing network states and learning policies that maximize long-term objectives such as convergence speed or final model accuracy.

3.4 Communication efficiency

Communication efficiency in cross-device Federated Learning (FL), which connects millions of devices, poses a significant challenge. Aggregating model weights becomes complex and time-consuming, especially with slow or disconnected devices (stragglers) causing delays (Zhang et al., 2024). To address this, techniques have been developed to optimize local model weights and reduce communication rounds, improving efficiency while maintaining model performance. Additionally, researchers are exploring decentralized training, model compression and pruning methods to further reduce communication costs in federated networks.

3.4.1 Decentralized training

Decentralized model training has emerged as a significant alternative to centralized FL, particularly in addressing concerns related to communication efficiency. As visually contrasted in Figure 4, the decentralized approach fundamentally transforms the traditional FL architecture by removing the central server bottleneck. In this innovative architecture, client devices are connected in a peer-to-peer (P2P) network, enabling direct communication between them and eliminating the need for a central server. This approach mitigates network bottlenecks and enhances scalability (Qi et al., 2024).

The architectural comparison illustrated in Figure 4 demonstrates how the centralized approach (left) creates a single point of failure and potential bottleneck, where all devices must communicate through a central server. In contrast, the decentralized approach (right) allows devices to share model updates directly with neighboring devices, creating a more resilient and efficient communication network. This peer-to-peer communication pattern significantly reduces the communication overhead and latency associated with traditional centralized FL systems.

Zhang et al. (2024) proposed EdgeFL, a decentralized FL framework that facilitates model aggregation without relying on a centralized server. The implementation of this architecture is achieved through the integration of a customized function within the PyTorch framework, which supports asynchronous joining of new client devices during the training process. Furthermore, the adoption of an API calling method simplifies the decentralized training procedure, reducing its complexity. Evaluations conducted on the CIFAR-10 and MNIST datasets have demonstrated that EdgeFL significantly reduces weight updating latency and model evaluation time by approximately 50% compared to existing decentralized FL methods. Additionally, EdgeFL shows improvements in accuracy, achieving a 2% increase for CIFAR-10 and a 5% increase for MNIST, when compared to the decentralized FedAvg approach.

3.4.2 Model compression and model pruning

To reduce communication volumes, three primary model compression techniques are employed: upstream compression (reducing client model size before aggregation), downstream compression (compressing the centralized model before downloading it to local devices) and local computation reduction (modifying the training algorithm to create more generalized models with limited resources). Shah and Lau (2023) introduced a compression technique combining both downstream and upstream methods with L1 regularization. They proposed two upstream compression algorithms, using an auxiliary binary mask and a sparse subnetwork for CNN models in federated networks, that effectively reduce model size without compromising accuracy. Evaluations of this approach on the CIFAR-10, CIFAR-100 and Fashion-MNIST datasets demonstrated an improvement of 0.75%–3.9% in accuracy, with fewer communication rounds compared to existing methods.

In addition, model pruning, both unstructured and structured, is applied in FL to enhance communication efficiency. Unstructured pruning removes individual weights, while structured pruning eliminates entire layers, including batch normalizers. The effect of this process transforms a dense neural network into a sparser, more computationally efficient model by selectively removing nodes and connections, as illustrated in Figure 5. This transformation process demonstrates how pruning techniques can significantly reduce the model's complexity while maintaining its essential functionality. The pruned network exhibits fewer parameters and connections, resulting in reduced memory requirements and faster computation times during both training and inference phases.

Figure 5 visually demonstrates the pruning process, where some nodes and connections are selectively removed (shown in lighter colors) from the original dense network (dark green) to create a sparser, yet functional network. This systematic refinement of the network architecture eliminates redundant or less important connections, which not only enhances communication efficiency by reducing the amount of data that needs to be transmitted between devices but also improves the overall computational efficiency of the federated learning process.

Jiang et al. introduced an adaptive pruning method, "FedMp," which utilizes structured pruning, allowing the central server to adjust pruning ratios according to the capabilities of heterogeneous worker devices. This adaptive approach recognizes that different devices in a federated network may have varying computational capabilities and communication constraints. Evaluations of this method showed up to a 4.1× speed improvement over other FL methods when trained on datasets like MNIST, CIFAR-10 and Tiny-ImageNet using models such as AlexNet, VGG-19 and ResNet-50 (Jiang et al., 2024). The significant performance improvements demonstrate the effectiveness of adaptive pruning in optimizing federated learning systems for diverse device capabilities while maintaining model accuracy.

3.5 Statistical heterogeneity

Statistical heterogeneity in Federated Learning (FL) refers to the variation in data distributions across local devices (clients), where each client collects and processes its own data. Unlike centralized learning, where data is from a common pool, FL faces non-IID (independent and identically distributed) data, where feature, label, or task distributions differ significantly across clients. For example, in healthcare, hospitals may have patient data with different demographics and disease distributions, making it challenging to train a global model that generalizes well across all clients.

To address this fundamental challenge, various strategies have been developed to reduce the effects of non-IID data. One approach is data sharing, where a small global dataset is distributed to clients. Zhao et al. (2018) showed that sharing 5% of global data increased accuracy by 30% on non-IID data. This method is useful in applications like mobile phone personalization, where data distributions vary and privacy concerns prevent data centralization.

Liu L. et al. (2020) proposed a client-edge-cloud hierarchical FL system with the Hier-FAVG algorithm, which uses edge servers to partially aggregate client models before sending them to the cloud. This approach reduces communication overhead and mitigates the effects of non-IID data. Experiments on MNIST and CIFAR-10 datasets showed faster convergence and reduced energy consumption compared to traditional cloud-based FL.

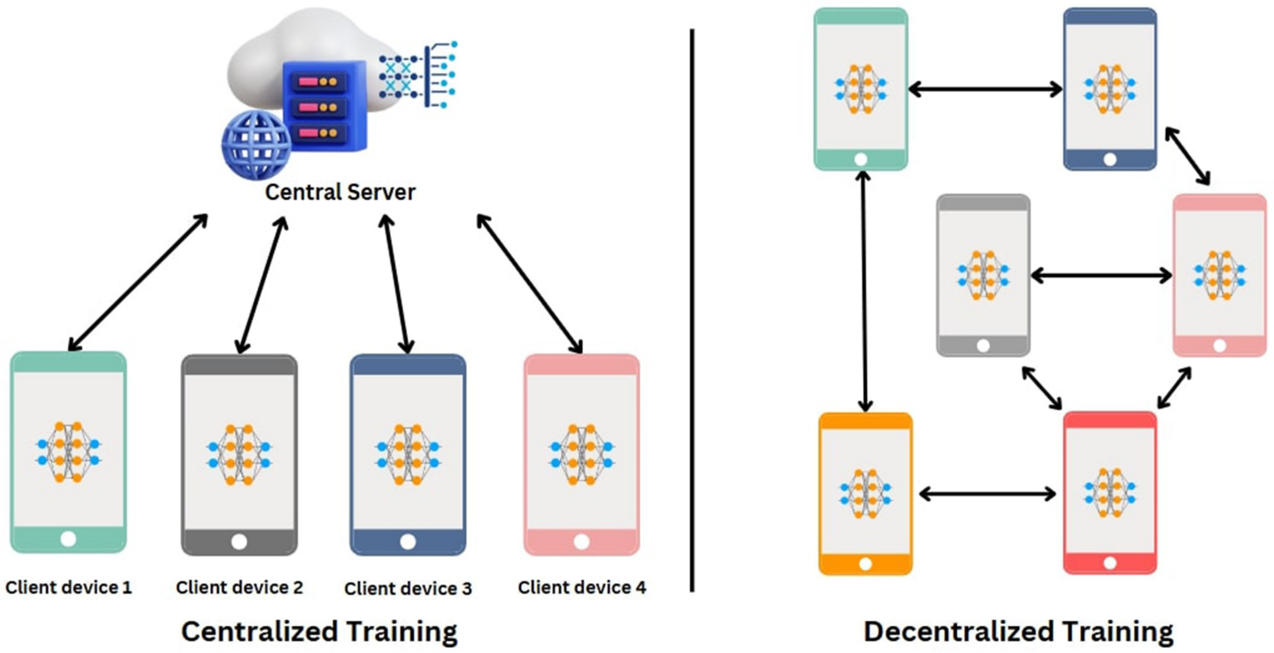

FIGURE 4
FL architectures: centralized training **(left)**, where devices send model updates to a central server creating a potential bottleneck and decentralized training **(right)**, where devices share updates directly with each other through peer-to-peer communication, eliminating the central server dependency and enhancing scalability.

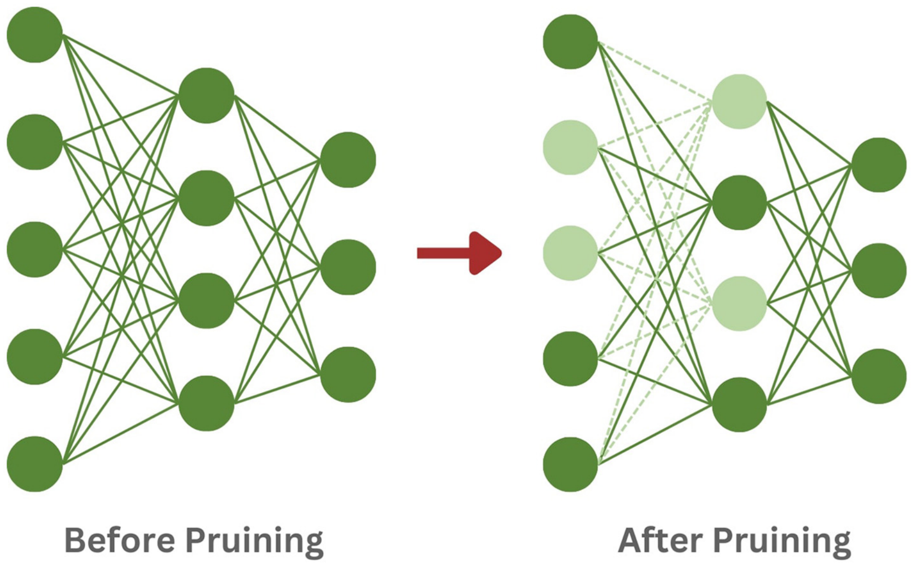

FIGURE 5
This figure shows neural network pruning, where some nodes and connections are selectively removed (shown in lighter colors) from the original dense network (dark green) to create a sparser, yet functional network.

Sattler et al. (2019) introduced Clustered Federated Learning (CFL), which groups clients based on similarities in data distributions using cosine similarity of gradient updates, preserving privacy. Experiments on MNIST and CIFAR-10 demonstrated that CFL significantly improved model accuracy, effectively doubling accuracy on CIFAR-10 under non-IID conditions. A real-life application could be in personalized healthcare, where patient data from different demographics is clustered for more accurate predictive modeling.

Model Regularization helps maintain a balance between local and global models during training. By adding regularization terms to the local optimization objective, this method prevents excessive divergence between client models. FedProx, for example, introduces a proximal term to the local objective, encouraging local models to stay close to the global model, thus facilitating better convergence in heterogeneous settings (Sahu et al., 2018).

Further advancements have explored adaptive client participation and sampling strategies to jointly manage statistical and system heterogeneity. Reisizadeh et al. (2020) proposed Federated Learning with Adaptive Node Participation (FLANP), a meta-algorithm that initiates training with only the fastest clients and progressively incorporates slower clients as the model achieves sufficient accuracy on the existing set. At each stage, the model from the previous round serves as a warm start, enabling efficient integration of additional clients. This staged participation allows the global model to learn from increasingly diverse data while maintaining efficiency, with experiments demonstrating up to a 6× reduction in wall-clock training time compared to conventional FL benchmarks.

Luo et al. (2021) introduced an adaptive client sampling algorithm that directly addresses statistical and system heterogeneity by optimizing client selection probabilities. Instead of uniform sampling, their method formulates a non-convex optimization problem based on a convergence bound for FL with arbitrary sampling. Clients with both faster communication speeds and statistically valuable data, as measured by local gradient information, are given higher probabilities of selection. Experiments on the EMNIST dataset showed that this adaptive sampling approach reduced the time required to reach a target loss by 73% compared to uniform sampling.

Li et al. (2024) focused on mitigating the bias introduced by incomplete local updates due to device heterogeneity. FedLGA operates at the aggregator side, using a Taylor expansion to approximate the full local gradient update that resource-constrained clients would have produced if they had completed all local training epochs. This ensures that contributions from limited-capacity clients on non-IID data are not underestimated or biased. On a non-IID partitioned CIFAR-10 dataset, FedLGA improved the testing accuracy from 60.91% (FedAvg baseline) to 64.44%, highlighting its effectiveness in handling heterogeneous updates.

3.6 System heterogeneity

As described by Mang et al., system heterogeneity in FL refers to variations in computational resources, network connectivity and energy constraints among distributed devices. These devices, including smartphones, tablets, drones and edge devices, vary in processing power, memory capacity and network bandwidth, creating challenges in coordinating and synchronizing training for a central model (Ye et al., 2023). For instance, while some devices may complete computations quickly, others, constrained by slower processors, limited connectivity, or power shortages, introduce delays and inefficiencies, potentially biasing the global aggregation process.

To address system heterogeneity in FL, several techniques have been developed. Asynchronous FL allows clients to send updates independently, reducing delays caused by slower devices. Faster clients send updates to the server as soon as they finish, without waiting for slower clients. This prevents delays in training caused by waiting for slower devices, allowing faster clients to progress. Federated averaging with partial participation selects a subset of clients for each training round based on available resources, ensuring underperforming devices do not hinder overall progress (McMahan et al., 2017).

Moreover, Heterogeneity-aware optimization techniques dynamically adjust learning rates and local updates based on client constraints. For instance, the FedProx algorithm extends FedAvg by adding a regularization term to prevent large deviations in model updates from weaker clients, improving convergence stability across heterogeneous devices (Sahu et al., 2018). Another method, hierarchical FL, reduces the communication burden by introducing a middle layer (e.g., edge servers) to aggregate updates locally before sending them to the central server, thereby managing the challenges of network heterogeneity (Menegatti et al., 2023). The important aspect of system heterogeneity is its impact on model performance and convergence. Disparities in device capabilities can lead to imbalanced contributions, where more powerful devices dominate training, potentially biasing the global model and reducing its ability to generalize effectively. Addressing these issues ensures that all clients, regardless of their limitations, contribute to an unbiased and robust FL system.

3.7 Privacy

Although Federated Learning (FL) was developed to address privacy concerns in distributed model training, the risk of data pattern leakage persists due to shared model weights and parameters with the central server. Data leakage can occur through inversion attacks, which enable attackers to reconstruct training data by querying shared model updates. Additionally, membership inference attacks can determine whether specific data points were used in training, while property inference techniques can reveal statistical properties of the training dataset (Zhang et al., 2023). To mitigate these risks, privacy-preserving methods such as Differential Privacy (DP) and Homomorphic Encryption (HE) have been adopted in FL.

Differential Privacy (DP) protects data by adding controlled noise to model weights during the training process. The privacy budget parameter (ϵ) determines the level of noise added, with smaller ϵ values ensuring higher privacy at the cost of model accuracy. A study done by Wei et al. (2020), highlights that DP is computationally efficient compared to traditional encryption methods like Secure Multiparty Computation (SMC), particularly for large datasets. Applying Gaussian noise to aggregated updates effectively prevents data leakage but decreases model accuracy as noise levels increase. However, DP faces challenges due to the trade-off between privacy and performance; excessive noise from lower ϵ values can impair model accuracy and delay convergence.

In contrast, Homomorphic Encryption (HE) ensures privacy by allowing encrypted model weights to undergo operations without decryption. Unlike Differential Privacy, Homomorphic Encryption preserves model accuracy since it does not modify weight values with noise. Zhang et al. (2020) proposed the Batch Crypt system, which quantizes and encodes gradients into low-bit integers, significantly reducing computational costs during aggregation. Experiments demonstrated a 23×–93× training speedup with less than 1% accuracy loss. Additionally, frameworks like the Paillier Federated Multi-Layer Perceptron, optimized by Fang and Qian (2021), reduce encryption and decryption times by 25%–28% compared to traditional HE methods, as shown in experiments using the MNIST dataset. These advancements underscore the effectiveness of DP and HE in addressing privacy concerns in FL systems, balancing security, accuracy and computational efficiency.

3.8 Advanced architectures in federated learning

While the foundational principles of Federated Learning (FL) address key privacy concerns, the practical demands of large-scale, heterogeneous and security-sensitive environments have necessitated the development of more sophisticated architectures. Traditional "flat" FL models, which rely on a single central server to coordinate all clients, often encounter bottlenecks in communication, scalability and security. In response, recent research has explored advanced architectural paradigms, prominently featuring hierarchical structures to manage complexity and blockchain integration to enhance security and decentralization. These approaches move beyond the initial conception of FL to create more robust, efficient and trustworthy systems suitable for real-world deployment in domains like the Internet of Things (IoT) and vehicular networks.

3.8.1 Hierarchical structures for scalability and heterogeneity

Hierarchical Federated Learning introduces multiple levels of aggregation to create a more scalable and communication-efficient system. Instead of all clients communicating directly with a single, often distant, cloud server, a multi-tiered structure is employed. A common implementation is the client-edge-cloud architecture, where intermediate edge servers perform partial model aggregation for clients within their proximity (Liu L. et al., 2020). This significantly reduces the communication overhead on the core network, as only the aggregated models from edge servers are sent to the central cloud. This structure not only accelerates training time and reduces energy consumption for end-user devices but also provides a better trade-off between communication and computation (Liu L. et al., 2020).

Beyond improving efficiency, hierarchical structures offer a powerful mechanism for managing statistical heterogeneity, a persistent challenge in FL where client data is non-independent and identically distributed (non-IID). One innovative approach involves integrating hierarchical clustering with the FL process. In this model, instead of training a single global model for all clients, the system first clusters clients based on the similarity of their local model updates (Briggs et al., 2020). Once clustered, specialized models are trained independently and in parallel for each cluster of similar clients. This method has been shown to achieve convergence in fewer communication rounds and allows a greater percentage of clients to reach a target accuracy, especially in non-IID settings, by tailoring models to specific data distributions (Briggs et al., 2020).

Further advancing this hierarchical concept, recent frameworks like Federated Learning with Hierarchical Neural Architecture Search (FINCH) automate the search for optimal model architectures within these clustered, hierarchical structures (Liu et al., 2024b). FINCH first divides clients into clusters based on their data distribution and then allocates different subnets from a pre-trained supernet to each cluster for parallel architecture searching and training. This approach significantly reduces the completion time and improves accuracy by narrowing the search space and tailoring architectures to specific data patterns within each cluster (Liu et al., 2024b). Complementing this, other strategies such as experience-driven model migration, as seen in the FedMigr framework, use deep reinforcement learning to intelligently guide the transfer of local models between clients. This migration is equivalent to training on more diverse data, which helps to reduce the parameter divergence caused by non-IID data and further enhances the performance of the global model (Liu et al., 2024a).

3.8.2 Blockchain integration for enhanced security and decentralization

A fundamental limitation of classical FL is its reliance on a central server, which acts as a single point of failure and requires clients to place their trust in the coordinating entity. Blockchain technology offers a transformative solution by enabling a fully decentralized, secure and auditable FL process. This integration, often termed "FLchain," replaces the central server with a distributed ledger where model updates are managed through a consensus mechanism among participating nodes (Nguyen et al., 2021).

In a blockchain-enabled FL system, local model updates from devices are treated as transactions. These transactions are then verified, aggregated and recorded in blocks by miners (e.g., edge servers). The resulting blockchain creates an immutable and transparent record of the entire training process, which enhances security against poisoning attacks and provides a clear audit trail (Qu et al., 2020). The FL-Block framework, for instance, allows end devices to exchange local learning updates via a blockchain-based global model, coordinated by a Proof-of-Work consensus mechanism. This eliminates the need for any centralized authority, enhances privacy protection and increases resistance to malicious attacks (Qu et al., 2020).

The synergy between hierarchical structures and blockchain technology creates even more powerful and scalable frameworks. For instance, in the context of the Internet of Vehicles (IoVs), a hierarchical blockchain framework can be designed to manage knowledge sharing across different geographic regions (Chai et al., 2021). In such a system, vehicles and roadside units (RSUs) form local "Ground-Chains" for primary FL, while RSUs and base stations (BSs) form a "Top-Chain" for higher-level aggregation. This layered approach not only accommodates the dynamic and large-scale nature of vehicular networks but also uses a lightweight Proof-of-Knowledge (PoK) consensus mechanism to ensure the security and reliability of the shared knowledge without the computational burden of traditional consensus algorithms (Chai et al., 2021). By integrating these advanced architectures, FL can overcome many of its practical limitations, paving the way for more secure, scalable and efficient decentralized intelligence in complex, real-world systems.

4 Current applications of federated learning

Federated learning (FL) has been increasingly adopted across various domains due to its ability to perform decentralized machine/deep learning while maintaining data privacy and low communication costs. The scenarios discussed below highlight the unique applications and benefits of FL in different domains. As per the review done by Brick et al., in the military sector, specially for Unmanned Air Vehicles (UAVs), FL provides several potential benefits. It enhances privacy and security by keeping raw data on UAVs while sharing only the model updates, reducing the risk of exposing sensitive information like UAV locations or identities. By enabling local computations, FL also reduces communication overhead which is crucial for missions deployed in bandwidth-constrained territories and time-critical situations where faster decision making is a must. This approach improves energy efficiency by conserving resources through reduced data transmissions. These benefits enhance performance of various UAV applications such as wireless channel modeling, trajectory planning, content caching and routing in flying *Ad-Hoc* Networks (Brik et al., 2020).

As per the comprehensive survey done by Li et al., FL in healthcare allows hospitals to collaboratively train models for disease diagnosis (e.g., cancer and tumor detection or segmentation in MRI and CT scans, COVID-19 diagnosis etc.) by sharing model parameter updates instead of electronic health records (EHRs). This ensures privacy and compliance with regulations like Health Insurance Portability and Accountability Act (HIPAA). This approach improves model accuracy by using diverse data representing diverse demographics and enabling cross institutional collaboration (Li L. et al., 2020). Figure 6 illustrates a federated learning architecture in healthcare where multiple healthcare institutions collaborate to train a global model. The diagram shows three different healthcare facilities, each with their own local data sources including patient records, medical imaging data and diagnostic information. Each institution has a local model that processes their private data locally, while a centralized federated server coordinates the training process. The key feature shown is that patient data remains localized and private within each institution, while only model updates and parameters are shared with the federated server, enabling collaborative learning across institutions while maintaining patient privacy and regulatory compliance.

Another use of FL in healthcare is for wearable devices and remote patient monitoring. FL enables training models on data from wearable devices, such as smartwatches or fitness trackers, to detect health conditions like arrhythmia, diabetes and heart strokes. This allows for personalized healthcare recommendations and early warnings while protecting user privacy (Lim et al., 2020). Further extending its utility, FL is applied to drug discovery, where models can be trained across different pharmaceutical institutions without sharing proprietary chemical data, and for analyzing functional MRI (fMRI) data to identify biomarkers for neurological disorders (Aledhari et al., 2020).

As per the research done by Du et al., in the Vehicular IoT, FL addresses privacy concerns and optimizes resource usage by enabling vehicles to collaboratively train machine learning models without sharing raw data. This is very useful for applications such as cooperative autonomous driving and intelligent transport systems (ITS), where vast amounts of data from vehicle sensors (e.g., LIDAR) are used to make decisions in real-time. FL reduces communication overhead by only transmitting model updates instead of raw data, which is crucial considering the limited bandwidth and dynamic nature of vehicular networks. It supports the integration of diverse sets of data from multiple vehicles to enhance decision-making processes, enabling improved traffic management, law enforcement and safety measures (Du et al., 2020). Figure 7 presents a comprehensive decentralized Internet of Things (IoT) training architecture specifically designed for traffic systems. The diagram illustrates a smart city traffic scenario where multiple vehicles equipped with LiDAR sensors are distributed across various road intersections and traffic networks. Each vehicle collects local data through its LiDAR sensors and other IoT devices, which is then processed locally to train independent machine learning models. The architecture shows how these local models can securely communicate and update a global model while maintaining privacy, including traffic infrastructure elements such as traffic lights, road networks and communication towers that facilitate the federated learning process.

The domain of mobility, which includes autonomous vehicles, benefits significantly from FL's ability to decouple latency-sensitive applications. For instance, a self-driving car acting as a learner can use its local model to react instantly to environmental changes without waiting for a server response (Gecer and Garbinato, 2024). This is critical for safety-related tasks such as collision avoidance and traffic sign classification. Moreover, FL is being applied to optimize ride-hailing services and predict energy demand for electric vehicle networks, all while protecting user location and travel data (Shaheen et al., 2022).

As demonstrated by Google's Gboard, FL has been effectively applied in mobile keyboard prediction. This approach allows for the training of language models directly on user devices, enhancing next word prediction capabilities while respecting and protecting user privacy. By utilizing FedAvg algorithm, model updates are computed locally on client devices and then aggregated on a central server to improve the global model without transferring sensitive data (Hard et al., 2018). This application highlights a core benefit of FL: personalization. Since training a single global model on all user data may not be optimal for any individual, FL provides a natural infrastructure for learning personalized models. Advanced personalization techniques within FL include user clustering, data interpolation (combining local and global data), and model interpolation (combining local and global models) to create intermediate models that balance generalization with individual user needs (Mansour et al., 2020).

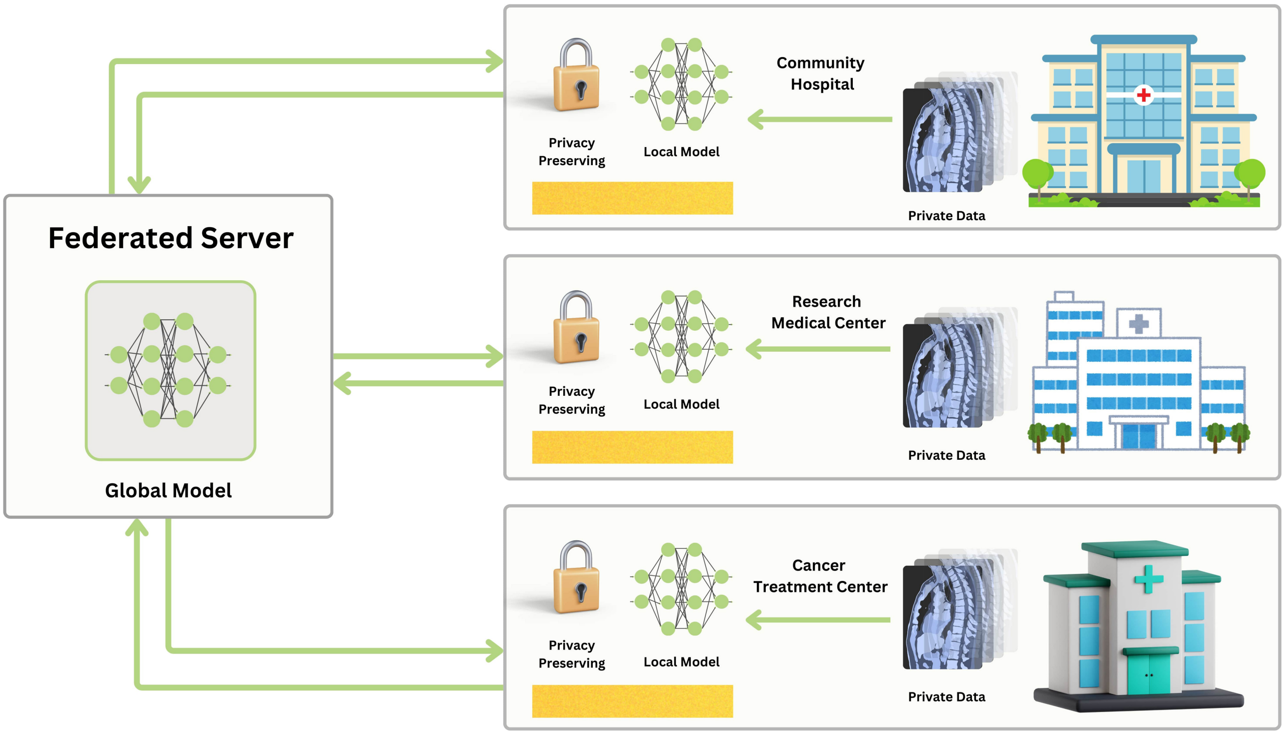

FIGURE 6
A use case illustrating that multiple healthcare institutions can work together to train a global model using a federated server, collecting findings while protecting patient privacy by preserving patient data localized and private.

Pandya et al., demonstrates how FL offers significant benefits for smart city applications by enabling decentralized data processing and there by preserving privacy and reducing communication overhead. Key use cases include smart transportation, healthcare, grid management, governance, disaster response and industries. FL allows these sectors to process data locally while sharing model updates for global insights. Other emerging applications span industrial IoT, for tasks like visual inspection and anomaly detection (Shaheen et al., 2022), and finance, for credit card fraud detection and anti-financial crime processes, where data sharing between institutions is highly restricted (Li L. et al., 2020). Across all the domains discussed above and many other domains that weren't mentioned, FL provides benefits such as enhanced privacy and compliance with privacy regulations by keeping raw data local, reduced communication overhead through local computations, model update sharing, improved resource efficiency and management by minimizing data transmission needs.

4.1 Practical implementation challenges in real-world federated learning systems

While Federated Learning (FL) presents a robust theoretical framework for privacy-preserving, decentralized machine learning, its transition from concept to real-world application is filled with significant practical challenges. These issues often stem from the inherent characteristics of the environments where FL is most needed, such as the Internet of Things (IoT), which involves a massive number of diverse and resource-constrained devices (Diba et al., 2025). Overcoming these practical hurdles is crucial for the successful and widespread deployment of FL systems. Key implementation challenges identified in recent literature include limited on-device resources, network bandwidth and connectivity issues, system and statistical heterogeneity (Zhang et al., 2022) and the lack of standardization and robust development tools (Wen et al., 2023).

A primary practical barrier is the limited on-device resources of the participating clients, especially in IoT networks (Zhang et al., 2022). Many IoT devices are designed with constraints on computational power, memory and energy budgets (Rjoub et al., 2025). Training machine learning models, particularly deep neural networks, is a computationally intensive task that can be inefficient and time-consuming on such hardware. Furthermore, these models require substantial memory not only for storing model weights and parameters but also for the intermediate results of computations (Zhang et al., 2022). The limited energy budgets of embedded processors further restrict hardware performance, posing a significant challenge to the local training required in FL (Wen et al., 2023). This resource scarcity severely impedes the deployment of complex models and can result in low training efficiency on edge devices (Zhang et al., 2022).

Closely related to on-device limitations is the challenge of limited network bandwidth and connectivity (Kairouz et al., 2021). Most IoT devices communicate over wireless networks, which typically have much lower bandwidth compared to wired

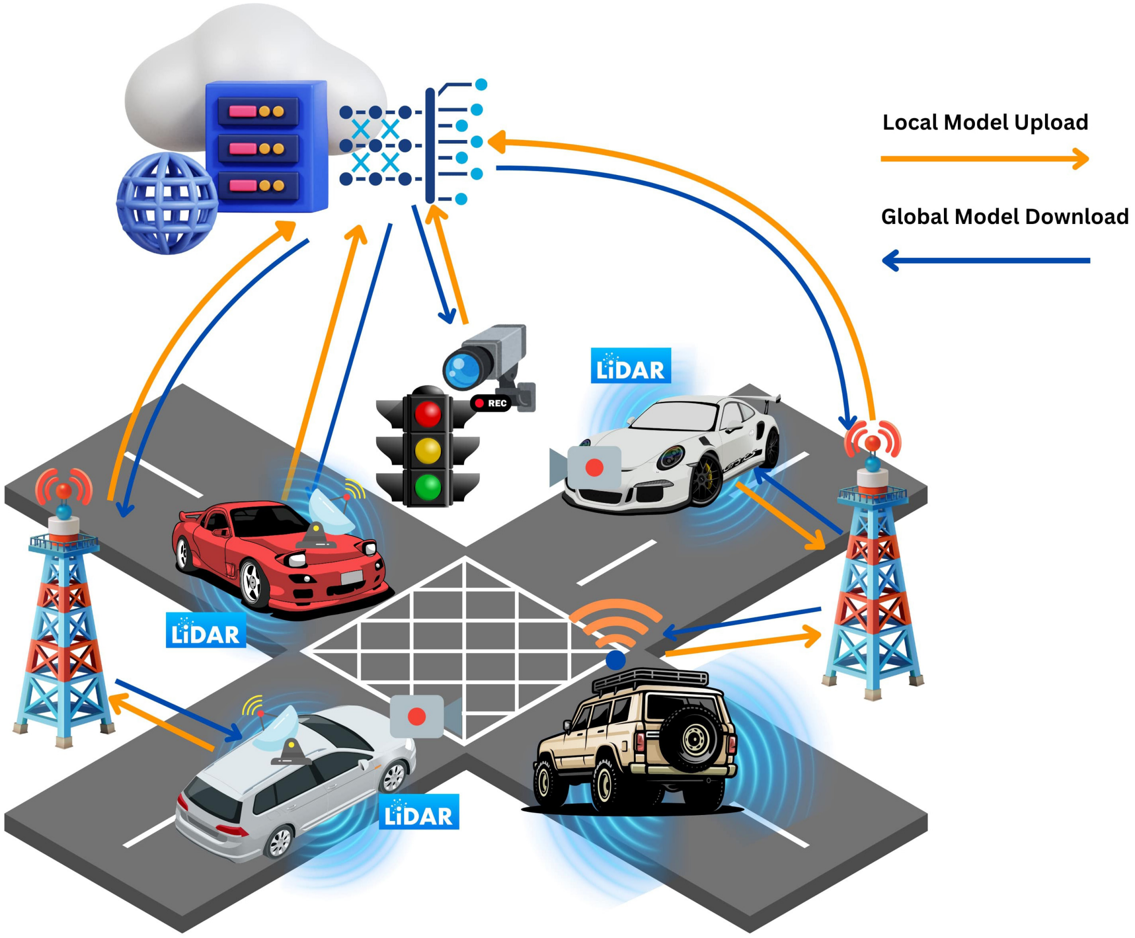

FIGURE 7
A decentralized Internet of Things (IoT) training architecture in a traffic system is illustrated in a diagram. Local data is collected by cars equipped with LiDAR sensors, which then train independent models and securely update a global model while maintaining privacy.

data center networks (Zhang et al., 2022). This communication bottleneck can make the transfer of model updates between clients and the central server inefficient, especially as the number of participating devices grows (Rjoub et al., 2025). In scenarios like a smart home, the total network bandwidth is a fixed resource; as more devices join the system, the available bandwidth per device decreases, making the communication problem worse (Zhang et al., 2022). Beyond bandwidth limitations, real-world settings are characterized by intermittent connectivity and availability (Kairouz et al., 2021). Devices may drop out of the network in the middle of a training round due to unstable connections, leading to the problem of "stragglers" who fail to report their updates in time (Zhang et al., 2022). This unreliability makes synchronous update protocols, where the server waits for all selected clients to respond, nearly impossible to implement effectively in large-scale IoT systems and can endanger the convergence of the training process (Zhang et al., 2022).

Another significant practical issue is the widespread heterogeneity across the system (Wen et al., 2023). This heterogeneity manifests in several forms. System heterogeneity refers to the diversity in hardware, operating systems and software APIs across the vast number of client devices (Zhang et al., 2022). Clients may use different deep learning frameworks (e.g., TensorFlow, PyTorch), resulting in different model formats that need to be aggregated. This diversity complicates system design and exacerbates asynchronous communication challenges (Zhang et al., 2022). Furthermore, statistical heterogeneity, where the data on each device is not independent and identically distributed (non-IID) (Rjoub et al., 2025), is a defining characteristic of FL (Kairouz et al., 2021). This challenge goes beyond simple non-IID distributions to include issues like data imbalance (where data size varies significantly across clients), local imbalance (where data distributions differ), and global class imbalance (Shaheen et al., 2022). The data collected by different devices can vary significantly in terms of features, dimensions and temporal patterns, creating discrepancies in local data structures among participants (Zhang et al., 2022). For instance, a surveillance camera might record video continuously, while a smart doorbell generates data intermittently. This non-IID data distribution can lead to the global model drifting away from the optimal solution, thereby degrading its performance (Wen et al., 2023).

Finally, the practical implementation of FL is hindered by a lack of standardization and system development tools (Zhang et al., 2022). As an emerging field, FL lacks standardized protocols for communication, data flow models and network configurations (Diba et al., 2025). This absence of standards impedes the widespread deployment of FL systems and makes it difficult to create an open environment for content sharing and interoperability (Zhang et al., 2022). From a development perspective, there is a need for user-friendly, integrated simulation environments that can help researchers and developers design and evaluate FL systems at a large scale (Kairouz et al., 2021) without requiring full implementation in a real-world setting (Li et al., 2025). Existing tools for edge computing often focus on model inference rather than the complexities of distributed training, leaving an under-explored area that is critical for the FL community (Zhang et al., 2022). Without mature, system-level frameworks and development tools, accomplishing tasks such as load balancing, resource management and task scheduling remains a significant challenge for practitioners (Zhang et al., 2022). Tackling these multifaceted practical issues is essential to bridge the gap between theoretical promise and real-world viability for Federated Learning.

5 Experimental results and benchmarking

Since this comprehensive analysis discusses the core challenges and solutions for federated learning based on communication cost, statistical and system heterogeneity and privacy concerns, we can evaluate their performance based on different metrics. Table 3 analyzes the performance done by each discussed aggregation technique study with different neural network architectures and datasets. As the table illustrates, the foundational FL model, FedAvg achieved 99.44% accuracy on MNIST and 85% accuracy on CIFAR-10 with 100 devices, despite requiring 300 and 2,000 communication rounds to achieve these accuracies (McMahan et al., 2017). However, FedProx has shown a notable improvement over FedAvg when it was trained with the MNIST Non-IID dataset by achieving 22% better accuracy with 1,000 devices, while it has demonstrated a better convergence on the FEMNIST and Shakespeare datasets (Sahu et al., 2018). However, the FedNova approach has demonstrated a 69.92% accuracy with only 100 communication rounds and using only 16 devices on the non-IID CIFAR-10 dataset, showing a higher performance with non-IID data with a smaller amount of communication rounds (Wang et al., 2020). The MOON framework has achieved 69.1% accuracy on the CIFAR-10 dataset in just 100 communication rounds with 50 devices (Li et al., 2021), while FedSim, which has introduced weight similarities-based local device clustering, has demonstrated strong performance across diverse datasets, reaching approximately 85% accuracy on MNIST with 1,000 devices in just 30 rounds (Palihawadana et al., 2022). Additionally, SCAFFOLD employed control variates to correct for client drift at both server and client levels, making it robust to statistical heterogeneity and reaching up to 84.2% accuracy on nonconvex EMNIST tasks while converging in fewer communication rounds.

Advanced personalization techniques have shown remarkable results in addressing heterogeneous federated environments. FedGPA, which decomposes models into feature extractors and classifiers with personalized aggregation weights based on prototype similarity, achieved state-of-the-art results across five challenging non-IID datasets, including 76.73% on CIFAR-10 and 90.04% on FMNIST. Hybrid approaches combining multiple strategies have demonstrated outstanding performance, with FedHybrid integrating FedAvg, FedProx and SCAFFOLD by combining proximal regularization with control variates, achieving exceptional accuracy of 94.12% on MNIST and 93.52% on CIFAR-10 using 100 clients (Niu and Wei, 2023). In application-specific domains, FEDMWAD targeted seizure prediction in healthcare using hypernetworks to generate module-wise aggregation weights, achieving 90.6% accuracy and a 0.942 AUC on the non-IID CHB-MIT EEG dataset (Ding et al., 2025).

Beyond the benchmarked results shown in Table 3, this paper has discussed experiments focused on privacy preservation techniques that revealed significant insights on privacy in federated architectures. Differential Privacy implementations demonstrated that a privacy budget (added bias) of models ensures privacy of federated systems, but it will lead to a decrement of performance accuracy of the models maintained in the federated architecture (Ouadrhiri and Abdelhadi, 2022). In addition, the BatchCrypt homomorphic encryption system has achieved a 23×–93× speedup in training time with less than 1% accuracy loss by showing a significant improvement in performance compared to other traditional encryption methods in federated learning (Fang and Qian, 2021). Furthermore, the communication efficiency experiments demonstrated that decentralized communication architectures have also been able to reduce communication-based network bottlenecks significantly. The EdgeFL method has shown a 50% reduction in weight updating latency and model evaluation time compared to the common centralized architecture (Zhang et al., 2024). Moreover, the model compression technique related works have achieved 0.75%–3.9% higher average accuracy compared to existing compression algorithms while reducing communication overhead (Shah and Lau, 2023).

The client-edge-cloud hierarchical system has shown a big boost in the model convergence process speed and training efficiency on the MNIST and CIFAR-10 datasets when dealing with non-IID data distributions, especially when the communication frequency with the edge server is increased (Menegatti et al., 2023). This is because it addresses system heterogeneity-based concerns in federated learning. In addition, clustered federated learning has also made a significant stride in handling statistical heterogeneity by achieving a 2× improvement in accuracy on the CIFAR-10 dataset under non-IID data conditions by applying gradient similarity-based clustering (Sattler et al., 2019). Moreover, in addressing the system heterogeneity concerns in the federated architecture, asynchronous federated learning methods have shown improvements in training efficiency by allowing local devices to contribute updates at different times based on the device's computational capabilities (Kairouz et al., 2021). The aggregation method FedProx has also addressed the system heterogeneity challenge through its regularization approach, while hierarchical federated learning has reduced the communication overhead by introducing intermediate aggregation at edge servers. These results

TABLE 3 Comparison of federated learning methods and performance.

Paper	Method used	Model used	Evaluated datasets	Evaluations	Communication rounds	Device count
McMahan et al. (2017)	FedAvg	CNN	MNIST	99.44% Accuracy	300	100
			CIFAR-10	85% Accuracy	2,000	100
Sahu et al. (2018)	FedProx	Multinomial logistic regression	MNIST (10 classes) (non-IID)	Training loss: FedProx converges to ∼0.5, FedAvg diverges. +22% better accuracy than FedAvg with non-IID data	200–400	1,000
			FEMNIST (62 classes)	Training loss: FedProx converges to ∼1.0, FedAvg reaches ∼2.0	200	200
		Two-layer LSTM (100 hidden units)	Shakespeare	Training loss: FedProx converges to ∼2.0, FedAvg reaches ∼3.0	20	143
Wang et al. (2020)	FedNova	VGG-11	CIFAR-10 (non-IID)	69.92% Accuracy	100	16
Karimireddy et al. (2021)	SCAFFOLD	Logistic regression, 2-layer NN, quadratic functions	EMNIST, simulated data	Consistently outperforms FedAvg and SGD, especially on heterogeneous (non-IID) data. Achieved up to 84.2% accuracy on nonconvex EMNIST task	4–286	100
Li et al. (2021)	MOON	CNN	CIFAR-10	69.1% Accuracy	27	50
		ResNet-50	CIFAR-100	67.5% Accuracy	100	50
Niu and Wei (2023)	FedHybrid	CNN (incorporates FedAvg + FedProx + FedScaffold components)	MNIST, CIFAR-10 (non-IID)	MNIST: 94.12% accuracy, CIFAR-10: 93.52% accuracy	100	100
Palihawadana et al. (2022)	FedSim	CNN-2D	FEMNIST	∼80% Accuracy	500	200
			MNIST	∼85% Accuracy	30	1,000
		MLP-3	Fed-MEx	∼93% Accuracy	200	30
		RNN	Fed-GoodReads	∼61% Accuracy	250	100
Kairouz et al. (2021)	FedGPA	CNN (2 versions): 2 conv + 2 FC (FMNIST/EMNIST), 3 conv + 2 FC (CIFAR)	FMNIST, EMNIST, CIFAR-10, CIFAR-100, CINIC-10 (non-IID)	Data Heterogeneity Setting 1 ($\sigma = 20$): FMNIST: 90.04%, EMNIST: 83.45%, CIFAR-10: 76.73%, CIFAR-100: 55.79%, CINIC-10: 64.40%	150	20
Ding et al. (2025)	FEDMWAD	GRU-based model with 1D CNN and fully connected layers	CHB-MIT (EEG seizure data, non-IID)	Accuracy: 90.6%, AUC: 0.942	200	13

suggest that while various approaches have been developed to address the federated learning based challenges, there is still a need for implementing more efficient methods.

6 Conclusion

This systematic review analyzed the current state of Deep Federated Learning, highlighting key advancements and the methods developed to solve its core challenges. We found that significant progress has been made in several areas. Advanced model aggregation methods, ranging from regularization techniques like FedProx (Sahu et al., 2018) to drift-correction mechanisms like SCAFFOLD (Karimireddy et al., 2021) and clustering-based strategies like FedSim (Palihawadana et al., 2022), have significantly improved model performance. Communication efficiency has been enhanced by techniques such as model compression (Shah and Lau, 2023) and decentralized training (Zhang et al., 2024). The challenges of varied data and device capabilities statistical and system heterogeneity are being addressed by these adaptive algorithms and personalized frameworks like FedGPA (Han et al., 2025). Privacy has been strengthened using formal methods like Differential Privacy (Wei et al., 2020) and Homomorphic Encryption (Zhang et al., 2020). The review also covered emerging areas like Federated Meta-Learning for rapid model adaptation, Federated Reinforcement Learning for distributed decision-making (Qi et al., 2021) and advanced hierarchical (Menegatti et al., 2023) and blockchain (Nguyen et al., 2021) architectures. These developments have enabled FL to be applied in critical domains such as healthcare (Li L. et al., 2020) and vehicular networks (Du et al., 2020).

While the field is advancing rapidly, important challenges and limitations remain. Achieving higher accuracy often requires complex algorithms that increase computational and communication costs. Strong privacy-preserving methods such as differential privacy and homomorphic encryption can degrade model performance or impose significant resource demands, restricting their adoption on resource-constrained devices. Many studies continue to rely on benchmark datasets or simulated environments, which may not reflect the complexity of real-world deployments (Wen et al., 2023). In addition, the absence of standardized frameworks and evaluation protocols hinders reproducibility and complicates meaningful cross-study comparisons. Practical deployment is further constrained by device heterogeneity, unstable connectivity, limited energy resources, and interoperability issues across platforms. These barriers highlight that, despite notable progress, federated learning still faces significant obstacles before it can achieve consistent, large-scale, and reliable real-world adoption.

Looking ahead, future research must prioritize solving these practical deployment challenges. This includes developing hybrid algorithms that are not only accurate but also resource-efficient and robust to the network instability common in real-world environments. Crucial work is needed to create more efficient implementations of privacy-preserving techniques like Homomorphic Encryption to reduce their computational burden on client devices. Furthermore, the development of standardized frameworks and tools is essential to simplify deployment and ensure interoperability.

A significant emerging research frontier is Green Federated Learning (GFL), which emphasizes sustainability by designing energy-efficient algorithms. Since machine learning is among the most energy-intensive computational domains, future research must prioritize developing FL systems with minimal carbon footprints. Recent work advocates for this paradigm by proposing energy-aware client selection, model compression, and standardized metrics to evaluate the environmental impact of FL (Thakur et al., 2025). This is particularly critical for IoT applications, where energy-efficient models can deliver practical benefits. For instance, a hybrid approach applied to intelligent transportation systems demonstrated notable energy savings through lightweight optimization and personalized, energy-aware model aggregation, underscoring the feasibility of GFL in real-world deployments (Kaleem et al., 2024).

Another important direction involves addressing the distinctive challenges of cross-silo FL, where a relatively small number of organizations such as hospitals or banks collaborate. Future research must move beyond technical optimizations to incorporate game-theoretic and incentive-based mechanisms that can govern cooperation and competition among strategic participants. This includes frameworks for data valuation and fair profit allocation to ensure long-term, stable collaboration (Huang et al., 2022).

In addition, cybersecurity in IoT represents a promising opportunity for FL adoption. By enabling privacy-preserving, collaborative intrusion detection systems (IDSs), FL can significantly enhance the security of distributed IoT networks (Arisdakessian et al., 2023). However, for these security systems to be trustworthy, they must also be transparent. A critical future direction is the integration of Explainable AI (XAI) into FL, allowing models to remain privacy-preserving while also interpretable, thereby enhancing expert trust in decision-making for sensitive IoT environments.

Alongside this focus on practical issues, promising research directions include exploring the integration of FL with next-generation technologies, such as leveraging 5G networks to reduce communication latency (Liu Y. et al., 2020) and developing Quantum Federated Learning (QFL) for secure quantum tasks (Chehimi and Saad, 2022). In conclusion, federated learning is a rapidly advancing field and addressing both its current practical limitations and future technological opportunities through continued innovation will be essential to unlocking its full potential.

Author contributions

LC: Writing – review & editing, Writing – original draft. JS: Writing – review & editing, Writing – original draft. PV: Writing – review & editing, Writing – original draft. YP: Writing – original draft, Writing – review & editing, Supervision.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aledhari, M., Razzak, R., Parizi, R. M., and Saeed, F. (2020). Federated learning: a survey on enabling technologies, protocols, and applications. *IEEE Access* 8, 140699–140725. doi: 10.1109/ACCESS.2020.3013541

Arisdakessian, S., Wahab, O. A., Mourad, A., Otrok, H., and Guizani, M. (2023). A survey on iot intrusion detection: federated learning, game theory, social psychology, and explainable ai as future directions. *IEEE Internet Things J.* 10, 4059–4092. doi: 10.1109/JIOT.2022.3203249

Briggs, C., Fan, Z., and Andras, P. (2020). "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *2020 International Joint Conference on Neural Networks (IJCNN)* (Glasgow: IEEE), 1–9. doi: 10.1109/IJCNN48605.2020.9207469

Brik, B., Ksentini, A., and Bouaziz, M. (2020). Federated learning for uavs-enabled wireless networks: use cases, challenges, and open problems. *IEEE Access* 8, 53841–53849. doi: 10.1109/ACCESS.2020.2981430

Chai, H., Leng, S., Chen, Y., and Zhang, K. (2021). A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* 22, 3975–3986. doi: 10.1109/TITS.2020.3002712

Chehimi, M., and Saad, W. (2022). "Quantum federated learning with quantum data," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Singapore: IEEE), 8617–8621. doi: 10.1109/ICASSP43922.2022.9746622

Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. (2019). Federated meta-learning with fast convergence and efficient communication. *arXiv [Preprint].* arXiv: 1802.07876. Available online at: https://arxiv.org/abs/1802.07876

Diba, B. S., Plabon, J. D., Mowla, T. J., Nahar, N., Mistry, D., Sarker, S., et al. (2025). Open problems and challenges in federated learning for IOT: a comprehensive review and strategic guide. *Comput. Electr. Eng.* 126:110515. doi: 10.1016/j.compeleceng.2025.110515

Ding, Y., Zhao, W., and Huang, K. (2025). Fedmwad: module-wise weighted aggregation federated learning combined with ditto for patient-independent seizure prediction. *Inf. Process. Manag.* 62:104253. doi: 10.1016/j.ipm.2025.104253

Du, Z., Wu, C., Yoshinaga, T., Yau, K.-L. A., Ji, Y., Li, J., et al. (2020). Federated learning for vehicular internet of things: recent advances and open issues. *IEEE Open J. Comput. Soc.* 1, 45–61. doi: 10.1109/OJCS.2020.2992630

Fang, H., and Qian, Q. (2021). Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet* 13:94. doi: 10.3390/fi13040094

Gecer, M., and Garbinato, B. (2024). Federated learning for mobility applications. *ACM Comput. Surv.* 56:133. doi: 10.1145/3637868

Han, Z., Feng, Y., Zhu, Y., Tian, Z., Hao, F., Song, M., et al. (2025). Fedgpa: federated learning with global personalized aggregation. *AI Open* 6, 82–92. doi: 10.1016/j.aiopen.2025.03.001

Hard, A. S., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., et al. (2018). Federated learning for mobile keyboard prediction. *arXiv [Preprint].* arXiv:1811.03604. doi: 10.48550/arXiv.1811.03604

Huang, C., Huang, J., and Liu, X. (2022). Cross-silo federated learning: challenges and opportunities. *arXiv [Preprint].* arXiv: 2206.12949. Available online at: https://arxiv.org/abs/2206.12949

Jiang, Z., Xu, Y., Xu, H., Wang, Z., Liu, J., Chen, Q., et al. (2024). Computation and communication efficient federated learning with adaptive model pruning. *IEEE Trans Mob Comput*, 23, 2003–2021. doi: 10.1109/TMC.2023.3247798

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., et al. (2021). Advances and open problems in federated learning. *Found. Trends Mach. Learn.* 14, 1–210. doi: 10.1561/9781680837896

Kaleem, S., Sohail, A., Babar, M., Ahmad, A., and Tariq, M. U. (2024). A hybrid model for energy-efficient green internet of things enabled intelligent transportation systems using federated learning. *Internet Things* 25:101038. doi: 10.1016/j.iot.2023.101038

Karimireddy, S. P., Kale, S., Mohri, M., Rddi, S. J., Stich, S. U., and Suresh, A. T. (2021). SCAFFOLD: stochastic controlled averaging for federated learning. *arXiv [Preprint].* arXiv: 1910.06378. Available online at: https://arxiv.org/abs/1910.06378

Li, L., Fan, Y., Tse, M., and Lin, K.-Y. (2020). A review of applications in federated learning. *Comput. Ind. Eng.* 149:106854. doi: 10.1016/j.cie.2020.106854

Li, M., Xu, P., Hu, J., Tang, Z., and Yang, G. (2025). From challenges and pitfalls to recommendations and opportunities: implementing federated learning in healthcare. *Med. Image Anal.* 101:103497. doi: 10.1016/j.media.2025.103497

Li, Q., He, B., and Song, D. (2021). "Model-contrastive federated learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA: IEEE Computer Society), 10708–10717. doi: 10.1109/CVPR46437.2021.01057

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: challenges, methods, and future directions. *IEEE Signal Process. Mag.* 37, 50–60. doi: 10.1109/MSP.2020.2975749

Li, X., Qu, Z., Tang, B., and Lu, Z. (2024). Fedlga: toward system-heterogeneity of federated learning via local gradient approximation. *IEEE Trans. Cybern.* 54, 401–414. doi: 10.1109/TCYB.2023.3247365

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., et al. (2020). Federated learning in mobile edge networks: a comprehensive survey. *IEEE Commun. Surv. Tutor.* 22, 2031–2063. doi: 10.1109/COMST.2020.2986024

Liu, J., Wang, S., Xu, H., Xu, Y., Liao, Y., Huang, J., et al. (2024a). Federated learning with experience-driven model migration in heterogeneous edge networks. *IEEE/ACM Trans. Netw.* 32, 3468–3484. doi: 10.1109/TNET.2024.3390416

Liu, J., Yan, J., Xu, H., Wang, Z., Huang, J., Xu, Y., et al. (2024b). Finch: enhancing federated learning with hierarchical neural architecture search. *IEEE Trans. Mob. Comput.* 23, 6012–6026. doi: 10.1109/TMC.2023.3315451

Liu, L., Zhang, J., Song, S., and Letaief, K. B. (2020). "Client-edge-cloud hierarchical federated learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)* (Dublin: IEEE), 1–6. doi: 10.1109/ICC40277.2020.9148862

Liu, X., Deng, Y., Nallanathan, A., and Bennis, M. (2024). Federated learning and meta learning: approaches, applications, and directions. *IEEE Commun. Surv. Tutor.* 26, 571–618. doi: 10.1109/COMST.2023.3330910

Liu, Y., Peng, J., Kang, J., Iliyasu, A. M., Niyato, D., El-Latif, A. A. A., et al. (2020). A secure federated learning framework for 5g networks. *IEEE Wirel. Commun.* 27, 24–31. doi: 10.1109/MWC.01.1900525

Luo, B., Xiao, W., Wang, S., Huang, J., and Tassiulas, L. (2021). "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications* (London: IEEE). doi: 10.1109/INFOCOM48880.2022.9796935

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. (2020). Three approaches for personalization with applications to federated learning. *arXiv [Preprint].* arXiv: 2002.10619. Available online at: https://arxiv.org/abs/2002.10619

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017). "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics*. Available online at: https://api.semanticscholar.org/CorpusID:14955348

Menegatti, D., Manfredi, S., Pietrabissa, A., Poli, C., and Giuseppi, A. (2023). Hierarchical federated learning for edge intelligence through average consensus. *IFAC-PapersOnLine* 56, 862–868. doi: 10.1016/j.ifacol.2023.10.1673

Nguyen, D. C., Ding, M., Pham, Q.-V., Pathirana, P. N., Le, L. B., Seneviratne, A., et al. (2021). Federated learning meets blockchain in edge computing: opportunities and challenges. *IEEE Internet Things J.* 8, 12806–12825. doi: 10.1109/JIOT.2021.3072611

Niu, X., and Wei, E. (2023). Fedhybrid: a hybrid federated optimization method for heterogeneous clients. *IEEE Trans. Signal Process.* 71, 150–163. doi: 10.1109/TSP.2023.3240083

Ouadrhiri, A. E., and Abdelhadi, A. (2022). Differential privacy for deep and federated learning: a survey. *IEEE Access* 10, 22359–22380. doi: 10.1109/ACCESS.2022.3151670

Palihawadana, C., Wiratunga, N., Wijekoon, A., and Kalutarage, H. (2022). Fedsim: similarity guided model aggregation for federated learning. *Neurocomputing* 483, 432–445. doi: 10.1016/j.neucom.2021.08.141

Qi, J., Zhou, Q., Lei, L., and Zheng, K. (2021). Federated reinforcement learning: techniques, applications, and open challenges. *Intell. Robot.* 1, 18–57. doi: 10.20517/ir.2021.02

Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., Piccialli, F., et al. (2024). Model aggregation techniques in federated learning: a comprehensive survey. *Future Gener. Comput. Syst.* 150, 272–293. doi: 10.1016/j.future.2023.09.008

Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Yu, S., Li, B., et al. (2020). Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet Things J.* 7, 5171–5183. doi: 10.1109/JIOT.2020.2977383

Reisizadeh, A., Tziotis, I., Hassani, H., Mokhtari, A., and Pedarsani, R. (2020). Straggler-resilient federated learning: leveraging the interplay between statistical accuracy and system heterogeneity. *arXiv [Preprint].* arXiv: 2012.14453. Available online at: https://arxiv.org/abs/2012.14453

Rjoub, G., Elmekki, H., Islam, S., Bentahar, J., and Dssouli, R. (2025). A hybrid swarm intelligence approach for optimizing multimodal large language models deployment in edge-cloud-based federated learning environments. *Comput. Commun.* 237:108152. doi: 10.1016/j.comcom.2025.108152

Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V., et al. (2018). Federated optimization in heterogeneous networks. *arXiv [Preprint]*. arXiv:1812.06127. doi: 10.48550/arXiv.1812.06127

Saifullah, S., Mercier, D., Lucieri, A., Dengel, A., and Ahmed, S. (2024). The privacy-explainability trade-off: unraveling the impacts of differential privacy and federated learning on attribution methods. *Front. Artif. Intell.* 7:1236947. doi: 10.3389/frai.2024.1236947

Sattler, F., Müller, K.-R., and Samek, W. (2019). Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural. Netw. Learn. Syst.* 32, 3710–3722. doi: 10.1109/TNNLS.2020.3015958

Shah, S. M., and Lau, V. K. N. (2023). Model compression for communication efficient federated learning. *IEEE Trans. Neural. Netw. Learn. Syst.* 34, 5937–5951. doi: 10.1109/TNNLS.2021.3131614

Shaheen, M., Farooq, M. S., Umer, T., and Kim, B.-S. (2022). Applications of federated learning; taxonomy, challenges, and research trends. *Electronics* 11:670. doi: 10.3390/electronics11040670

Thakur, D., Guzzo, A., Fortino, G., and Piccialli, F. (2025). Green federated learning: a new era of green aware ai. *ACM Comput. Surv.* 57:194. doi: 10.1145/3718363

Tugwell, P., and Tovey, D. (2021). Prisma 2020. *J. Clin. Epidemiol.* 134, A5-A6. doi: 10.1016/j.jclinepi.2021.04.008

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. (2020). "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20* (Red Hook, NY: Curran Associates Inc).

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., et al. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* 15, 3454–3469. doi: 10.1109/TIFS.2020.2988575

Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., Zhang, W., et al. (2023). A survey on federated learning: challenges and applications. *Int. J. Mach. Learn. Cybern.* 14, 513–535. doi: 10.1007/s13042-022-01647-y

Ye, M., Fang, X., Du, B., Yuen, P., and Tao, D. (2023). Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput. Surv.* 56, 1–44. doi: 10.1145/3625558

Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., and Liu, Y. (2020). "BatchCrypt: efficient homomorphic encryption for cross-silo federated learning," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)* (USENIX Association), 493506. Available online at: https://www.usenix.org/conference/atc20/presentation/zhang-chengliang

Zhang, H., Bosch, J., and Olsson, H. H. (2024). "EdgeFL: a lightweight decentralized federated learning framework," in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)* (Los Alamitos, CA: IEEE Computer Society), 556–561. doi: 10.1109/COMPSAC61105.2024.00081

Zhang, L., Xu, J., Vijayakumar, P., Sharma, P. K., and Ghosh, U. (2023). Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system. *IEEE Trans. Netw. Sci. Eng.* 10, 2864–2880. doi: 10.1109/TNSE.2022.3185327

Zhang, T., Gao, L., He, C., Zhang, M., Krishnamachari, B., Avestimehr, A. S., et al. (2022). Federated learning for the internet of things: applications, challenges, and opportunities. *IEEE Internet Things Mag.* 5, 24–29. doi: 10.1109/IOTM.004.2100182

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V., et al. (2018). Federated learning with non-IID data. *arXiv [Preprint]*. arXiv:1806.00582. doi: 10.48550/arXiv.1806.00582

Zhuo, H. H., Feng, W., Lin, Y., Xu, Q., and Yang, Q. (2020). Federated deep reinforcement learning. *arXiv [Preprint].* arXiv: 1901.08277. Available online at: https://arxiv.org/abs/1901.08277